



POLSKO-JAPOŃSKA  
WYŻSZA SZKOŁA  
TECHNIK KOMPUTEROWYCH

Ewa Stemposz  
Andrzej Jodłowski  
Alina Stasiecka

**Zarys metodyki  
wspierającej naukę  
projektowania systemów  
informacyjnych**



WYDAWNICTWO  
PWSTK

**Dr inż. Ewa Stemposz** prowadzi działalność naukowo-dydaktyczną w Polsko-Japońskiej Wyższej Szkole Technik Komputerowych w Warszawie, w Katedrze Inżynierii Oprogramowania.

**Dr inż. Andrzej Jodłowski** jest adiunktem w Katedrze Zastosowań Informatyki (w Zakładzie Systemów Rozproszonych) w Szkole Głównej Gospodarstwa Wiejskiego w Warszawie.

**Mgr inż. Alina Stasiecka** pracuje na stanowisku asystenta w Katedrze Inżynierii Oprogramowania w Polsko-Japońskiej Wyższej Szkole Technik Komputerowych w Warszawie. Ewa Stemposz, Andrzej Jodłowski i Alina Stasiecka prowadzą wspólnie działalność badawczą i dydaktyczną zajmując się szeroko pojętą problematyką inżynierii oprogramowania. Ich zainteresowania badawcze dotyczą w szczególności dziedzin takich jak: obiektość, modelowanie pojęciowe, modelowanie biznesowe, analiza i projektowanie systemów informatycznych, nauczanie zdalne, jakość w procesach nauczania, systemy zarządzania bazami danych, zarządzanie projektami informatycznymi. Wyniki badań z wyżej wymienionych dziedzin publikują w kraju i zagranicą w postaci monografii, artykułów w czasopismach i artykułów konferencyjnych.

Publikacja została przygotowana z myślą o osobach interesujących się modelowaniem i/lub projektowaniem, a w szczególności dla tych, którzy zajmują się analizą wymagań na systemy informacyjne i aktualnie wykorzystują albo planują wykorzystywać w przyszłości popularny obiektowy graficzny język UML. Opracowanie to może być również użyteczne dla osób prowadzących działalność związaną z modelowaniem i/lub projektowaniem, w ramach innych dziedzin problemowych. Przez osoby interesujące się rozumiane są tu zarówno osoby uczące się, nauczające, jak i osoby podejmujące ten rodzaj aktywności w życiu zawodowym. Uogólniając, zaprezentowana w rozdziale pierwszym publikacji metodyka, zarówno jeśli chodzi o schemat struktury zasobów, jak i określenie sposobów postępowania w trakcie nauczania, może być wykorzystana w nauczaniu w innych dziedzinach niekoniecznie związanych z informatyką czy z symulowaniem środowiska, do nauczania tzw. umiejętności miękkich, za jakie można uznać modelowanie/projektowanie. Publikacja może stanowić także pomoc dydaktyczną do przedmiotów związanych z analizą i projektowaniem systemów informacyjnych. W aspekcie tego przeznaczenia, szczególnie przydatne są rozdziały poświęcone jedenastu zadaniom, dla których dołączono przykładowe rozwiązania, omówienia błędów oraz opisy zdań w aspekcie ich użyteczności w procesie nauczania. Innym ważnym celem, ze względu na powszechnie wykorzystanie UML w organizacjach związanych z produkcją oprogramowania, było użycie w zadaniach jak największej liczby rodzajów diagramów dostarczanych przez ten język. Przeważająca większość popularnych pozycji wydawniczych o UML bazuje na przykładach, które są albo na zbyt wysokim poziomie abstrakcji, albo odnoszą się do bardzo ograniczonego wycinka dziedziny problemowej. Dotyczy to nie tylko informacji związanej z modelowaniem i/lub projektowaniem w ogóle, ale także z wykorzystywaniem konstrukcji UML. Ostatnim z założonych celów była próba zilustrowania sposobu budowy oprogramowania z wykorzystaniem najlepszych praktyk zidentyfikowanych na potrzeby inżynierii oprogramowania, czyli budową oprogramowania w oparciu o model iteracyjno-przyrostowy, na bazie architektury opartej na komponentach z wykorzystaniem modelowania wizualnego.



POLSKO-JAPOŃSKA  
WYŻSZA SZKOŁA  
TECHNIK KOMPUTEROWYCH

Ewa Stemposz  
Andrzej Jodłowski  
Alina Stasiecka

# Zarys metodyki wspierającej naukę projektowania systemów informacyjnych



Wydawnictwo  
PWSTK

© Copyright by Wydawnictwo PJWSTK  
Warszawa 2013

© Copyright by Ewa Stemposz, Andrzej Jodłowski, Alina Stasiecka  
Warszawa 2013

Wszystkie nazwy produktów są zastrzeżonymi nazwami handlowymi lub znakami towarowymi odpowiednich firm.

Książki w całości lub w części nie wolno powielać ani przekazywać w żaden sposób, nawet za pomocą nośników mechanicznych i elektronicznych (np. zapis magnetyczny) bez uzyskania pisemnej zgody Wydawnictwa.

**Edytor**

prof. zw. dr hab. Leonard Bolc

**Recenzent**

dr hab. Agnieszka Mykowiecka

**Korekta**

Anna Bittner

**Komputerowy skład tekstu**

Andrzej Jodłowski, Alina Stasiecka, Ewa Stemposz

**Projekt okładki**

Rafał Małyk

**Wydawnictwo**

**Polsko-Japońskiej Wyższej Szkoły Technik Komputerowych**

ul. Koszykowa 86, 02-008 Warszawa

tel. +48 22 58-44-526, fax +48 22 58-44-503

e-mail: oficyna@pjwstk.edu.pl

ISBN 978-83-63103-39-2

e-ISBN 978-83-63103-66-8

*Pamięci Profesora Leonarda Bolca*

# **Spis treści**

<b>Rozdział 1 Zarys metodyki wspierającej naukę projektowania systemów informacyjnych .....</b>	3
<i>Ewa Stemposz</i>	
<b>Rozdział 2 Fundacja „UFO” .....</b>	61
<i>Ewa Stemposz, Alina Stasiecka, Andrzej Jodłowski</i>	
<b>Rozdział 3 Wytwórnia porcelany stołowej .....</b>	141
<i>Ewa Stemposz, Alina Stasiecka</i>	
<b>Rozdział 4 Dom aukcyjny „Stare auta” .....</b>	181
<i>Ewa Stemposz, Andrzej Jodłowski</i>	
<b>Rozdział 5 System „Najlepsze gry” .....</b>	217
<i>Ewa Stemposz, Andrzej Jodłowski</i>	
<b>Rozdział 6 Muzeum .....</b>	251
<i>Ewa Stemposz, Andrzej Jodłowski</i>	
<b>Rozdział 7 Kwatery do wynajęcia .....</b>	279
<i>Ewa Stemposz, Andrzej Jodłowski</i>	
<b>Rozdział 8 Osiedlowy klub kartingowy .....</b>	321
<i>Ewa Stemposz, Andrzej Jodłowski</i>	
<b>Rozdział 9 Internetowa książka kucharska .....</b>	369
<i>Ewa Stemposz, Andrzej Jodłowski</i>	
<b>Rozdział 10 Klinika jednego dnia .....</b>	415
<i>Ewa Stemposz, Andrzej Jodłowski</i>	
<b>Rozdział 11 Hurtownia .....</b>	461
<i>Ewa Stemposz, Alina Stasiecka, Andrzej Jodłowski</i>	
<b>Rozdział 12 Klub judo .....</b>	525
<i>Ewa Stemposz, Alina Stasiecka, Andrzej Jodłowski</i>	
<b>Bibliografia .....</b>	557

## Spis treści rozdziału

Rozdział 1 Zarys metodyki wspierającej naukę projektowania systemów informacyjnych.....	4
1.1 Wprowadzenie .....	4
1.2 Model efektywnego nauczania .....	5
1.2.1 Podział procesu dydaktycznego na etapy .....	6
1.2.2 Struktura zasobów wykorzystywanych w metodyce .....	8
1.2.3 Aktywności wyróżnione w metodyce .....	12
1.2.4 Aktywność złożona: Zarządzaj grupą tematyczną.....	13
1.3 Grupy tematyczne i tematy .....	15
1.4 Zgrupowane omówienia najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	21
1.4.1 Omówienia błędów dla grupy: <i>Wprowadzenie do przedmiotu</i> .....	22
1.4.2 Omówienia błędów dla grupy: <i>Analiza funkcjonalna</i> .....	24
1.4.3 Omówienia błędów dla grupy: <i>Analiza strukturalna</i> .....	28
1.4.4 Omówienia błędów dla grupy: <i>Analiza dynamiczna</i> .....	35
1.5 Polecenia proste .....	37
1.5.1 Polecenia proste dla grupy: <i>Wprowadzenie do przedmiotu</i> .....	37
1.5.2 Polecenia proste dla grupy: <i>Analiza funkcjonalna</i> .....	38
1.5.3 Polecenia proste dla grupy: <i>Analiza strukturalna</i> .....	38
1.5.4 Polecenia proste dla grupy: <i>Analiza dynamiczna</i> .....	42
1.6 Szablony służące do tworzenia opisów zadań projektowo/problemowych .....	43
1.7 Metryki do oceny jakości rozwiązań .....	46
1.7.1 Metryki do oceny jakości rozwiązań w grupie: <i>Analiza funkcjonalna</i> ....	47
1.7.2 Metryka do oceny jakości rozwiązań w grupie: <i>Analiza strukturalna</i> .....	50
1.7.3 Metryka do oceny jakości rozwiązań w grupie: <i>Analiza dynamiczna</i> ....	53
1.8 Wzorzec dokumentacji projektowej .....	54
1.9 Podsumowanie.....	59

## Rozdział 1

# Zarys metodyki wspierającej naukę projektowania systemów informacyjnych

*Ewa Stemposz*

## 1.1 Wprowadzenie

Nauczanie modelowania/projektowania, niezależnie od dziedziny, której dotyczy, jest zadaniem trudnym i wydaje się, że bezpośredni kontakt osoby uczącej się z osobą nauczającą jest w tym wypadku nie do przecenienia. Wieloletnie doświadczenie autorów, nabycie dzięki prowadzeniu przedmiotów związanych z analizą i projektowaniem systemów informacyjnych, potwierdzałoby taką obserwację. Nauczane były osoby studiujące na różnych kierunkach i o różnym doświadczeniu zawodowym: studenci studiów informatycznych inżynierskich dziennych (zazwyczaj bez żadnego jeszcze doświadczenia), studenci studiów informatycznych zaocznych oraz podyplomowych (osoby pracujące), studenci studiów informatycznych internetowych (zarówno inżynierskich, jak i podyplomowych, zazwyczaj z pewnym, chociaż niekoniecznie związanym z projektowaniem doświadczeniem). Analiza systemowa, bazując na pojęciach z dziedziny problemowej, z założenia nie powinna być związana z żadnym środowiskiem implementacji, dlatego zdarzało się też, że na studiach podyplomowych uczone były osoby posiadające wykształcenie w dyscyplinach innych niż informatyka, jak na przykład prawo, zarządzanie itd.

Wszędzie tam, gdzie kontakt osoby uczącej z osobą nauczającą był bezpośredni i miał miejsce częściej, efekty były bardziej widoczne. Najlepsze wyniki uzyskiwano na kierunkach informatycznych, szczególnie na kursach prowadzonych w trybie stacjonarnym. Zdecydowanie gorsze były osiągnięcia uczestników studiów internetowych, gdzie największą rolę odgrywa zazwyczaj motywacja osoby uczącej się, ale niemałe znaczenie ma też jakość materiałów przeznaczonych do pracy indywidualnej, jakość zarówno w aspekcie merytorycznym i technicznym, jak i w aspekcie poprawnej konstrukcji dydaktycznej. Poprawna konstrukcja dydaktyczna materiałów jest znacząco istotna szczególnie wtedy, gdy kontakt uczącego się z nauczycielem jest utrudniony, stąd materiał dostarczany do samodzielnego pracy powinien być wyjątkowo starannie przemyślany pod względem wspierania (a czasami i narzucania) prawidłowych procesów dydaktycznych [29].

Niniejszy rozdział prezentuje zarys metodyki wspierającej naukę modelowania/projektowania, niezależnie od docelowej dziedziny problemowej. Nie mniej jednak, badania były przeprowadzane dla przedmiotu poświęconego

nauczaniu projektowania systemów informacyjnych z wykorzystaniem języka UML [35-50].

Metodykę oparto na modelu efektywnego nauczania [9, 29]. Zostały zidentyfikowane rodzaje zasobów niezbędne do realizowania procesu dydaktycznego, w tym: grupy tematyczne, tematy, ścieżki dydaktyczne oraz materiały dydaktyczne. Wśród materiałów dydaktycznych wyróżniono takie jak: omówienia błędów najczęściej popełnianych przez osoby uczące się w trakcie prac realizowanych przez nich samodzielnie, listę przykładowych poleceń prostych obejmujących zasięgiem prezentowane treści, propozycję szablonu służącego do tworzenia opisu zadań projektowo/problemowych, propozycję zbiorów miar dla szacowania jakości artefaktów, takich jak: diagramy przypadków użycia, scenariusze przypadków użycia, diagramy klas, diagramy dynamiczne. Określenie zbiorów miar stanowi próbę ujednolicenia sposobów szacowania osiągnięć osób nauczanych/uczących się, a także próbę dostarczenia metod do oceny jakości procesów nauczania.

W publikacji zamieszczono także przykładowe rozwiązania dla jedenastu zadań projektowo/problemowych (rozdziały 2-12), łącznie z ich opisami zawierającymi: określenie stopnia trudności dla każdego zadania traktowanego jako całość, określenie stopnia trudności zadania w obrębie wybranej grupy tematycznej oraz określenie użyteczności zadania zarówno w grupie tematycznej, jak i na etapie nauczania, zgodnym z modelem efektywnego nauczania.

Ponadto w rozdziale zawarto też propozycję wzorca dokumentacji dla projektu, z założenia realizowanego przez osoby nauczane/uczące się samodzielnie, w oparciu o dostarczone przez siebie wymagania na system o wielkości od dziesięciu do piętnastu klas. Rozwiązywanie jednego z przykładowych zadań, jest nim zadanie *Klub judo* (rozdział 12), stanowi ilustrację wykorzystania tego wzorca.

## 1.2 Model efektywnego nauczania

W badaniach przeprowadzonych przez autorów i poświęconych identyfikowaniu pożądanych własności materiałów dydaktycznych, pożądanych w aspekcie wspierania osób nauczanych/uczących się w trakcie nauki (opisane w [15-34]), wzięto pod uwagę rezultaty prac dydaktyków nad metodami nauczania [1-14]. Dla nauczania tradycyjnego, dydaktycy wyróżnili dwa podstawowe rodzaje podejść do tego procesu: *nauczanie podające* i *nauczanie problemowe*, gdzie pierwsze z nich bazuje wyłącznie na przekazywaniu informacji, natomiast w drugim, jako podstawowe zadanie, przyjęto skupienie uwagi na procesie budowania wiedzy u osoby nauczanej/uczącej się. Z prac dydaktyków wynikało także, że środowisko, które uwzględnia oba ww. podejścia, jest środowiskiem ułatwiającym i zachęcającym do uczenia się. Prowadząc badania, autorzy poszukiwali takiej struktury dla materiałów dydaktycznych, która opierała by się na zbalansowaniu obu zasad, co ma szczególnie znaczenie dla nauczania na odległość, gdzie praca osoby

uczącej się polega głównie na samodzielnym studiowaniu, z niewielkim wsparciem ze strony nauczyciela. Badania efektywności opisanych w literaturze form nauczania, przeprowadzono dla różnych modeli dydaktycznych. W efekcie, jako najbardziej odpowiedni dla realizowania założonych celów, wybrano *model efektywnego nauczania*. Model ten wydawał się być najlepszy do symulowania środowiska do nauczania tzw. *umiejętności miękkich*, za jakie można uznać modelowanie/projektowanie. Jedną z cech wyróżniających model efektywnego nauczania spośród innych było usystematyzowanie procesu nauczania, poprzez wyróżnienie w nim etapów/podetapów łącznie z określeniem dla każdego z nich aktywności, które powinny być realizowane w kolejnych krokach. Ponadto, w modelu efektywnego nauczania bardzo silnie jest akcentowana konieczność wykorzystywania różnego rodzaju materiałów dydaktycznych, m.in. takie jak pytania/polecenia proste (sprawdzające efektywność procesu nabywania wiedzy), a także zadania typu projektowo/problemowego (wykorzystywane do oceny stopnia zrozumienia problemów), co odpowiadało założonym celom i było zgodne z definicją proponowaną przez specjalistów od Instructional Design, gdzie Instructional Design jest wyspecjalizowaną dziedziną dydaktyki. Instructional Design zostało zdefiniowane przez specjalistów jako „*systematyczny i refleksyjny proces przekładania zasad i metod nauczania/uczenia się na materiały instruktażowe, ćwiczenia, zasoby informacyjne i ewaluacyjne*” [11, 30, 31].

Wyniki badań autorów przeprowadzanych dla nauczania na odległość skutkowały też systematycznym wprowadzaniem modyfikacji do procesów dydaktycznych realizowanych w trakcie nauczania stacjonarnego.

### **1.2.1 Podział procesu dydaktycznego na etapy**

W procesie dydaktycznym przeprowadzanym zgodnie z modelem efektywnego nauczania wyróżniono cztery podstawowe etapy (patrz Rys. 1-1), z których każdy obejmuje pewien określony zestaw kolejno realizowanych czynności:

- Etap 1: *Przygotowanie do nauki*

Oznacza czas przeznaczony na przygotowanie osoby nauczanej/uczącej się do efektywnej pracy w okresie następnym (na kolejnym etapie); przygotowanie do nauki powinno obejmować mniej więcej 10% całości kursu. W skład głównych aktywności realizowanych na tym etapie wchodzą takie jak:

- streszczenie wiedzy (która ma być nabywana w trakcie realizowania kursu) wraz ze wskazaniem na jej kluczowe elementy,
- krótkie omówienie najważniejszych/kluczowych elementów,
- motywowanie do zainteresowania się przekazywaną wiedzą,
- określenie celu dydaktycznego.

- Etap 2: *Nauka*

Określa okres poświęcony nauczaniu/uczeniu się; proces nabywania wiedzy ma być oparty zarówno o podejście podające, jak i podejście problemowe; etap nauki powinien obejmować mniej więcej 65% całości kursu. Wyróżnione zostały tu aktywności takie jak:

- przekazanie wiedzy bazowej niezbędnej dla realizacji następnego kroku,
- nabycie nowej wiedzy,
- dostarczenie przykładów umożliwiających zastosowanie nabyczej wiedzy.

- Etap 3: *Podsumowanie dotychczasowych działań*

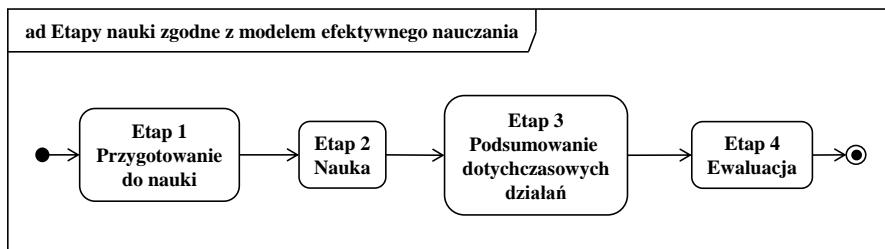
Związały jest z przeglądem tego, co podlegało nauce na poprzednim etapie; podsumowanie powinno obejmować mniej więcej 10% całości kursu. Do podstawowych aktywności realizowanych na tym etapie należą:

- streszczenie i przypomnienie nabyczej wiedzy,
- wskazanie na możliwy transfer tej wiedzy do nowego kontekstu,
- przegląd definicji kluczowych pojęć,
- przegląd literatury związanej tematycznie z nabycią wiedzą.

- Etap 4: *Ewaluacja*

Dotyczy okresu przeznaczonego na ocenianie; ewaluacja powinna obejmować mniej więcej 15 % całości kursu. Do podstawowych aktywności wyróżnionych na tym etapie należą:

- dokonanie oceny/samooceny nabyczej wiedzy,
- tworzenie rozwiązań do zadań projektowo/problemowych,
- realizacja sprzężenia zwrotnego z osobą nauczającą dla konsultowania rozwiązań do zadań projektowo/problemowych.



Rys. 1-1 Etapy nauki zgodne z modelem efektywnego nauczania

Etap ewaluacji dotyczy m.in. szacowania osiągnięć osób nauczanych/uczących się, takich jak na przykład: ocenianie stopnia zapamiętania przekazywanych w trakcie nauczania treści czy oceniania stopnia rozumienia problemów. Nie mniej jednak etap ewaluacji przede wszystkim jest przeznaczony na analizę procesów nauczania/uczenia się, tak aby w trybie stacjonarnym służyć pomocą osobie nauczającej w celu modyfikowania dydaktycznych elementów procesu, a w trybie

pracy na odległość ma dodatkowo wspomagać osobę uczącą się w samodzielnym nabywaniu wiedzy.

### **1.2.2 Struktura zasobów wykorzystywanych w metodyce**

Na Rys. 1-2 zaprezentowano organizację elementów strukturalnych, niezbędnych dla przetwarzania informacji w metodyce wspierającej nauczanie modelowania/projektowania.

Metodykę oparto na modelu efektywnego nauczania. Zidentyfikowane zostały następujące rodzaje zasobów dydaktycznych: ścieżki dydaktyczne, grupy tematyczne (łącznie z obejmowanymi przez nie tematami uporządkowanymi zgodnie z sugerowaną kolejnością nauczania), tematy oraz materiały dydaktyczne; podział ten nie jest kompletny (patrz Rys. 1-3).

Zasób dydaktyczny ma być opisywany własnościami takimi jak: unikatowa nazwa, charakterystyka, cel dydaktyczny oraz zakres odpowiedzialności. Zasób dydaktyczny może być zarówno agregatem, jak i elementem składowym innego zasobu; ma być określone uporządkowanie zasobów składowych w agregacie. Użyteczność tej relacji jest oceniana w cztero-stopniowej skali: *niska, średnia, wysoka i b. wysoka*.

Informacje zawarte w zasobie mogą być rozszerzane o informacje zawarte w innych zasobach. Dla każdego takiego rozszerzenia możliwe jest określenie stopnia jego użyteczności, w skali jak poprzednio.

Ścieżki dydaktyczne mogą być zarówno proste, jak złożone. Wiadomo, która ze ścieżek jest podstawowa, jest nią ta określona przez osobę odpowiedzialną za przedmiot. Tylko ścieżka złożona może (ale nie musi) obejmować całość przedmiotu. Dla każdej ścieżki ma być wiadomo, które grupy tematyczne i tematy obejmuje; mogą to być tylko tematy z grup tematycznych, które zostały przypisane do ścieżki.

Budowanie ścieżki dydaktycznej rozpoczyna się od wyboru grup tematycznych, które ma obejmować na danym etapie nauczania, zgodnym z modelem efektywnego nauczania. Każda ze zdefiniowanych ścieżek musi być powiązana z co najmniej jedną grupą tematyczną; podobnie każda grupa tematyczna musi mieć określony co najmniej jeden skojarzony z nią temat. Zarówno grupy tematyczne, jak i tematy mogą być powtarzane w obrębie ścieżki. Kolejność tematów objętych daną ścieżką musi być zgodna zarówno z kolejnością ustaloną wcześniej dla grup tematycznych (określoną na potrzeby danej ścieżki), jak i z kolejnością tematów w grupie, zdefiniowaną przez osobę odpowiedzialną za przedmiot.

Nauczanie jest prowadzone w oparciu o posiadane zasoby dydaktyczne. Dla każdego z zasobów ma być znana jego użyteczność na każdym z etapów modelu efektywnego nauczania (oceniana w skali jak wyżej), dla każdego z przedmiotów (może być ich więcej niż jeden), które dany zasób wykorzystują.

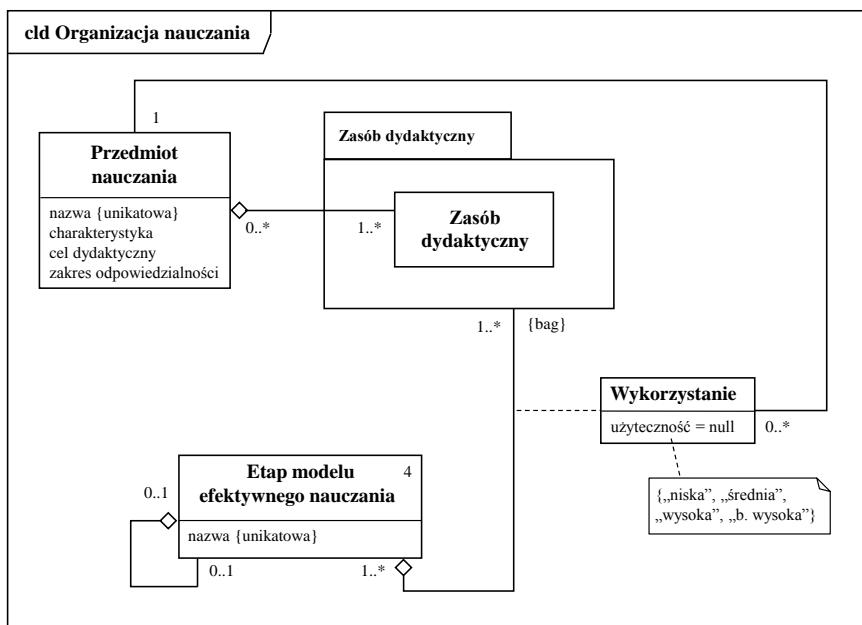
Podział materiałów dydaktycznych, dokonany na przykładzie przedmiotu poświęconego nauce projektowania systemów informacyjnych, zaprezentowano na Rys. 1-4.

Dla każdego z materiałów dydaktycznych ma być przechowywana informacja o jego związkach merytorycznych z innymi materiałami oraz o wszystkich poprzednich jego wersjach.

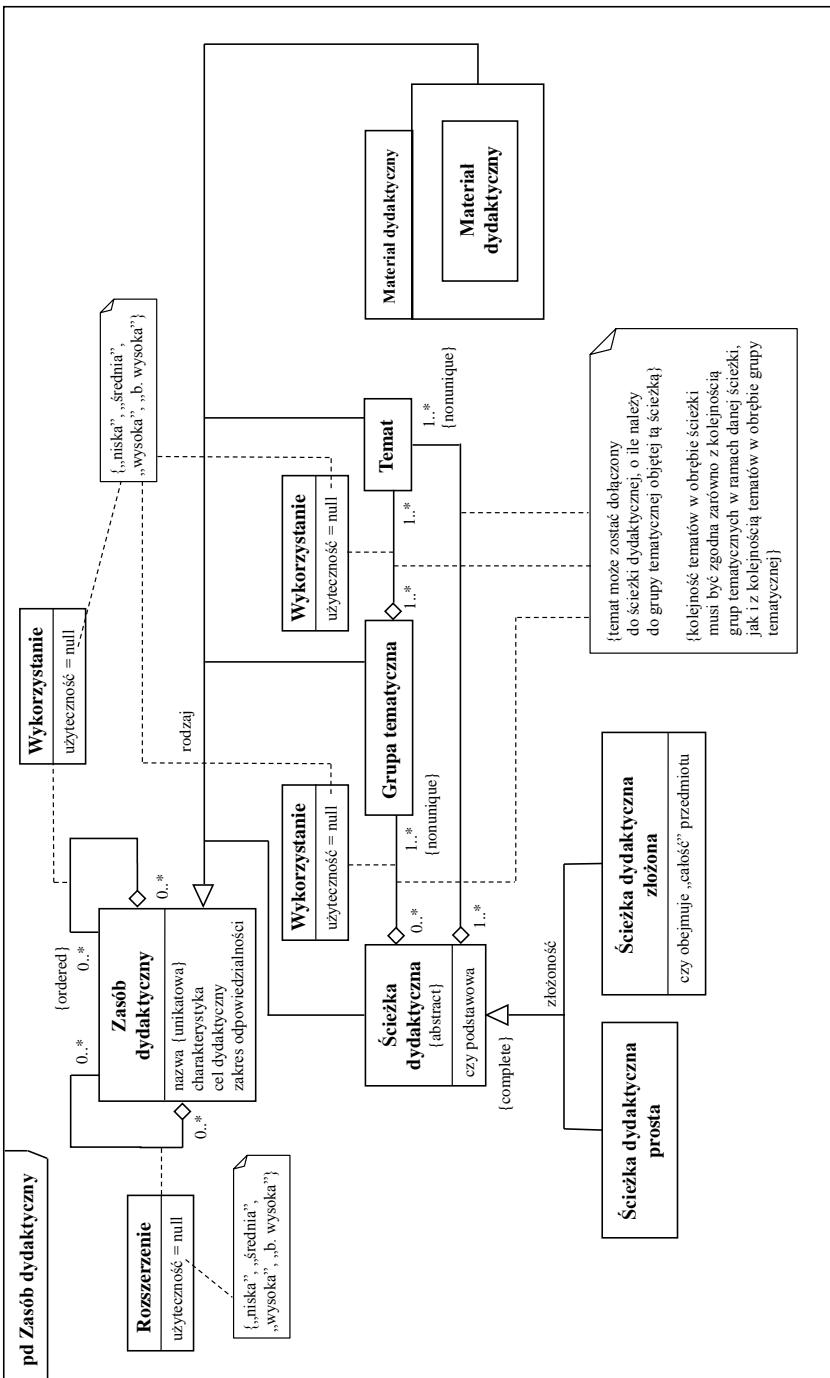
Materiały zostały podzielone ze względu na dwa aspekty: *grupa tematyczna* i *złożoność*. W wybranym do zilustrowania metodyki przedmiocie wyróżniono cztery grupy tematyczne: *Wprowadzenie do przedmiotu*, *Analiza funkcjonalna*, *Analiza strukturalna* oraz *Analiza dynamiczna*. Podział na grupy tematyczne nie jest rozłączny.

W drugim aspekcie podziału materiałów dydaktycznych (*złożoność*) wyróżniono dwie grupy: grupę materiałów prostych i grupę materiałów złożonych; materiały obu tych rodzajów mogą być elementami składowymi innych materiałów złożonych.

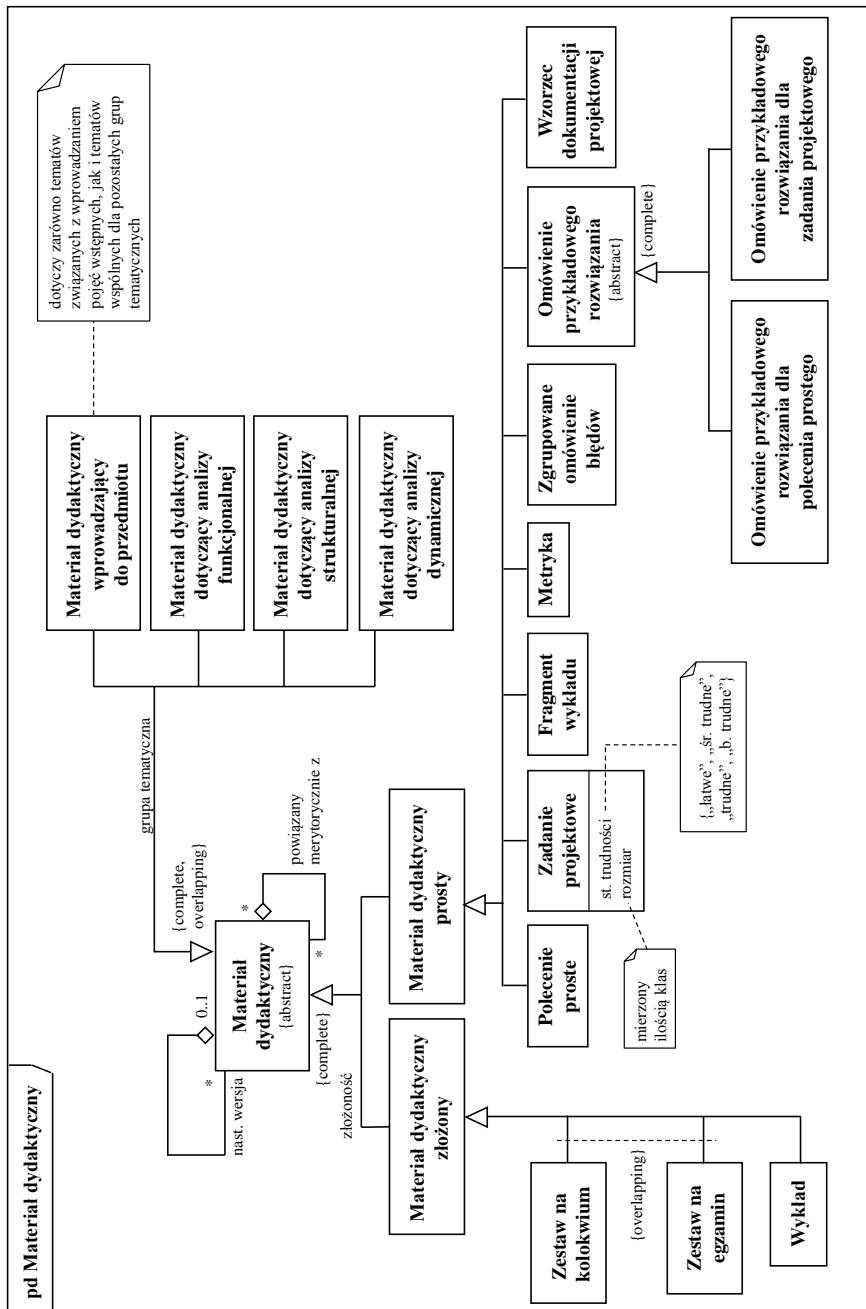
Do materiałów złożonych włączono: zestawy na kolokwium i zestawy na egzamin, które mogą być wykorzystywane w obydwu rolach, a także wykład traktowany jako całość.



Rys. 1-2 Organizacja elementów strukturalnych wyróżnionych w metodyce



Rys. 1-3 Struktura zasobów wykorzystywanych w metodyce

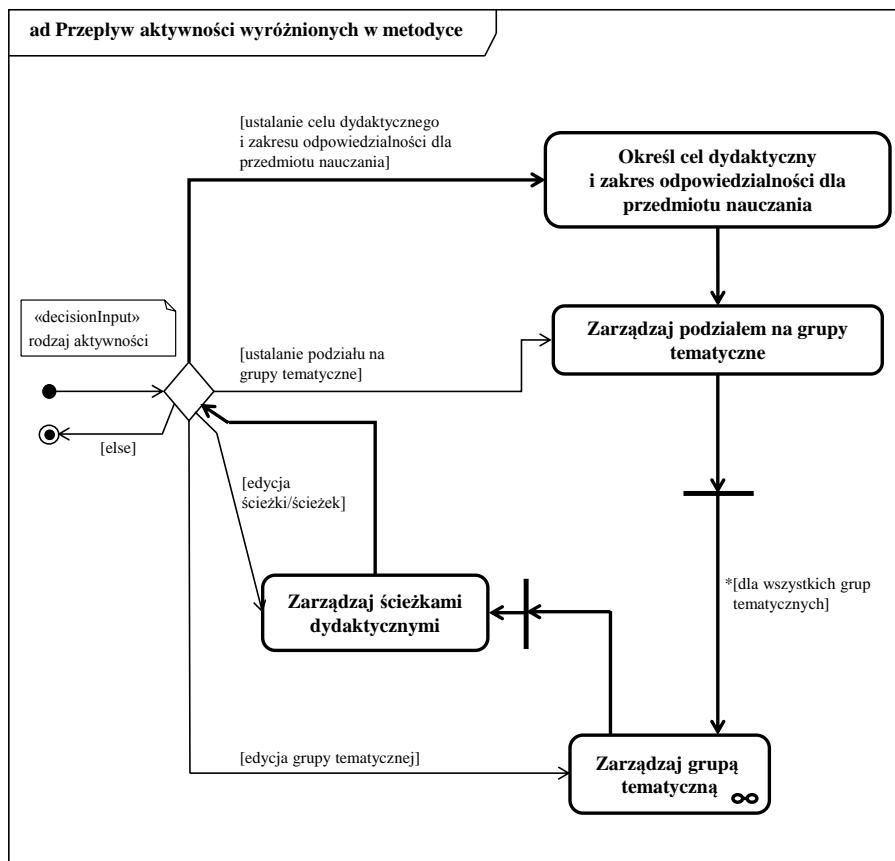


Rys. 1-4 Struktura materiałów dydaktycznych na przykładzie przedmiotu poświęconego nauce projektowania systemów informacyjnych

Z kolei dla materiałów prostych wyróżniono rodzaje takie jak: polecenia proste, zadania projektowo/problemowe, fragmenty wykładów, metryki (zbiory miar), zgrupowane omówienia błędów, omówienia przykładowych rozwiązań oraz wzorzec dokumentacji projektowej. Podział materiałów prostych jest rozłączny. Oba podziały, tzn. podział materiałów prostych i podział materiałów złożonych, nie są kompletne. Oznacza to, że stosownie do dalszych potrzeb mogą być wyróżniane kolejne rodzaje materiałów z obu grup, użyteczne w nauczaniu przedmiotu.

### 1.2.3 Aktywności wyróżnione w metodyce

Na Rys. 1-5 zaprezentowano przepływ aktywności w metodyce, aktywności zostały wyróżnione odpowiednio do zidentyfikowanych w punkcie 1.2.2 rodzajów zasobów.



Rys. 1-5 Przepływ aktywności wyróżnionych w metodyce

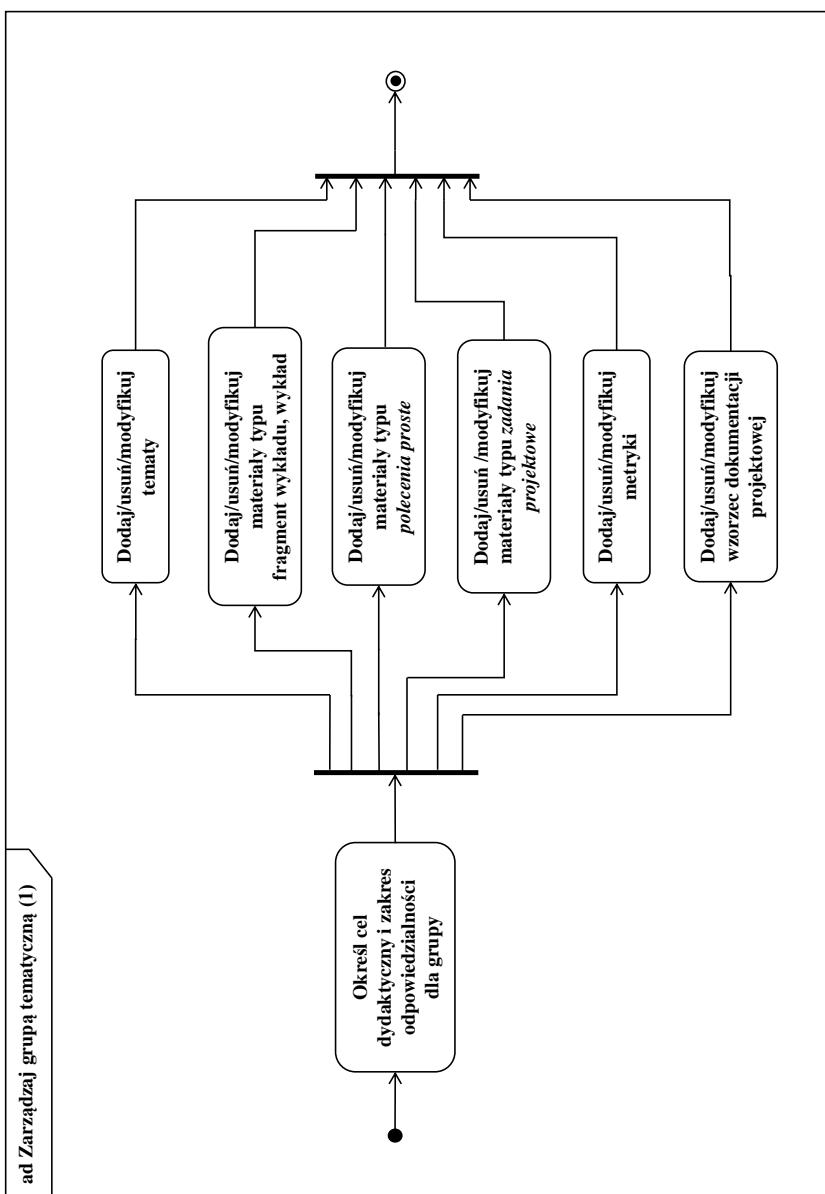
Do diagramu wprowadzono aktywność złożoną *Zarządzaj grupą tematyczną* (omówioną w 1.2.4). Przez *zarządzaj* rozumiane są (w całości publikacji) operacje typu CRUD.

Przepływ aktywności wyróżniony przez pogrubienie ścieżek przepływu sterowania wyznacza kolejność kroków realizowanych w trakcie początkowej organizacji nauczania danego przedmiotu.

#### **1.2.4 Aktywność złożona: Zarządzaj grupą tematyczną**

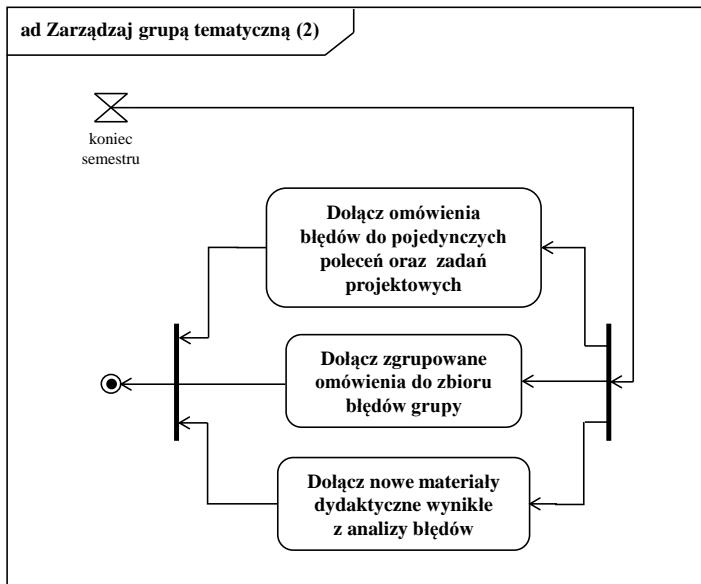
Dla aktywności złożonej *Zarządzaj grupą tematyczną*, zaprezentowanej na Rys. 1-6 i Rys. 1-7, wyróżniono kroki, takie jak:

- określ cel dydaktyczny i zakres odpowiedzialności (dla danej grupy tematycznej),
- zarządzaj tematami,
- przygotuj materiał teoretyczny, są nim na przykład fragmenty wykładów czy wykład jako całość,
- zarządzaj materiałami kontrolnymi typu polecenia proste,
- zarządzaj materiałami kontrolnymi typu zadania projektowo/problemowe, specyfikowanymi w postaci spójnych zbiorów wymagań na systemy o wielkości mniej więcej od dziesięciu do piętnastu klas, dla określonej dziedziny problemowej,
- zarządzaj metrykami (zbiorami miar) użytecznymi zarówno w szacowaniu osiągnięć osoby nauczanej/uczącej się, jak i w pomiarze jakości nauczania w obrębie danej grupy tematycznej; obie te aktywności są realizowane w oparciu o przygotowane na potrzeby przedmiotu polecenia proste oraz zadania projektowo/problemowe,
- zarządzaj wzorcem dokumentacji projektowej,
- przetestuj osoby nauczane/uczące się, wykorzystując zbiór przygotowanych materiałów kontrolnych,
- identyfikuj błędy,
- dołącz analizę błędów do poleceń/zadań obejmujących zakresem konkretne błędy,
- dołącz analizę błędów (w postaci zgrupowanej) do grupy tematycznej,
- dołącz do zbioru materiałów pomocniczych dodatkowy materiał teoretyczny i/lub polecenia oraz zadania kontrolne uwzględniające analizę błędów popełnianych przez osoby nauczane/uczące się,



Rys. 1-6 Schemat zarządzania grupą tematyczną (diagram pierwszy)

- modyfikuj cel dydaktyczny i zakres, dla danej grupy tematycznej, a także dla innych oraz dla całości przedmiotu nauczania.



Rys. 1-7 Schemat zarządzania grupą tematyczną (diagram drugi)

### 1.3 Grupy tematyczne i tematy

Podział na grupy tematyczne, określenie celu dydaktycznego i zakresu odpowiedzialności dla każdej z grup wyróżnionych w punkcie 1.2.2 została przedstawiony w Tab. 1-1. Pominięto element opisu grupy tematycznej, nazwany *charakterystyka* (umieszczony na diagramie na Rys. 1-3). W Tab. 1-2, Tab. 1-3, Tab. 1-4 oraz Tab. 1-5 przedstawiono listy tematów w obrębie każdej z grup tematycznych oddzielnie. Uwzględniono głównie te tematy, które zostały wykorzystane w jedenastu zadaniach projektowych prezentowanych w kolejnych rozdziałach publikacji (od 2 do 12). Dla każdego tematu, podobnie jak dla grup tematycznych, określono cel dydaktyczny i obejmowany przez dany temat zakres. Zaproponowano uporządkowanie tematów w obrębie każdej z grup, odpowiednio do aktualnie realizowanego toku nauczania.

Tab. 1-1 Podział na grupy tematyczne

Grupa tematyczna	Cel dydaktyczny	Zakres odpowiedzialności
<i>Wprowadzenie do przedmiotu</i>	omówienie zarówno pojęć wprowadzających do przed-	– omówienie etapów i głównych aktywności wyróżnianych

	miotu, jak i pojęć wspólnych dla pozostałych grup tematycznych	w procesie wytwarzania oprogramowania  <ul style="list-style-type: none"> <li>– prezentacja rodzajów modeli i diagramów wytwarzanych w trakcie konstruowania oprogramowania; skupienie uwagi na diagramie pakietów</li> <li>– objaśnienie podstawowych pojęć wykorzystywanych w trakcie budowy modeli i diagramów</li> <li>– omówienie mechanizmów rozszerzalności</li> <li>– omówienie reguł nazewnictwa dla elementów składowych diagramów</li> <li>– omówienie struktury diagramów w aspekcie poprawy ich czytelności</li> </ul>
<i>Analiza funkcjonalna</i>	nabycie umiejętności prowadzenia analizy wymagań na system, o wielkości od 10 do 15 klas, w oparciu o zadany tekst wymagań z możliwością zastosowania dekompozycji poziomej (model opisywany jest przez więcej niż jeden diagram) i pionowej (gdy do modelu/diagramu zostały wprowadzone elementy zagnieżdżone)	<ul style="list-style-type: none"> <li>– określanie wymagań na system, w oparciu o zadany tekst wymagań</li> <li>– budowa modeli przypadków użycia</li> <li>– tworzenie słownika pojęć</li> <li>– konstruowanie scenariuszy dla wybranych przypadków użycia</li> <li>– dekomponowanie modelu/diagramu, na przykład z wykorzystaniem diagramów pakietów lub podprzypadków (czyli przypadków wynikłych z podziału przypadków z wyższego poziomu abstrakcji)</li> <li>– tworzenie planu iteracji w oparciu o model przypadków użycia</li> </ul>

<i>Analiza strukturalna</i>	<p>nabycie umiejętności konstruowania modeli/diagramów opisujących strukturę wybranego fragmentu dziedziny problemowej, w oparciu o zadany tekst wymagań, z możliwością zastosowania dekompozycji poziomej i pionowej</p>	<ul style="list-style-type: none"> <li>– konstruowanie schematów pojęciowych (dla warstwy biznesowej), w oparciu o zadany tekst wymagań</li> <li>– przejście ze schematu pojęciowego na schemat logyczny (projektowy) uwzględniający ograniczenia przykładowego środowiska implementacji</li> <li>– przejście ze schematu pojęciowego na schemat relacyjny (tylko dla encji biznesowych)</li> <li>– dekomponowanie modelu/diagramu, na przykład z wykorzystaniem diagramów pakietów</li> <li>– konstruowanie diagramu komponentów łącznie ze specyfikowaniem interfejsów</li> </ul>
<i>Analiza dynamiczna</i>	<p>nabycie umiejętności prowadzenia analizy dynamicznej w celu modyfikowania zawartości modeli/diagramów strukturalnych, z możliwością wprowadzania dekompozycji poziomej i pionowej</p>	<ul style="list-style-type: none"> <li>– przeprowadzenie analizy dynamicznej dla pojedynczych przypadków użycia (co najwyżej kilku, dla których zależności pomiędzy nimi wynikają z tekstu wymagań)</li> <li>– konstruowanie diagramów aktywności</li> <li>– konstruowanie maszyn stanowych, głównie behawioralnych</li> <li>– konstruowanie diagramów interakcji dla operacji realizowanych na schematach pojęciowych i projektowych</li> <li>– konstruowanie diagramów przeglądu interakcji</li> <li>– konstruowanie diagramów następstwa stanów</li> </ul>

		– wprowadzanie modyfikacji do modeli/diagramów strukturalnych w efekcie przeprowadzonej analizy dynamicznej
--	--	---

Tab. 1-2 Wprowadzenie do przedmiotu – tematy

<b>Wprowadzenie do przedmiotu</b>	
<b>Temat</b>	<b>Zakres odpowiedzialności; lista pojęć</b>
<i>proces wytwarzania oprogramowania</i>	– omówienie kolejnych etapów procesu wytwarzania oprogramowania w modelu wodospadowym – pojęcia: <i>model wodospadowy</i> , <i>model iteracyjno-przyrostowy</i>
<i>analiza a projektowanie</i>	– opis biznesu a środowisko implementacji – pojęcia: <i>model pojęciowy</i> , <i>model logiczny (projektowy)</i>
<i>model a diagram</i>	– omówienie rodzajów modeli wykorzystywanych w trakcie modelowania; rodzaje diagramów; <i>wyróżnik diagramu</i> w UML
<i>klasyfikatory a instancje</i>	– pojęcia: <i>klasyfikator</i> , <i>instancja klasyfikatora</i> , <i>relacja pomiędzy klasyfikatorami</i> ; przykłady wykorzystania klasyfikatorów i instancji
<i>mechanizmy rozszerzalności</i>	– pojęcia: <i>stereotyp</i> , <i>ograniczenie</i> , <i>komentarz</i>
<i>dekompozycja diagramów</i>	– dekompozycja diagramów z wykorzystaniem diagramów pakietów, podziału modelu na diagramy, zagnieżdżanie
<i>reguły nazewnictwa</i>	– omówienie reguł nazewnictwa dla bytów biznesowych oraz modelujących je elementów składowych diagramów
<i>poprawa percepcji diagramu/diagramów</i>	– pojęcia: <i>czytelność</i> , <i>reguły nazewnictwa</i>

Tab. 1-3 Analiza funkcjonalna – tematy

<b>Analiza funkcjonalna</b>	
<b>Temat</b>	<b>Zakres odpowiedzialności; lista pojęć</b>
<i>wymagania na system</i>	<ul style="list-style-type: none"> <li>– pojęcia: <i>analiza wymagań na system, wymaganie funkcjonalne, wymaganie niefunkcjonalne</i></li> <li>– określenie przykładowych wymagań funkcjonalnych oraz wymagań niefunkcjonalnych (łącznie ze specyfikacją metryk do ich pomiarów)</li> </ul>
<i>aktorzy</i>	<ul style="list-style-type: none"> <li>– pojęcia: <i>aktor, rodzaje aktorów</i> (osoby, organizacje, inne systemy, urządzenia zewnętrzne, ...), <i>hierarchia dla aktorów</i></li> <li>– <i>aktor abstrakcyjny</i></li> <li>– aktor <i>Podsystem czasu</i></li> </ul>
<i>przypadki użycia</i>	<ul style="list-style-type: none"> <li>– pojęcia: <i>przypadek użycia, interakcja aktora z przypadkiem użycia, relacja pomiędzy przypadkami użycia, blok ponownego użycia</i></li> </ul>
<i>słownik pojęć</i>	<ul style="list-style-type: none"> <li>– budowanie słownika pojęć w oparciu o opis fragmentu dziedziny problemowej</li> </ul>
<i>scenariusze</i>	<ul style="list-style-type: none"> <li>– budowanie scenariuszy dla wybranych wymagań funkcjonalnych</li> </ul>
<i>dekompozycja modelu/diagramu przypadków użycia</i>	<ul style="list-style-type: none"> <li>– pojęcia: <i>dzielenie przypadków na podprzypadki, poziomy abstrakcji w modelu przypadków użycia</i></li> <li>– pojęcia: <i>dekompozycja pozioma</i> (np. podział modelu na więcej niż jeden diagram), <i>dekompozycja pionowa</i> (zaginieżdżanie funkcji)</li> </ul>
<i>plan iteracji</i>	<ul style="list-style-type: none"> <li>– tworzenie przykładowego planu iteracji w oparciu o model przypadków użycia</li> </ul>

Tab. 1-4 Analiza strukturalna – tematy

<b>Analiza strukturalna</b>	
<b>Temat</b>	<b>Zakres odpowiedzialności; lista pojęć</b>
<i>klasa a obiekt</i>	– pojęcia: <i>klasa jako klasyfikator, obiekt jako wystąpienie klasyfikatora, inwarianty klasy</i>
<i>atrybuty klasy</i>	– pojęcia: <i>atrybut prosty, atrbut złożony, atrbut opcjonalny, atrbut powtarzalny, atrbut pochodny, atrbut klasowy</i>
<i>metody klasy</i>	– pojęcia: <i>metoda a procedura, sygnatura metody, ciało metody, metoda abstrakcyjna, metoda zaimplementowana, metoda obiektowa, metoda klasowa, przesłanianie metod, przeciążanie metod, polimorfizm metod, operacja a metoda i komunikat</i>
<i>struktury generalizacji-specjalizacji</i>	– pojęcia: <i>dziedziczenie klas, rodzaje dziedziczenia</i>
<i>klasa abstrakcyjna a klasa konkretna</i>	– pojęcia: <i>wystąpienie bezpośrednie klasy, wystąpienie pośrednie klasy, klasa abstrakcyjna, klasa konkretna</i>
<i>asocjacje</i>	– pojęcia: <i>asocjacja jako klasyfikator, powiązanie jako wystąpienie asocjacji, nazwa asocjacji a rola asocjacji, liczności asocjacji, klasa asocjacji, asocjacja skierowana, agregacja, kompozycja, asocjacja kwalifikowana, asocjacja rekursywna (wykorzystywana do budowania grafów)</i>
<i>analiza wartości</i>	– pojęcia: <i>wartość początkowa (domyślna), wartość pochodna, wartość graniczna</i>
<i>diagramy struktur złożonych</i>	– wykorzystanie diagramów dla modelowania atrybutów złożonych
<i>przejście na schemat projektowy</i>	– pojęcia: <i>ograniczenia środowiska implementacji, schemat pojęciowy, schemat projektowy</i>
<i>przejście na schemat relacyjny</i>	– pojęcia: <i>tabela a klasa i ekstensja klasy, obiekt a wiersz tabeli, klucze główne i klucze obce, zamiana atrybutów i metod, przejście dla dziedziczenia klas,</i>

	<i>zamiana asocjacji</i>
<i>diagram komponentów</i>	– budowanie przykładowego diagramu komponentów łącznie ze specyfikowaniem interfejsów

Tab. 1-5 Analiza dynamiczna – tematy

<b>Analiza dynamiczna</b>	
<b>Temat</b>	<b>Zakres odpowiedzialności; lista pojęć</b>
<i>diagramy aktywności</i>	– pojęcia: <i>aktywność prosta, aktywność złożona, romb decyzyjny, sztabka synchronizacyjna, etykieta przejścia</i>
<i>diagramy stanów</i>	– pojęcia: <i>stan, zdarzenie, przejście, rodzaje stanów, rodzaje zdarzeń, rodzaje przejść, struktury generalizacji - specjalizacji dla maszyn stanowych, maszyna stanowa behawioralna a maszyna protokołowa, stan początkowy i stan końcowy na diagramach aktywności i w maszynach stanowych</i>
<i>diagramy interakcji</i>	– pojęcia: <i>diagramy sekwencji, diagramy komunikacji, diagramy przeglądu interakcji, fragmenty wyodrębnione na diagramach sekwencji, przywoływanie wystąpień interakcji, diagramy następstwa stanów</i>
<i>podsumowanie analizy dynamicznej</i>	– wprowadzanie modyfikacji do modeli/diagramów strukturalnych w efekcie przeprowadzonej analizy dynamicznej

## **1.4 Zgrupowane omówienia najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się**

W rozdziale zamieszczono przykładowe zgrupowane omówienia tych błędów, które są najczęściej popełniane przez osoby nauczane/uczące się w trakcie realizowania przedmiotu poświęconego nauce projektowania systemów informacyjnych. Podobne omówienia można tworzyć dla dowolnego przedmiotu, nie tylko dla tych związanych z nauką modelowania/projektowania – jest to szczególnie użyteczne dla

przedmiotów realizowanych w nauczaniu na odległość. Omówienia zostały zaprezentowane w postaci list zawierających opisy błędów.

Listy błędów zostały podzielone na cztery części – zgodnie z podziałem na grupy tematyczne wyróżnione w punkcie 1.2.2 : *Wprowadzenie do przedmiotu*, *Analiza funkcjonalna*, *Analiza strukturalna* oraz *Analiza dynamiczna*.

### **1.4.1 Omówienia błędów dla grupy: *Wprowadzenie do przedmiotu***

*Dla przypomnienia:* Należy odróżniać pojęcia takie jak: *model* i *diagram*, które nie będąc synonimami, są jednak często wykorzystywane zamiennie:

- Model jest semantycznie spójnym, kompletnym opisem projektowanego systemu utworzonym z określonej perspektywy (funkcjonalnej, strukturalnej, dynamicznej, ...) na pewnym poziomie szczegółowości. „Pewien poziom szczegółowości” oznacza, że niektóre elementy systemu zostały ukryte, a inne wyeksponowane. Z kolei, fraza „kompletny opis” zaświdała, że żadna dodatkowa informacja nie jest potrzebna dla zrozumienia systemu z danej perspektywy, na danym poziomie szczegółowości. Pojedynczy model zazwyczaj nie wystarcza ani do zrozumienia wszystkich aspektów złożonego systemu jednocześnie ani do znalezienia odpowiedniego rozwiązania, zwykle potrzebujemy ich wiele, ale stanowiąc razem kompletny opis systemu, muszą być wzajemnie spójne i nie redundantne [39].
- Diagram służy do opisania modelu, dany model może być opisany przy pomocy wielu diagramów, zaś dany element modelu może pojawiać się na wielu diagramach opisujących ten model.

*Uwagi ogólne:*

1. *Często litery na diagramach są zbyt małe, rysunki niestaranne, źle zaplanowane i eksponujące niezbyt ważne elementy.* Wszystko to utrudnia percepcję, co jest sprzeczne z ideą wykorzystywania diagramów do polepszenia komunikacji pomiędzy uczestnikami projektu (uczestnikiem projektu jest również klient) w trakcie prac nad systemem. Poprawa percepcji jest uzyskiwana zgodnie z regułą, że „jeden obraz wart jest tysiąca słów”.
2. *Nazwy elementów diagramów powinny „przenosić” semantykę tych bytów/operacji/relacji biznesowych,* które zostały zidentyfikowane w danym fragmencie dziedziny problemowej.
3. *Często brakuje nagłówków i/lub ram dla diagramów.* Z kolei w nagłówkach brakuje wyróżników diagramów.

*Dla przypomnienia:* forma obramowana, wspomagająca zarówno podział złożonych elementów, jak i – dzięki nagłówkom z wyróżnikami określającymi rodzaj diagramu i nazwami diagramów – ułatwiająca wyszukiwanie zależności

pomiędzy różnymi diagramami, jest zdecydowanie bardziej użyteczna niż nieobramowana.

4. *Uwaga:* Warto dzielić model/diagram na części (dotyczy to wszystkich ich rodzajów), ponieważ im mniej elementów dany model/diagram zawiera, tym bardziej jest czytelny, co wydatnie wspomaga komunikowanie się osób. Ma to szczególne znaczenie dla dużych projektów.
5. *Dla większych systemów* warto jest podjąć próbę grupowania jego elementów składowych w pakiety. W takiej sytuacji należy skupić uwagę na podziale na pakiety o wysokiej kohezji i słabych wzajemnych sprzężeniach. Proces grupowania może zostać oparty o różne przesłanki (jak np. wspólna funkcjonalność), zawsze jednak należy kierować się zasadą, że dane zachowanie systemu powinno być implementowane tylko w jednym miejscu, dla uniknięcia problemów aktualizacyjnych, np. takich jak anomalie podczas wstawiania, usuwania czy modyfikowania.

*Uwaga:* pakiet może być wykorzystany dla diagramów każdego rodzaju.

*Błędy związane z wykorzystaniem mechanizmów rozszerzalności:*

1. *Brak jakichkolwiek ograniczeń czy komentarzy na diagramach.*
2. *Błędem jest mylenie oznaczeń ograniczeń i komentarzy* i nie zawsze jest to związane jedynie z nieznajomością notacji. W UML ograniczenia umieszczamy w nawiasach klamrowych (mogą być dodatkowo umieszczane w kopertach, ale nie muszą), a komentarze w kopertach bez nawiasów klamrowych. Ograniczenie jest warunkiem (*warunek wstępny*, *warunek końcowy*, *invariant*), który musi być spełniony, aby można było wykonać pewną operację. Ograniczenia są implementowane, dlatego zawsze należy określić metodę (oraz klasę), w której należałoby zawrzeć odpowiedni fragment kodu.

*Dla przypomnienia:* metody typu *sprawdź coś ...* nie służą do implementacji ograniczeń. Wynik sprawdzenia zostanie zwrócony do tej metody, której zadaniem jest właśnie implementacja. Z kolei, komentarze są wykorzystywane do bardziej szczegółowego objaśnienia pewnych elementów na diagramie, np. mogą informować o sposobie wyliczania dochodu miesięcznego (dla metody *wylicz zarobek miesięczny()*). Nie podlegają zamianie na kod.

3. *Uwaga:* w przeciwnieństwie do ograniczeń, których liczba pośrednio określa jakość diagramu, umieszczanie zbyt wielu komentarzy nie jest zalecane, ponieważ pogarszają percepcję diagramu. Zamiast wprowadzania komentarzy lepiej jest skupić uwagę na właściwym doborze nazw dla elementów diagramu, tak by uwypuklać ich istotę. W ten sposób powstają tak zwane samo-opisujące się diagramy.

### 1.4.2 Omówienia błędów dla grupy: *Analiza funkcjonalna*

*Uwaga:* Konstruowanie modelu funkcjonalnego należy w pierwszym kroku rozpocząć od zidentyfikowania potencjalnych użytkowników (aktorów) z otoczenia systemu traktowanego, jako całość (najwyższy poziom abstrakcji modelu) i od tworzenia diagramów przypadków obejmujących te usługi systemu, które są bezośrednio dostępne dla aktorów z jego otoczenia, biorąc pod uwagę możliwą ewolucję systemu w przyszłości. Zawsze też należy uwzględnić hierarchie dla aktorów i relacje pomiędzy przypadkami (o ile mają/mogłyby mieć miejsce). Warto jest również oznaczać aktora abstrakcyjnego.

*Podsumowując:* w budowie modelu przypadków użycia można wyróżnić kolejne etapy, takie jak:

- *Zidentyfikowanie potencjalnych użytkowników z otoczenia systemu traktowanego jako całość* (co można prezentować, wykorzystując diagram kontekstowy);
- *Określenie zbioru usług oferowanych przez system dla użytkowników wyróżnionych w pierwszym kroku.* Należy zbudować model przypadków użycia, składający się z jednego lub więcej diagramów. Model ten powinien zawierać zarówno wszystkie najważniejsze z perspektywy potencjalnych aktorów przypadki użycia (tzw. przypadki bazowe), jak i relacje występujące pomiędzy przypadkami bazowymi (o ile konieczność ich wprowadzenia wynika z tekstu wymagań);
- *Sporządzenie dokumentacji.* Dla każdego z przypadków bazowych wyróżnionych w kroku drugim, należy sporządzić dokumentację, która powinna zawierać scenariusze dla przepływu zdarzeń pomiędzy aktorami a systemem w trakcie wykonania każdego z przypadków; dla przypadków nietrywialnych należy dołączyć opisy algorytmów dla ich części wykonawczych (dostarczone np. w postaci pseudokodu lub diagramów aktywności);
- *Dokonanie podziału wyróżnionych przypadków na podprzypadki* w oparciu o dokumentację sporzązoną w poprzednim kroku. Podział przypadków jest realizowany w celu zmniejszania ich złożoności, identyfikacji bloków ponownego użycia oraz podziału modelu ze względu na poziomy abstrakcji.

*Uwaga:* Dla najwyższego poziomu abstrakcji, czyli dla diagramów konstruowanych z perspektywy aktorów z otoczenia systemu, nie skupiamy uwagi na podziale danego przypadku użycia na podprzypadki. Podprzypadki nie są istotne dla klienta zamawiającego system, natomiast ważnym jest prawidłowe określenie zbioru usług, które system dostarczałby przyszłym użytkownikom. Podział przypadków na podprzypadki ma znaczenie dla zespołu, który ten system buduje, ponieważ pozwala na zmniejszenie złożoności. Złożony przypadek, podzielony na części, jest łatwiejszy do „ogarnięcia”. Proces dekompozycji przypadków ma też drugi aspekt, oprócz zmniejszania złożoności. Próbujemy zidentyfikować elementy nadające się do

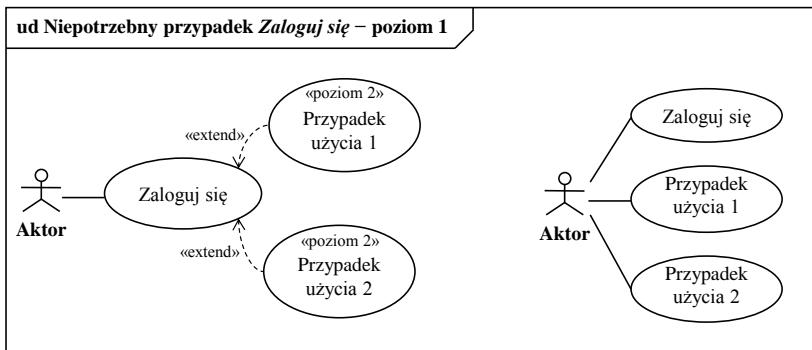
wielokrotnego wykorzystania, czyli tzw. bloki ponownego użycia. *Dla przypomnienia:* UML nie posiada specjalnego symbolu dla oznaczania bloków ponownego użycia (a szkoda, bo pozwalało to na łatwe wyróżnienie funkcji wewnętrznych, pomocniczych). Taki symbol ponownego użycia istniał w metodyce OOSE, której twórcą był Ivar Jacobson i którą wykorzystano w UML.

Poniżej została zamieszczona lista błędów popełnianych najczęściej przez osoby uczące się specyfikowania wymagań funkcjonalnych na system:

1. *Niezgodność diagramu kontekstowego z innymi diagramami modelu*, co oznacza, że do diagramów przypadków wprowadzono aktorów, których nie wyróżniono na diagramie kontekstowym i vice versa. Jedyne wyjątki to aktor abstrakcyjny (którego nie należy wprowadzać do diagramu kontekstowego, ponieważ stanowi uogólnienie aktorów „konkretnych”) oraz aktor *Podsystem czasu*, dla którego można w dalszej kolejności (na diagramach przypadków) bardziej szczegółowo określać momenty wywoływania skojarzonych z tym aktorem usług jak np.: *Codziennie, Na początku miesiąca, Codziennie 23:59*.
2. *Niezrozumienie idei pojęcia aktor*, czego dowodzi np. wprowadzenie do modelu na najwyższym poziomie abstrakcji aktora o nazwie *System*. System nie jest bytem ze swojego otoczenia, nie wchodzi w interakcję sam ze sobą! Na tym poziomie pomijamy też aktora takiego jak *Podsystem zarządzania bazą danych*, w ogóle jakikolwiek podsystem czy moduł projektowanego systemu. Jedynym wyjątkiem jest aktor *Podsystem czasu*, ponieważ w tym wypadku przyczyną sprawczą dla uruchomienia usługi systemu traktowanego jako całość jest upływ czasu. Nadejście określonego momentu jest traktowane jak wystąpienie zdarzenia w otoczeniu systemu. Z kolei, na niższych poziomach abstrakcji, gdzie jeden przypadek użycia może wywoływać inny przypadek, aktorem dla *przypadku wywoływanego* jest *przypadek wywołujący*.
3. Aktor *Administrator systemu*: w analizie systemowej raczej rzadko zajmujemy się tego rodzaju aktorem, a jeśli w ogóle, to nie przypisujemy do niego usług biznesowych – chyba że w danej instytucji administrator posiada obowiązki związane z działalnością w dziedzinie problemowej, co nie wydaje się być najlepszym rozwiązaniem.
4. Aktor *Operator/Użytkownik*: zadania dla modelu przypadków wymagają określenia możliwie jak największej grupy potencjalnych użytkowników danego systemu, a *Operator/użytkownik* jest aktorem zbyt ogólnym.
5. *Wprowadzanie do diagramu przypadków hierarchii dziedziczenia dla aktorów z aktorem nadzędnym* (np. *Osoba*), który nie posiada ani jednego bezpośrednio przypisanego do siebie przypadku użycia. Ponieważ hierarchie są tworzone po to, by możliwe było dziedziczenie przez klasyfikator „bardziej” wyspecjalizowane własności z klasyfikatora „bardziej” ogólnego, rozwiązanie jak wyżej jest błędne. Klasyfikator ogólny nie posiada w takim przypadku żadnych własności.

- Dla przypomnienia:* w przypadku hierarchii budowanych dla aktorów, dziedziczeniu podlegają prawa dostępu do usług dostarczanych przez system.
6. *Niezrozumienie idei budowy diagramu przypadków wyłącznie z perspektywy aktorów z otoczenia systemu traktowanego jako całość.* Uwaga: ta perspektywa oznacza, że nie należy włączać do diagramu takich przypadków, z którymi żaden aktor z otoczenia systemu nie wchodzi w  bezpośrednią interakcję (tzw. przypadków wewnętrznych), które zazwyczaj powstają na skutek podziału kodu przypadków bazowych. Powyższa reguła obowiązuje również dla niższych poziomów abstrakcji.
  7. *Niezbędny korzystny – z perspektywy percepcji diagramu – wzajemny układ tworzących go elementów.* Lepiej jest, zamiast umieszczenia aktorów w centrum diagramu, z przypadkami ułożonymi wokół nich, rysować aktorów po lewej i prawej stronie diagramu, a przypadki lokować w środkowej jego części. Przy takim układzie bardziej uwidaczniana jest istota wykorzystywanych pojęć: aktor (byt z otoczenia tego fragmentu systemu, dla którego został skonstruowany diagram) i przypadek (kod reprezentujący usługę systemową, wewnętrzne systemu).
  8. *Wprowadzanie duplikatów do diagramu:* w obrębie danego diagramu (czyli w jednej przestrzeni nazw), dopuszczalne jest umieszczanie kopii aktorów (w celu poprawienia czytelności diagramu, poprzez redukowanie liczby przecinających się linii), natomiast nie wolno wprowadzać kopii przypadków użycia.
  9. *Zbyt mało przypadków,* wykorzystywanie wyłącznie funkcjonalności sugerowanej w ostatnim punkcie tekstu wymagań dla zadania projektowo/problemowego, a są to wyłącznie przykłady potencjalnego wykorzystania systemu.
  10. *Niezbędny dobrze dobrane nazwy dla przypadków użycia* – przypadek powinien mieć nazwę zadania (a nie nazwę struktury danych), które kod implementujący fragment systemu z danego poziomu abstrakcji (podsystem, moduł, przypadek użycia, pojedyncza funkcja, procedura) ma wykonać na zlecenie aktora. Nazwa przypadku musi wyraźnie wskazywać na obserwowalny rezultat, czyli na coś, co zostanie zrobione w efekcie zrealizowania przypadku i co ma konkretną wartość dla aktora (jest potrzebne, użyteczne itd.).
  11. *Nieodpowiedni podział danego przypadku na podprzypadki, w oparciu o sporządzony dla tego przypadku scenariusz.* Dokonujemy podziału kodu implementującego przypadek w celu zmniejszenia jego złożoności albo wyróżnienia bloków ponownego użycia, a nie w celu naśladowania, określonej w scenariuszu, kolejności przepływu zdarzeń.
  12. *Niepotrzebne wprowadzanie do diagramu przypadku o nazwie Zaloguj się,* powiązanego z pozostałymi przypadkami relacjami «extend» (jak po lewej stronie Rys.1-7). Domyślnie przyjmuje się, że autentykacja i autoryzacja zostały już wykonane przed wywołaniem przypadku. Dodatkowym błędem, dzięki takiej organizacji diagramu, jest wprowadzenie przypadków, które są

przypadkami wewnętrznymi na tym poziomie abstrakcji. Przypadki połączone z *Zaloguj się* relacją «extend», nie będą widoczne dla aktorów z otoczenia systemu. Konstrukcja zaprezentowana po prawej stronie rysunku Rys. 1-8 jest bardziej poprawna, aczkolwiek w tym momencie przypadek *Zaloguj się* (dla którego potrzeba wprowadzenia w dalszej kolejności jest oczywista), w trakcie analizy początkowej w ogóle nie powinien być umieszczany na diagramie.



Rys. 1-8 Niepotrzebny przypadek *Zaloguj się*

13. *Nadużywanie przypadków o nazwie Zarządzaj czymś tam*. Często zdarzają się diagramy zawierające same *Zarządzaj*, i to bez określenia, na czym właściwie to zarządzanie ma polegać.
14. *Błędne rozumienie sekcji Warunek wstępny w dokumentacji przypadku*. W tej sekcji umieszczane są warunki nakładane na „świat przed wywołaniem przypadku”. Są one zazwyczaj sprawdzane na początku realizowania danego przypadku (jeszcze przed rozpoczęciem dialogu systemu z aktorem). Na przykład, nie można zarejestrować wypożyczenia płyty DVD, o ile wcześniej nie zarejestrowano w systemie ani jednej płyty.
15. *Podobnie, jak poprzednio: sekcja Warunek końcowy służy określению stanu systemu po zrealizowaniu danego przypadku*. Czyli, jeżeli zadaniem danego przypadku jest wyłącznie dostarczenie pewnej informacji dla aktora, to stan systemu nie ulega zmianie po zakończeniu tego przypadku.
16. *Niezrozumienie istoty scenariusza*. Scenariusz to opis dialogu/komunikacji pomiędzy aktorem a wywoływanym przez niego przypadkiem użycia. Dialog służy do pozyskania od aktora informacji potrzebnych do zrealizowania przypadku. Przypadek może także informować aktora o akcjach realizowanych/zrealizowanych w trakcie wykonywania. W każdym bądź razie scenariusz nie jest opisem algorytmu, na którym ma być oparta realizacja przypadku.

*Dla przypomnienia:* w scenariuszu łączymy w pary „akcję” z „reakcją”, czyli np. system odpytuje aktora o pewne informacje (akcja), aktor wprowadza odpowiednie dane (reakcja) – dzięki temu scenariusz jest bardziej zwarty i przejrzysty.

17. *Brak wyraźnego oddzielenia scenariusza głównego od scenariuszy alternatywnych.* Jest to ważne, ponieważ implementacja przypadku jest zazwyczaj rozpoczęta od implementacji jego scenariusza głównego, najbardziej użytecznego z perspektywy potencjalnych użytkowników systemu. W dalszych etapach scenariusz główny może być rozszerzany o przepływy alternatywne.

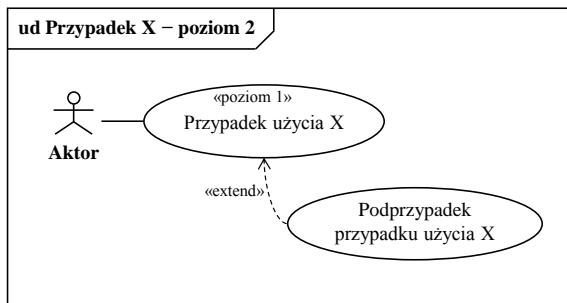
*Dla przypomnienia:* istnieje tylko jeden scenariusz główny – kojarzony z osiągnięciem pożdanego (pozytywnego) dla użytkownika, obserwowalnego rezultatu. Natomiast, zazwyczaj można wyróżnić więcej niż jeden scenariusz alternatywny.

18. *Brak numeracji kolejnych kroków/błędny numerowanie* zarówno w scenariuszu głównym, jak i w scenariuszach alternatywnych, co może znacząco utrudnić określenie związków przyczynowo-skutkowych pomiędzy aktywnościami wyróżnionymi w scenariuszu.

19. *Prezentowanie scenariusza w postaci oddzielnych wątków* utrudnia wydzielenie scenariusza głównego.

20. *Stosunkowo częstym błędem jest też umieszczanie kilku przypadków bazowych w jednym scenariuszu,* mimo że polecenie dotyczy jednego przypadku i na diagramach nie widać przesłanek do takiego postępowania.

21. *Błędy w budowaniu diagramu z podziałem przypadku na podprzypadki* – prawidłowa konstrukcja została zilustrowana na Rys. 1-9. Wprowadzenie do diagramu przypadku X nie jest błędem, o ile nie jest to sprzeczne z diagramami z wyższego poziomu abstrakcji.



Rys. 1-9 Poprawna konstrukcja dla podziału przypadku X na podprzypadki

### 1.4.3 Omówienia błędów dla grupy: *Analiza strukturalna*

Poniżej podano listę typowych błędów popełnianych w trakcie przeprowadzania analizy strukturalnej. Błędy dotyczą głównie schematu pojęciowego/diagramu klas.

Lista błędów została podzielona na grupy, stosownie do podziału kategorii *Poprawność*, który zaproponowano w tabeli miar służącej do oceny diagramu klas

(podrozdział 1.7.2). Są to grupy takie jak: uwagi ogólne, błędy związane z identyfikowaniem klas, atrybutów i metod oraz oznaczaniem ich rodzajów, z konstruowaniem hierarchii klas, z określaniem związków asocjacji (również kompozycji i asocjacji kwalifikowanej) oraz z wykorzystywaniem mechanizmów rozszerzalności (przede wszystkim komentarza i ograniczeń).

#### *Uwagi ogólne:*

1. *Błędem jest wprowadzanie do diagramów informacji, które ani bezpośrednio, ani pośrednio nie zostały umieszczone w tekście wymagań, jak np. id, jako atrybut obiektu, czy też elipsa dla dziedziczenia w sytuacji, gdy w tekście wymagań zasugerowano wszystkie potrzebne podklasy. Jest to traktowane jako nadinterpretacja wymagań (patrz punkt 4 tabeli miar dla oceny diagramu klas, Nadmiarowość).*
2. *Nie wolno wprowadzać do diagramu klas duplikatów, np. klas o tych samych nazwach w obrębie jednego diagramu, metod lub atrybutów o tych samych nazwach w obrębie jednej klasy – dla przypomnienia, zarówno diagram, jak i klasa stanowią jedną przestrzeń nazw.*
3. *Zapominanie o fakcie, że na tym etapie budowania systemu (analiza) skupiamy uwagę nie na typach danych (np. atrybutów klas, argumentów metod), ale na ich nazwach. To właśnie dobrze dobrana nazwa ma przenosić informację, co ma przechowywać dany atrybut, co ma przekazać argument metody czy też, co ma zrobić/zwrócić metodą. Działalność związana z przypisywaniem typów odpowiednich dla środowiska wybranego do implementowania systemu będzie miała miejsce w fazie projektowania (w modelu kaskadowym wytwarzania oprogramowania).*
4. *Umieszczanie atrybutów/metod w kilku wierszach – trzeba tak planować rysunek, aby każdy atrybut czy metoda zmieściły się w dokładnie jednym wierszu. Natomiast nazwa klasy, tekst komentarza czy ograniczenia mogą zajmować kilka wierszy.*

#### *Błędy związane z identyfikowaniem klas:*

1. *Zapominanie o fakcie, że nazwa klasy to zazwyczaj rzeczownik w liczbie pojedynczej. Należy postępować zgodnie z regułą: klasę nazywa się odpowiednio do tego, co ma przechowywać jej obiekt. Przykładowo, jeżeli pojedynczy obiekt klasy ma przechowywać informację o jednym produkcie, to klasie powinna być nadana nazwa Produkt, ale gdyby pojedynczy obiekt miał przechowywać informacje o zbiorze produktów, to w takim wypadku klasę należałoby nazwać: Produkty.*

Ponadto, należy dobrze rozumieć, co opisuje obiekt danej klasy: czy by pełniący rolę klasyfikatora czy może jedną instancję (egzemplarz) danego

klasyfikatora. Na przykład, jeżeli pojedynczy obiekt klasy *Produkt* opisuje rodzaj produktów (czyli klasyfikator), a nie jeden egzemplarz produktu danego rodzaju, to klasa *Produkt* powinna posiadać atrybut związany z przechowywaniem informacji o stanie produktu, czyli o liczbie aktualnie posiadanych sztuk/egzemplarzy danego produktu.

2. *Błędne oznaczanie klas i metod abstrakcyjnych.* Klasy i metody abstrakcyjne są oznaczane na diagramach na dwa sposoby: poprzez umieszczenie w sekcji służącej do przechowywania nazwy klasy (za nazwą klasy/metody) wyrażenia {abstract} lub poprzez napisanie nazwy klasy/metody kursywą. W przypadku ręcznego sporządzania diagramów, oczywiście tylko pierwszy z podanych sposobów jest wykorzystywany. Ponadto, używamy albo pierwszej albo drugiej notacji, a nie obu jednocześnie.

*Dla przypomnienia:* klasa abstrakcyjna może nie posiadać ani jednej metody abstrakcyjnej. To nie jest warunek konieczny dla oznaczenia danej klasy jako abstrakcyjnej. Klasa abstrakcyjna może posiadać metody abstrakcyjne, ale nie musi. Klasa konkretna nie może ich posiadać w ogóle. Definicja klasy abstrakcyjnej została oparta na pojęciu *wystąpienia bezpośredniego* (czyli na pojęciu: *obiekt klasy*). Klasa abstrakcyjna to taka klasa, dla której nie planuje się tworzenia jej wystąpień bezpośrednich (czyli obiektów) w czasie run-time'u. Zadaniem klasy abstrakcyjnej jest przechowywanie inwarariantów wspólnych dla jej podklas. Wyróżnianie klas abstrakcyjnych (w ogóle nadklas) ma bardzo duże znaczenie dla ponownego wykorzystania zdefiniowanego wcześniej kodu, np. kodu zakupionego w postaci biblioteki klas.

*Uwaga:* Klasa abstrakcyjna musi posiadać podklasy!

3. *Niezrozumienie istoty pojęć takich jak instancja klasy i ekstensja klasy.* Pierwsze z nich oznacza obiekt klasy (wystąpienie bezpośrednie klasy), a drugie skład obiektów danej klasy. Ekstensja klasy może zawierać zarówno wystąpienia bezpośrednie, jak i wystąpienia pośrednie danej klasy.

*Błędy związane z identyfikowaniem atrybutów i oznaczaniem ich rodzajów:*

1. *Nieoznaczanie rodzajów atrybutów.* Oznaczanie rodzaju powinno być wykonywane zawsze dla atrybutów opcjonalnych, powtarzalnych, pochodnych i klasowych. Brak oznaczeń może skutkować złym doborem struktur danych, implementujących atrybuty w środowisku implementacji.
2. *Niewprowadzanie do diagramu (w ogóle) lub brak oznaczeń dla atrybutów klasowych* (dla przypomnienia, oznaczamy je poprzez podkreślenie). Warto korzystać z atrybutów klasowych, ponieważ umożliwiają zmianę wartości granicznych, sugerowanych w tekście wymagań, bez konieczności wykonywania ponownej komplikacji kodu.
3. *Niezrozumienie istoty atrybutu pochodnego,* którego wartość można wyliczyć lub wyprowadzić w oparciu o tzw. dane bazowe. Dane bazowe mogą być

przechowywane w całym systemie (a nie tylko w tym samym obiekcie, co atrybut pochodny). Wprowadzanie atrybutów pochodnych ma znaczenie optymalizacyjne. Analityk wie, że atrybut pochodny jest bytem redundantnym, czyli nie powinien być umieszczony w modelu ze względu na pogorszenie percepcji, nie mniej jednak sugeruje jego przechowywanie. Atrybuty pochodne są wprowadzane zazwyczaj w sytuacji, gdy pewne dane rzadko się zmieniają, wyliczenie ich wartości jest kosztowne, a oprócz tego są one często wykorzystywane. Na przykład nie tylko atrybut *wiek* (wyliczany z daty urodzenia osoby) jest atrybutem pochodnym, ale też atrybut *liczba zawodów*, w których wziął udział dany zawodnik w sytuacji, gdy wprowadzono do diagramu asocjację łączącą klasy *Zawodnik* i *Zawody*.

4. *Nagminne wykorzystywanie atrybutów do modelowania ścieżek przesyłania komunikatów pomiędzy obiektami klas.* Jeżeli do diagramu zostały wprowadzone klasy, których obiekty mają zgodnie z tekstem wymagań przesyłać między sobą komunikaty, to do modelowania ścieżek przesyłania komunikatów wykorzystujemy wyłącznie asocjacje (a nie atrybuty i nie atrybuty łącznie z asocjacjami). Na etapie projektowania asocjacje zostaną zamienione na konstrukcje umożliwiające ich implementowanie w wybranym środowisku programistycznym, np. w popularnych językach obiektowych do tego celu mogą być wykorzystywane zmienne referencyjne lub kolekcje tego rodzaju zmiennych (wybór zmiennej prostej czy kolekcji jest uzależniony od liczności asocjacji).

*Dla przypomnienia:* w modelowaniu obiektowym *komunikat* oznacza żądanie wykonania operacji na obiekcie.

*Błędy związane z identyfikowaniem metod i oznaczaniem ich rodzajów:*

1. *Mylenie pojęć: sygnatura operacji/metody i nazwa operacji/metody.*

*Dla przypomnienia:* nazwa operacji/metody jest częścią sygnatury operacji/metody. Sygnatura zawiera elementy takie jak: *nazwa, lista argumentów i typ wartości zwracanej*. W przypadku identyfikowania polimorfizmu metod w danej hierarchii, wymagana jest zgodność sygnatur metod, a nie tylko ich nazw.

2. *Nieumieszczenie argumentów dla operacji/metod na listach argumentów.*

*Dla przypomnienia:* dla operacji/metody obiektowej typu *zapytanie* (np. *podaj()*, *odczytaj()* czy *wyświetl()*) nie umieszczamy atrybutów obiektu na liście argumentów operacji/metody, ponieważ operacja/metoda obiektowa ma bezpośredni dostęp do atrybutów obiektu, dla którego została wywołana.

3. *Błędy w definiowaniu metod polimorficznych w danej hierarchii klas.*

*Dla przypomnienia:* polimorfizm metod nie oznacza istnienia jednej metody, która realizowana jest na różne sposoby w zależności od tego, do obiektu której

klasy został wysłany komunikat, ale wielu metod (co najmniej dwóch), które mają pewne (to samo) zadanie do wykonania. Zadanie jest określone poprzez sygnaturę operacji, będącej elementem składowym wyrażenia specyfikującego komunikat, nazwa operacji odzwierciedla semantykę zadania (dlatego właściwy dobór nazw jest tak ważny w analizie). Zadanie jest realizowane przez każdą z metod polimorficznych w inny sposób w zależności od obiektu, dla którego zażądano wykonania operacji. Wniosek – metody polimorficzne różnią się ciałami, różnica w kodzie metod polimorficznych stanowiących implementację danej operacji wynika z różnych algorytmów, o które oparto ich ciała.

*Dla przypomnienia:* pojęcia takie jak *przesłanianie* i *przeciążanie* nie są synonimami. Metody przesłaniane muszą mieć identyczne sygnatury, natomiast metody przeciążane – tylko identyczne nazwy. Z tego wynikają między innymi następujące fakty: a) polimorfizm metod jest związany z przesłananiem metod, a nie z przeciążaniem, b) w danej klasie (ogólniej, w jednej przestrzeni nazw) można umieścić kilka metod przeciążonych, c) przesłanianie metod wymaga umieszczenia odpowiednich metod w jednej, wspólnej hierarchii.

4. *Uwaga:* nie istnieją ani metody opcjonalne, ani metody pochodne!
5. *Błędy związane z identyfikowaniem metod klasowych:* metody klasowe wykorzystywane są w trzech sytuacjach: (1) tworzenie nowych obiektów danej klasy, (2) operacje na invariantach klasy (atributach klasowych) oraz (3) operacje na ekstensji klasy. Metody klasowe należy podkreślać na diagramie.
6. *Nieznajomość rodzaju metod (abstrakcyjne, zaimplementowane), które wolno umieszczać w klasie abstrakcyjnej.* *Dla przypomnienia:* klasa abstrakcyjna może zawierać oba rodzaje metod, natomiast klasa konkretna tylko metody zaimplementowane. Ponadto, jeśli do klasy abstrakcyjnej zostanie wprowadzona metoda abstrakcyjna, to w podklassach tej klasy należy umieścić metody dostarczające implementacje dla niej. Metody implementujące muszą mieć dokładnie taką samą sygnaturę (a nie tylko taką samą nazwę), jak metoda abstrakcyjna. Stąd nie można uznać metody w podklasie, która ma wyłącznie tę samą nazwę co metoda w nadklasie (ale np. różną liczbę argumentów czy różne typy argumentów), za jej implementację.

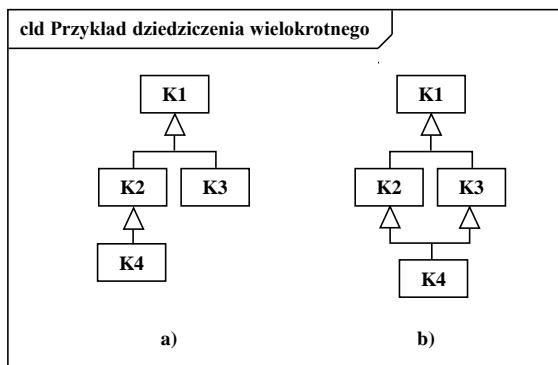
*Błędy związane z konstruowaniem hierarchii klas:*

1. *Błędne rozmieszczanie atrybutów i metod w hierarchii.*
2. *Nieodróżnianie rodzajów dziedziczenia:* na przykład *dziedziczenia kompletnego* (ang. complete) od *niekompletnego* (ang. incomplete), *rozłącznego* (ang. disjoint) od *nierozłącznego* (ang. overlapping) oraz *dziedziczenia wielokrotnego* (ang. multiple inheritance) od *wieloaspektowego* (ang. multiple aspect inheritance).

Na przykład, dziedziczenie wielokrotne ma miejsce wtedy, gdy w hierarchii występuje klasa, która bezpośrednio (!) dziedziczy po więcej niż jednej

nadklasie. Czyli na Rys. 1-10a) nie występuje dziedziczenie wielokrotne, ponieważ klasa *K4* dziedziczy bezpośrednio tylko po jednej klasie (jest nią klasa *K2*), natomiast na Rys. 1-10b) zostało wprowadzone dziedziczenie wielokrotne: klasa *K4* dziedziczy bezpośrednio po klasach *K2* i *K3*.

*Dla przypomnienia:* Rodzaje dziedziczenia oznaczamy przy symbolu dziedziczenia (symbolem jest niezamalowany trójkąt).



Rys. 1-10 Przykład bez dziedziczenia wielokrotnego i z dziedziczeniem wielokrotnym

3. *Nieodróżnianie elipsy od dziedziczenia niekompletnego.* Elipsa oznacza, że wszystkie podklasy zostały zdefiniowane, ale niektóre z nich nie zostały wprowadzone do diagramu, np., dlatego że w danym momencie interesujemy się tylko pewnym wycinkiem dziedziny problemowej. Z kolei *dziedziczenie niekompletne* oznacza, że proces identyfikacji podklas ma jeszcze miejsce (suma ekstensji podklas nie pokrywa ekstensji nadklasty). W tekście wymagań elipso sugerują sformułowania, przykładowo takie jak: *np., itd., między innymi.*

*Dla przypomnienia:* Jeśli dla danego podziału klasy wybrano dziedziczenie *niekompletne*, to w takim przypadku nadklasa nie może zostać oznaczona jako abstrakcyjna.

4. *Często, w celu uniknięcia niejednoznaczności, nazwy podklas powinny zawierać nazwę nadklasty.* Na przykład klasy dziedziczące po klasie *Serwis* powinny zostać nazwane *Serwis typowy* czy *Serwis kompletowany na zamówienie*, a nie *Typowy* i *Kompletowany na zamówienie*. W niektórych sytuacjach może to utrudniać zrozumienie, o jaki rodzaj bytów chodzi (np. w kodzie źródłowym), ponieważ nie tylko serwis może być typowy czy kompletowany na zamówienie.

*Błędy związane z określaniem związków asocjacji:*

1. *Błędna identyfikacja asocjacji.*

*Dla przypomnienia:* nie należy wprowadzać do diagramu asocjacji modelujących czynności, np. takich, że ktoś rejestruje czy przegląda pewne dane. Informacja o tym, kto ma uprawnienia do zarejestrowania czy przeglądania tych danych, powinna być modelowana z wykorzystaniem diagramów przypadków użycia. Asocjację należy wprowadzać tylko wtedy, gdy chcemy pamiętać *kto, co i kiedy* rejestrował lub przeglądał.

2. *Brak nazw dla asocjacji* często utrudnia zrozumienie diagramu i zmusza do zaglądania do tekstu wymagań.
3. *Błędnie oznaczane liczności asocjacji.*

*Dla przypomnienia:* określając liczność asocjacji, trzeba brać pod uwagę gromadzenie danych przez system na przestrzeni czasu. Fakt ten pomijamy tylko wtedy, gdy w tekście wymagań wyraźnie sformułowano, że należy pamiętać wyłącznie stan aktualny.

4. *Niewykorzystywanie dziedziczenia asocjacji*, które polega na tym, że zamiast łączyć daną klasę oddziennie z każdą z podklas innej klasy, łączymy ją z nadklassą, a ewentualne różnice w licznościach asocjacji modelujemy, wykorzystując ograniczenia.

*Uogólniając:* dla asocjacji, które podobnie jak klasy są klasyfikatorami, także można tworzyć struktury generalizacji-specjalizacji.

5. *Nadużywanie symbolu zamalowanego „trójkątka”*, który wskazuje kierunek czytania nazwy asocjacji. Symbol ten zaburza percepcję diagramu, co jest sprzeczne z ideą wprowadzenia informacji w trakcie modelowania/projektowania systemów za pomocą obrazów. Przyjawszy założenie, że w naszej kulturze tekst jest czytany z lewa na prawo i z góry na dół, te same reguły należy stosować do asocjacji na diagramach, wykorzystując symbol zmiany kierunku czytania tylko wtedy, gdy jest to naprawdę potrzebne. Jest to dodatkowo zgodne z regułą, że w celu poprawienia czytelności/percepcji diagramów nie wprowadzamy do diagramów pojęć, które zostały uznane za domyślne, np. nie umieszczamy na diagramach klas wyrażeń *{disjoint}* czy *{incomplete}*.

6. *Nadużywanie agregacji/kompozycji:* brak nazw dla asocjacji powoduje, że traciona jest informacja o charakterze relacji pomiędzy obiektami klas – nie wszystkie związki semantyczne zostały opisane w tekście wymagań użytkownika jako „*należy do*”, „*zawiera się w ...*”. Nazwy asocjacji powinny wynikać z tekstu wymagań.

7. *Niezrozumienie różnicy między agregacją a kompozycją*, niezrozumienie stwierdzenia, że „w przypadku kompozycji cykl życiowy części zawiera się w cyklu życiowym całości”. Stwierdzenie odwrotne nie jest prawdziwe, nie jest prawdą, że „usunięcie wszystkich części skutkuje usunięciem całości”.

Ponadto, silny związek częścь-całość to synonim kompozycji, a nie agregacji.

8. *Brak asocjacji kwalifikowanej* (punkt 1.6 tabeli miar dla oceny diagramu klas) – należy ją od razu umieścić na diagramie, a w zadaniu dotyczącym tej asocjacji najlepiej jest przerysować fragment diagramu z konstrukcją pierwotną (czyli bez asocjacji kwalifikowanej) oraz dołączyć objaśnienie, dlaczego właśnie w tym miejscu diagramu należało taką asocjację umieścić.

#### 1.4.4 Omówienia błędów dla grupy: *Analiza dynamiczna*

- Uwaga:* Graf aktywności i maszyna stanowa są grafami skierowanymi, co oznacza, że przejścia pomiędzy wierzchołkami muszą być opatrzone grotami określającymi kierunki przechodzenia.
- Etykię przejścia* pomiędzy stanami dla maszyny stanowej behawioralnej można specyfikować na różne sposoby – w oparciu o definicję etykiety.

*Dla przypomnienia:* Etykieta przejścia dla maszyny behawioralnej składa się z trzech opcjonalnych elementów, takich jak: *nazwa zdarzenia*, *[warunek przejścia]* i */ akcja*. Warunki na diagramach dynamicznych wprowadzane są w nawiasach kwadratowych i oznaczają wyrażenia typu Boolean, tzn. mogą przyjmować dwie wartości: „*prawda*” albo „*falsz*”.

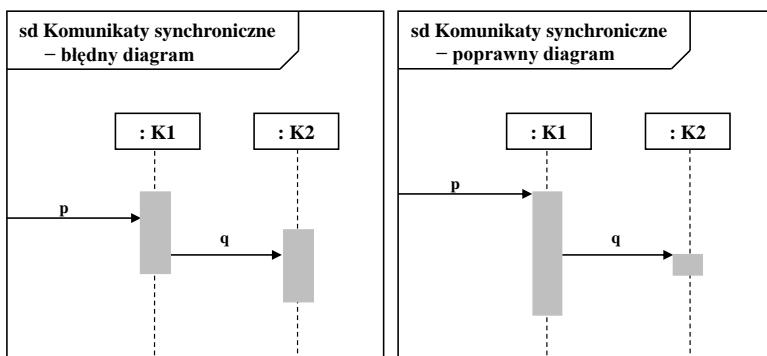
- Często występują problemy z prawidłowym wykorzystaniem symboli: stanu początkowego i stanu końcowego* przy budowaniu maszyny stanowej. Tab. 1-6 zawiera porównanie wykorzystania obu symboli dla diagramów aktywności i diagramów stanów.

Tab. 1-6 Różnice znaczeń symboli dla stanu początkowego i stanu końcowego dla diagramów aktywności i diagramów stanów

Symbol	Diagram aktywności – znaczenie symbolu	Diagram stanów – znaczenie symbolu
●	początek sekwencji aktywności	<p><i>dla najwyższego poziomu zagieźdżenia:</i> powołanie do życia nowego obiektu tej klasy, dla której zbudowana została dana maszyna stanowa</p> <p><i>dla kolejnych poziomów zagieźdżenia:</i> osiągnięcie stanu początkowego oznacza przejście do tego podstanu stanu złożonego, który został oznaczony jako jego stan początkowy lub oznacza przejście do stanu początkowego</p>

		diagramu zagnieżdzonego w danym diagramie
●	koniec sekwencji aktywności	<p><i>dla najwyższego poziomu zagnieżdżenia:</i> osiągnięcie stanu końcowego oznacza koniec życia obiektu, czyli usunięcie z systemu obiektu tej klasy, dla której zbudowano daną maszynę stanową</p> <p>jeśli w tekście wymagań nie ma informacji o usuwaniu obiektów danej klasy z systemu, to stan końcowy w ogóle nie powinien być wprowadzany do diagramu</p> <p><i>dla kolejnych poziomów zagnieżdżenia:</i> osiągnięcie stanu końcowego oznacza opuszczenie stanu złożonego lub opuszczenie diagramu zagnieżdzonego w danym diagramie</p>

4. Diagram zamieszczony na Rys. 1-11 (po lewej stronie) *prezentuje podstawowe błędy w komunikacji synchronicznej* – takiej, gdzie nadawca komunikatu zawiesza działanie do momentu zakończenia pracy odbiorcy komunikatu (synchroniczność oznaczana jest za pomocą linii ciągłej i zamalowanego grotu).



Rys. 1-11 Podstawowe błędy popełniane dla komunikatów synchronicznych

Realizacje operacji  $p, q$  na instancjach klasyfikatorów  $K1, K2$ , rozpoczęły się zanim odpowiednie komunikaty zawierające żądania wykonania  $p, q$  zostały w ogóle przesłane. Ponadto, operacja  $q$  zakończyła się później niż operacja  $p$ , co nie jest możliwe w komunikacji synchronicznej, ponieważ  $p$  jest zawieszone do momentu, dopóki  $q$  się nie skończy i nie przekaże sterowania z powrotem do  $p$ .

*Uwaga:* Instancje kwalifikatorów biorące udział w interakcji, o ile istniały w momencie rozpoczęcia interakcji, muszą być umieszczane na tej samej wysokości – oś pionowa diagramu jest wykorzystywana do modelowania upływu czasu.

## 1.5 Polecenia proste

Poniżej zamieszczono listy przykładowych poleceń prostych, umożliwiających sprawdzenie stopnia zapamiętania treści przekazywanych w trakcie realizowania przedmiotu i wykorzystujących pojedyncze, proste konstrukcje języka modelowania. Polecenia zostały podzielone na cztery grupy, przewidziane oddzielnie dla każdej z wyróżnionych grup tematycznych, czyli dla: *Wprowadzenia do przedmiotu*, *Analizy funkcjonalnej*, *Analizy strukturalnej* oraz *Analizy dynamicznej*.

### 1.5.1 Polecenia proste dla grupy: *Wprowadzenie do przedmiotu*

1. Objaśnij pojęcie: *stereotyp*. Podaj odpowiedni przykład dla schematu pojęciowego skonstruowanego na podstawie tekstu wymagań.
2. Objaśnij pojęcie: *wartość etykietowana*. Podaj odpowiedni przykład dla schematu pojęciowego skonstruowanego na podstawie tekstu wymagań.
3. Objaśnij różnicę w zastosowaniu ograniczeń i komentarzy.
4. Objaśnij różnice pomiędzy pojęciami: *klasyfikator* i *instancja klasyfikatora*. Podaj przykłady zastosowania w modelu funkcjonalnym, strukturalnym i dynamicznym.
5. Objaśnij pojęcie: *asocjacja*. Podaj przykład zastosowania w modelu funkcjonalnym i strukturalnym.
6. Objaśnij pojęcie: *generalizacja/specjalizacja*. Podaj przykład zastosowania w modelu funkcjonalnym, strukturalnym i dynamicznym.
7. Objaśnij pojęcie: *realizacja*. Podaj przykład zastosowania w modelu strukturalnym.
8. Objaśnij pojęcie: *zależność*. Podaj przykład zastosowania w modelu funkcjonalnym, strukturalnym i dynamicznym.

9. Objąsnij, w jakim celu i dla jakiego rodzaju diagramów mogą być wykorzystywane pakiety.
10. Objąsnij, jakie korzyści przynosi obramowywanie diagramów i wykorzystywanie nagłówków łącznie z wyróżnikami i nazwami diagramów.

### **1.5.2 Polecenia proste dla grupy: *Analiza funkcjonalna***

1. Dla podanego tekstu wymagań skonstruuj diagram kontekstowy.
2. Czy system zewnętrzny lub podsystem wewnętrzny mogą być aktorami? Jeśli tak, odpowiedź uzupełnij przykładami spójnymi z podanym tekstem wymagań.
3. Objąsnij, dlaczego aktor *Podsystem czasu* jest traktowany inaczej niż inne podsystemy w modelu przypadków użycia?
4. W oparciu o podany tekst wymagań określ kilka przykładowych przypadków użycia łącznie z wykorzystaniem występujących między nimi relacji. Jeśli występują zależności pomiędzy aktorami (czyli dostęp do pewnych usług systemu jest możliwy dla więcej niż jednego aktora), wyspecyfikuj je w postaci odpowiedniej hierarchii.
5. Na podstawie podanego tekstu wymagań, dla wybranego przypadku użycia systemu, sformułuj scenariusz.
6. Na podstawie podanego tekstu wymagań, dla wybranego przypadku użycia systemu, sformułuj scenariusz oraz zilustruj podział tego przypadku na podprzypadki.

### **1.5.3 Polecenia proste dla grupy: *Analiza strukturalna***

1. W oparciu o podane wymagania skonstruuj schemat pojęciowy nie redukując liczności asocjacji.
2. Wyjaśnij, jak przebiega proces identyfikacji potencjalnych klas w trakcie budowy modelu struktury systemu.
3. Objąsnij różnicę między dziedziczeniem *nierożłącznym* (ang. overlapping) a *dziedziczeniem wielokrotnym* (ang. multiple inheritance). Czy któryś z tych obu rodzajów dziedziczenia wystąpił na schemacie pojęciowym skonstruowanym na podstawie podanego tekstu wymagań?
4. Objąsnij różnicę między *dziedziczeniem nierożłącznym* (ang. overlapping) a *dziedziczeniem wieloaspektowym* (ang. multiple aspect inheritance)? Czy któryś z tych obu rodzajów dziedziczenia wystąpił na schemacie pojęciowym skonstruowanym na podstawie podanego tekstu wymagań?

5. Objąśnij różnicę między *dziedziczeniem wielokrotnym* (ang. multiple inheritance) a *wieloaspektowym* (ang. multiple aspect inheritance)? Czy któryś z tych obu rodzajów dziedziczenia wystąpił na schemacie pojęciowym skonstruowanym na podstawie podanego tekstu wymagań?
6. Co oznacza znak *elipsy* (inaczej znak opuszczenia, ang. ellipsis) znajdujący się na diagramie? Objąśnij, czym różni się wykorzystanie tego symbolu od ograniczenia *{incomplete}*. Podaj odpowiedni przykład dla schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań.
7. Podaj definicję *klasy abstrakcyjnej*. Czy klasa abstrakcyjna może być zdefiniowana wyłącznie w korzeniu hierarchii klas? Czy klasa w korzeniu hierarchii klas zawsze musi być klasą abstrakcyjną? Oznacz klasy abstrakcyjne na schemacie pojęciowym skonstruowanym na podstawie podanego tekstu wymagań.
8. Objąśnij różnicę między *klasą* a *ekstensją klasy*. Podaj przykładowe ekstensje dla klas wchodzących w skład najbardziej rozwiniętej hierarchii na schemacie pojęciowym skonstruowanym w oparciu o podany tekst wymagań.
9. Odpowiedz, czym różnią się pojęcia: *instancja klasy* i *instancja obiektu*? Które z nich nie jest poprawne?
10. Zdefiniuj pojęcie *metoda abstrakcyjna*. Wprowadź, co najmniej jedną metodę abstrakcyjną do schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań.
11. Czy klasa abstrakcyjna może zawierać metody zaimplementowane? Odpowiedź uzasadnij. Jeśli odpowiedź jest na „tak”, podaj odpowiedni przykład dla schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań.
12. Dla schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań podaj przykłady: *operacji, metody i komunikatu*.
13. Wyjaśnij pojęcie: *polimorfizm metod*. Wskaż przykład jego zastosowania na schemacie pojęciowym skonstruowanym na podstawie podanego tekstu wymagań.
14. Objąśnij pojęcie: *przesłanianie metod*. Wskaż przykład jego zastosowania na schemacie pojęciowym skonstruowanym na podstawie podanego tekstu wymagań. Czy przesłanianie jest rodzajem polimorfizmu metod?
15. Wyjaśnij pojęcie: *przeciążanie metod*. Wskaż przykład jego zastosowania na schemacie pojęciowym skonstruowanym na podstawie podanego tekstu wymagań. Czy przeciążanie jest rodzajem polimorfizmu metod?
16. Objąśnij różnicę między *przeciążaniem* a *przesłanianiem metod*. Wskaż przykłady wykorzystania tych pojęć na schemacie pojęciowym skonstruowanym na podstawie podanego tekstu wymagań. Czy przeciążanie jest rodzajem polimorfizmu metod?

17. Wyjaśnij różnicę pomiędzy *atrybutem klasowym* a *atrybutem obiektu*, *metodą klasową* a *metodą obiektu*. Dla schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań wskaz (bądź dopisz, o ile nie ma) przykład metody o zasięgu klasowym i atrybutu klasowego.
18. Czy *zasadę zamienialności* można stosować do obiektów należących do różnych hierarchii klas? Odpowiedz uzasadnij. Jeśli odpowiedź jest na „tak”, wskaz przykład jej zastosowania na schemacie pojęciowym skonstruowanym na podstawie podanego tekstu wymagań.
19. Czy *zasadę zamienialności* można stosować do obiektów klas tworzących agregację? Odpowiedz uzasadnij. Jeśli odpowiedź jest na „tak”, wskaz przykład jego zastosowania na schemacie pojęciowym skonstruowanym na podstawie podanego tekstu wymagań.
20. Wskaz na schemacie pojęciowym skonstruowanym na podstawie podanego tekstu wymagań asocjację wiele-do-wielu z niesymetrycznymi licznosciami dla obu końców asocjacji oraz ze zdefiniowaną klasą lub atrybutem asocjacji. Na jakim etapie cyklu życia systemu powinno się zamienić klasę asocjacji na konstrukcję zawierającą klasę pośredniczącą? Narysuj odpowiedni przykład (z atrybutami i metodami) dla wskazanego uprzednio fragmentu schematu.
21. Wskaz na schemacie pojęciowym skonstruowanym na podstawie podanego tekstu wymagań taki fragment, gdzie *role asocjacji* mogłyby być użyteczne. Zamień ten fragment (zachowując semantykę) tak, aby role asocjacji mogły zostać opuszczone.
22. Wskaz na schemacie pojęciowym skonstruowanym na podstawie podanego tekstu wymagań asocjację, która w naturalny sposób kwalifikuje się do zamiany na *kompozycję* lub *agregację* (być może już taką umieściłeś). Uzasadnij swój wybór (dlaczego aggregacja? dlaczego kompozycja?). Ponadto, dlaczego zdecydowałeś się na wykorzystanie aggregacji (lub kompozycji), a nie „zwyczajnej asocjacji”?
23. Podaj przykład *agregacji rekurencyjnej* dla schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań. Czy jest możliwe wykorzystanie *kompozycji rekurencyjnej* zamiast tej aggregacji? Odpowiedz uzasadnij, ilustrując odpowiednimi przykładami.
24. Czym różni się wykorzystanie *agregacji rekurencyjnej* od *asocjacji rekurencyjnej*? Objasnij odpowiedź na przykładzie schematu pojęciowego skonstruowanego dla podanego tekstu wymagań.
25. Objasnij różnicę pomiędzy *komentarzem* a *ograniczeniem*. Podaj odpowiednie przykłady dla schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań.
26. Dla schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań podaj przykład *ograniczenia statycznego* i *ograniczenia dynamicznego*. Uzasadnij swój wybór. Na jakie elementy modelu zostały

- nałożone te ograniczenia? W jakiej metodzie i w jakiej klasie umieściłbyś ich implementację?
27. Dla schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań podaj przykład ograniczenia, które odnosi się do dwóch asocjacji. W jakiej metodzie i w jakiej klasie umieściłbyś implementację tego ograniczenia?
  28. Dla schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań podaj przykład ograniczenia na istnienie obiektu. Czy jest ograniczenie statyczne czy dynamiczne? W jakiej metodzie i w jakiej klasie umieściłbyś implementację tego ograniczenia?
  29. Dla schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań podaj przykład ograniczenia, którego implementacja mogłaby być elementem struktury systemu (schematu pojęciowego).
  30. Dla schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań asocjację wiele-do-wielu (posiadającą atrybut lub klasę asocjacji i niesymetryczne liczności) zamień na *asocjację kwalifikowaną*. Uzasadnij wybór asocjacji.
  31. Dla schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań wskaż asocjację wiele-do-wielu, posiadającą atrybut lub klasę asocjacji i niesymetryczne liczności. Zamień tę asocjację na *asocjację kwalifikowaną*. Wyjaśnij, czy oba rozwiązania są równoważne. Wyjaśnij, w jakim celu stosuje się asocjacje kwalifikowane na schemacie pojęciowym.
  32. Dla schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań znajdź miejsce, gdzie zastosowanie *asocjacji n-arnej* wydaje Ci się oczywiste. Narysuj zmieniony fragment schematu, nie zapominając o licznościach, nazwie i klasie asocjacji dla asocjacji n-arnej.
  33. Dla schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań fragment schematu zawierający asocjację binarną z klasą asocjacji przerysuj jako trzy klasy połączone *asocjacją ternarną*. Dobierz stosowne liczności i skomentuj ewentualne wady/zalety takiego rozwiązania.
  34. Schemat pojęciowy skonstruowany na podstawie podanego tekstu wymagań zamień na schemat projektowy dla popularnego środowiska obiektowego (na przykład w języku Java).
  35. Dla schematu pojęciowego skonstruowanego na podstawie podanego tekstu wymagań wybierz klasę posiadającą przynajmniej dwie podklasy (w razie potrzeby dorysuj brakującą podkласę). Zamodeluj wybrany fragment diagramu w systemie relacyjnym na dwa sposoby.
  36. Dla schematu pojęciowego skonstruowanego w oparciu o podany tekst wymagań wybierz fragment z klasą posiadającą co najmniej dwie podklasy (w razie potrzeby dorysuj brakującą podkласę) oraz asocjację pomiędzy jedną

z klas wybranej hierarchii z inną klasą z diagramu. Zamodeluj wybrany fragment diagramu w systemie relacyjnym.

#### 1.5.4 Polecenia proste dla grupy: *Analiza dynamiczna*

1. Dla przypadku użycia systemu wybranego z podanego tekstu wymagań sformułuj scenariusz i narysuj diagram aktywności, zgodny z tym scenariuszem. Określ przynajmniej dwie aktywności (nie licząc początkowej i końcowej).
2. Zaproponuj algorytm dla metody wybranej z ostatniego punktu podanego tekstu wymagań. Dla opisu algorytmu wykorzystaj pseudokod.
3. Dla metody wybranej z ostatniego punktu podanego tekstu wymagań zaproponuj algorytm zilustrowany za pomocą diagramu aktywności.
4. Dla metody wybranej z ostatniego punktu podanego tekstu wymagań zaproponuj algorytm zilustrowany za pomocą diagramów aktywności z wykorzystaniem aktywności złożonej.
5. Dla metody wybranej z ostatniego punktu podanego tekstu wymagań zaproponuj algorytm zilustrowany za pomocą diagramów aktywności z wykorzystaniem przepływów współbieżnych.
6. Dla metody wybranej z ostatniego punktu podanego tekstu wymagań zaproponuj algorytm zilustrowany za pomocą diagramów aktywności z wykorzystaniem przepływów obiektów.
7. Dla przypadku użycia wybranego z modelu przypadków skonstruowanego w oparciu o podany tekst wymagań narysuj taki fragment diagramu sekwencji, który przedstawia interakcje pomiędzy obiektami co najmniej dwóch różnych klas.
8. Dla przypadku użycia wybranego z modelu przypadków skonstruowanego w oparciu o podany tekst wymagań, narysuj taki fragment diagramu komunikacji, który przedstawia interakcje pomiędzy obiektami co najmniej dwóch różnych klas.
9. Zamień diagram sekwencji dla przypadku użycia wybranego z modelu przypadków skonstruowanego w oparciu o podany tekst wymagań na izomorficzny diagram komunikacji.
10. Zamień diagram komunikacji dla przypadku użycia wybranego z modelu przypadków skonstruowanego w oparciu o podany tekst wymagań na izomorficzny diagram sekwencji.
11. Dla przypadku użycia wybranego z modelu przypadków skonstruowanego w oparciu o podany tekst wymagań narysuj taki fragment diagramu sekwencji, który wykorzystuje co najmniej jeden fragment wyodrębniony (dowolnego rodzaju).

12. Dla przypadku użycia wybranego z modelu przypadków skonstruowanego w oparciu o podany tekst wymagań narysuj taki fragment diagramu sekwencji, który wykorzystuje przywołanie wystąpienia interakcji.
13. Dla przypadku użycia wybranego z modelu przypadków skonstruowanego w oparciu o podany tekst wymagań skonstruuj diagram przeglądu interakcji (jest to diagram o wyróżniku *iod*).
14. Dla schematu projektowego powstałego w oparciu o schemat pojęciowy skonstruowany zgodnie z podanym tekstem wymagań utwórz – dla przypadku użycia wybranego z modelu funkcjonalnego – diagram sekwencji.
15. Dla dowolnej klasy ze schematu pojęciowego skonstruowanego w oparciu o podany tekst wymagań wskaż dwa stany (wyłączając stan początkowy i stan końcowy), w jakich mogą znajdować się obiekty tej klasy. Przedstaw maszynę zachowania (behawioralną), zawierającą te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.
16. Dla dowolnej klasy ze schematu pojęciowego skonstruowanego w oparciu o podany tekst wymagań przedstaw maszynę zachowania (behawioralną), zawierającą co najmniej jeden stan złożony sekwencyjny.
17. Dla dowolnej klasy ze schematu pojęciowego skonstruowanego w oparciu o podany tekst wymagań przedstaw maszynę zachowania (behawioralną), zawierającą co najmniej jeden stan złożony wspólnie.
18. Dla dowolnej klasy ze schematu pojęciowego skonstruowanego w oparciu o podany tekst wymagań wskaż dwa stany (oprócz stanu początkowego i stanu końcowego), w jakich mogą znajdować się obiekty tej klasy. Przedstaw maszynę protokołową, zawierającą te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.
19. Dla dowolnej klasy ze schematu pojęciowego skonstruowanego w oparciu o podany tekst wymagań zamień maszynę zachowania (behawioralną) na maszynę protokołową.

## **1.6 Szablony służące do tworzenia opisów zadań projektowo/problemowych**

W Tab. 1-7 i w Tab. 1-8 zaprezentowano elementy składowe szablonów dla tworzenia zarówno charakterystyk ogólnych, jak i szczegółowych dla zadań projektowo/problemowych. Przykłady wykorzystania szablonów zamieszczone w rozdziałach 2-12 publikacji. Stopnie trudności dla poszczególnych zadań traktowanych jako całość, a także stopnie oznaczające trudność zagadnień czy pojęć omawianych w danej grupie tematycznej dla każdego z zadań oddzielnie, określone zostały z perspektywy osoby rozpoczynającej naukę modelowania/projektowania systemów informacyjnych. Podobną zasadę przyjęto dla ustalania użyteczności

każdego z zadań, oddziennie dla każdej z grup tematycznych w obrębie danego zadania oraz oddziennie dla każdego z etapów nauczania zgodnego z modelem efektywnego nauczania.

Tab. 1-7 Ogólna charakterystyka zadania

<b>Ogólna charakterystyka zadania</b>		
<i>Nazwa</i>		
<i>Dziedzina problemowa</i>		
<i>Cel</i>		
<i>Zakres odpowiedzialności</i>		
<i>Stopień trudności</i>		
<i>Rozmiar</i>		
<i>Użyteczność w grupach tematycznych</i>		
<i>Grupa tematyczna</i>	<i>Użyteczność w grupie tematycznej</i>	<i>Użyteczność na etapie zgodnym z modelem efektywnego nauczania</i>
<i>Wprowadzenie do przedmiotu</i>		<i>Przygotowanie do nauki: Nauka: Podsumowanie dotychczasowych działań: Ewaluacja:</i>
<i>Analiza funkcjonalna</i>		<i>Przygotowanie do nauki: Nauka: Podsumowanie dotychczasowych działań: Ewaluacja:</i>
<i>Analiza strukturalna</i>		<i>Przygotowanie do nauki: Nauka: Podsumowanie dotychczasowych działań: Ewaluacja:</i>
<i>Analiza dynamiczna</i>		<i>Przygotowanie do nauki: Nauka: Podsumowanie dotychczasowych działań: Ewaluacja:</i>

Tab. 1-8 Szczegółowa charakterystyka zadania w obrębie każdej z grup tematycznych

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>		
<b>Stopień trudności grupy w zadaniu</b>		
<b>Poruszane tematy</b>		
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>

*Dla przypomnienia:* Ostatni punkt tekstu wymagań w zadaniach projektowo/problemowych został przewidziany do szacowania stopnia nabycia dwojakiego rodzaju umiejętności: (1) poprawnego umieszczania metod w klasach oraz (2) do budowy modelu przypadków użycia.

W pierwszym przypadku jest oceniane, czy użyto dobrej nazwy dla metody (czy nazwa przenosi semantykę metody), czy lista argumentów jest poprawna oraz czy metoda została właściwie oznaczona (*obiektowa, klasowa, abstrakcyjna, zaimplementowana*). Na początkowym etapie analizy nie należy specyfikować typów ani dla argumentów metod ani dla wartości zwracanych przez metody. Tego rodzaju aktywność związana jest z konkretnym środowiskiem implementacji i będzie realizowana podczas etapu projektowania (w modelu wodospadowym wytwarzania oprogramowania).

Z kolei, w trakcie budowy modelu przypadków użycia, jest możliwe, że do zrealizowania nietrywialnego, bardziej złożonego przypadku będzie potrzebna więcej niż jedna metoda, ale ich identyfikowanie będzie miało miejsce dopiero w trakcie przeprowadzania analizy dynamicznej. Analiza funkcjonalna i strukturalna wymagają jedynie określenia, od jakiej metody i umieszczonej w jakiej klasie rozpocznie się realizacja każdego z przypadków, o który zapytano w ostatnim punkcie tekstu wymagań w zadaniu projektowo/problemowym.

## 1.7 Metryki do oceny jakości rozwiązań

Elementy składowe zbiorów miar (nazwanych w tabelach Tab. 1-9, Tab. 1-10, Tab. 1-11 oraz w Tab. 1-12 przedmiotami oceny) tworzą metryki służące do oceny jakości rozwiązań i mogą być wykorzystywane zarówno samodzielnie (do oceniania poleceń prostych, gdzie rozwiązaniami są pojedyncze konstrukcje modelowania), jak i łącznie – na przykład do szacowania jakości rozwiązań dla zadań projektowo/problemowych, czyli do oceny jednego lub więcej diagramów czy scenariuszy opisujących spójne fragmenty dziedziny problemowej i odnoszących się do wybranej grupy tematycznej. Objasnienia znaczenia niektórych z miar wykorzystanych w metrykach podano poniżej.

*Poprawność:* diagram jest *poprawny*, jeżeli występujące na nim pojęcia zostały użyte we właściwy sposób. Wyróżnia się dwa podstawowe rodzaje poprawności:

- *poprawność syntaktyczną* – diagram jest poprawny syntaktycznie, jeżeli pojęcia są przedstawione zgodnie z regułami składniowymi języka wykorzystanego do opisu,
- *poprawność semantyczną* – diagram jest poprawny semantycznie, jeżeli odpowiada sytuacji z rozważanej dziedziny problemowej.

*Kompletność:* diagram jest *kompletny*, jeżeli przedstawia wszystkie relevantne cechy rozważanego fragmentu dziedziny problemowej.

Poniższe czynności mogą być wykorzystane w celu ułatwienia stwierdzenia, czy dany diagram jest kompletny:

- przeprowadzenie analizy wymagań użytkownika i sprawdzenie, czy wszystkie zawarte w nich pojęcia zostały wyrażone na diagramie;
- przeprowadzenie analizy pojęć zobrazowanych na diagramie i porównanie zgodności ich opisów z opisami zawartymi w wymaganiach;
- przeprowadzenie jednoczesnej analizy diagramów wchodzących w skład różnych modeli, np. diagramów klas i diagramów dynamicznych, ze szczególnym uwzględnieniem ich wzajemnej spójności.

*Minimalność:* diagram jest *minimalny*, jeżeli każdy z aspektów dziedziny problemowej jest przedstawiony na nim dokładnie jeden raz. Usunięcie dowolnego elementu z diagramu minimalnego prowadzi do utraty informacji.

Niemniej jednak, w niektórych sytuacjach redundancja informacji może być przydatna i warto ją wykorzystać, konstrując diagram. Na przykład, już na etapie analizy można zasugerować przechowywanie pewnego pochodnego (wyliczalnego) atrybutu w sytuacji, gdy:

- jest kosztowny obliczeniowo,
- jego wartość rzadko się zmienia,
- jest często wykorzystywany.

Z kolei, asocjacje pochodne można wykorzystać do skrócenia ścieżki dostępu do pewnych informacji. Zawsze należy jednak udokumentować, które elementy są pochodne oraz w jaki sposób się je wylicza lub algorytmicznie wprowadza.

*Nadmiarowość:* diagram jest nadmiarowy jeśli wprowadzono do niego informacje, które ani bezpośrednio, ani pośrednio nie wynikają z tekstu wymagań.

### 1.7.1 Metryki do oceny jakości rozwiązań w grupie: Analiza funkcjonalna

W Tab. 1-9 zamieszczono propozycję zbioru miar wykorzystywanych do określania jakości diagramów przypadków użycia.

Tab. 1-9 Tabela miar do oceny diagramu przypadków użycia

Przedmiot oceny	Zakres ocen	Waga	Ocena
<b>1. Poprawność</b> (suma ocen z pozycji 1.1 do 1.5)	0..15	1	
1.1 <i>aktorzy</i> : czy poprawnie określono aktorów? czy nazwy wybrane dla nich zgodnie z regułami i tekstem wymagań? czy zidentyfikowano aktora <i>Podsystem czasu</i> ? czy zidentyfikowano aktorów abstrakcyjnych?	0..3	1	
1.2 <i>hierarchia/hierarchie dla aktorów</i> : czy wprowadzono do diagramu taką strukturę (struktury) i czy została ona poprawnie zbudowana?	0..3	1	
1.3 <i>przypadki użycia</i> : czy zidentyfikowane przypadki użycia są zgodne z informacjami zawartymi w tekście wymagań? czy nazwy zostały wybrane w oparciu o reguły nazewnictwa?  na przykład: błędem jest wprowadzenie do diagramu tzw. <i>przypadków wewnętrznych na danym poziomie abstrakcji</i> , czyli takich, które nie są bezpośrednio wywoływane przez aktorów z otoczenia danego fragmentu systemu (czy też całości systemu na najwyższym poziomie abstrakcji)  nazwy przypadków powinny sugerować zadania, zlecane przez aktora systemowi do wykonania,	0..3	1	

powinny więc budzić skojarzenia z opercjami realizowanymi na danych, a nie ze strukturami danych			
1.4 <i>ustalanie praw dostępu aktorów do przypadków użycia:</i> czy prawa dostępu zostały ustalone zgodnie z tekstem wymagań, czyli zgodnie z zakresem odpowiedzialności pracowników biznesowych z rozważanego fragmentu dziedziny problemowej?	0..3	1	
1.5 <i>relacje pomiędzy przypadkami:</i> czy poprawnie zidentyfikowano i oznaczono relacje występujące pomiędzy przypadkami użycia?	0..3	1	
<b>2. Kompletność</b>	0..3	1/3	
<b>3. Minimalność</b>	0..3	1/3	
<b>4. Nadmiarowość</b>	0..3	1/3	
<b>5. Samo tłumaczenie:</b> czy nazwy wprowadzone do diagramu prawidłowo „przenoszą” semantykę opisywanych przez nie aktorów, operacji oraz relacji?	0..3	1/3	
<b>6. Organizacja/czytelność:</b> oceniane jest rozmieszczenie przypadków użycia względem siebie, rozmieszczenie aktorów względem przypadków, wprowadzenie do diagramu przecinających się linii (co pogarsza percepcję), a także czy litery w nazwach nie są zbyt małe oraz czy wyróżniono nazwy aktorów i przypadków (poprzez pogrubienie liter)	0..6	1/3	
<b>7. Znajomość notacji języka modelowania:</b> m.in. sprawdzane jest czy diagram ma postać obramowaną? czy wprowadzono do diagramu nagłówki z wyróżnikiem i nazwą diagramu? czy symbole dla oznaczania relacji dziedziczenia pomiędzy aktorami są prawidłowe? czy poprawnie oznaczono aktorów abstrakcyjnych? czy poprawnie oznaczono nie tylko relacje występujące pomiędzy aktorami (aktorami z otoczenia fragmentu systemu lub systemu traktowanego jako całość) a przypadkami użycia, ale także relacje występujące pomiędzy	0..3	1/3	

przypadkami użycia?			
<b>8. Ocena łączna</b>	0..22	1	

W Tab. 1-10 zamieszczono propozycję zbioru miar wykorzystywanych do oceny scenariuszy dla przypadków użycia.

Tab. 1-10 Tabela miar dla oceny scenariusza przypadku użycia

Przedmiot oceny	Zakres ocen	Waga	Ocena
<b>1. Poprawność</b> (suma ocen z pozycji 1.1 do 1.4)	0..10	1	
1.1 <i>specyfikacja warunków</i> : oceniany jest zarówno fakt wprowadzenia do scenariusza obu rodzajów warunków (początkowego i końcowego), jak też i poprawność sformułowań dla wyrażeń liczycznych określających te warunki	0..2	1	
1.2 <i>podział scenariusza</i> : odejmowane są punkty za brak podziału scenariusza na przepływ główny (jest tylko jeden taki) i przepływy alternatywne; oceniana jest także poprawność wzajemnego uporządkowania (numeracji) kolejnych kroków w obu rodzajach przepływów	0..3	1	
1.3 <i>zgodność zawartości przepływów z informacjami zawartymi w tekście wymagań i podstawowym celem scenariusza</i> : odejmowane są punkty za wprowadzenie do scenariusza operacji związanych nie z dialogiem aktora z systemem, ale z etapem wykonawczym danego przypadku – skutkiem czego scenariusz w większym stopniu przypomina opis algorytmu wykonania danego przypadku użycia niż opis dialogu aktora z systemem	0..3	1	
1.4 <i>forma specyfikowania kolejnych kroków</i> : poprawność sformułowań jest oceniana zarówno dla przepływu głównego, jak i dla przepływów alternatywnych	0..2	1	
<b>2. Kompletność</b>	0..3	1/3	

<b>3. Minimalność</b>	0..3	1/3	
<b>4. Nadmiarowość</b>	0..3	1/3	
<b>5. Samo tłumaczenie:</b> oceniane jest czy nazwy wprowadzone do scenariusza prawidłowo „przenoszą” semantykę opisywanych przez nie bytów	0..3	1/3	
<b>6. Organizacja/czytelność</b>	0..3	1/3	
<b>7. Ocena łączna</b>	0..15	1	

### 1.7.2 Metryka do oceny jakości rozwiązań w grupie: *Analiza strukturalna*

W Tab. 1-11 przedstawiono propozycję elementów składowych metryki, która może być zastosowana do oceny zadań projektowo/problemowych w grupie tematycznej *Analiza strukturalna* (czyli dla oceny diagramu klas ze schematu pojęciowego lub projektowego). Poszczególne miary mogą być wykorzystywane także oddzielnie dla oceny struktur stanowiących rozwiązania do poleceń prostych.

Tab. 1-11 Tabela miar do oceny diagramu klas

Przedmiot oceny	Zakres ocen	Waga	Ocena
<b>1. Poprawność</b> (suma ocen z punktów 1.1 - 1.7)	0..20		
1.1 <i>identyfikacja klas</i> : odejmowanie punktów za na przykład umieszczenie klasy zamiast atrybutu czy asocjacji (także w sytuacji odwrotnej), wprowadzenie do diagramu aktora systemu (?), błędą nazwę klasy (z reguły rzeczownik w liczbie pojedynczej), brak nazwy nadklasy w nazwie podklasty	0..3	1	
1.2 <i>identyfikacja atrybutów oraz specyfikacja rodzaju każdego z nich (opcjonalny, powtarzalny, pochodny, klasowy)</i> : odejmowanie punktów np. za wykorzystanie atrybutu zamiast asocjacji (lub odwrotnie) czy też za zbyt szczegółową specyfikację (np. specyfikację typu atrybutu w języku programowania)	0..3	1	

<p>1.3 <i>identyfikacja metod i specyfikacja ich rodzajów (obiektowe, klasowe, abstrakcyjne)</i>: odejmowanie punktów, np. za umieszczenie na diagramie metod generycznych (takich, jak: <i>usuń</i>, <i>utwórz</i>, <i>czytaj</i>, <i>modyfikuj</i>), za wprowadzenie metody zamiast atrybutu pochodnego (już w pierwszym kroku analizy), za umieszczenie na diagramie metody pochodnej lub opcjonalnej (nie istnieją!), za zbyt szczegółową specyfikację (podobnie jak dla atrybutów), za umieszczanie metod w miejscu ich potencjalnego wywołania (np. w klasie modelującej aktora systemu) zamiast w tej klasie, której obiekty przechowują dane potrzebne do wykonania metody</p>	0..3	1	
<p>1.4 <i>poprawna identyfikacja/budowa struktur generalizacji -specjalizacji</i>: oceniane jest czy prawidłowo zidentyfikowano klasy połączone związkami dziedziczenia (należy opierać się wyłącznie o podobieństwo semantyk, a nie o podobieństwo atrybutów czy metod w klasach)? czy rodzaje dziedziczenia (<i>rozłączne</i>, <i>nierozłączne</i>, <i>kompletne</i>, <i>niekompletne</i>, <i>jednoaspektowe</i>, <i>wieloaspektowe</i>, <i>jednokrotne</i>, <i>wielokrotne</i>, <i>dynamiczne</i>, <i>elipsa</i>) zostały określone zgodnie z informacjami zawartymi w tekście wymagań? czy atrybuty i metody w ramach jednej hierarchii zostały poprawnie rozmieszczone?</p> <p>punkty są odejmowane za brak hierarchii w ogóle, nieprawidłową strukturę hierarchii (np. za umieszczenie klas o różnej semantycie w jednej hierarchii, zamianę ról podklasa-nadklaśa, obecność pętli w strukturze), wykorzystywanie tzw. <i>obejść</i>, czyli ograniczeń środowiska implementacji, jak na przykład użycie asocjacji, agregacji czy kompozycji dla modelowania dziedziczenia nietrywialnego (dziedziczenie <i>nierozłączne</i>, <i>wielokrotne</i>, <i>wieloaspektowe</i>, <i>dynamiczne</i>) – tego rodzaju aktywność powinna mieć miejsce na etapie projektowania, a nie na etapie analizy (w modelu wodoszadowym wytwarzania oprogramowania)</p>	0..3	1	

1.5 <i>identyfikacja asocjacji:</i> czy nazwy asocjacji wybrano stosownie do informacji o związkach między bytami (zawartych w tekście wymagań)? czy liczności asocjacji zostały określone prawidłowo? czy wykorzystano role i atrybuty asocjacji (lub klasy asocjacji)?  punkty są odejmowanie za uchybienia w ww. kwestiach, a także za modelowanie czynności z wykorzystaniem asocjacji, wprowadzanie asocjacji redundantnych (np. jako skutek niewykorzystywania dziedziczenia asocjacji czy nie oznaczania asocjacji pochodnych) oraz wprowadzanie do diagramu elementów przynależnych do fazy projektowania (jak np. asocjacji skierowanych czy kluczy obcych zamiast asocjacji)	0..3	1	
1.6 <i>identyfikacja agregacji, kompozycji i asocjacji kwalifikowanej</i>	0..2	1	
1.7 <i>wprowadzanie ograniczeń i komentarzy:</i> ma tu znaczenie ich liczba oraz w jakiej postaci zostały umieszczone – dla osób rozpoczynających naukę podstawową trudnością jest zazwyczaj podjęcie decyzji, czy dana informacja powinna być w ogóle wprowadzona do diagramu, a jeśli tak, to czy w postaci komentarza czy ograniczenia	0..3	1	
<b>2. Kompletność</b>	0..3	1	
<b>3. Minimalność</b>	0..3	1/3	
<b>4. Nadmiarowość:</b>	0..3	1/3	
<b>5. Samo tłumaczenie:</b> oceniane jest czy nazwy elementów diagramu dobrze opisują semantykę zarówno modelowanych bytów z dziedziny problemowej, jak i semantykę relacji zachodzących pomiędzy bytami oraz czynności realizowanych na bytach	0..2	1	
<b>6. Organizacja diagramu</b>	0..5	1	
<b>7. Czytelność</b>	0..3	1/3	
<b>8. Znajomość notacji języka modelowania:</b> m.in. czy wykorzystywano formę obramowaną	0..3	1/3	

diagramu z nagłówkiem zawierającym wyróżnik oraz nazwę diagramu? czy prawidłowo oznaczono rodzaje dziedziczenia? czy prawidłowo oznaczono liczności asocjacji?			
<b>9. Ocena łączna</b>	0..34	1	

### 1.7.3 Metryka do oceny jakości rozwiązań w grupie: *Analiza dynamiczna*

W Tab. 1-12 zamieszczono propozycję zbioru miar wykorzystywanych do oceny diagramów dynamicznych.

Tab. 1-12 Tabela miar dla oceny diagramu aktywności, stanu, interakcji

Przedmiot oceny	Zakres ocen	Waga	Ocena
<b>1. Poprawność</b> (suma ocen z pozycji 1.1 do 1.2)	0..6	1	
1.1 <i>identyfikacja wierzchołków w grafach aktywności i w maszynach stanowych; identyfikacja uczestników interakcji:</i>  w maszynach stanowych często niepotrzebnie, niezgodnie z tekstem wymagań, wprowadzane są stany początkowe i końcowe  częstą przyczyną błędów jest też niezrozumienie istoty różnicy pomiędzy stanami początkowymi i końcowymi na diagramach aktywności i w maszynach stanowych  na diagramach interakcji problemem jest identyfikacja uczestników interakcji zgodna ze strukturą zaproponowaną na schematach ją opisujących (czyli na schemacie pojęciowym lub projektowym)	0..3	1	
1.2 <i>etykietowanie przejść pomiędzy wierzchołkami grafów aktywności/stanów; oznaczanie rodzajów komunikatów na diagramach interakcji:</i> m.in. brakuje oznaczeń kierunków przejść, co jest wymagane zarówno na diagramach aktywności, jak i w maszynach stanowych, ponieważ w obu	0..3	1	

przypadkach wykorzystywane są grafy skierowane; przejścia warunkowe często nie są opatrywane warunkami			
<b>2. Kompletność</b>	0..3	1/3	
<b>3. Minimalność</b>	0..3	1/3	
<b>4. Nadmiarowość</b>	0..3	1/3	
<b>5. Samo tłumaczenie:</b> oceniane jest czy nazwy wprowadzone do scenariusza prawidłowo „przenoszą” semantykę opisywanych przez nie bytów	0..3	1/3	
<b>6. Organizacja/czytelność</b>	0..3	1/3	
<b>7. Znajomość notacji języka modelowania:</b> m.in. oceniane jest czy wprowadzono do diagramu ramę z nagłówkiem (z wyróżnikiem i nazwą diagramu)? czy symbole dla oznaczania wierzchołków i etykiet przejść w grafach aktywności/maszynach stanowych są prawidłowe? czy rodzaje i etykiety komunikatów na diagramach interakcji są zgodne z informacjami zawartymi w tekście wymagań?	0..3	1/3	
<b>8. Ocena łączna</b>	0..12	1	

## 1.8 Wzorzec dokumentacji projektowej

W Tab. 1-13 zamieszczono wzorzec dokumentacji projektowej, gdzie dla kolejnych pozycji zawarto wskazówki objaśniające, jakiego rodzaju informacje powinny zostać zamieszczone w efekcie przeprowadzonej analizy funkcjonalnej i analizy strukturalnej. Obie analizy mają być przeprowadzone dla systemu o wielkości od dziesięciu do piętnastu klas, w oparciu o wymagania dostarczone przez osobę nauczaną/uczącą się, będącą autorem projektu.

Praca projektowa z założenia ma być realizowana przez każdą z osób nauczanych/uczących się oddzielnie i oddawana na zakończenie semestralnego kursu.

Tab. 1-13 Wzorzec dokumentacji projektowej

<i>Nagłówek</i>	nazwa systemu, dane autora, data utworzenia
-----------------	---

	dokumentacji, numer wersji
<i>Dziedzina problemowa</i>	krótko o tym, gdzie projektowany system mógłby znaleźć zastosowanie, w jakiej dziedzinie/dziedzinach działalności ludzkiej
<i>Cel</i>	krótko (i bardzo ogólnie) dlaczego zdecydowano się na budowę tego systemu, jakiego rodzaju działalność ma ułatwić jego wykorzystywanie, w jaki sposób mógłby on wspomagać przyszłych użytkowników w ich pracy w dziedzinie problemowej <i>uwaga:</i> nie należy mylić zawartości tego punktu z zawartością punktu następnego, którego zadaniem jest określenie zakresu odpowiedzialności systemu
<i>Zakres odpowiedzialności systemu</i>	krótko o tym, jakie usługi mają być dostarczone przez projektowany system, innymi słowy ten punkt zawiera bardzo uproszczone omówienie oczekiwanej funkcjonalności ze zwróceniem uwagi na funkcjonalności najważniejsze z perspektywy celu budowy systemu
<i>Użytkownicy systemu</i>	należy wymienić listę potencjalnych użytkowników systemu (bez wymieniania przypisanych do nich funkcjonalności) – np. wykorzystując diagram kontekstowy
<i>Wymagania użytkownika</i>	<i>wymagania użytkownika</i> nazywane też czasami <i>wymaganiami wstępymi</i> na system, należy podzielić na 3 części (oddzielone od siebie np. jedną pustą linią) <i>część pierwsza</i> ma dotyczyć omówienia struktury tego fragmentu dziedziny problemowej, którym zajmuje się system; należy w niej zamieścić zarówno specyfikacje bytów z dziedziny problemowej (łącznie ze szczegółowymi opisami ich własności), jak również informacje o relacjach zachodzących pomiędzy tymi bytami; część pierwsza stanowi podstawę, w oparciu o którą należy skonstruować schemat pojęciowy systemu <i>część druga</i> powinna specyfikować oczekiwana funkcjonalność systemu; zbiór udostępnianych usług powinien tworzyć spójną całość, realizowaną na

	<p>opisanej w części pierwszej strukturze (tzn. na schemacie pojęciowym); specyfikację usług można rozpoczęć np. sformułowaniem (po pustej linii oddzielającej ją od części pierwszej), takim jak: <i>Oczekuje się, że system będzie wspomagał użytkowników w...</i></p> <p>każdą ze zidentyfikowanych usług należy powiązać z użytkownikiem (lub grupą użytkowników), których działalność ma wspierać; użytkownicy wymienieni w tej części <i>Wymagań użytkownika</i> muszą też występować na zbiorczej liście użytkowników systemu umieszczonej w punkcie <i>Użytkownicy systemu</i></p> <p>wskazane jest, aby na liście usług znalazły się takie, których realizacja będzie wymagała wykorzystania metod klasowych oraz takie, dzięki którym będzie możliwe zademonstrowanie hierarchii dla aktorów z otoczenia systemu</p> <p>część druga posłuży za podstawę do zbudowania diagramu/diagramów przypadków użycia</p> <p>część trzecia, podobnie jak poprzednio, powinna być oddzielona pustą linią od części drugiej</p> <p>należy wymienić co najmniej kilka przykładowych ograniczeń (minimum 3), które system powinien wypełniać, np. można określić środowisko sprzętowe czy programowe, w którym miałyby pracować system, ustalić wymagania na niezawodność, wydajność itd.</p>
<i>Wymagania funkcjonalne</i>	<p>należy skonstruować diagram kontekstowy oraz model przypadków użycia zgodny z funkcjonalnością wyspecyfikowaną w drugiej części punktu <i>Wymagania użytkownika</i></p> <p>model przypadków użycia (złożony z jednego lub więcej diagramów przypadków) powinien być utworzony z perspektywy użytkownika z otoczenia systemu, czyli na najwyższym poziomie abstrakcji</p> <p>ze względu na czytelność diagramu, nie jest zalecane wprowadzanie zbyt dużej liczby przypadków do jednego diagramu; nie jest zalecane również prezentowanie rozbudowanych zależności pomiędzy przypadkami, w tym włączanie przypadków z niższych poziomów abstrakcji do poziomów</p>

	wyższych, ponieważ potencjalnego użytkownika nie interesuje wewnętrzna organizacja funkcji systemu
<i>Opis struktury systemu (schemat pojęciowy)</i>	w oparciu o opis struktury systemu umieszczony w części pierwszej punktu <i>Wymagania użytkownika</i> , należy zbudować schemat pojęciowy (diagram klas); schemat powinien zawierać metody, od których rozpocznie się realizacja każdej z funkcjonalności wyspecyfikowanych w modelu przypadków użycia
<i>Wymagania niefunkcjonalne</i>	należy określić ograniczenia, przy których ma pracować system (wymienione w trzeciej części punktu <i>Wymagania użytkownika</i> ); dla każdego z ograniczeń należy podać propozycję pojedynczej miary lub metryki (będącej miarą złożoną), której zadaniem będzie ułatwienie dokonywania pomiarów w celu kontroli wypełniania ograniczeń i szacowania jakości produktu
<i>Opis przyszłej ewolucji systemu</i>	krótko o tym, czy planowana jest w przyszłości rozbudowa systemu i jeśli tak, to których jego elementów miałaby dotyczyć i w jakim zakresie
<i>Słownik pojęć</i>	ta pozycja zawiera definicje pojęć z dziedziny problemowej; pojęcia w słowniku należy uporządkować alfabetycznie

W Tab. 1-14 zamieszczono rozszerzenie wzorca dokumentacji projektowej, o prezentację wyników analizy dynamicznej oraz o prezentację propozycji architektury systemu, wynikającej zarówno z podziału funkcjonalności, jak i podziału struktury warstwy biznesowej na moduły.

Tab. 1-14 Rozszerzenie wzorca dokumentacji projektowej

<i>Analiza dynamiczna dla wybranego przypadku użycia</i>	wykorzystując informacje zawarte w punkcie <i>Wymagania użytkownika</i> należy przeprowadzić analizę dynamiczną dla wybranego nietrywialnego przypadku użycia, czyli w oparciu o przepływy zdarzeń wyspecyfikowane w scenariuszu napisanym dla tego przypadku, trzeba zbudować diagram aktywności oraz co najmniej oba podstawowe diagramy interakcji (sekwencji i komunikacji)
--	---

	<p>analiza wybranego przypadku użycia powinna zakończyć się podsumowaniem, które metody i w jakich klasach są niezbędne dla realizacji tego przypadku: zidentyfikowane metody należy wprowadzić do schematu pojęciowego</p> <p>jeśli w efekcie przeprowadzonej analizy dynamicznej pojawią się nowe atrybuty, klasy czy asocjacje, należy również wprowadzić je do schematu pojęciowego</p>
<i>Analiza dynamiczna dla wybranej klasy obiektów</i>	<p>należy zbudować maszynę stanową dla wybranej klasy obiektów</p> <p>podobnie jak poprzednio, analiza powinna zakończyć się podsumowaniem i wprowadzeniem nowych elementów do schematu pojęciowego</p>
<i>Diagram przypadków użycia rozszerzony o elementy wynikające z analizy scenariusza dla wybranego nietrywialnego przypadku użycia</i>	<p>dla zilustrowania relacji zachodzących pomiędzy przypadkami, w oparciu o scenariusz dla wybranego nietrywialnego przypadku użycia i wykonaną wcześniej analizę dynamiczną, należy dokonać podziału tego przypadku na podprzypadki, uwzględniając wyniki w skonstruowanym uprzednio modelu przypadków</p> <p>taka działalność stanowi podstawę do wyróżniania modułów oprogramowania, czyli do określania elementów architektury systemu</p> <p>w celu identyfikacji bloków ponownego użycia, pożąданie byłoby, aby analiza dynamiczna została przeprowadzona dla więcej niż jednego nietrywialnego przypadku użycia</p>
<i>Diagram pakietów dla modelu przypadków użycia</i>	w oparciu o wyniki analizy dynamicznej, należy przeprowadzić dekompozycję modelu przypadków – w celu ułatwienia realizacji aktywności związanych z projektowaniem interfejsu użytkownika
<i>Diagram pakietów dla schematu pojęciowego</i>	należy przeprowadzić dekompozycję schematu pojęciowego, aby wyróżnić pakiety, stanowiące na etapie późniejszym bazę dla określania architektury warstwy biznesowej,
<i>Diagramy komponentów</i>	należy wyróżnić komponenty zarówno dla interfejsu użytkownika, jak i dla warstwy biznesowej, łącznie

	ze specyfikacją interfejsów i operacji związanych z każdym z nich
--	---

## 1.9 Podsumowanie

Niniejsze opracowanie jest przeznaczone głównie dla osób interesujących się modelowaniem i/lub projektowaniem, a w szczególności dla tych, którzy zajmują się analizą wymagań na systemy informacyjne i aktualnie wykorzystują albo planują wykorzystywać w przyszłości popularny obiektowy graficzny język UML. Nie mniej jednak wydaje się, że może też być to opracowanie użyteczne dla osób prowadzących działalność związaną z modelowaniem i/lub projektowaniem w ramach innych dziedzin problemowych. Przez osoby *interesujące się* rozumiane są tu zarówno osoby uczące się, nauczające, jak i osoby podejmujące ten rodzaj aktywności w życiu zawodowym. Uogólniając jeszcze bardziej, wydaje się, że zaprezentowana metodyka, zarówno jeśli chodzi o schemat struktury zasobów, jak i określenie sposobów postępowania w trakcie nauczania, z powodzeniem mogłaby zostać wykorzystana w nauczaniu w innych dziedzinach, niekoniecznie związanych z informatyką.

Opracowanie może stanowić także pomoc dydaktyczną do przedmiotów związanych z analizą i projektowaniem systemów informacyjnych. W aspekcie tego przeznaczenia, szczególnie użyteczne wydają się być rozdziały poświęcone przykładowym jedenastu zadaniom, dla których dołączono rozwiązania i omówienia błędów (dla każdego z zadań oddziennie). Zadania, zróżnicowane pod względem trudności, zostały oparte na opisach spójnych fragmentów dziedzin problemowych wybranych do analizy; warto nadmienić, że analizowane fragmenty, o wielkości od 10 do 15 klas, mogą stanowić bazę do budowy realnego, użytecznego oprogramowania. Dla wszystkich zamieszczonych zadań kolejno przeprowadzono mniej lub bardziej zaawansowaną analizę funkcjonalną, strukturalną oraz dynamiczną. Dla analizy dynamicznej, w większości przypadków ograniczono się do pojedynczych diagramów aktywności, maszyn stanowych i diagramów interakcji. Ponadto, w opracowaniu przedstawiono niektóre z technik wykorzystywanych w fazie projektowania do eliminowania tzw. ograniczeń środowiska implementacji oraz zaprezentowano transformację fragmentów schematów pojęciowych do diagramów encja-związek z użyciem trzech sposobów opisu tego rodzaju transformacji.

Ponieważ zadania w większości obejmują swym zakresem początkowe etapy prac realizowanych w trakcie budowy oprogramowania (analizę systemową i wstępne projektowanie), zamieszczone diagramy w sposób oczywisty wymagają dalszego opracowywania, czyli między innymi poprawiania popelnionych błędów, rozszerzania czy uszczegóławiania modeli/diagramów, tak aby w efekcie uzyskać dla każdego z zadań spójną całość, wypełniającą zarówno wymagania funkcjonalne, jak i ograniczenia wynikające z treści zadania. Ogół prac w trakcie budowy systemu

powinien być realizowany w trakcie kolejnych nawrotów w iteracyjno-przyrostowym modelu tworzenia oprogramowania [39]. Jedno z zadań, jest nim zadanie *Hurtownia*, opisuje proces tworzenia przykładowego planu iteracji, który jest tworzony w oparciu o modele zbudowane w pierwszym podejściu do analizy systemowej i może być modyfikowany w kolejnych iteracjach.

Zawartość każdego pojedynczego zadania, jak i wzajemny układ wszystkich zadań jest próbą określenia reguł postępowania dla nauczania przeprowadzania analizy wymagań oraz tworzenia wstępnego projektu (na poziomie pojęciowym i logicznym) dla systemów informacyjnych. Zasadnicze aktywności, które stanowiły podstawę dla badań, to: analiza wymagań łącznie z konstruowaniem schematu pojęciowego opisującego „biznes”, którego dany system dotyczy (opis ten powinien być niezależny od środowisk implementacji) oraz projektowanie, gdzie do podstawowych aktywności należą budowa schematu projektowego (uwzględniającego ograniczenia wybranego do implementacji środowiska informatycznego) oraz zaproponowanie schematu bazy danych (dla encji biznesowych). Próba identyfikacji reguł dla nauczania ww. aktywności, a także określenie rodzajów i struktury materiałów pomocniczych, które mogłyby znacząco wspomagać ten proces, to jedno z najważniejszych zadań, jakie miała wypełnić ta publikacja.

Innym ważnym celem – ze względu na powszechnie wykorzystanie UML w organizacjach związanych z produkcją oprogramowania – było użycie w zadaniach jak największej liczby rodzajów diagramów dostarczanych przez ten język. Przeważająca większość popularnych pozycji wydawniczych o UML bazuje na przykładach, które są albo na zbyt wysokim poziomie abstrakcji, albo odnoszą się do bardzo ograniczonego wycinka dziedziny problemowej. Dotyczy to nie tylko informacji związanej z modelowaniem i/lub projektowaniem w ogóle, ale także z wykorzystywaniem konstrukcji UML. Nadmierne uproszczenie przykładów skutkuje zazwyczaj niemożnością zastosowania nabitej wiedzy w praktyce.

Kolejnym celem była próba zilustrowania sposobu budowy oprogramowania z wykorzystaniem najlepszych praktyk zidentyfikowanych na potrzeby inżynierii oprogramowania, czyli budowę oprogramowania w oparciu o model iteracyjno-przyrostowy na bazie architektury opartej na komponentach z wykorzystaniem modelowania wizualnego, co oznacza budowę elementów bardziej złożonych z elementów prostszych, posiadających pewien potencjał ponownego użycia, w sposób przyrostowy. Problem ten dobrze podsumowuje sformułowanie użyte w publikacji poświęconej metodyce RUP: „jak zjeść słonia? kawałek po kawałku” [49]. Zagregowana całość stanowi spójny fragment i może zostać wykorzystana jako element innej, większej całości w kolejnym kroku. Potencjalne elementy składowe najłatwiej jest zidentyfikować już na wczesnych etapach budowy oprogramowania, w trakcie prac analitycznych i projektowych; w zadaniu *Hurtownia* (rozdział 11) podjęto próbę zilustrowania takiego podejścia.

## Spis treści rozdziału

Rozdział 2 Fundacja “UFO” .....	62
2.1 Tekst wymagań.....	62
2.2 Analiza funkcjonalna .....	64
2.2.1 Diagram kontekstowy .....	64
2.2.2 Model przypadków użycia.....	65
2.2.3 Scenariusze dla wybranych przypadków użycia łącznie z podziałem przypadku na podprzypadki.....	67
2.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	73
2.3 Analiza strukturalna.....	75
2.3.1 Tekst wymagań z komentarzami do analizy strukturalnej.....	75
2.3.2 Schemat pojęciowy .....	80
2.3.3 Analiza elementów pochodnych .....	85
2.3.4 Analiza wartości początkowych .....	86
2.3.5 Analiza wartości granicznych .....	88
2.3.6 Prezentacja struktur elementów złożonych.....	89
2.3.7 Wykorzystanie typu <i>enumeration</i> dla określania zbioru wartości .....	91
2.3.8 Zadania podstawowe .....	92
2.3.9 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	95
2.3.10 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji .....	98
2.3.11 Schemat relacyjny dla fragmentu diagramu klas .....	100
2.4 Analiza dynamiczna .....	105
2.4.1 Zadania podstawowe .....	106
2.4.2 Omówienie zmian na schemacie pojęciowym i schemacie projektowym wynikających z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań.....	131
2.5 Zadania do samodzielnej pracy .....	133
2.6 Podsumowanie zadania.....	134

## Rozdział 2

# Fundacja “UFO”

Ewa Stemposz, Alina Stasiecka, Andrzej Jodłowski

### 2.1 Tekst wymagań

Fundacja „UFO” zajmuje się m.in. katalogowaniem kontaktów ludzi z niezidentyfikowanymi obiektami latającymi (ang. *Unidentified Flying Object*), w czym ma jej pomóc projektowany system komputerowy. Pracownik fundacji, otrzymując, drogą telefoniczną lub za pośrednictwem e-maila, zgłoszenie dotyczące prawdopodobnego kontaktu z UFO, podejmuje decyzję, czy wprowadzić do systemu nowy kontakt, czy też dołączyć kolejnego uczestnika do wprowadzonego już wcześniej kontaktu, bądź też w ogóle zignorować zgłoszenie.

1. Zgodnie z projektem, w systemie mają być przechowywane dane osobowe i numery telefonów (co najmniej 1) zarówno uczestników kontaktów z UFO, jak i ekspertów fundacji. Ekspert może być również uczestnikiem kontaktu.
2. Ekspert charakteryzowany jest przez listę posiadanych przez niego specjalizacji (co najmniej 1) oraz miesięczne wynagrodzenie podstawowe. Wynagrodzenie to nie może jednorazowo wzrosnąć o więcej niż 10% poprzedniej wartości oraz nie może przekroczyć wartości maksymalnej, jednakowej dla wszystkich ekspertów.
3. Informacje, które należy przechowywać dla każdego kontaktu, to: opis, data, godzina, czas trwania, miejsce, kto był jego uczestnikiem (mogło być ich wielu) i jaka jest relacja z tego kontaktu każdego z jego uczestników (w postaci tekstu). Relacja jest tu rozumiana jako zeznanie naocznego świadka tego wydarzenia.
4. Kontakty dzielą się ze względu na typ oraz na średni stopień wiarygodności. Pierwszy podział, który jest kompletny i rozłączny, obejmuje m.in. obserwacje i uprowadzenia. Obserwacja rozumiana jest jako zauważenie przez uczestnika czegoś, co podejrzewa, że jest niezidentyfikowanym obiektem latającym. Dla obserwacji ma być przechowywana liczba uczestników kontaktu, a dla uprowadzeń – opis badania medycznego, o ile takie zostało przeprowadzone na danym uczestniku. Dla drugiego podziału kontaktów, czyli podziału ze względu na średni stopień wiarygodności, wyróżniono następujące grupy: kontakty o bardzo wysokim średnim stopniu wiarygodności ( $\geq 80\%$ ), kontakty o wysokim średnim stopniu wiarygodności ( $\geq 70\%$  i  $< 80\%$ ) itd. Podział ten jest, podobnie jak poprzedni, kompletny. Wprowadzenie nowego kontaktu automatycznie skutkuje przypisaniem go do grupy kontaktów o najniższym

średnim stopniu wiarygodności (ponieważ średni stopień wiarygodności, przy braku w tym momencie jakichkolwiek opinii ekspertów, wynosi 0%).

5. W systemie należy przechowywać informację, któremu ekspertowi przydzielono do zaopiniowania dany kontakt. Jest możliwe, że kontakt został już zarejestrowany, ale nikt go nie zaopiniował. Przechowywane mają być informacje tylko o tych ekspertach, którym przydzielono co najmniej jeden kontakt.
6. Dla opinii wystawionej przez eksperta należy przechowywać dane takie jak: numer (unikatowy w ramach danego kontaktu), data wystawienia, koszt (określony przez eksperta) oraz treść. Wystawiając opinię, ekspert określa również stopień wiarygodności kontaktu (wyrażony w procentach) oraz ewentualnie wprowadza informacje o zdjęciach wykonanych na miejscu zaistnienia kontaktu, o ile takie wykonał. Zdjęcia mogą też być dołączone później, w innym terminie. Dla każdego z nich należy pamiętać: opis, datę wykonania oraz adres pliku (tylko dla zdjęć przechowywanych w wersji cyfrowej). Ze względu na ograniczenia finansowe fundacji, każdy z ekspertów może mieć w danym momencie przydzielone nie więcej niż 3 kontakty do zaopiniowania.
7. Planuje się, aby dla każdego kontaktu było możliwe obliczenie średniego stopnia wiarygodności, przy czym dla wszystkich typów kontaktów oprócz obserwacji algorytm ma być oparty na tej samej prostej formule: *średni stopień wiarygodności jest równy średniej ze stopni wiarygodności określonych przez ekspertów, którzy opiniowali kontakt*; w przypadku obserwacji dodatkowo należy uwzględnić liczbę uczestników kontaktu. Średni stopień wiarygodności jest wyliczany po każdym wprowadzeniu nowej opinii o kontakcie, co może skutkować zmianą przydziału kontaktu do innej kategorii.
8. Z każdym kontaktem związana jest teczką z dokumentacją, zawierającą różnego rodzaju papierowe dokumenty. W systemie mają być przechowywane następujące informacje o niej: numer (unikatowy), data założenia (równoznaczna z terminem rozpoczęcia prac nad danym kontaktem), data zakończenia tych prac oraz status przyjmujący jedną z następujących wartości: „w trakcie opracowywania”, „opracowana”, „do ponownej weryfikacji”, „zweryfikowana pozytywnie”.
9. Ponadto ma być pamiętane, kiedy i któremu ekspertowi przydzielono do weryfikacji daną teczkę z dokumentacją kontaktu. Weryfikacja teczki polega na sprawdzeniu formalnej poprawności dokumentów w niej zawartych (bez pośrednictwa systemu), a następnie zarejestrowaniu w systemie informacji takiej jak: kto i kiedy weryfikował daną teczkę, czy proces ten zakończył się sukcesem oraz komentarz eksperta. Jeśli ekspert miał zastrzeżenia do któregoś z dokumentów teczki, to otrzymuje ona status „do ponownej weryfikacji”. Teczka może być weryfikowana przez wielu ekspertów (również tych samych, którzy uprzednio to robili), ale nie mogą oni należeć do grupy opiniującej kontakt, którego teczka dotyczy. Pierwsza pozytywna weryfikacja powoduje

zmianę statusu teczki na „zweryfikowana pozytywnie”.

10. System powinien umożliwiać m.in:

- 10.1 zgłoszenie nowego kontaktu;
- 10.2 wystawienie opinii o kontakcie przez eksperta;
- 10.3 sporządzenie zestawienia kosztów wszystkich opinii wystawionych w zadany okresie przez wybranego eksperta;
- 10.4 obliczenie miesięcznego wynagrodzenia eksperta, łącznie ze sprawdzeniem, czy wystawiał opinie o kontaktach w danym miesiącu; jeśli tak, to należy obliczyć koszty tych opinii i dodać je do miesięcznego wynagrodzenia podstawowego;
- 10.5 stworzenie listy *nieaktywnych* ekspertów, czyli takich, którzy przez rok nie zaopiniowali żadnego kontaktu i nie zweryfikowali żadnej teczki z dokumentacją.

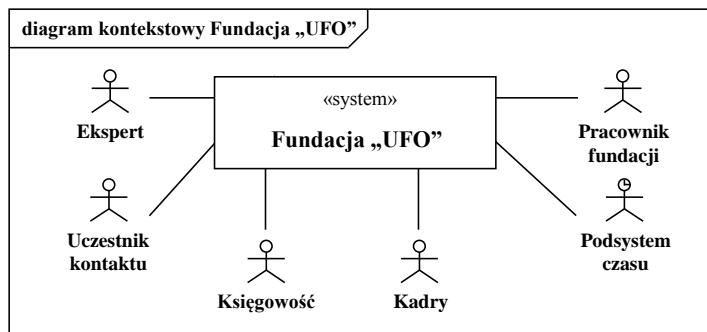
## 2.2 Analiza funkcjonalna

### 2.2.1 Diagram kontekstowy

*Polecenie:* Zbuduj diagram kontekstowy w oparciu o tekst wymagań zamieszczony w 2.1.

Diagram kontekstowy przedstawiający aktorów wybranych dla systemu w oparciu o analizę tekstu wymagań został zaprezentowany na Rys. 2-1. Należy zwrócić uwagę na fakt, że nie wszyscy aktorzy zostali bezpośrednio wymienieni w ww. tekście wymagań. Ich wyboru dokonano na podstawie znajomości dziedziny problemowej, biorąc pod uwagę możliwą ewolucję systemu w przyszłości.

*Dla przypomnienia:* Tworzenie modelu funkcjonalnego systemu wymaga zachowania spójności kolejno następujących po sobie diagramów, zarówno diagramu kontekstowego, jak i diagramu/ów przypadków użycia. Dotyczy to wyboru aktorów i wyboru przypadków użycia.



Rys. 2-1 Diagram kontekstowy

*Uwaga:* Aktor *Podsystem czasu* symbolizuje wszystkich aktorów związanych ze zdarzeniami wywoływanymi przez upływ czasu, takich jak np.: *Codziennie 23:59* czy *Początek roku*.

## 2.2.2 Model przypadków użycia

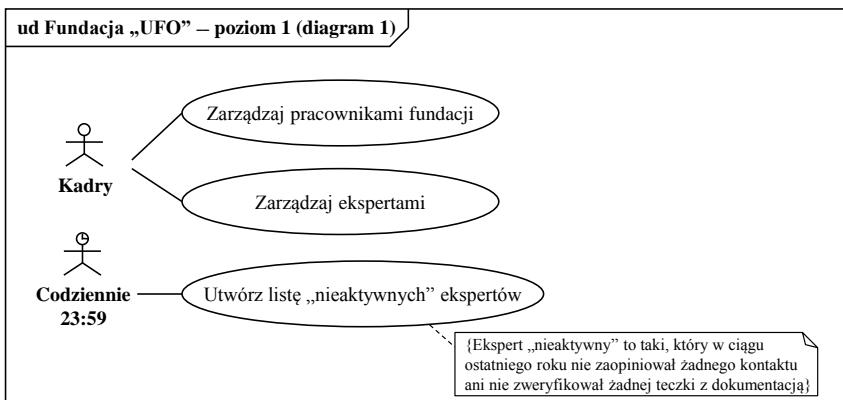
*Polecenie:* Zbuduj model przypadków użycia z perspektywy aktorów z otoczenia systemu, czyli na najwyższym poziomie abstrakcji (poziomie 1), uwzględniając przede wszystkim funkcjonalność sugerowaną w ostatnim punkcie tekstu wymagań (pkt 10 tego tekstu).

Dla przejrzystości, model przypadków użycia [36, 38, 41, 45] został opisany za pomocą dwóch diagramów przedstawionych na Rys. 2-2 i Rys. 2-3.

W nagłówkach oznaczono poziom abstrakcji dla każdego z obu rozpatrywanych diagramów. Ponieważ zostały one zbudowane z perspektywy aktorów z otoczenia systemu, jest to *poziom 1*.

Należy pamiętać o tym, że zaproponowany w kolejnych krokach wstępny zbiór usług oferowanych przez system powinien podlegać dalszej weryfikacji, modyfikacji, rozszerzeniu itd.

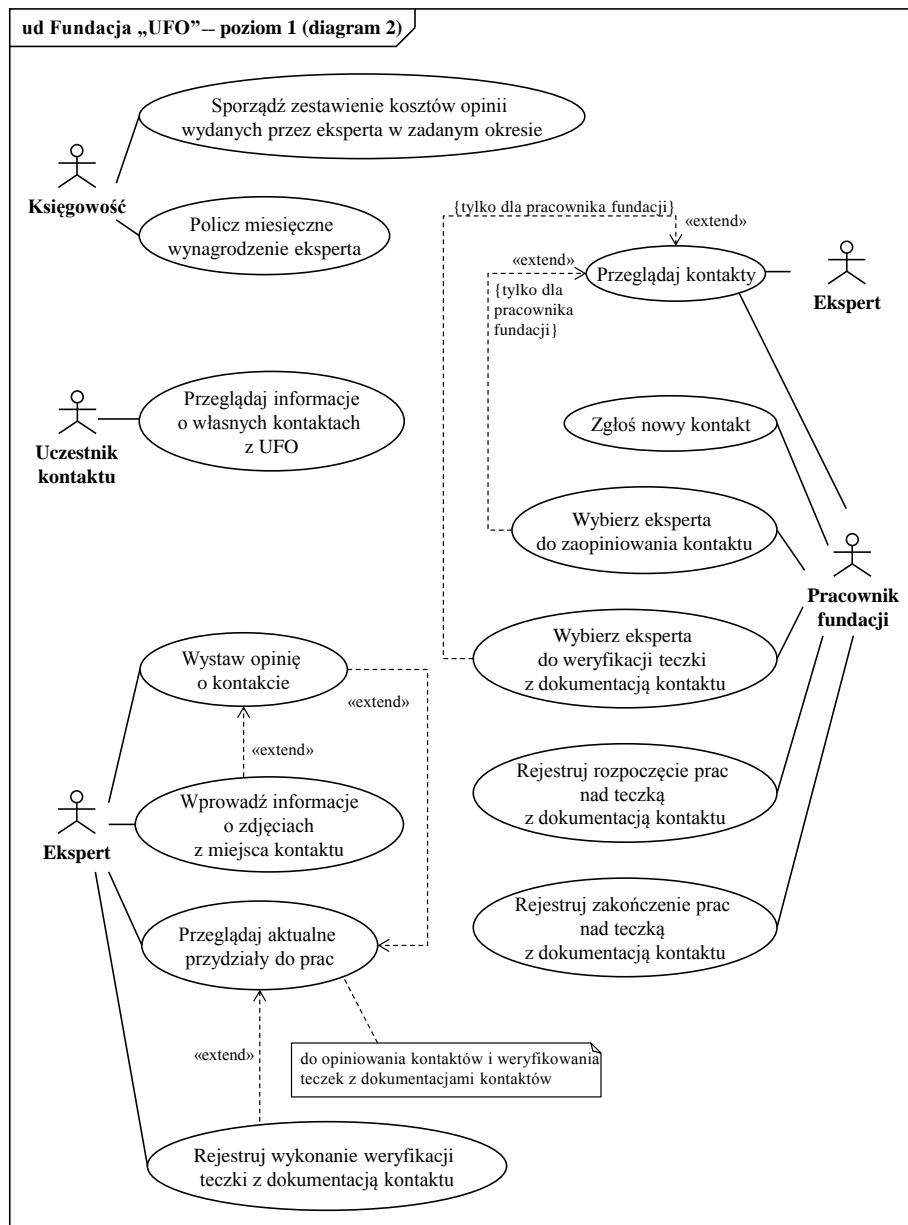
Rys. 2-2 zawiera przypadki użycia typu CRUD (Create, Read, Update, Delete). Przypadki te mogą być na tym etapie analizy w ogóle pominięte lub narysowane na oddzielnym diagramie (tak jak zostało to zrobione w proponowanym modelu).



Rys. 2-2 Model przypadków użycia (diagram 1)

*Uwaga:* Ponieważ moment wywołania przypadku związanego z tworzeniem listy nieaktywnych ekspertów nie został określony *explicite* w tekście wymagań, wybrano przykładowy.

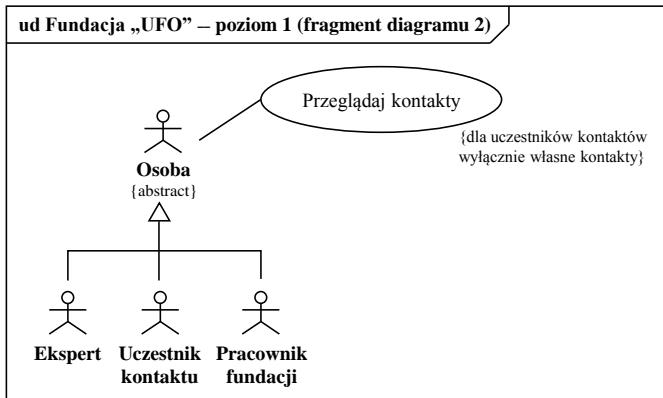
Diagram przedstawiony na Rys. 2-3 prezentuje pozostałą, określona na podstawie tekstu wymagań, funkcjonalność systemu.



Rys. 2-3 Model przypadków użycia (diagram 2)

Fragment modelu przypadków z diagramu 2 dotyczący przeglądania kontaktów można narysować, wykorzystując dziedziczenie aktorów, jak na Rys. 2-4. Do diagramu wprowadzono aktora abstrakcyjnego *Osoba*, któremu przypisano funkcjonalność *Przeglądaj kontakty*. Dzięki takiemu modelowaniu aktorzy *Ekspert*,

*Uczestnik kontaktu* oraz *Pracownik fundacji* dziedziczą wspólną funkcjonalność po aktorze *Osoba*, zachowując własne przypadki użycia. Jest to rozwiązanie bardziej elastyczne niż propozycja hierarchii aktorów, w której przykładowo *Ekspert* dziedziczyłby funkcjonalność *Uczestnika kontaktu*.



Rys. 2-4 Dziedziczenie aktorów dla funkcjonalności dotyczącej przeglądania kontaktów

### 2.2.3 Scenariusze dla wybranych przypadków użycia łącznie z podziałem przypadku na podprzypadki

*Polecenie 1:* Dla przypadku użycia *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu*:

- napisz scenariusz,
- zaproponuj podział tego przypadku na podprzypadki.

*Dla przypomnienia:* Scenariusz powinien być opisem dialogu aktora z systemem, a nie opisem algorytmu dla realizacji danego przypadku użycia. Dialog aktora z systemem, służący m.in. zidentyfikowaniu danych wejściowych/wyjściowych, niezbędnych do wykonania przypadku użycia, stanowi bazę dla prac projektanta interfejsu użytkownika, które to prace powinny i mogą być przeprowadzane jednocześnie z pracami analitycznymi.

Przykładowy scenariusz dla wybranego przypadku użycia został przedstawiony w Tab. 2-1. Przypadek ten może też być wywołany z przypadku *Przeglądaj aktualne przydziały do prac*, co nie zostało uwzględnione w scenariuszu.

Tab. 2-1 Scenariusz dla przypadku użycia *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu*

<b>Nazwa przypadku użycia</b>	Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu
<b>Warunek początkowy</b>	Co najmniej 1 teczka z dokumentacją o statusie

	„opracowana” lub „do ponownej weryfikacji” została zarejestrowana.						
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Aktor <i>Ekspert</i> uruchamia przypadek użycia.</li> <li>2. System wyświetla listę teczek z dokumentacją aktualnie przydzielonych aktorowi do zweryfikowania. Aktor wybiera z listy teczkę, dla której chce zarejestrować wykonanie weryfikacji.</li> <li>3. System odpytuje o dane przeprowadzonej weryfikacji: o komentarz i o to, czy weryfikacja zakończyła się sukcesem. Aktor wprowadza potrzebne informacje.</li> <li>4. System wyświetla komunikat o zarejestrowaniu wykonania weryfikacji teczki oraz ewentualnie o zmianie jej statusu.</li> </ol>						
<b>Alternatywne przepływy zdarzeń</b>	2a. Ani jedna teczka z dokumentacją nie została przydzielona aktorowi do zweryfikowania, system wyświetla odpowiedni komunikat i kończy przypadek użycia.						
<b>Zakończenie</b>	W dowolnym momencie.						
<b>Warunek końcowy</b>	Zostaje zarejestrowana informacja dotycząca przeprowadzenia weryfikacji teczki z dokumentacją wybranego kontaktu – przez danego eksperta oraz ewentualnie będzie też zmieniony jej status, zgodnie z regułą:						
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="padding: 5px;"><b>Wynik weryfikacji</b></th> <th style="padding: 5px;"><b>Status końcowy teczki</b></th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">sukces</td> <td style="padding: 5px;">„zweryfikowana pozytywnie”</td> </tr> <tr> <td style="padding: 5px;">brak sukcesu</td> <td style="padding: 5px;">„do ponownej weryfikacji”</td> </tr> </tbody> </table>	<b>Wynik weryfikacji</b>	<b>Status końcowy teczki</b>	sukces	„zweryfikowana pozytywnie”	brak sukcesu	„do ponownej weryfikacji”
<b>Wynik weryfikacji</b>	<b>Status końcowy teczki</b>						
sukces	„zweryfikowana pozytywnie”						
brak sukcesu	„do ponownej weryfikacji”						

*Uwaga:* Zakłada się, że data weryfikacji jest aktualną datą systemową, więc tej informacji nie trzeba wprowadzać.

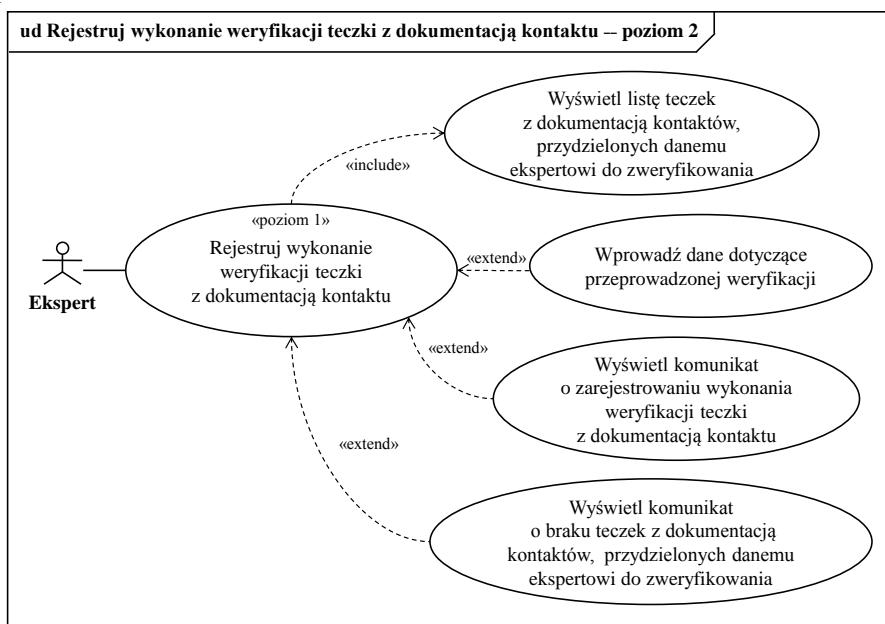
Diagramy z przykładowym podziałem przypadku *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu*, wykorzystujące różne notacje dla aktora, gdzie aktorem jest przypadek użycia, zostały przedstawione na Rys. 2-5 i Rys. 2-6. Zawarto w nich przypadki użycia powstałe na skutek podziału kodu pochodzącego z wyższego poziomu abstrakcji.

*Dla przypomnienia:* Podział danego przypadku na podprzypadki jest wykonywany w oparciu o sporządzony dla tego przypadku scenariusz i polega na podziale

bardziej złożonego kodu na kilka prostszych, dobrze wyizolowanych z otoczenia, współpracujących ze sobą części. Warto podkreślić, że robi się to w celu zmniejszenia złożoności kodu czy też identyfikacji bloków ponownego użycia, a nie w celu naśladowania określonej w scenariuszu kolejności przepływu zdarzeń.

Na pierwszym z diagramów przedstawionym na Rys. 2-5 umieszczono aktora *Ekspert* wywołującego przypadek *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu*. Przypadek ten pochodzi z najwyższego poziomu abstrakcji, na którym dostęp do usług systemu mają wyłącznie aktorzy wyróżnieni na diagramie kontekstowym.

Zbiór usług aktorów z otoczenia systemu tworzą przypadki użycia z najwyższego poziomu abstrakcji modelu. Kolejne poziomy abstrakcji, na których umieszcza się przypadki będące efektem podziału kodu, mają znaczenie wyłącznie dla poprawy efektywności prac zespołu projektowego. Wszystkie wprowadzone do diagramu przypadki użycia (oprócz tego połączonego bezpośrednio z aktorem) należą do kolejnego, niższego poziomu abstrakcji (tu: poziomu 2), gdzie rolę aktora pełni przypadek pochodzący z poziomu wyższego, co jest zgodne z definicją aktora: *aktor to sprawca przepływu zdarzeń*. Powyższa reguła obowiązuje na każdym z kolejnych poziomów.



Rys. 2-5 Diagram z przykładowym podziałem przypadku *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu* – z aktorem *Ekspert*

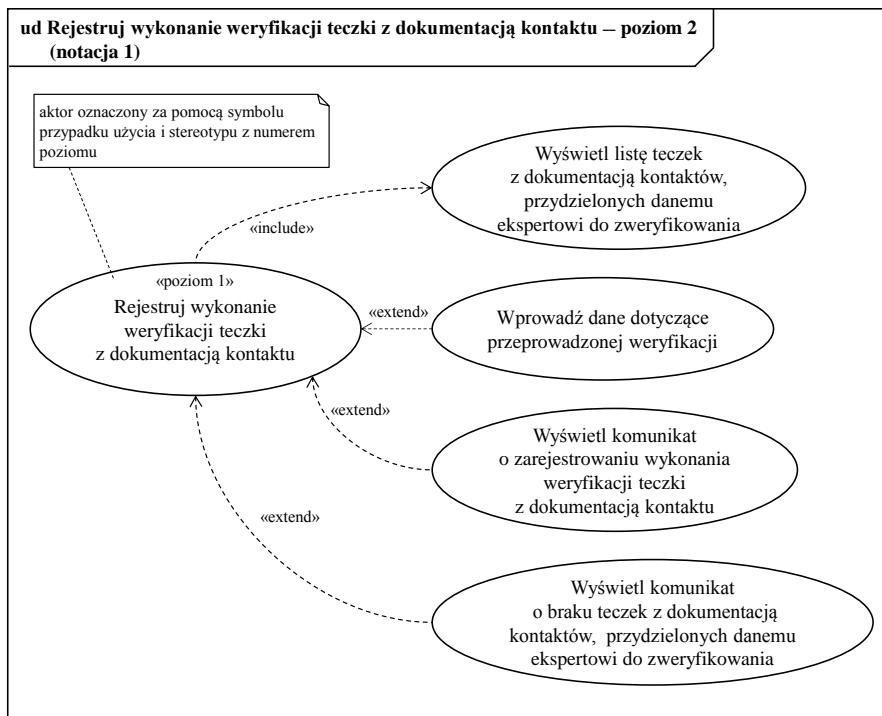
Rozwiązanie przedstawione na Rys. 2-5 może okazać się niewystarczające, dla analizowania coraz niższego poziomu abstrakcji modelu (coraz większego zagnieżdżania kodu), ze względu na to, że wprawdzie przypadki użycia są tam

również wywoływanie przez aktorów, lecz aktorem dla danego przypadku jest inny przypadek, a nie byt z otoczenia systemu czy też aktor *Podsystem czasu*.

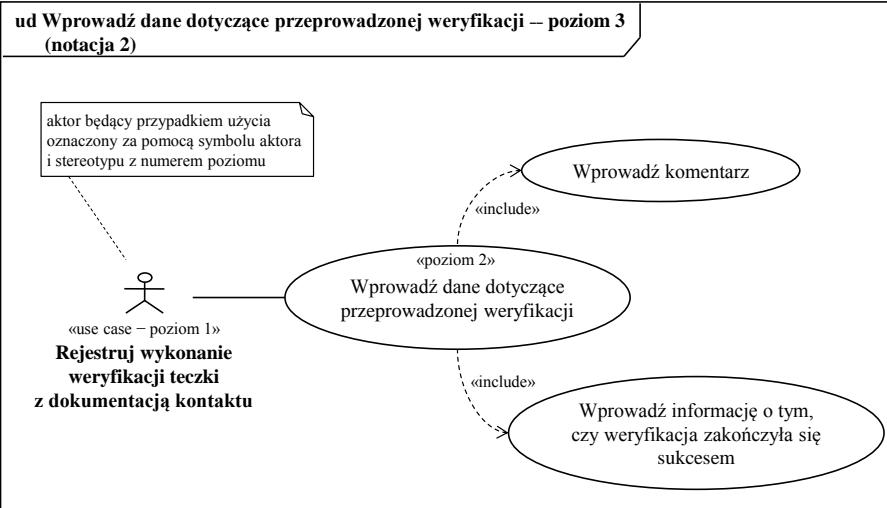
W związku z powyższym, zaproponowano różne notacje, zgodnie z którymi, na kolejnych niższych poziomach abstrakcji modelu, w roli aktora jest wykorzystywany przypadek oznaczony stereotypem z numerem poziomu, na którym aktor-przypadek został wprowadzony.

Na Rys. 2-6 został zaprezentowany diagram z przykładowym podziałem przypadku *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu*, a Rys. 2-7 i Rys. 2-8 przedstawiają podział przypadku *Wprowadź dane dotyczące przeprowadzonej weryfikacji*.

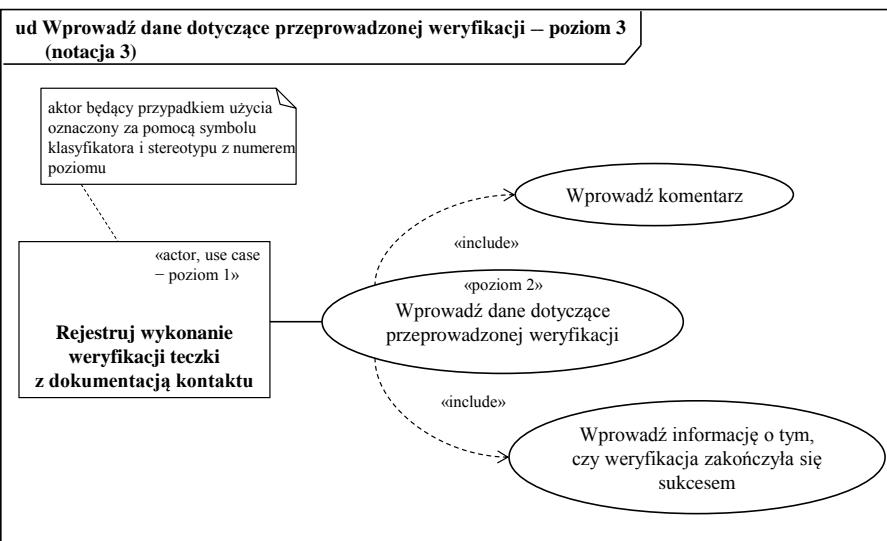
Na Rys. 2-6, dla oznaczenia aktora będącego przypadkiem użycia z wyższego poziomu abstrakcji, wchodzącym w interakcję z czterema innymi przypadkami, wykorzystano notację dla przypadków użycia, nazywaną dalej *notacją 1 dla oznaczania aktorów* (będących przypadkami). Z kolei na Rys. 2-7 posłużyono się podstawowym symbolem wykorzystywanym dla oznaczania aktorów w UML z dodatkowym użyciem stereotypu «use case» (notacja 2), a do Rys. 2-8 wprowadzono aktora będącego przypadkiem użycia z wykorzystaniem standardowej notacji dla oznaczania klasyfikatorów (notacja 3).



Rys. 2-6 Diagram z przykładowym podziałem przypadku *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu* – notacja 1



Rys. 2-7 Diagram z przykładowym podziałem przypadku *Wprowadź dane dotyczące przeprowadzonej weryfikacji* – notacja 2



Rys. 2-8 Diagram z przykładowym podziałem przypadku *Wprowadź dane dotyczące przeprowadzonej weryfikacji* – notacja 3

**Polecenie 2:** Dla funkcjonalności *Wyświetl opinię o kontakcie* napisz scenariusz.

Przykładowy scenariusz dla wybranego przypadku użycia został przedstawiony w Tab. 2-2. Przypadek ten może też być wywołany z przypadku *Przeglądaj aktualne przydziały do prac*, co nie zostało uwzględnione w scenariuszu.

Tab. 2-2 Scenariusz dla przypadku użycia *Wystaw opinię o kontakcie*

<b>Nazwa przypadku użycia</b>	Wystaw opinię o kontakcie
<b>Warunek początkowy</b>	Został zarejestrowany co najmniej 1 kontakt.
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Aktor <i>Ekspert</i> uruchamia przypadek użycia.</li> <li>2. System wyświetla listę kontaktów przypisanych ekspertowi do zaopiniowania. Aktor wybiera kontakt.</li> <li>3. System odpytuje o dane dotyczące opinii o kontakcie, takie jak: treść, koszt i stopień wiarygodności kontaktu. Aktor wprowadza dane.</li> <li>4. System odpytuje o chęć wprowadzenia informacji o zdjęciach. Aktor potwierdza.</li> <li>5. System odpytuje o dane dotyczące zdjęć. Aktor wprowadza dane.</li> <li>6. System informuje aktora o zarejestrowaniu nowej opinii dotyczącej wybranego kontaktu.</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	<ol style="list-style-type: none"> <li>2a. Aktor nie ma przypisanych żadnych kontaktów do zaopiniowania, system wyświetla odpowiedni komunikat i kończy przypadek użycia.</li> <li>2b. Aktor nie wybrał kontaktu do zaopiniowania, system kończy przypadek użycia.</li> <li>3a. Podana wartość stopnia wiarygodności nie należy do przedziału (0% –100%), system informuje o tym aktora i ponownie odpytuje o dane dotyczące opinii o kontakcie.</li> <li>4a. Aktor rezygnuje z wprowadzania informacji o zdjęciach, system przechodzi do punktu 6 scenariusza.</li> </ol>
<b>Zakończenie</b>	Zgodnie ze scenariuszem.
<b>Warunek końcowy</b>	Zostaje zarejestrowana nowa opinia dla danego kontaktu. W przypadku zmiany średniego stopnia wiarygodności kontaktu, może być on przypisany do nowej grupy kontaktów o odpowiednim średnim stopniu wiarygodności.

*Uwaga:* Oczywiście można było napisać inny scenariusz, ale należy zwrócić głównie uwagę na to, czy scenariusz jest skonstruowany zgodnie z regułami.

Można było również sformułować inny, bardziej szczegółowy warunek początkowy, co spowodowałoby zmianę alternatywnych przepływów zdarzeń. Przykładowo, dla warunku początkowego „Zarejestrowano co najmniej 1 kontakt do zaopiniowania dla danego eksperta”, w alternatywnych przepływach zdarzeń nie pojawiłby się punkt „2a. Aktor nie ma przypisanych żadnych kontaktów do zaopiniowania, system wyświetla odpowiedni komunikat i kończy przypadek użycia.”.

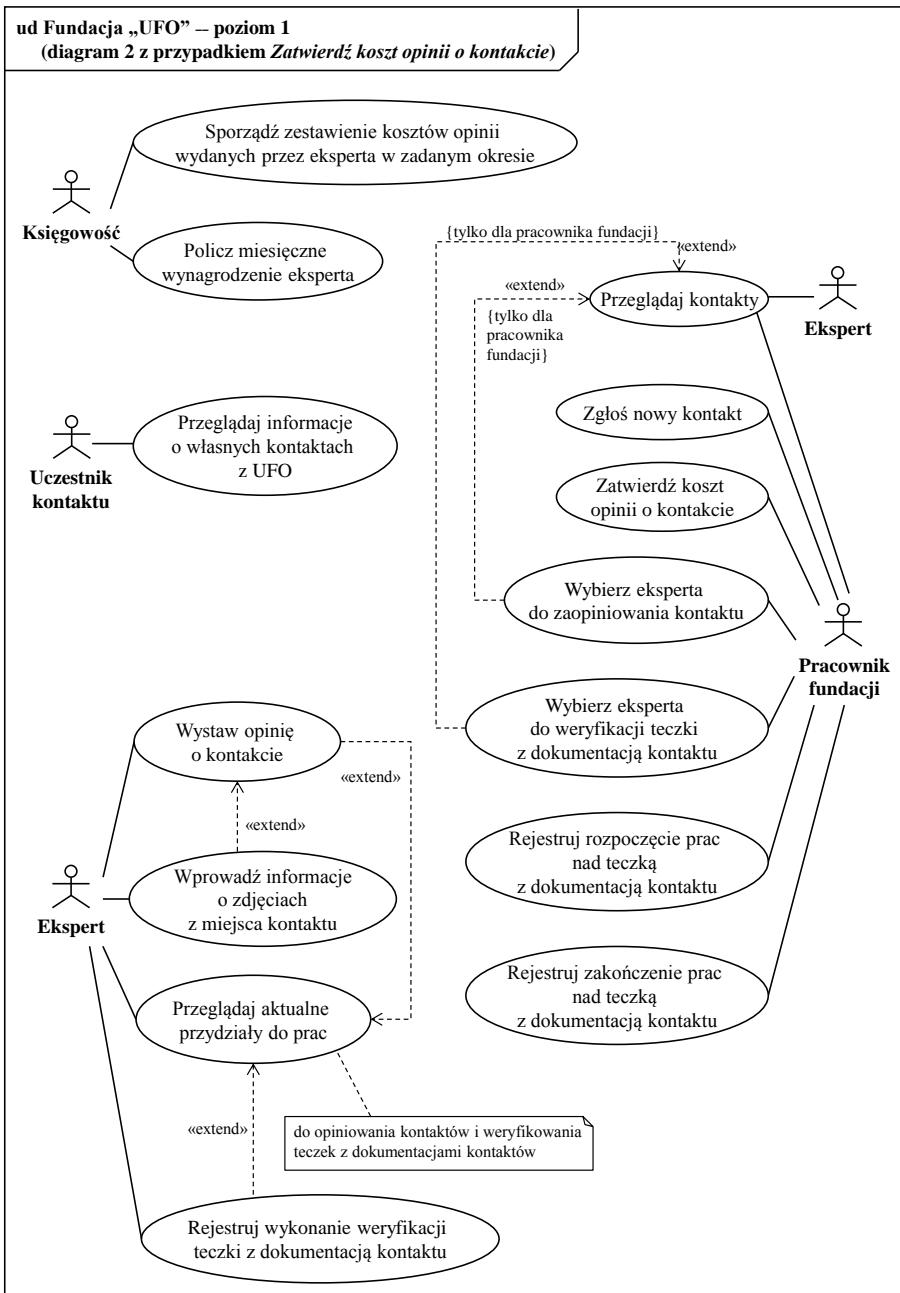
Analizując scenariusz podany w Tab. 2-2, należy rozważyć problem związany z kosztem podawanym przez eksperta w trakcie wprowadzania opinii o kontakcie. W tekście wymagań nie ma informacji o żadnych ograniczeniach nałożonych na wysokość tej kwoty, więc system pozwoliłby na wprowadzenie dowolnej wartości. W celu uniknięcia takiej sytuacji zaproponowano rozszerzenie tekstu wymagań. Przyjęto założenie, że podczas wprowadzania opinii ekspert podaje koszt wstępny opinii wraz z jego uzasadnieniem. Koszt ten powinien zostać zweryfikowany przez pracownika fundacji, który wprowadza koszt zatwierdzony.

Takie założenie powoduje konieczność zmiany scenariusza (pkt 3 głównego przepływu zdarzeń – musiałby zostać rozszerzony o wprowadzenie uzasadnienia dla określonego przez eksperta kosztu opinii). Należałoby również uzupełnić model przypadków użycia o przypadek *Zatwierdź koszt opinii o kontakcie*. Uzupełniony model zaprezentowano na Rys. 2-9.

Oczywiście nie jest to jedyne rozwiązanie ww. problemu. Przykładowo, można przechowywać wartość maksymalnego kosztu opinii, jednakową dla wszystkich opinii i porównywać ją z wartością kosztu wstępnego wprowadzaną przez eksperta.

## 2.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

- Częstym błędem jest brak zrozumienia idei pojęcia *przypadek użycia* – nazwa przypadku nie powinna być kojarzona ze strukturą danych, ale z operacjami realizowanymi na tej strukturze. Innymi słowy, nazwa przypadku powinna sugerować zadanie, które system ma wykonać na zlecenie aktora. Tak więc nazwy takie jak: *Opinia o kontakcie*, *Miesięczne wynagrodzenie eksperta* są błędne, gdyż nie wiadomo, jakiego rodzaju operacje mają być na ww. danych przeprowadzone.
- Podobnie, często widoczne jest niezrozumienie idei budowania hierarchii dziedziczenia dla aktorów, np. nieporozumieniem jest dziedziczenie aktora *Ekspert* po aktorze *Uczestnik kontaktu*. Jeżeli *Jan Kowalski*, który jest zatrudniony przez fundację w roli eksperta, będzie też uczestnikiem jakiegoś kontaktu, to chcąc zgłosić ten kontakt, zarejestruje się w systemie jako *Uczestnik kontaktu*.



Rys. 2-9 Model przypadków użycia (diagram 2), z dodatkowym przypadkiem *Zatwierdź koszt opinii o kontakcie*

- Zgodnie z tekstem wymagań, dokumentacja jest przechowywana w postaci „papierowej” i teczka z dokumentacją istnieje fizycznie poza systemem. W związku z tym sprawdzanie formalnej poprawności dokumentacji zawartej w tej teczce odbywa się poza systemem, a w systemie należy wyłącznie zarejestrować fakt jej przeprowadzenia. Przypadek użycia *Weryfikuj dokumentację* nie byłby więc wybrany/nazwany prawidłowo, ponieważ jego nazwa nie jest adekwatna do tego, co ten przypadek powinien modelować.

## 2.3 Analiza strukturalna

### 2.3.1 Tekst wymagań z komentarzami do analizy strukturalnej

Poniżej został przedstawiony tekst wymagań uzupełniony o komentarze wspomagające proces konstruowania schematu pojęciowego. Ponadto rozszerzono go o informacje związane z wprowadzaniem kosztu opinii o kontakcie (pkt 6 tekstu wymagań), wynikające z analizy funkcjonalnej.

#### Tekst wymagań z komentarzami

Fundacja „UFO” zajmuje się m.in. katalogowaniem kontaktów ludzi z niezidentyfowanymi obiektami latającymi (ang. *Unidentified Flying Object*), w czym ma jej pomóc projektowany system komputerowy. Pracownik fundacji, otrzymując, drogą telefoniczną lub za pośrednictwem e-maila, zgłoszenie dotyczące prawdopodobnego kontaktu z UFO, podejmuje decyzję, czy wprowadzić do systemu nowy kontakt, czy też dołączyć kolejnego uczestnika do wprowadzonego już wcześniej kontaktu, bądź też w ogóle zignorować zgłoszenie.

1. Zgodnie z projektem, w systemie mają być przechowywane dane osobowe<sup>1</sup> i numery telefonów (co najmniej jeden)<sup>2</sup> zarówno uczestników kontaktów

---

<sup>1</sup> Nie należy wymyślać własnych atrybutów – skoro na tym etapie analizy wiemy tylko tyle, że mają być pamiętane *dane osobowe*, to tylko taki atrybut powinien pojawić się na diagramie. Jest to atrybut będący strukturą złożoną i w toku dalszego postępowania powinno zostać określone, jakie elementy zawiera.

<sup>2</sup> Jeżeli dla jednej osoby ma być pamiętany więcej niż jeden numer telefonu, modelujemy tę informację w postaci atrybutu powtarzalnego, gdzie każdy jego składnik posiada taką samą organizację (strukturę). Należy także zwrócić uwagę na nazwę atrybutu: *numer*, a nie *numery*.

- z UFO, jak i ekspertów fundacji<sup>3</sup>. Ekspert może być również uczestnikiem kontaktu<sup>4</sup>.
2. Ekspert charakteryzowany jest przez listę posiadanych przez niego specjalizacji<sup>5</sup> (co najmniej 1) oraz miesięczne wynagrodzenie podstawowe. Wynagrodzenie to nie może jednorazowo wzrosnąć o więcej niż 10% poprzedniej wartości oraz nie może przekroczyć wartości maksymalnej jednakowej dla wszystkich ekspertów<sup>6</sup>.
  3. Informacje, które należy przechowywać dla każdego kontaktu, to: opis, data, godzina, czas trwania, miejsce, kto był jego uczestnikiem<sup>7</sup> (mogło być ich

<sup>3</sup> Należy tu wyróżnić dwie klasy: *Ekspert* i *Uczestnik kontaktu*. Ponieważ zawierają one takie same atrybuty: *dane osobowe* i *numer telefonu :[1..\*]*, trzeba utworzyć dla nich nadklasę (klasa *Osoba*) będącą klasą abstrakcyjną (podział jest kompletny), zawierającą wspólne własności.

<sup>4</sup> Ze względu na możliwość istnienia obiektów będących równocześnie instancjami obu podklaś: *Ekspert* i *Uczestnik kontaktu*, należy oznaczyć hierarchię jako nieroziłączną (*overlapping*). *Uwaga:* Jeśli w hierarchii znajduje się więcej podklaś, a nieroziłączność dotyczy tylko niektórych z nich, wtedy nie umieszcza się oznaczenia *overlapping* przy symbolu dziedziczenia, ale wykorzystując linię przerywaną, trzeba pokazać, których konkretnie podklaś dotyczy, lub też umieścić przy jej oznaczeniu stosowne ograniczenie.

<sup>5</sup> Na tym etapie analizy nie widać żadnego związku opisu kontaktu ze specjalizacjami ekspertów oraz nie widać konieczności wykorzystywania specjalizacji, np. do przeszukiwania zbioru ekspertów. W związku z powyższym, zaproponowano rozwiązanie zawierające atrybut powtarzalny *specjalizacja:[1..\*]*, a nie klasę *Specjalizacja* (w tym momencie byłaby ona pozbawiona invariantów). Proponowane rozwiązanie jest lepsze w świetle tekstu wymagań, ponieważ specjalizacja jest to po prostu jeden z elementów składowych opisu eksperta, podobnie jak dane osobowe, czy numer telefonu. Rozwiązanie z wykorzystaniem klasy *Specjalizacja* w przyszłości mogłoby ułatwić np. wyszukiwanie ekspertów po specjalizacjach, ale dla aktualnego tekstu wymagań jest rozwiązaniem nadmiarowym. Należy także zwrócić uwagę na nazwę atrybutu: *specjalizacja*, a nie *lista specjalizacji*.

<sup>6</sup> Z tekstu wymagań (pkt 2) wynikają dwa ograniczenia nałożone na wartość atrybutu *miesięczne wynagrodzenie podstawowe*, które powinny zostać wprowadzone do schematu pojęciowego. Ograniczenia te to: {*miesięczne wynagrodzenie podstawowe nie może jednorazowo wzrosnąć o więcej niż 10% poprzedniej wartości*} oraz {*miesięczne wynagrodzenie podstawowe nie może przekroczyć miesięcznego wynagrodzenia maksymalnego*}. W klasie *Ekspert* powinien zostać umieszczony atrybut *maks. miesięczne wynagrodzenie*. Jest to atrybut klasowy, ponieważ posiada jednakową wartość dla wszystkich obiektów tej klasы.

<sup>7</sup> Ze względu na to, że została już wcześniej wyróżniona klasa *Uczestnik kontaktu*, informacja o tym, *kto był uczestnikiem kontaktu*, powinna zostać zamodelowana jako asocjacja łącząca klasę *Uczestnik kontaktu* i *Kontakt*. Informacje o związkach zachodzących między instancjami klas, w większości przypadków należy modelować jako asocjacje, a nie atrybuty. Będem byłoby umieszczenie atrybutu powtarzalnego *uczestnik kontaktu* w klasie *Kontakt*, ponieważ sugerowałoby to powielanie danych uczestników kontaktów (te same dane w klasie *Uczestnik kontaktu* i w klasie *Kontakt*). Na etapie analizy asocjacja ma pierwszeństwo przed atrybutami modelującymi relacje zachodzące pomiędzy obiektami

wielu) i jaka jest relacja z tego kontaktu każdego z jego uczestników<sup>8</sup> (w postaci tekstu). Relacja jest tu rozumiana jako zeznanie naocznego świadka tego wydarzenia.

4. Kontakty dzielą się ze względu na typ oraz ze względu na średni stopień wiarygodności<sup>9</sup>. Pierwszy podział, który jest kompletny i rozłączny, obejmuje m.in.<sup>10</sup> obserwacje i uprowadzenia<sup>11</sup>. Obserwacja rozumiana jest jako zauważenie przez uczestnika czegoś, co podejrzewa, że jest niezidentyfikowanym obiektem latającym. Dla obserwacji ma być przechowywana liczba uczestników kontaktu, a dla uprowadzeń – opis badania medycznego, o ile takie zostało przeprowadzone<sup>12</sup> na danym uczestniku. Dla drugiego podziału kontaktów, czyli podziału ze względu na średni stopień

---

klas. Również niedopuszczalne jest umieszczenie asocjacji i dodatkowo atrybutów, służących temu samemu celowi.

<sup>8</sup> Relacja uczestnika z kontaktu (czyli zeznanie naocznego świadka) powinna zostać zamodelowana jako klasa asocjacji między klasami *Uczestnik kontaktu* i *Kontakt*. Jeśli uczestnik brał udział w kontakcie, to pamiętamy jego relację z tego kontaktu, co sugeruje klasę asocjacji.

<sup>9</sup> W tekście wymagań wymieniono dwa aspekty podziału kontaktów (*typ* oraz *średni stopień wiarygodności*), dlatego na diagramie powinno pojawić się dziedziczenie dwuaspektowe. *Uwaga:* W hierarchii wieloaspektowej wymagane jest oznaczanie aspektów. Błędem jest umieszczenie, zamiast aspektów dziedziczenia, takich podklas klasy *Kontakt* jak: *Typ* i *Średni stopień wiarygodności*. Takie rozwiązanie oznaczałoby, że *Typ* lub *Średni stopień wiarygodności* jest szczególnym przypadkiem *Kontaktu*, co nie jest prawdą. Nazwa podklasy powinna zachować „semantyczną zgodność” z nazwą nadklasty, a więc nie klasa nazwana: *Wysoki stopień wiarygodności*, a klasa: *Kontakt o bardzo wysokim średnim stopniu wiarygodności*, co oznacza, że obiektem podklasy klasy *Kontakt* musi być kontakt, a nie stopień wiarygodności. Nazwa klasy powinna sugerować, co będzie przechowywane w jednym obiekcie danej klasy, a obiekt podklasy powinien być rodzajem, szczególnym przypadkiem wystąpienia (w tym wypadku pośredniego) nadklasty.

<sup>10</sup> Skrót *m.in.* sugeruje konieczność wykorzystania dziedziczenia typu *elipsa* (podział jest kompletny, wszystkie pozostałe podklasy są znane, a elipsa oznacza, że niektóre ze zidentyfikowanych klas nie zostały narysowane na diagramie). W związku z tym, że podział jest skończony, nadklasa jest klasa abstrakcyjna, co zostało już powiedziane wcześniej (w przypisie 3).

<sup>11</sup> Nazwy klas powinny być rzeczownikami w liczbie pojedynczej, a więc *Uprowadzenie*, *Obserwacja*, a nie *Uprowadzenia*, *Obserwacje*.

<sup>12</sup> *Opis badania medycznego* nie może zostać zamodelowany w postaci atrybutu w klasie *Uprowadzenie*, gdyż nie byłoby wiadomo, który z uczestników kontaktu opisuje przeprowadzone badanie medyczne (jedno uprowadzenie może być związane z wieloma uczestnikami kontaktu) – dlatego należy wprowadzić go do klasy asocjacji pomiędzy klasami *Uczestnik kontaktu* i *Kontakt*. Powinien to być atrybut opcjonalny, ponieważ nie dla wszystkich relacji z uprowadzeń pojawił się opis takiego badania. Należy również umieścić ograniczenie *{o ile kontaktem było uprowadzenie}*, ponieważ asocjacja łączy uczestnika kontaktu z dowolnym kontaktem, nie tylko z uprowadzeniem. Nie ma sensu wprowadzanie ograniczenia *{o ile przeprowadzono badanie medyczne}*, gdyż takiego warunku w projektowanym systemie nie da się zaimplementować, a ograniczenia muszą być implementowalne.

wiarygodności, wyróżniono następujące grupy: kontakty o bardzo wysokim średnim stopniu wiarygodności ( $\geq 80\%$ ), kontakty o wysokim średnim stopniu wiarygodności ( $\geq 70\%$  i  $< 80\%$ ) itd. Podział ten jest, podobnie jak poprzedni, kompletny. Wprowadzenie nowego kontaktu automatycznie skutkuje przypisaniem go do grupy kontaktów o najniższym średnim stopniu wiarygodności (ponieważ średni stopień wiarygodności, przy braku w tym momencie jakichkolwiek opinii ekspertów, wynosi 0%).

5. W systemie należy przechowywać informację, któremu ekspertowi przydzielono do zaopiniowania dany kontakt. Jest możliwe, że kontakt została już zarejestrowany, ale nikt go nie zaopiniował. Przechowywane mają być informacje tylko o tych ekspertach, którym przydzielono co najmniej jeden kontakt<sup>13</sup>.
6. Dla opinii wystawionej przez eksperta należy przechowywać dane takie jak: numer (unikatowy w ramach danego kontaktu<sup>14</sup>), data wystawienia, wstępny koszt (określony przez eksperta), uzasadnienie kosztu, koszt zatwierdzony przez pracownika fundacji oraz treść. Wystawiając opinię, ekspert określa również stopień wiarygodności kontaktu (wyrażony w procentach) oraz ewentualnie wprowadza informacje o zdjęciach<sup>15</sup> wykonanych na miejscu zaistnienia kontaktu, o ile takie wykonał. Zdjęcia mogą też być dołączone później, w innym terminie. Dla każdego z nich należy pamiętać: opis, datę wykonania oraz adres pliku (tylko dla zdjęć przechowywanych w wersji cyfrowej). Ze względu na ograniczenia finansowe fundacji, każdy z ekspertów

<sup>13</sup> Do schematu pojęciowego należy wprowadzić asocjację *ma przydzielony do zaopiniowania* między klasami *Ekspert* i *Kontakt*, z licznością  $1..*$ , umieszczoną po stronie klasy *Kontakt* i licznością  $0..*$  przy klasie *Ekspert*. Oznaczenia liczności  $0..*$  i  $*$  są równoważne. Zamiast nazwy asocjacji (która musi być frazą czasownikową), można wprowadzić rolę (fraza rzeczownikowa), np. *opiniadowca* po stronie klasy *Ekspert*. Fraza rzeczownikowa jest zdecydowanie łatwiejsza do wymyślenia i z reguły nie zmienia znaczenia wraz z upływem czasu, w przeciwieństwie do frazy czasownikowej. Porównajmy: asocjacja nazwana *opiniuje* (teraz) różni się znaczeniowo od asocjacji *zaopiniował* (kiedyś). Natomiast asocjacja nazwana z wykorzystaniem roli *opiniadowca* zachowią swoje znaczenie mimo upływu czasu.

<sup>14</sup> Numer opinii jest unikatowy w ramach kontaktu, dlatego może on być potencjalnie kwalifikatorem asocjacji między klasami *Ekspert* i *Kontakt* (klasa *Opinia* jest klasą tej asocjacji), umieszczonym po stronie klasy *Kontakt* (co oznacza, że jest unikatowy w obrębie zbioru opinii dla danego kontaktu) i jednoznacznie wskazującym na eksperta, który wystawił opinię o danym numerze.

<sup>15</sup> Z tekstu wymagań wynika, że informacje o zdjęciach mogą być wprowadzane podczas wystawiania opinii, tak więc klasę *Zdjęcie* należy połączyć relacją z klasą *Opinia*. Ze względu na fakt, że nie ma sensu przechowywanie zdjęć, o ile nie została wprowadzona informacja o opinii, która te zdjęcia zawiera, relacja ta powinna być kompozycją. W takim rozwiązaniu usunięcie opinii spowoduje usunięcie zdjęć z nią związanych, a zdjęcie będzie związane wyłącznie z jedną opinią. Ponieważ zdjęcia mogą być dołączone do opinii później, w innym terminie, jest możliwe, że niektóre opinie mogą w danym momencie (lub nawet w ogóle) nie posiadać zdjęć (co jest dopuszczalne dla kompozycji).

może mieć w danym momencie przydzielon4 nie więcej niż 3 kontakty do zaopiniowania<sup>16</sup>.

7. Planuje się, aby dla każdego kontaktu było możliwe obliczenie średniego stopnia wiarygodności, przy czym dla wszystkich typów kontaktów oprócz obserwacji algorytm ma być oparty na tej samej prostej formule<sup>17</sup>: średni stopień wiarygodności jest równy średniej ze stopni wiarygodności określonych przez ekspertów, którzy opiniowali kontakt; w przypadku obserwacji dodatkowo należy uwzględnić liczbę uczestników kontaktu. Średni stopień wiarygodności jest wyliczany po każdym wprowadzeniu nowej opinii o kontakcie, co może skutkować zmianą przydziału kontaktu do innej kategorii<sup>18</sup>.
8. Z każdym kontaktem związana jest teczka z dokumentacją, zawierająca różnego rodzaju papierowe dokumenty<sup>19</sup>. W systemie mają być przechowywane następujące informacje o niej: numer (unikatowy), data założenia (równoznaczna z terminem rozpoczęcia prac nad danym kontaktem), data zakończenia tych prac oraz status przyjmujący jedną z następujących wartości: „w trakcie opracowywania”, „opracowana”, „do ponownej weryfikacji”, „zweryfikowana pozytywnie”.
9. Ponadto ma być pamiętane, kiedy i któremu ekspertowi przydzielono do weryfikacji daną teczkę z dokumentacją kontaktu. Weryfikacja teczki polega na sprawdzeniu formalnej poprawności dokumentów w niej zawartych (bez pośrednictwa systemu), a następnie zarejestrowaniu w systemie informacji

<sup>16</sup> Informacji zawartej w tym fragmencie tekstu wymagań nie można zamodelować za pomocą oznaczenia liczności asocjacji 0..3. Oznaczałoby to „zgubienie” danych historycznych i zapamiętanie maksymalnie trzech ostatnich opinii przydzielonych ekspertowi do wykonania, a nie o to w tekście chodziło. Można więc zamodelować to np. jako ograniczenie *{w danym momencie maks. 3 kontakty przydzielone do zaopiniowania}* nałożone na asocjację, a dokładniej, na rolę asocjacji przechowującą informacje o kontaktach przydzielonych ekspertowi do zaopiniowania.

<sup>17</sup> Metoda *obl. średni stopień wiarygodności*, zaimplementowana w nadklasie *Kontakt*, będzie dziedziczona przez wszystkie podklasy w tym fragmencie hierarchii, który powstał na skutek podziału kontaktów, odpowiednio do ich typów (aspekt *typ*). Do klasy *Obserwacja* należy wprowadzić metodę właściwą dla tej podklasy, przesłaniającą metodę dziedziczoną z nadklasy. Jest to przykład polimorfizmu metod związanego z przesłananiem. Należy zwrócić uwagę na fakt, że polimorfizm metod nie dotyczy tego fragmentu hierarchii, który powstał na skutek podziału kontaktów, odpowiednio do ich średniego stopnia wiarygodności.

<sup>18</sup> Zmiana przypisania kontaktu do podklasy, dla podziału związanego ze średnim stopniem wiarygodności, sugeruje wykorzystanie w modelowaniu specjalizacji dynamicznej.

<sup>19</sup> Termin *papierowe dokumenty* oznacza dokumenty przechowywane poza systemem. Nie ma tu mowy o elektronicznej wersji dokumentów, dlatego też błędem byłoby umieszczenie klasy *Dokument* na schemacie pojawiowym. Oczywiście, gdyby w przyszłości w systemie miałyby być przechowywane dokumenty w wersji elektronicznej, taka klasa zostałaby wprowadzona do schematu pojawiowego. Z zamieszczonego tekstu wymagań nie wynika jednak takie rozwiązanie.

takiej jak:<sup>20</sup> kto i kiedy weryfikował daną teczkę, czy proces ten zakończył się sukcesem oraz komentarz eksperta. Jeśli ekspert miał zastrzeżenia do któregoś z dokumentów teczki, to otrzymuje ona status „do ponownej weryfikacji”. Teczkę może być weryfikowana przez wielu ekspertów (również tych samych, którzy uprzednio to robili<sup>21</sup>), ale nie mogą oni należeć do grupy opiniującej kontakt, którego teczka dotyczy. Pierwsza pozytywna weryfikacja powoduje zmianę statusu teczki na „zweryfikowana pozytywnie”.

#### 10. System powinien umożliwiać<sup>22</sup> m.in:

- 10.1 zgłoszenie nowego kontaktu;
- 10.2 wystawienie opinii o kontaktcie przez eksperta;
- 10.3 sporządzenie zestawienia kosztów wszystkich opinii wystawionych w zadany okresie przez wybranego eksperta;
- 10.4 obliczenie miesięcznego wynagrodzenia eksperta, łącznie ze sprawdzeniem, czy wystawiał opinię o kontaktach w danym miesiącu; jeśli tak, to należy obliczyć koszty tych opinii i dodać je do miesięcznego wynagrodzenia podstawowego;
- 10.5 stworzenie listy *nieaktywnych* ekspertów, czyli takich, którzy przez rok nie zaopiniowali żadnego kontaktu i nie zweryfikowali żadnej teczki z dokumentacją.

### 2.3.2 Schemat pojęciowy

*Polecenie:* Dla podanego tekstu wymagań skonstruuj schemat pojęciowy.

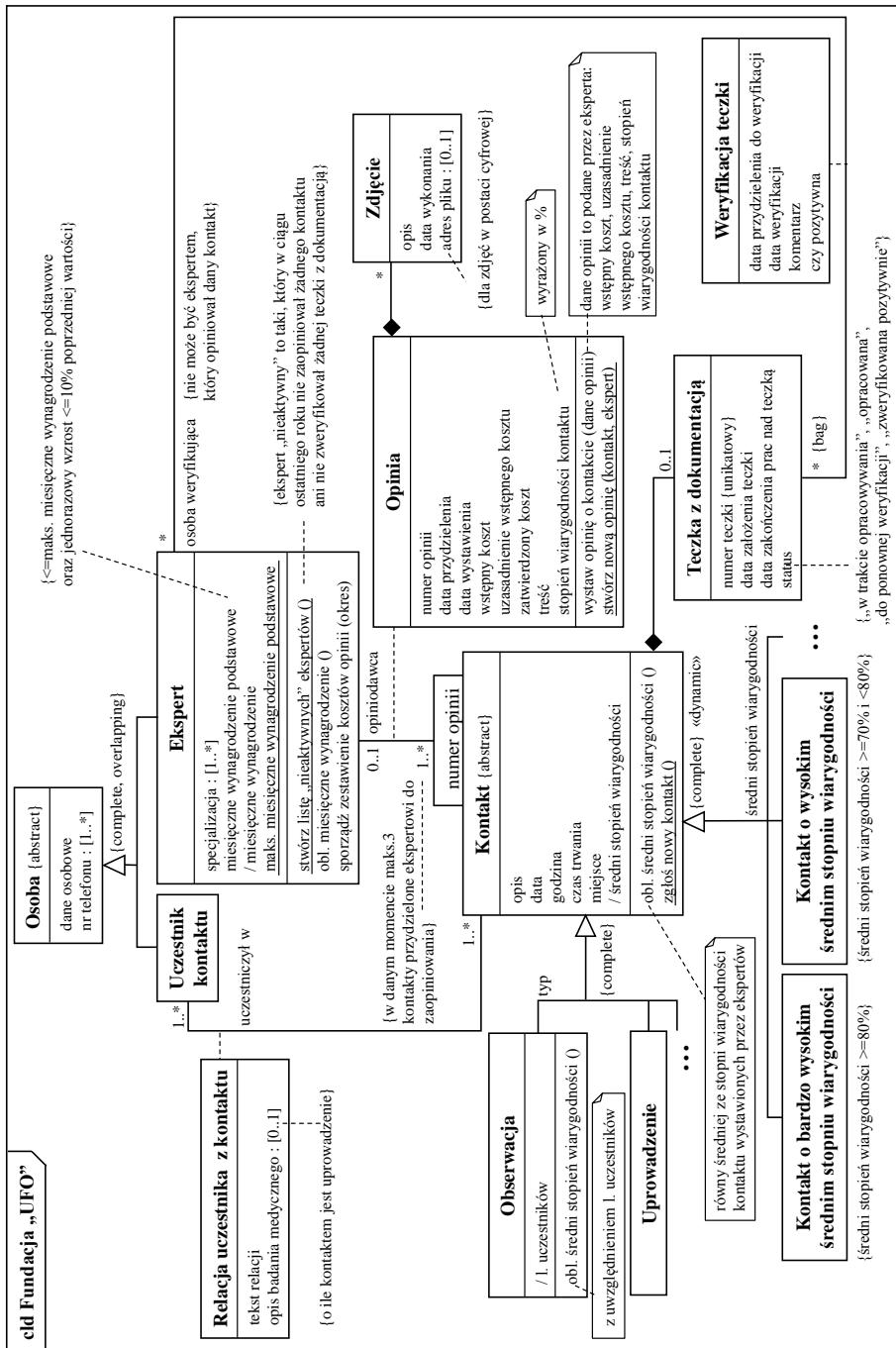
Schemat pojęciowy [35, 40, 41] przedstawiono na Rys. 2-10.

<sup>20</sup> W modelu powinna pojawić się klasa asocjacji *Weryfikacja teczki z dokumentacją*, której obiekty będą przechowywały informacje o przeprowadzeniu danej weryfikacji przez eksperta.

<sup>21</sup> Tekst wymagań sugeruje wystąpienie wielu powiązań pomiędzy dwoma tymi samymi obiektami (konkretny ekspert i konkretna teczka z dokumentacją), dlatego na diagramie należy umieścić ograniczenie *{bag}* przy jednym z końców asocjacji między klasami *Ekspert* i *Teczka z dokumentacją*.

<sup>22</sup> Do schematu pojęciowego należy wprowadzić metody implementujące funkcjonalności wymienione w tym punkcie tekstu wymagań. *Dla przypomnienia*, metody umieszcza się tam, gdzie znajdują się dane wejściowo/wyjściowe potrzebne do wykonania metody, a nie w miejscu ich potencjalnego wywołania, np. metoda *zgłoś nowy kontakt ()* tworzy nowy obiekt w klasie *Kontakt* i w tej klasie powinna zostać umieszczona (jako metoda klasowa), a nie np. w klasie *Uczestnik* (gdyby uczestnik jako aktor miał taką funkcjonalność). Ponadto, na tym etapie nie wprowadzamy również do diagramu klas aktorów, choć jest możliwe, że pojawi się klasa przechowująca obiekty, które są równocześnie aktorami systemu, jak na przykład klasa *Ekspert*. Ekspert jako aktor ma przypisany do siebie pewien zbiór usług systemu (tzn., że ma prawo zlecać je systemowi do wykonania), co zostało zaprezentowane w modelu przypadków użycia.

*Uwaga:* należy pamiętać o umieszczaniu list argumentów dla metod.

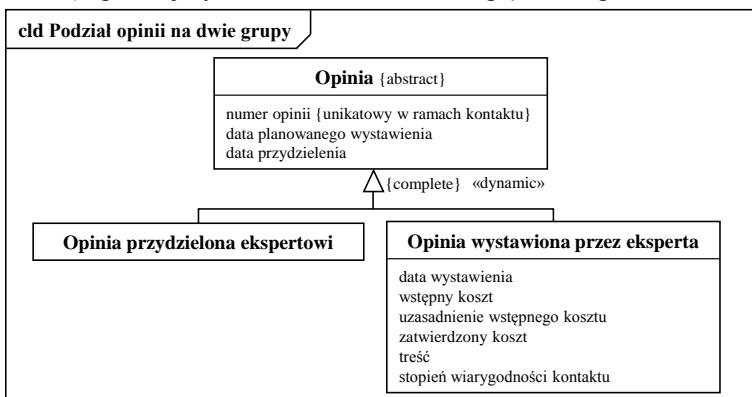


Rys. 2-10 Schemat pojęciowy systemu *Fundacja „UFO”*

Do schematu pojęciowego wprowadzono jedynie te metody, które zostały wymienione w ostatnim punkcie tekstu wymagań.

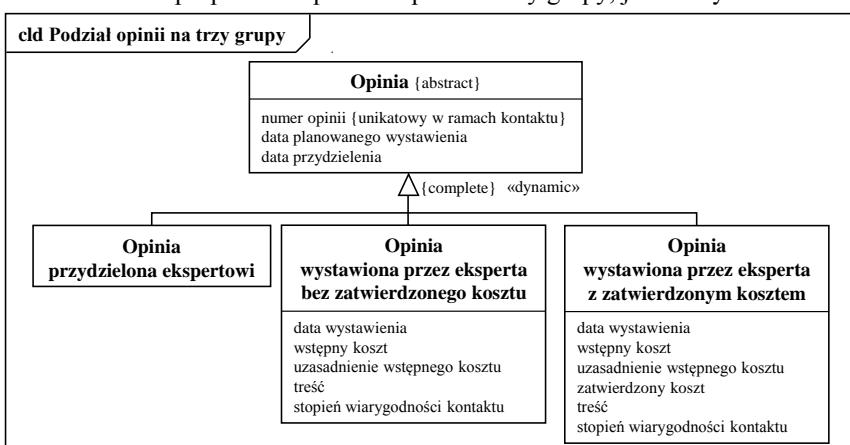
*Uwaga:* Na schemacie pojęciowym w nazwach atrybutów i metod przyjęto powszechnie stosowane skróty: *l.* (liczba), *maks.* (maksymalny), *min.* (minimalny), *obl.* (oblicz). Stosowanie skrótów jest dopuszczalne, o ile są one jednoznacznie interpretowane przez wszystkie osoby korzystające z danego schematu.

Rys. 2-11 zawiera alternatywną wersję dla fragmentu schematu, uwzględniającą podział opinii na dwie grupy: opinie, które już zostały przydzielone ekspertom, ale nie są jeszcze przez nich wystawione oraz opinie już wystawione przez ekspertów (pominieto tu metody oraz komentarze). W klasie *Opinia* umieszczono dodatkowy atrybut *data planowanego wystawienia*, który pozwoliłby na lepszą kontrolę procesu opiniowania (w późniejszych rozważaniach nie uwzględniano go).



Rys. 2-11 Fragment schematu pojęciowego z podziałem opinii na dwie grupy

Można również zaproponować podział opinii na trzy grupy, jak na Rys. 2-12.

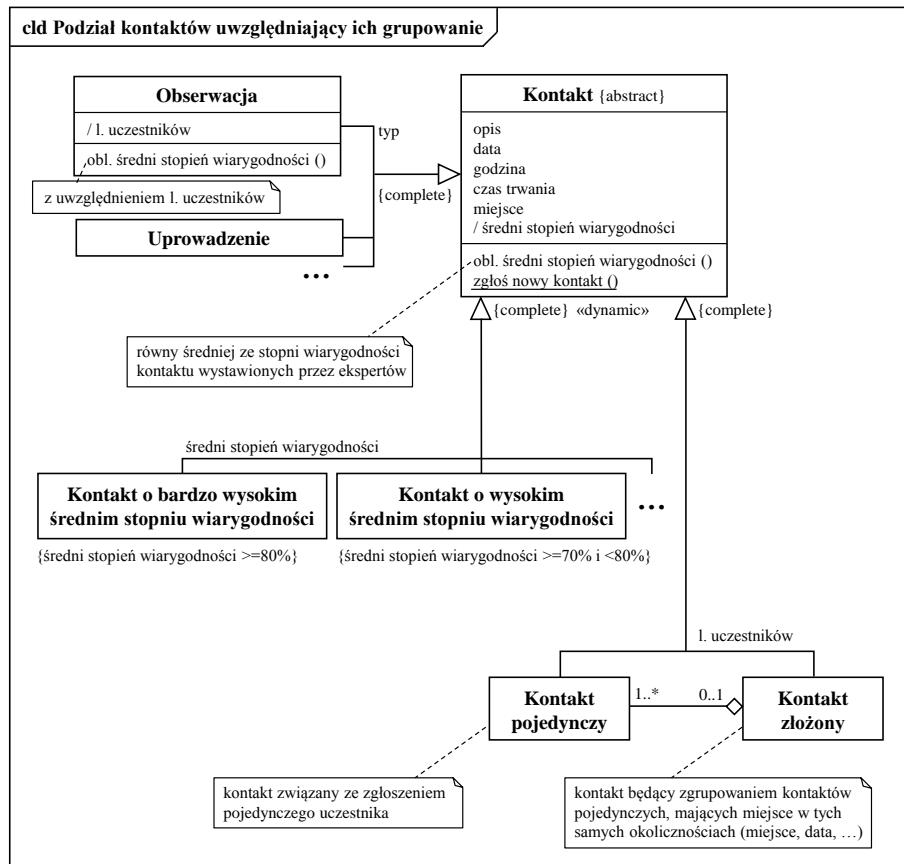


Rys. 2-12 Fragment schematu pojęciowego z podziałem opinii na trzy grupy

Takie rozwiązanie powinno zdecydowanie poprawić efektywność procesu przydzielania kontaktów do zaopiniowania czy zatwierdzania kosztów.

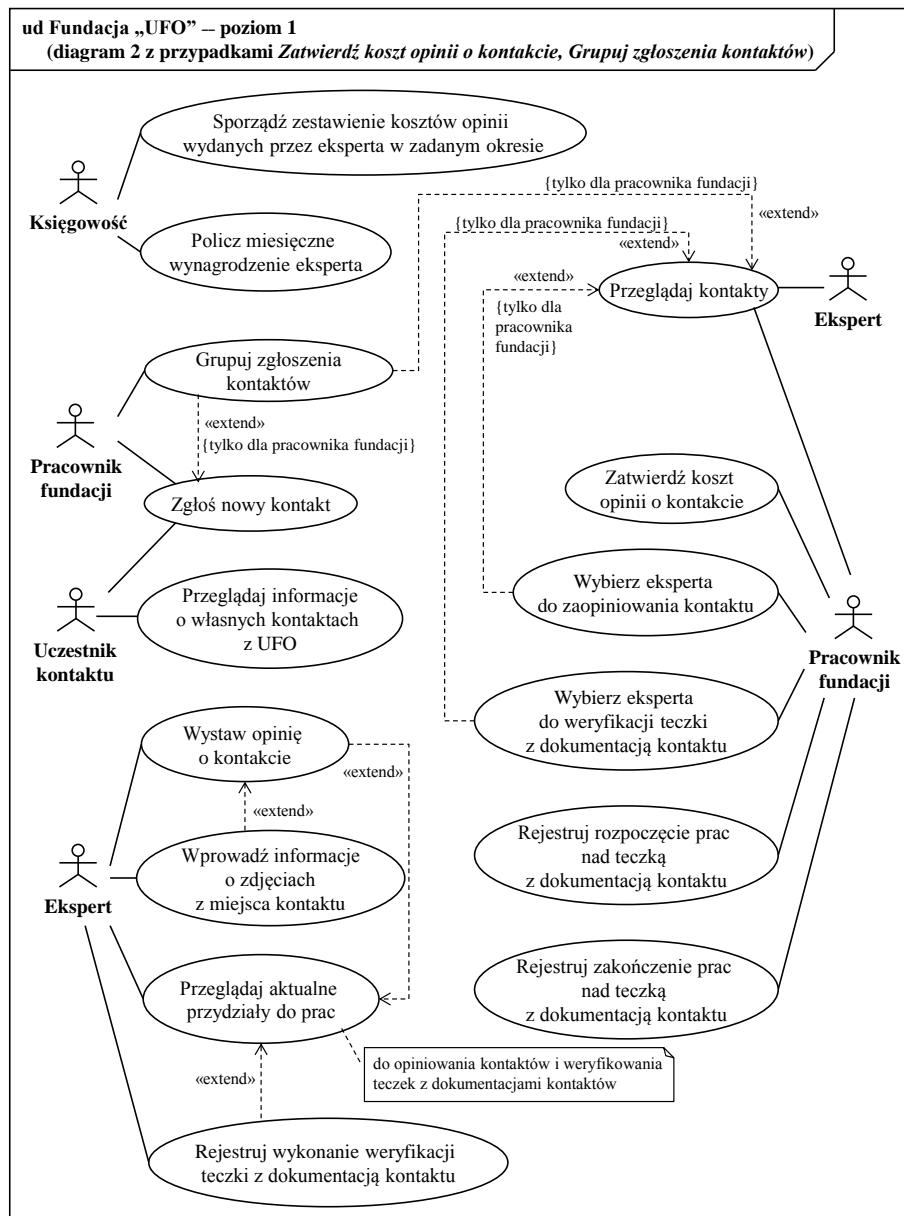
Ponadto, w trakcie analizy tekstu wymagań rozważano też, jak zmieniłby się proponowany schemat pojęciowy przy założeniu, że uczestnik mógłby sam zgłosić kontakt, np. przez stronę internetową. Pracownik fundacji musiałby wtedy posiadać możliwość przeglądania pojedynczych zgłoszeń i ewentualnego grupowania ich za pośrednictwem systemu. W podanym rozwiążaniu pracownik musi grupować kontakty poza systemem i jedynie rejestruje wyniki tego procesu. Grupowanie kontaktów mogłoby być przeprowadzane w oparciu o przesłanki takie jak: wspólne miejsce, ta sama lub zbliżona data, podobny opis itp.

Zmodyfikowany schemat pojęciowy uwzględniający te rozważania został przedstawiony na Rys. 2-13. Ze względu na przejrzystość rysunek zawiera jedynie ten fragment, który został zmieniony.



Rys. 2-13 Fragment schematu pojęciowego systemu z dodatkowym aspekiem podziału kontaktów

Modyfikacja schematu pojęciowego musiałaby być realizowana równocześnie z rozszerzeniem zbioru usług dostarczanych przez system o możliwość grupowania kontaktów, co ilustruje Rys. 2-14.



Rys. 2-14 Fragment modelu przypadków użycia (diagram 2), uzupełniony o przypadek *Grupuj zgłoszenia kontaktów*

Dalsze rozważania dotyczyć będą schematu pojęciowego zawartego na Rys. 2-10.

### 2.3.3 Analiza elementów pochodnych

Po skonstruowaniu schematu pojęciowego dobrze jest zastanowić się nad elementami pochodnymi zidentyfikowanymi w trakcie analizy i odpowiedzieć na pytanie, czy zachowają swoje wartości wraz ze zmianami zachodzącymi na przestrzeni czasu. Wprowadzanie do diagramu informacji redundantnej w postaci elementu pochodnego jest uzasadnione co najwyżej w przypadku, gdy wyliczanie jego wartości jest kosztowne obliczeniowo, wartość ta rzadko się zmienia, ale jest często wykorzystywana w trakcie innych obliczeń.

Oczywiście każdy element pochodny (atrybut, asocjacja, klasa) wymaga dostarczenia na późniejszym etapie analizy skojarzonego z nim fragmentu kodu/metody/metod, wyliczającego (wyprowadzającego) jego wartość w oparciu o elementy bazowe (czyli te, które nie są poprzedzone znakiem ukośnika).

Na schemacie pojęciowym przedstawionym na Rys. 2-10 oznaczono następujące atrybuty jako elementy pochodne: *miesięczne wynagrodzenie* w klasie *Ekspert*, *l. uczestników* w klasie *Obserwacja* i *średni stopień wiarygodności* w klasie *Kontakt*.

#### 1. Atrybut pochodny: *miesięczne wynagrodzenie* w klasie *Ekspert*

Wartość atrybutu miesięczne wynagrodzenie w klasie *Ekspert*, zgodnie z tekstem wymagań, jest wyliczana raz w miesiącu i stosunkowo rzadko wykorzystywana. Nie ma więc powodu dla jej przechowywania, gdyż zgodnie z tekstem wymagań nie jest planowana archiwizacja wynagrodzeń. Wydaje się więc, że wystarczy wprowadzić do schematu wyłącznie metodę *obl. miesięczne wynagrodzenie ()*.

#### 2. Atrybut pochodny: *l. uczestników* w klasie *Obserwacja*

Z rozważań wynika, że atrybut ten powinien pozostać pochodny. Wyliczenie jego wartości (w oparciu o licznosć roli *uczestnik* dla asocjacji pomiędzy klasami *Kontakt* i *Uczestnik kontaktu*) wprawdzie nie jest kosztowne obliczeniowo, ale raczej nie będzie ona ulegać zmianom na przestrzeni czasu, natomiast jest wykorzystywana przy wprowadzaniu nowej opinii dla kontaktu tego typu.

#### 3. Atrybut pochodny: *średni stopień wiarygodności* w klasie *Kontakt*

Wartość tego atrybutu jest wyliczana zawsze po wystawieniu nowej opinii i na tej podstawie kontakt jest przypisywany do odpowiedniej podklasy, po czym zgodnie z tekstem wymagań, wartość ta nie jest już dalej wykorzystywana. Tak więc i w tym przypadku nie widać potrzeby jej przechowywania, i jak poprzednio, wydaje się, że wystarczy wprowadzić do schematu pojęciowego wyłącznie metodę *obl. średni stopień wiarygodności ()* (wykonywaną w oparciu o algorytm podany w punkcie 7 tekstu wymagań).

Nie mniej jednak można by tu rozważyć pewną optymalizację (co nie należy do podstawowych obowiązków analityków, chociaż czasami muszą to robić, ze względu na wiedzę o modelowanej dziedzinie problemowej), dotyczącą

algorytmu dla tej metody. Wartość atrybutu *średni stopień wiarygodności* można by wyliczać ze wzoru:

- (1)  $\text{średni stopień wiarygodności} = (\text{n} * \text{średni stopień wiarygodności} + \text{stopień wiarygodności określony w wystawionej opinii}) / (\text{n}+1)$ ,

gdzie:

- $n$  oznacza liczbę dotychczas wystawionych opinii,
- zmienna *średni stopień wiarygodności* oznacza: po lewej stronie wyrażenia wartość atrybutu wyliczoną po wystawieniu opinii, po prawej stronie jego wartość poprzednią.

W związku z powyższym, aby poprawić efektywność obliczeń (co miałyby znaczenie szczególnie przy dużej liczbie opinii), można zasugerować przechowywanie dwóch atrybutów pochodnych: *średni stopień wiarygodności i l. dotychczas wystawionych opinii*.

W dalszych rozważaniach została przyjęta do realizacji pierwsza z obu tych propozycji (czyli wprowadzenie do schematu pojęciowego wyłącznie metody *obl. średni stopień wiarygodności* (), opartej na pierwotnym algorytmie), głównie ze względu na to, że nie przewiduje się, aby dostarczano dużo opinii dla pojedynczego kontaktu, jak również ze względu na to, że zniknęłaby informacja o innym algorytmie obliczania tej wartości dla kontaktów będących obserwacjami.

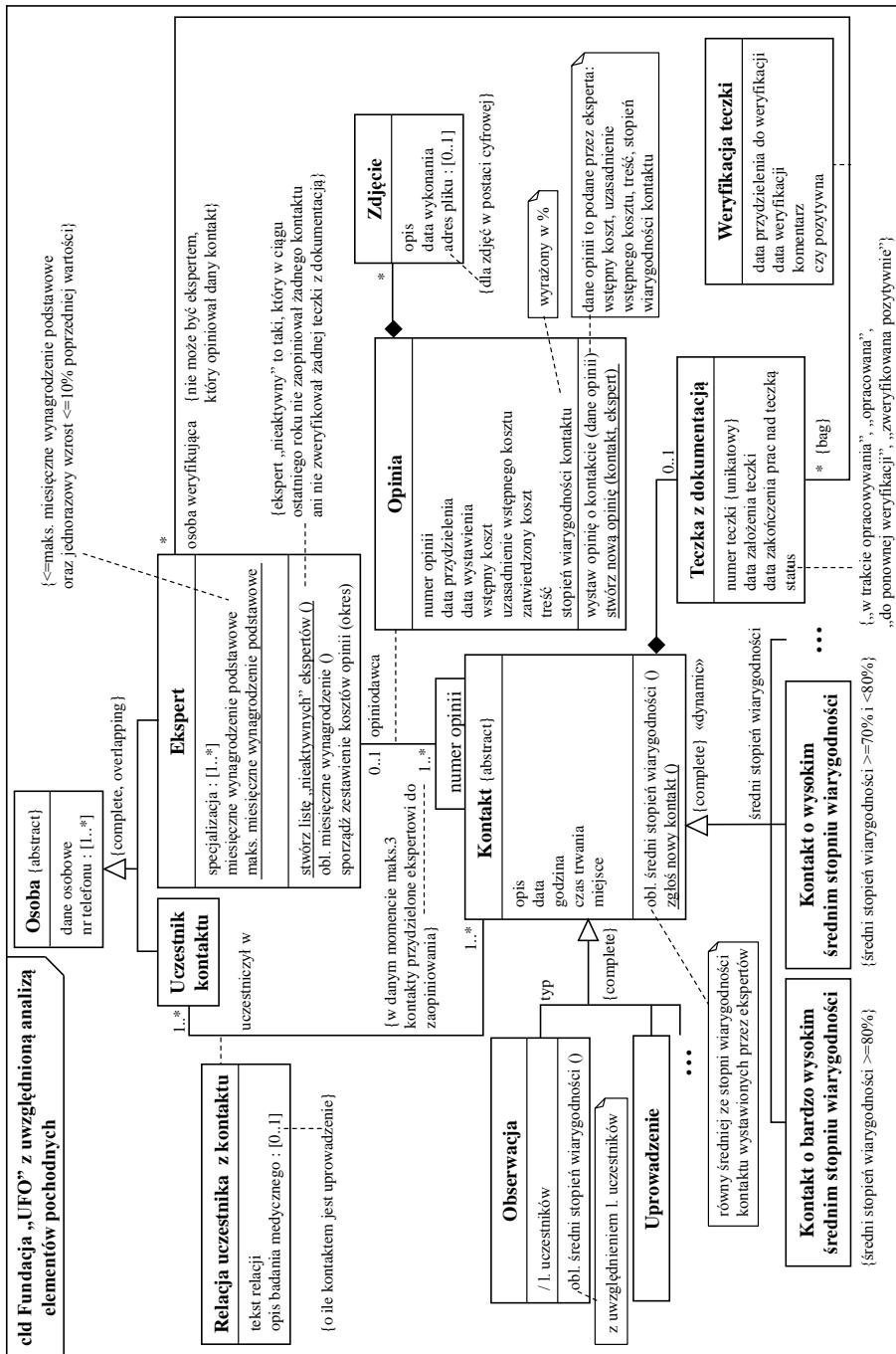
Podsumowując, wprowadzenie do schematu pojęciowego elementu pochodnego ma miejsce w dwóch sytuacjach:

- (1) wartość wyliczona/wyprowadzona przez metodę ma być dodatkowo przechowywana. Po przeanalizowaniu zasadności przechowywania możliwe są dwa rozwiązania: albo umieszcza się elementy pochodne łącznie z wyliczającymi je metodami (które są z reguły opuszczane na tym etapie analizy, ponieważ oczywiste jest, że i tak muszą być one ostatecznie wprowadzone), albo umieszcza się wyłącznie metody wyliczające wartości (bez sugerowania dodatkowego ich przechowywania);
- (2) w celu przypomnienia, że na etapie późniejszym trzeba będzie rozważyć, w jaki sposób wyliczać/wyprowadzać wartość elementu pochodnego.

Schemat pojęciowy, po modyfikacji uwzględniającej analizę elementów pochodnych, został przedstawiony na Rys. 2-15. Dalsze rozważania będą prowadzone w oparciu o ten schemat.

### **2.3.4 Analiza wartości początkowych**

Wartość początkowa jest to wartość, jaką przyjmuje atrybut w trakcie tworzenia obiektu. W sytuacji braku jawniej inicjalizacji może się zdarzyć, że jest ona zupełnie przypadkowa, co z kolei mogłoby skutkować błędny działaniem systemu.

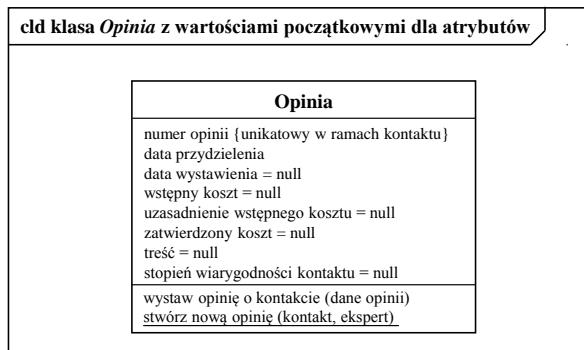


Rys. 2-15 Schemat pojęciowy systemu z uwzględnioną analizą elementów pochodnych

W związku z powyższym należy przeprowadzić analizę wartości początkowych i podjąć decyzję, które atrybuty wymagają domyślnej inicjalizacji.

*Uwaga:* Często zdarza się, że atrybuty są inicjalizowane wartością *null*, co może oznaczać: brak danych, wartość jeszcze nie została wprowadzona/wyliczona lub wartość nieistotna w rozważanym kontekście.

Przykład klasy z zainicjalizowanymi atrybutami przedstawiono na Rys. 2-16.



Rys. 2-16 Klasa *Opinia* z zainicjalizowanymi atrybutami

Na kolejnych schematach będą umieszczane tylko przykładowe wartości początkowe.

### 2.3.5 Analiza wartości granicznych

W fazie analizy, po skonstruowaniu wstępnego schematu pojęciowego, należało by przyjrzeć się także wprowadzonym do niego wartościom granicznym i zastanowić, czy na przestrzeni czasu wartości te mogą ulegać zmianom. Jeśli tak, to przyszły użytkownik systemu musi posiadać takie możliwości bez konieczności ponownej komplikacji kodu systemu.

Zgodnie z tekstem wymagań, mamy do rozważenia np. następującą wartość graniczną związaną z ograniczeniem *{w danym momencie maks. 3 kontakty przydzielone ekspertowi do zaopiniowania}* nałożonym na asocjację łączącą klasy *Ekspert* i *Opinia*.

Potencjalnie, w przyszłości sytuacja finansowa fundacji może ulec zmianie i będzie możliwe przydzielenie ekspertowi większej liczby kontaktów do zaopiniowania, niż to jest aktualnie. Implementacja wartości granicznej w postaci stałej (a nie zmiennej) nie pozwoli na zmianę tej wartości bez ponownej komplikacji kodu. Dobrze jest więc wprowadzić rozwiążanie bardziej elastyczne, polegające na umieszczeniu atrybutu klasowego *maks. l. kontaktów do zaopiniowania w danym momencie* (który będzie przechowywał żądaną wartość graniczną) oraz dostarczenie metody umożliwiającej zmianę tej wartości, obie umieszczone w klasie *Ekspert*. Atrybut powinien zostać zainicjalizowany wartością zgodną ze stanem aktualnym, czyli 3.

Zmienić się powinien także tekst ograniczenia, np. na następujący: {*w danym momencie l. kontaktów przydzielonych ekspertowi do zaopiniowania <= maks. l. kontaktów do zaopiniowania w danym momencie*}.

Na diagramie klas umieszczone również inne wartości graniczne, np. te odnoszące się do podziału kontaktów na grupy ze względu na średni stopień wiarygodności. Aktualnie kontakty o bardzo wysokim średnim stopniu wiarygodności muszą mieć wartość tego stopnia większą bądź równą 80%. Na przestrzeni czasu wartość ta może ulegać zmianom i system powinien ułatwiać realizacje tych zmian. Analogicznie jak w przypadku pierwszej z rozważanych wartości granicznych, należy wprowadzić do podklas klasy *Kontakt* (podział dokonany ze względu na średni stopień wiarygodności) atrybuty klasowe przechowujące wartości graniczne średniego stopnia wiarygodności, zainicjalizowane wartościami początkowymi wynikającymi z aktualnych wymagań. Przykładowo, w podklasie *Kontakt o bardzo wysokim stopniu wiarygodności* należy umieścić atrybut *min. stopień wiarygodności=80%*. W tej klasie nie ma potrzeby umieszczania atrybutu *maks. średni stopień wiarygodności*.

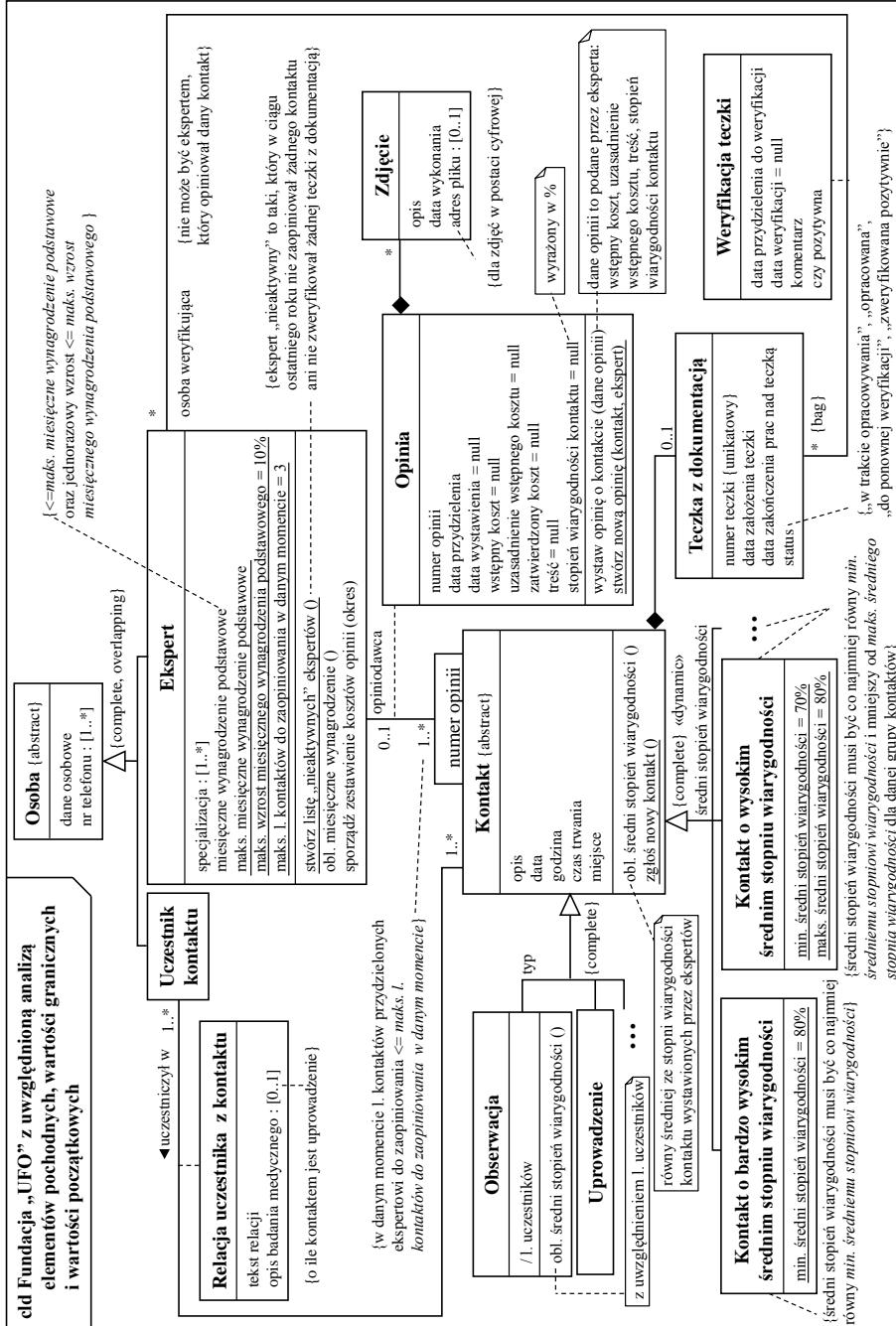
Kolejna wartość graniczna jest związana z atrybutem  *miesięczne wynagrodzenie podstawowe* w klasie *Ekspert*. Na wartość tego atrybutu zostało nałożone ograniczenie *{jednorazowy wzrost<=10%wartości}*. Po przeprowadzeniu rozważań analogicznych do wyżej wymienionych zaproponowano wprowadzenie do klasy *Ekspert* nowego atrybutu klasowego zainicjalizowaną wartością 10% – *maks. wzrost miesięcznego wynagrodzenia podstawowego=10%*, a zmodyfikowane ograniczenie ma postać *{jednorazowy wzrost<=maks. jednorazowy wzrost miesięcznego wynagrodzenia podstawowego}*.

Schemat pojęciowy, po modyfikacji uwzględniającej analizę wartości początkowych atrybutów, wartości pochodnych i wartości granicznych, został przedstawiony na Rys. 2-17. Dalsza analiza będzie bazować na tym właśnie schemacie.

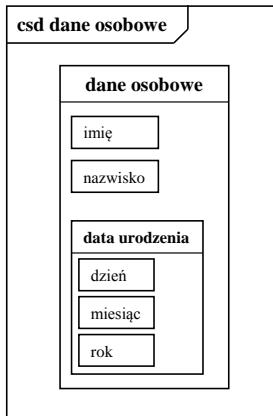
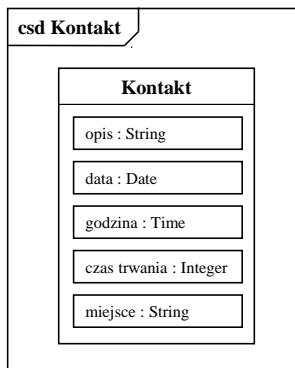
### 2.3.6 Prezentacja struktur elementów złożonych

Analizując schemat pojęciowy, warto przyjrzeć się bliżej sposobowi prezentowania struktur elementów złożonych. Rozważania przeprowadzono w oparciu o atrybut złożony *dane osobowe* umieszczony w klasie *Osoba*. Z tekstu wymagań nie wynika, co mają zawierać takie dane, ale przy założeniu, że w docelowym schemacie pojęciowym byłoby to sprecyzowane, w celu pokazania struktury wewnętrznej takiego atrybutu, można wykorzystać diagram struktur złożonych (ang. *Composite Structure Diagram*), jak na Rys. 2-18.

Inny przykład wykorzystania diagramu struktur złożonych (dla klasy *Kontakt* – z uwzględnieniem typów atrybutów) zaprezentowano na Rys. 2-19.



Rys. 2-17 Schemat pojęciowy systemu z uwzględnieniem analizy elementów pochodnych, wartości granicznych i wartości początkowych

Rys. 2-18 Diagram struktur złożonych dla atrybutu złożonego *dane osobowe*Rys. 2-19 Diagram struktur złożonych dla klasy *Kontakt*

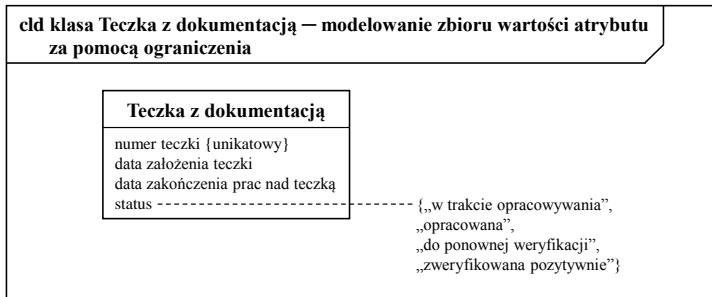
Użyteczność diagramów *csd* wydaje się być oczywista w sytuacji, gdy mamy do czynienia ze złożonym schematem pojęciowym zawierającym dużą liczbę elementów składowych. Wtedy na najwyższym poziomie abstrakcji można uwidoczyć klasy i relacje zachodzące pomiędzy nimi (generalizacje/specjalizacje i różnego rodzaju asocjacje), a na niższych poziomach prezentować struktury poszczególnych elementów.

### 2.3.7 Wykorzystanie typu *enumeration* dla określania zbioru wartości

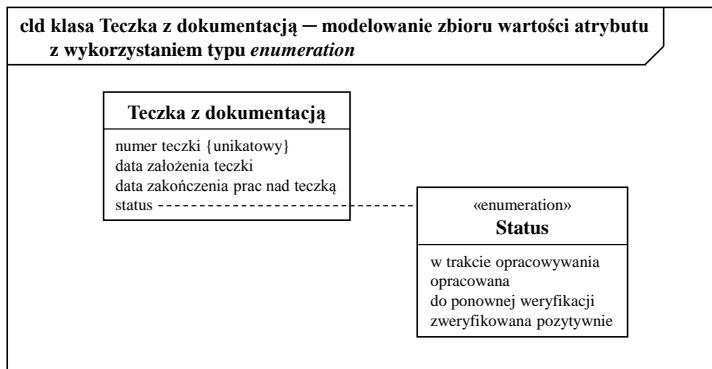
Ze względu na przejrzystość diagramów, upakowanych zazwyczaj maksymalnie na jednej stronie dokumentacji, autorzy przyjęli technikę specyfikowania zbioru wartości atrybutu w postaci ograniczeń (o ile wartości te zostały wymienione w tekście wymagań). Przykładowo, na atrybut *status* w klasie *Teczka z dokumentacją* nałożono na schemacie pojęciowym ograniczenie {„w trakcie opracowywania”, „opracowana”, „do ponownej weryfikacji”, „zweryfikowana”}

*pozytywnie”}. Można to zamodelować inaczej, wykorzystując typ wyliczeniowy *enumeration*, choć zajmuje on więcej miejsca na diagramach i pojęciowo jest bardziej związany z obiektowym środowiskiem implementacji.*

Wykorzystanie ograniczenia do specyfikowania zbioru wartości atrybutu ilustruje Rys. 2-20, a typu *enumeration* – Rys. 2-21.



Rys. 2-20 Modelowanie zbioru wartości atrybutu za pomocą ograniczenia



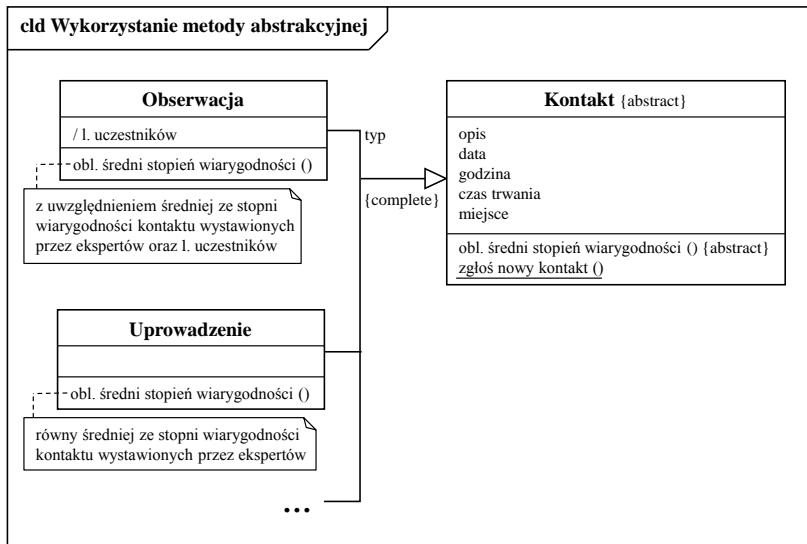
Rys. 2-21 Modelowanie zbioru wartości atrybutu z wykorzystaniem typu *enumeration*

### 2.3.8 Zadania podstawowe

Zaprezentowane poniżej rozwiązania zadań odnoszą się do schematu pojęciowego przedstawionego na Rys. 2-17.

**Zad. 1:** Wskaż metodę abstrakcyjną lub wprowadź ją, o ile nie została dotąd umieszczona.

Na przedstawionym schemacie pojęciowym żadna z metod nie jest metodą abstrakcyjną. Metodą abstrakcyjną mogłaby tu być metoda *obl. średni stopień wiarygodności ()* – przy założeniu, że dla każdego typu kontaktu wykorzystano by inny algorytm wyliczania takiego stopnia. Przykładowy fragment schematu, z umieszczoną na nim metodą abstrakcyjną, został przedstawiony na Rys. 2-22.



Rys. 2-22 Fragment schematu pojęciowego ilustrujący wykorzystanie metody abstrakcyjnej

**Zad. 2:** Wskaż klasy abstrakcyjne. Czy klasa abstrakcyjna może zawierać zaimplementowane metody? Jeśli odpowiedź jest na „tak”, wprowadź zaimplementowaną metodę.

Klasy abstrakcyjne wprowadzone do schematu to klasy: *Kontakt* i *Osoba*. Klasa abstrakcyjna może zawierać metody zaimplementowane. Przykładem metody zaimplementowanej w klasie abstrakcyjnej mogłaby być metoda *wprowadź opis (opis)* w klasie *Kontakt*.

**Zad. 3:** Podaj przykłady metody obiektowej, metody klasowej, atrybutu obiektowego i atrybutu klasowego.

- Metoda obiektowa: *obl. miesięczne wynagrodzenie ()* w klasie *Ekspert*;
- metoda klasowa: *usuń informacje o „nieaktywnych” ekspertach ()* w klasie *Ekspert*;
- atrybut obiektowy: *miesięczne wynagrodzenie podstawowe* w klasie *Ekspert*;
- atrybut klasowy: *maks. miesięczne wynagrodzenie podstawowe* w klasie *Ekspert*.

**Zad. 4:** Wskaż przykład zastosowania polimorfizmu metod.

Metody polimorficzne to metody *obl. średni stopień wiarygodności ()* umieszczone w hierarchii dla kontaktów (w aspekcie *typ*). Jest to również dobry przykład przesłaniania metod.

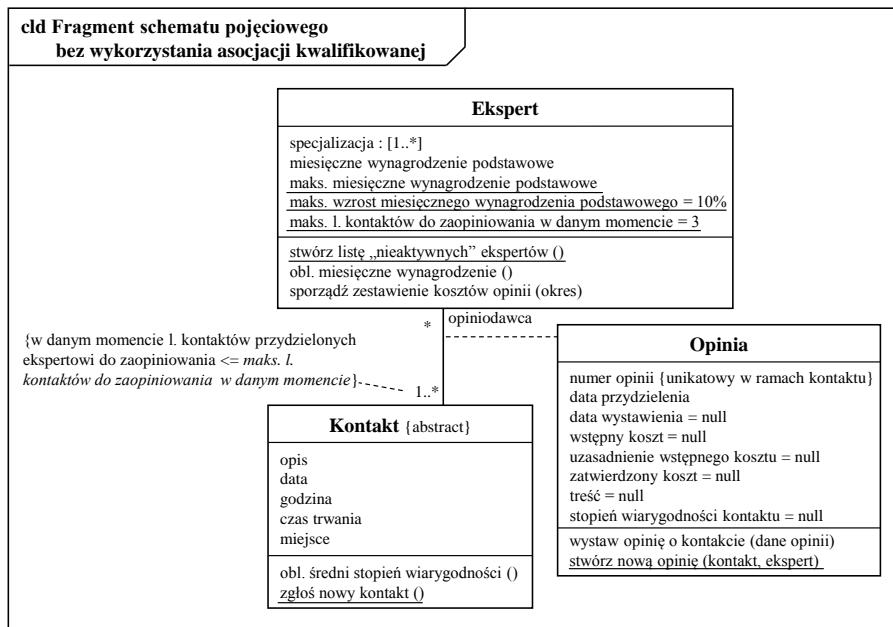
**Zad. 5:** Asocjację posiadającą niesymetryczne liczności oraz atrybut (lub klasę asocjacji) zamień na asocjację kwalifikowaną. Uzasadnij, dlaczego poddałeś zamianie właśnie tę asocjację.

Do schematu pojęciowego wprowadzono asocjację kwalifikowaną łączącą klasy *Ekspert* i *Kontakt*. Wskazuje ona, że kluczem do przeszukiwania zbioru ekspertów przypisanych do danego kontaktu może być numer opinii, który jest unikatowy w ramach kontaktu. Liczność asocjacji naprzeciwko kwalifikatora (równa 0..1) wskazuje na „siłę” klucza, dlatego z klasy *Opinia* usunięto tekst ograniczenia dotyczącego unikatowości atrybutu *numer opinii*.

*Dla przypomnienia:* Stwierdzenie *numer opinii jest unikatowy w ramach kontaktu* oznacza, że numer opinii jest unikatowy w zbiorze opinii dla danego kontaktu, a nie w całej ekstensji klasy *Kontakt*.

*Uwaga:* Po zamianie asocjacji na asocjację kwalifikowaną nie usuwamy kwalifikatora z klasy, w której był pierwotnie umieszczony, co wydaje się bardziej sprzyjać prawidłowej percepcji modelu niż w przypadku, gdyby został usunięty (takie rozwiążanie można spotkać w niektórych publikacjach dotyczących analizy obiektywej).

Rys. 2-23 przedstawia fragment schematu pojęciowego sprzed zamiany asocjacji „zwykłej” na asocjację kwalifikowaną.



Rys. 2-23 Fragment schematu pojęciowego bez wykorzystania asocjacji kwalifikowanej

**Zad. 6:** Wskaż asocjację, która jest agregacją lub kompozycją (lub może kwalifikować się do zamiany na takową). Objaśnij, dlaczego zdecydowałeś się na wybór agregacji/kompozycji, a nie „zwykłej” asocjacji. Ponadto, dlaczego wybrałeś agregację, a nie kompozycję (czy też odwrotnie)?

Do schematu pojęciowego wprowadzono dwie kompozycje. Pierwsza z nich to asocjacja pomiędzy klasami *Kontakt* i *Teczka z dokumentacją*. Klasa *Kontakt* jest całością, a klasa *Teczka z dokumentacją* stanowi element składowy opisu tej całości. Zastosowano kompozycję, ponieważ cykl życiowy teczki z dokumentacją zawiera się w cyklu życiowym kontaktu. Obiekt klasy *Teczka z dokumentacją* nie istnieje bez obiektu klasy *Kontakt* i należy wyłącznie do jednego obiektu klasy *Kontakt*.

Druga kompozycja to asocjacja pomiędzy klasami *Opinia* (całość) i *Zajęcie* (część składowa opisu opinii). Uzasadnienie wyboru kompozycji jw.

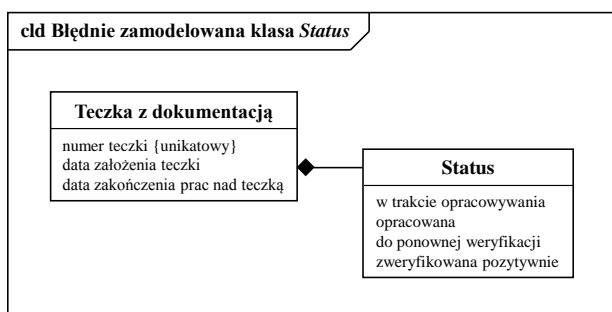
**Zad. 7:** Wskaż ograniczenie odnoszące się do asocjacji. Zaproponuj, jaka metoda i w jakiej klasie mogłaby być odpowiedzialna za jego zapewnienie.

Przykładowe ograniczenie to np. *{nie może być ekspertem, który opiniował dany kontakt}*, nałożone na asocjację pomiędzy klasami *Ekspert* i *Teczka z dokumentacją*, a dokładniej na rolę asocjacji *osoba weryfikująca*, przechowującą informację o ekspertach, którzy przeprowadzili weryfikację danej teczki z dokumentacją kontaktu. Jest to ograniczenie statyczne, a za jego implementację mogłaby być odpowiedzialna metoda *przydziel eksperta do weryfikacji teczki z dokumentacją (ekspert)* umieszczona w klasie *Teczka z dokumentacją*.

### 2.3.9 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

- Pierwsza część komentarzy odnoszących się do typowych błędów dotyczy wyboru klas i ich nazw. Niektóre z uwag zostały już wcześniej umieszczone w przypisach, w rozdziale 2.3.1. Zostaną jednak one powtórzone w tym miejscu, ponieważ zawierają opisy błędów szczególnie często popełnianych przez osoby nauczane/uczące się w ich pracach.
  - Błądem jest wprowadzenie klasy modelującej aktora do schematu pojęciowego (o ile nie jest to część informacji ze schematu pojęciowego), np. *Pracownik fundacji* jest wyłącznie aktorem i dlatego na tym etapie powinien zostać umieszczony jedynie na diagramach przypadków użycia.
  - Nazwy klas to z reguły rzeczowniki w liczbie pojedynczej, a więc *Uprowadzenie*, *Obserwacja*, a nie *Uprowadzenia*, *Obserwacje*. Nazwa klasy musi być skojarzona z informacją przechowywaną w jednym obiekcie tej klasy, czyli nazwa klasy *Uprowadzenie*, jeśli pojedynczy obiekt przechowuje informacje o jednym uprowadzeniu, z kolei *Uprowadzenia*, jeśli przechowuje informacje o grupie uprowadzeń. Należy zwrócić uwagę na fakt, że istnieją w języku polskim rzeczowniki występujące wyłącznie w liczbie mnogiej, jak np. drzwi, spodnie, zajęcia, ćwiczenia (przy czym zajęcia i ćwiczenia są tu rozumiane jako np. zajęcia prowadzone na uczelni, czyli tzw. jednostka lekcyjna, a nie np. ćwiczenie fizyczne), dla których nie znajduje zastosowania wyżej wymieniona reguła.

- Nazwa podklasy powinna zachowywać „semantyczną zgodność” z nazwą nadklasy, a więc *Kontakt o bardzo wysokim średnim stopniu wiarygodności*, a nie *Wysoki stopień wiarygodności*. Obiektem podklasy klasy *Kontakt* jest kontakt, a nie stopień wiarygodności. Inaczej mówiąc, stopień wiarygodności nie jest rodzajem kontaktu. Obiekt podklasy powinien być rodzajem (szczególnym przypadkiem) wystąpienia pośredniego nadklasy.
- Na diagramie nie należy umieszczać klas modelujących byty, dla których nie jest planowane przechowywanie informacji o nich w systemie. Przykładem może tu być pojęcie *papierowe dokumenty*, które oznacza dokumenty przechowywane poza systemem. W wymaganiach nie ma mowy o elektronicznej wersji dokumentów, dlatego też błędem jest wprowadzenie do modelu klasy *Dokument*.
- Częstym błędem jest umieszczanie klasy z listą atrybutów zamiast atrybutu z ograniczeniem zawierającym zbiór dopuszczalnych wartości dla tego atrybutu. Przykładem może tu być atrybut *status* w klasie *Teczka z dokumentacją*, z ograniczeniem {„w trakcie opracowywania”, „opracowana”, „do ponownej weryfikacji”, „zweryfikowana pozytywnie”} nałożonym na wartość tego atrybutu. Błąd jest wprowadzenie do schematu pojęciowego klasy *Status* z atrybutami takimi jak: *w trakcie opracowywania, opracowana, do ponownej weryfikacji, zweryfikowana pozytywnie* (Rys. 2-24). Jakie wartości miałyby przechowywać takie atrybuty?



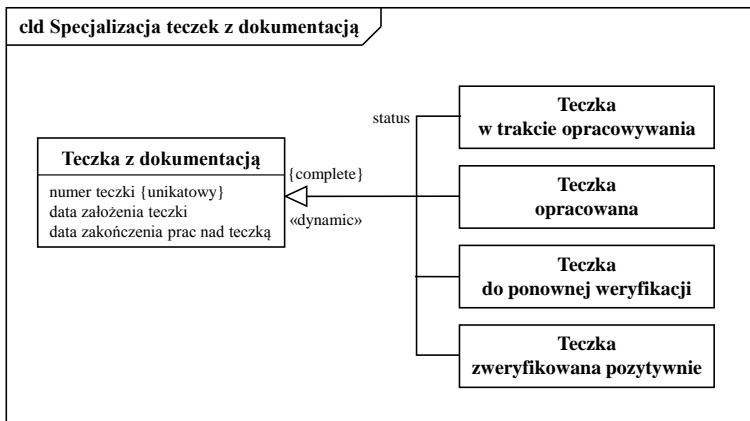
Rys. 2-24 Błędnie zamodelowana klasa *Status*

Rozwiążanie powyższe byłoby poprawne, gdyby nazwa klasyfikatora (typu) została poprzedzona stereotypem «enumeration», a także został on inaczej powiązany z klasą *Teczka z dokumentacją*, jak na Rys. 2-21.

W modelach proponowanych przez osoby nauczane/uczące się pojawiały się również rozwiązania, w których zamiast atrybutu *status*, wykorzystywano hierarchię skonstruowaną dla teczek z dokumentacją. Na tym etapie analizy, w świetle istniejących wymagań, byłoby to rozwiązanie nadmiarowe, aczkolwiek można by je w pewnych sytuacjach uznać za użyteczne, ponieważ poprawia percepcję diagramu – hierarchia jest zdecydowanie lepiej widoczna, niż atrybut łącznie z ograniczeniem na wartości, które może przyjmować. Gdyby jednak w przyszłości okazało się, że teczki

z dokumentacją opisywane byłyby dodatkowymi własnościami zależnymi od wartości statusu bądź też przyszły użytkownik chciałby przeszukiwać teczki po statusie, organizacja zawierająca hierarchię dla statusu z pewnością by to ułatwiła.

Na Rys. 2-25 przedstawiono fragment schematu pojęciowego zawierający specjalizację teczek z dokumentacją. Ponieważ, w zależności od wartości statusu, dana teczka będzie zmieniała przynależność do klasy, wykorzystano specjalizację dynamiczną.



Rys. 2-25 Specjalizacja dynamiczna dla teczek z dokumentacją kontaktu

- Nazwy metod muszą być czasownikami lub frazami czasownikowymi, sugerującymi komunikaty przesyłane do obiektów – nazwa taka, jak np. *zestawienie ekspertów* jest błędna, bo nie wiadomo, jakiego rodzaju operacja ma być przeprowadzona na obiektach klasy *Ekspert* (np. o odczyt czy zapis zestawienia).
- Asocjacje, dla których nie wprowadzono nazw, zazwyczaj są mało użyteczne i utrudniają percepcję diagramu – nie wiadomo, co modelują. Nazwę asocjacji można pominąć tylko wtedy, gdy jest oczywista dla wszystkich uczestników projektu.
- Nazwy asocjacji, jak np. *rodzaj*, *w*, *na* są niedopuszczalne.
- *Dla przypomnienia:* Nie należy wprowadzać elementów z fazy projektowania/implementacji, takich jak np. *id kontaktu*, które ani *explicite*, ani *implicite* nie wynikają z tekstu wymagań. Tworzony model dotyczy fazy analizy i ma opisywać wymagania użytkownika – niezależnie od potencjalnego przyszłego środowiska implementacji.
- *Dla przypomnienia:* Stosowanie klasy pośredniczącej zamiast klasy asocjacji, powoduje utratę informacji o rzeczywistym charakterze relacji między bytami w modelowanym fragmencie dziedziny problemowej. Przykładowo, klasy *Opinia*, *Relacja uczestnika z kontaktem*, *Weryfikacja teczki* powinny zostać

wprowadzone do schematu pojęciowego jako klasy asocjacji. Zamiana klas asocjacji na klasy pośredniczące jest przeprowadzana zazwyczaj dopiero w fazie projektowania, a nie w fazie analizy i tylko dlatego, że środowiska implementacyjne nie posiadają natywnych środków do tzw. „bezszwowego” implementowania klas asocjacji.

### **2.3.10 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji**

W trakcie transformowania schematu pojęciowego, który jest niezależny od środowiska implementacji, na schemat projektowy [37, 40] (przyjmując założenie, że docelowym środowiskiem implementacyjnym będzie popularny język obiektowy, np. Java, czy C#) muszą być wykonane takie czynności, jak: uszczegóławianie wyników analizy, dostosowanie do ograniczeń i możliwości środowiska implementacji oraz optymalizacja systemu.

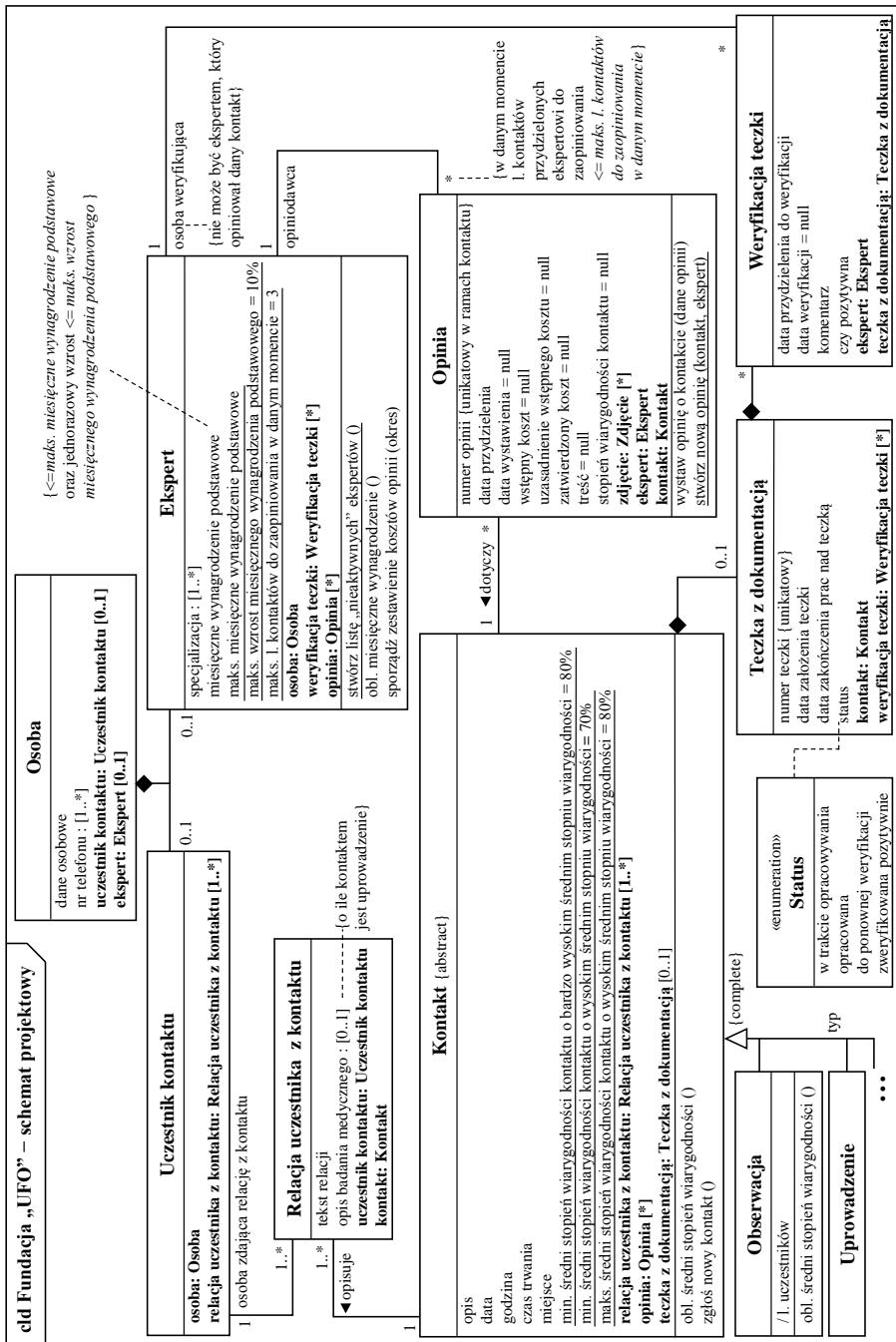
*Dla przypomnienia:* Uszczegóławianie wyników analizy polega na podaniu reguł odwzorowania struktur pojęciowych w struktury tego języka programowania, który będzie wykorzystany do implementacji systemu. Czynności, jakie należy wykonać, to m.in.: obejście braku dziedziczenia nietrywialnego, zamiana klas asocjacji na klasy pośredniczące, określenie szczegółowych specyfikacji atrybutów i metod (oznaczenie typów, dostępności, podanie pełnych sygnatur metod oraz wyspecyfikowanie typów argumentów i wartości zwracanej dla każdej metody), określenie sposobów implementacji asocjacji (z reguły poprzez wprowadzenie dodatkowych atrybutów), identyfikacja asocjacji skierowanych oraz określenie sposobów implementowania ekstensji klas (patrz [41]).

W rozwiążaniu proponowanym na Rys. 2-26 skupiono uwagę wyłącznie na eliminowaniu dziedziczenia nietrywialnego (nierozłącznego, wieloaspektowego, dynamicznego) i implementowaniu asocjacji poprzez umieszczenie dodatkowych atrybutów będących referencjami lub kolekcjami referencji do powiązanych obiektów, w zależności od liczności asocjacji (atrybuty te wyróżnione zostały przez pogrubienie), natomiast pominięto m.in. specyfikacje typów dla atrybutów i metod.

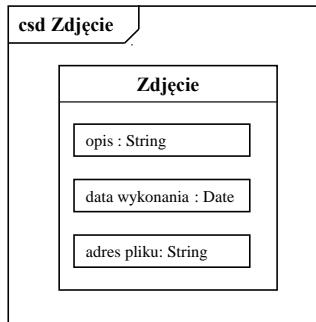
Schemat projektowy z Rys. 2-26 utworzono w oparciu o schemat pojęciowy po przeprowadzonej analizie elementów pochodnych, wartości początkowych i wartości granicznych (z Rys. 2-17).

Informację o zdjęciach dołączonych do opinii zamodelowano w postaci atrybutu powtarzanego *zdjęcie*, będącego kolekcją podobiektów obiektu klasy *Opinia*. Na Rys. 2-27 przedstawiono strukturę klasy *Zdjęcie* z wykorzystaniem diagramu struktur złożonych (*csd*).

Inaczej niż dla zdjęć, informacje o teczkach z dokumentacją i ich weryfikowaniu, ze względu na powiązania istniejące pomiędzy obiektami klas *Teczka* z dokumentacją i *Weryfikacja teczki* oraz *Weryfikacja teczki i Osoba*, zamodelowano z wykorzystaniem kompozycji i zmiennych referencyjnych.



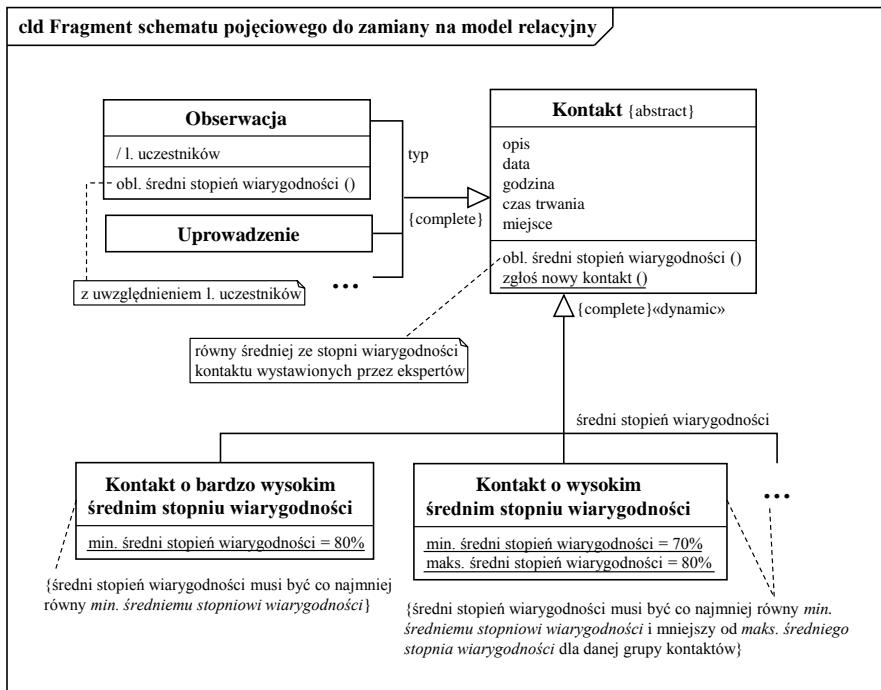
Rys. 2-26 Schemat projektowy

Rys. 2-27 Diagram struktur złożonych dla atrybutu *zdjęcie*

### 2.3.11 Schemat relacyjny dla fragmentu diagramu klas

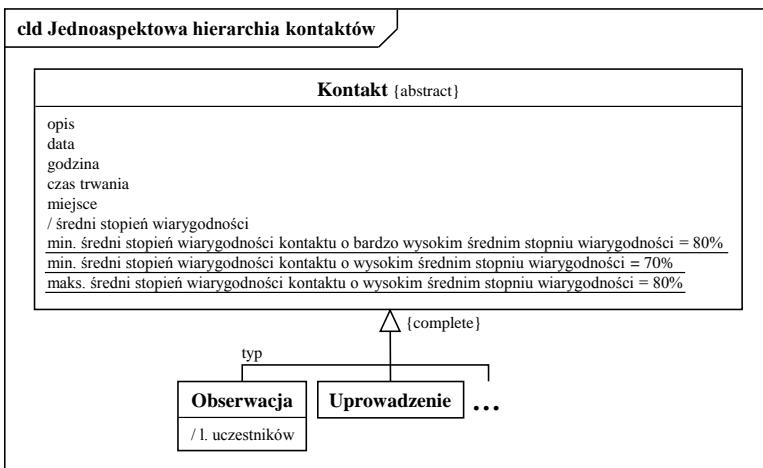
*Polecenie:* Fragment diagramu klas zawierający co najmniej trzy klasy oraz związek dziedziczenia przedstaw w modelu relacyjnym.

Fragment diagramu klas ze schematu pojęciowego, podlegający zamianie na model relacyjny [37, 47], przedstawiono na Rys. 2-28.



Rys. 2-28 Fragment schematu pojęciowego podlegający zamianie na model relacyjny

Przed zamianą na model relacyjny, podobnie jak dla schematu projektowego, dokonano transformacji hierarchii dla kontaktów – w celu wyeliminowania dziedziczenia dwuaspektowego oraz dziedziczenia dynamicznego. Do nadklasy *Kontakt* przeniesiono wszystkie własności z podziału kontaktów ze względu na średni stopień wiarygodności. Mówiąc kolokwialnie, ten aspekt dziedziczenia został „zwinięty do nadklasty”. W rezultacie zamianie na model relacyjny podlegała hierarchia kompletna, rozłączna i jednoaspektowa, jak na Rys. 2-29.



Rys. 2-29 Jednoaspektowa hierarchia kontaktów

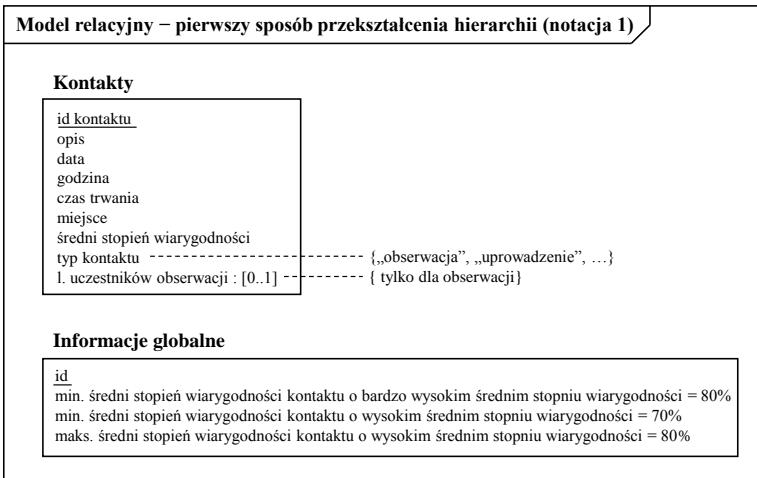
Model relacyjny, dla fragmentu diagramu klas przedstawionego na Rys. 2-29, zaprezentowano w trzech wersjach, różniących się sposobem przekształcenia hierarchii [41]. Każda z wersji została przedstawiona w trzech przykładowych notacjach, dwie pierwsze zostały omówione m.in. w: [41], a trzecią (ang. *Crow's Foot*) można uznać za jedną z częściej wykorzystywanych [44].

### Pierwszy sposób przekształcenia hierarchii – Utworzenie jednej tabeli dla całej hierarchii

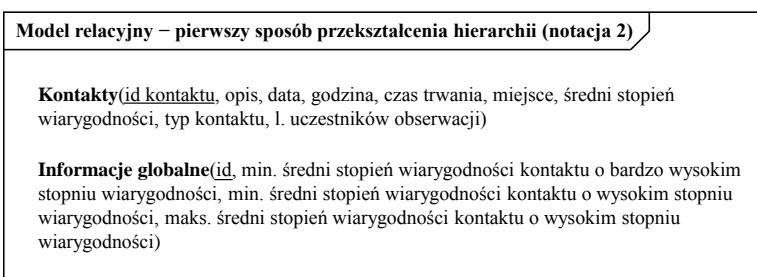
Pierwszy ze sposobów obejścia dziedziczenia, również dziedziczenia trywialnego, polega na utworzeniu jednej tabeli dla całej hierarchii klas poprzez przeniesienie do nadklasty wszystkich własności wynikających z podziału oraz dodanie dodatkowego atrybutu – dyskryminatora wariantu.

W rozważanej hierarchii występowały dwie metody polimorficzne *obl. średni stopień wiarygodności* () w klasach *Kontakt* i *Obserwacja*. Ponieważ zaproponowane rozwiązanie nie wspiera polimorfizmu metod, nie jest ono z tego powodu zalecane do wykorzystania.

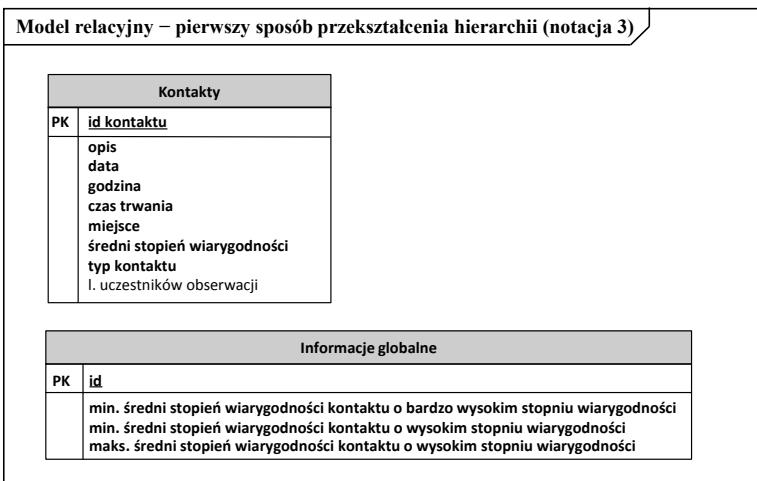
Odpowiedni model relacyjny został zaprezentowany kolejno na Rys. 2-30 (notacja 1), Rys. 2-31 (notacja 2) oraz na Rys. 2-32 (notacja 3).



Rys. 2-30 Model relacyjny – pierwszy sposób przekształcenia hierarchii (notacja 1)



Rys. 2-31 Model relacyjny – pierwszy sposób przekształcenia hierarchii (notacja 2)

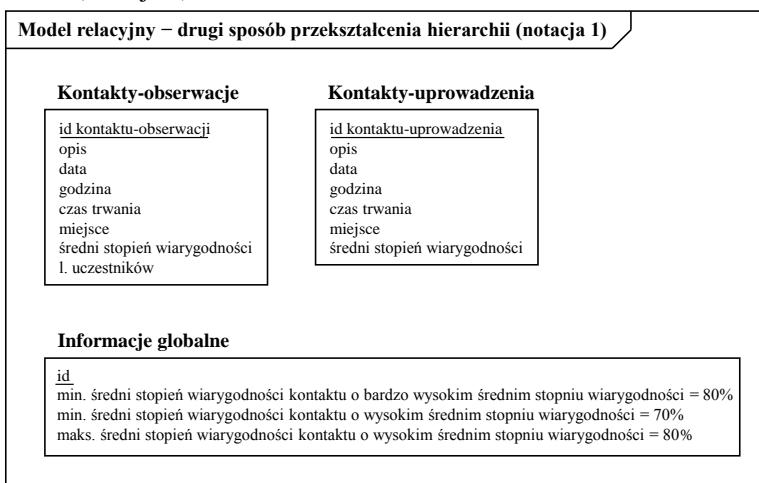


Rys. 2-32 Model relacyjny – pierwszy sposób przekształcenia hierarchii (notacja 3)

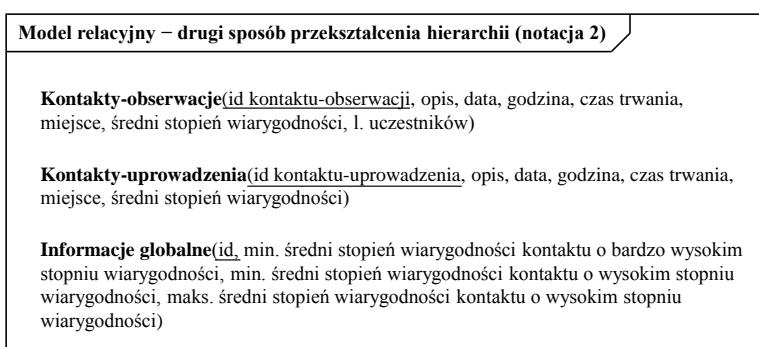
Należy zwrócić uwagę na fakt, że we wszystkich prezentowanych notacjach pojawiła się dodatkowa tabela *Informacje globalne*. Dla przypomnienia: Przy zamianie na model relacyjny wszystkie atrybuty klasowe zostają umieszczone w odrębnej tabeli, niezależnie od tego, w której klasie się znajdowały (na schemacie pojęciowym). Nazwa takiej tabeli powinna wyraźnie wskazywać, że zawiera ona informacje globalne. Powyższa reguła odnosi się także do dalszych rozważań.

### **Drugi sposób przekształcenia hierarchii – Utworzenie oddzielnych tabel dla każdej podklasy**

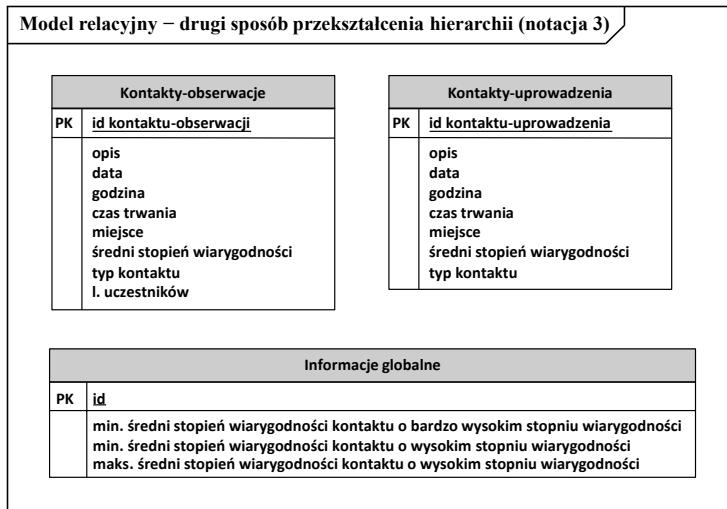
Drugi sposób obejścia dziedziczenia przy zamianie na model relacyjny polega na utworzeniu odrębnych tabel dla wszystkich podklas wyróżnionych w hierarchii, łącznie z atrybutami, które odziedziczyły po nadklasie, co oznacza usunięcie nadklasy i przesunięcie jej atrybutów do podklas. Odpowiedni model relacyjny został zaprezentowany kolejno na Rys. 2-33 (notacja 1), Rys. 2-34 (notacja 2) oraz na Rys. 2-35 (notacja 3).



Rys. 2-33 Model relacyjny – drugi sposób przekształcenia hierarchii (notacja 1)



Rys. 2-34 Model relacyjny – drugi sposób przekształcenia hierarchii (notacja 2)

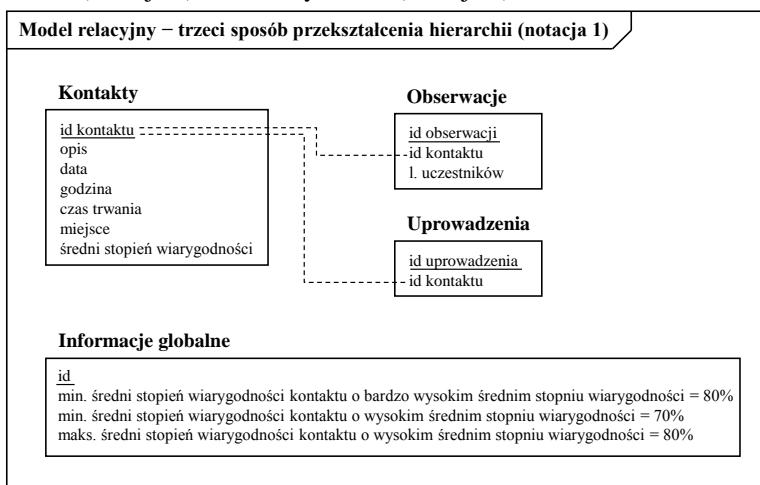


Rys. 2-35 Model relacyjny – drugi sposób przekształcenia hierarchii (notacja 3)

### Trzeci sposób przekształcenia hierarchii – Utworzenie oddzielnego tabel dla każdej klasy

Trzeci sposób obejścia dziedziczenia polega na zamianie każdej klasy występującej w hierarchii na odrębną tabelę.

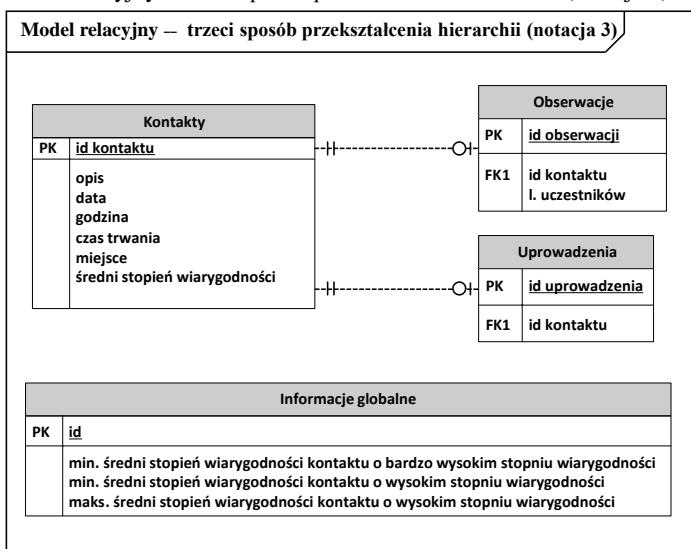
Odpowiedni model relacyjny został zaprezentowany kolejno na Rys. 2-36 (notacja 1), Rys. 2-37 (notacja 2) oraz na Rys. 2-38 (notacja 3).



Rys. 2-36 Model relacyjny – trzeci sposób przekształcenia hierarchii (notacja 1)

<b>Model relacyjny – trzeci sposób przekształcenia hierarchii (notacja 2)</b>	
<b>Kontakty</b>	( <u>id kontaktu</u> , opis, data, godzina, czas trwania, miejsce, średni stopień wiarygodności)
<b>Obserwacje</b>	( <u>id obserwacji</u> , id kontaktu, l. uczestników)
<b>Uprowadzenia</b>	( <u>id uprowadzenia</u> , id kontaktu)
<b>Informacje globalne</b>	( <u>id</u> , min. średni stopień wiarygodności kontaktu o bardzo wysokim stopniu wiarygodności, min. średni stopień wiarygodności kontaktu o wysokim stopniu wiarygodności, maks. średni stopień wiarygodności kontaktu o wysokim stopniu wiarygodności)

Rys. 2-37 Model relacyjny – trzeci sposób przekształcenia hierarchii (notacja 2)



Rys. 2-38 Model relacyjny – trzeci sposób przekształcenia hierarchii (notacja 3)

*Uwaga:* Przedstawione rozwiązania obejścia dziedziczenia w różnym stopniu wspierają polimorfizm metod. Przyjmując dla wsparcia polimorfizmu skalę wartości {„słabe”, „średnie”, „duże”}, zastąpienie hierarchii jedną tabelą nie wspiera polimorfizmu metod („słabe” wsparcie), w rozwiążaniu polegającym na stworzeniu oddzielnych tabel dla każdej z podklas, można określić je jako „średnie”, a „duże” – w przypadku stworzenia odrębnych tabel dla każdej z klas hierarchii.

## 2.4 Analiza dynamiczna

Zaprezentowane poniżej rozwiązania zadań dotyczących analizy dynamicznej (patrz [36, 40, 41]) odnoszą się do schematu pojęciowego przedstawionego na Rys. 2-17.

### 2.4.1 Zadania podstawowe

**Zad. 1:** Dla klasy *Teczka z dokumentacją* wskaż kilka stanów (co najmniej dwa stany, nie licząc stanów początkowego i końcowego), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

*Uwaga:* W tym zadaniu nie jest wymagana cała maszyna stanowa, wystarczy zaprezentować dwa stany wraz ze specyfikacją przejścia.

*Dla przypomnienia:*

- *Maszyna stanowa* jest grafem skierowanym, reprezentowanym za pomocą notacji diagramów stanów. Wierzchołki grafu stanowią stany obiektu/obiektów, a krawędzie opisują przejścia między stanami. Przejście między stanami jest odpowiedzią na zdarzenie.
- *Maszyna stanowa zachowania* – może dotyczyć zmian stanów nie jednego, lecz wielu obiektów w aspekcie pewnego fragmentu zachowania systemu.
- *Protokołowa maszyna stanowa* to rodzaj maszyny stanów, która przypisana jest do jednej klasy i specyfikuje reakcje obiektów tej klasy na zdarzenia, które do nich przychodzą (najczęściej są to komunikaty), stanowiąc w ten sposób *model historii życia* obiektów klasy, będący opisem wszystkich możliwych stanów i przejść.

Składnia etykiety przejścia jest różna dla różnych rodzajów maszyny stanowej:

dla maszyny stanowej zachowania

*nazwa zdarzenia [warunek] / operacja,*

a dla maszyny protokołowej

*[warunek początkowy] operacja / [warunek końcowy].*

Wszystkie elementy składowe etykiet są opcjonalne. Przykładowo, w maszynie stanowej zachowania, etykieta dla przejścia ze stanu „*Teczka z dokumentacją opracowana*” do stanu „*Teczka z dokumentacją do ponownej weryfikacji*” mogłaby mieć postać:

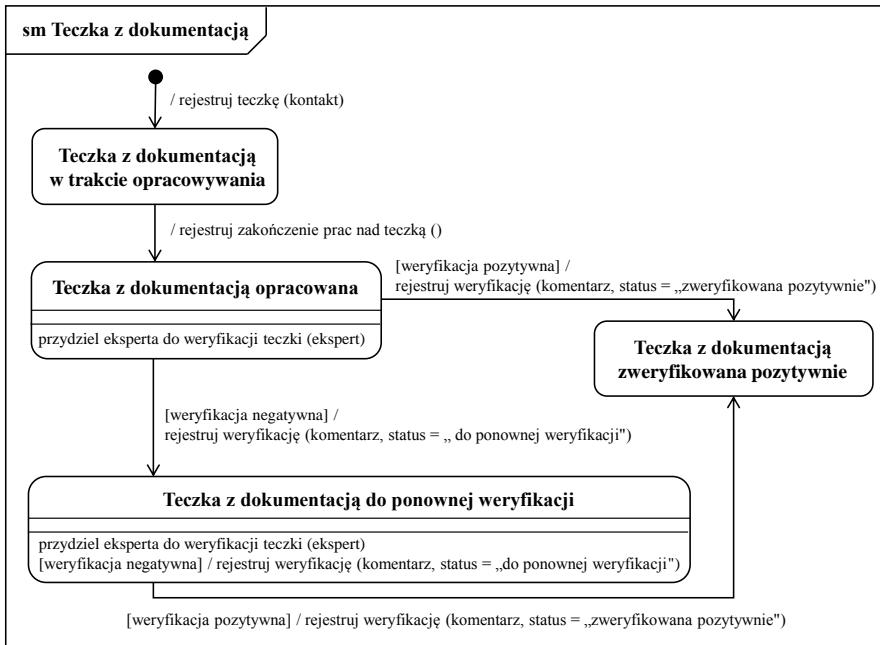
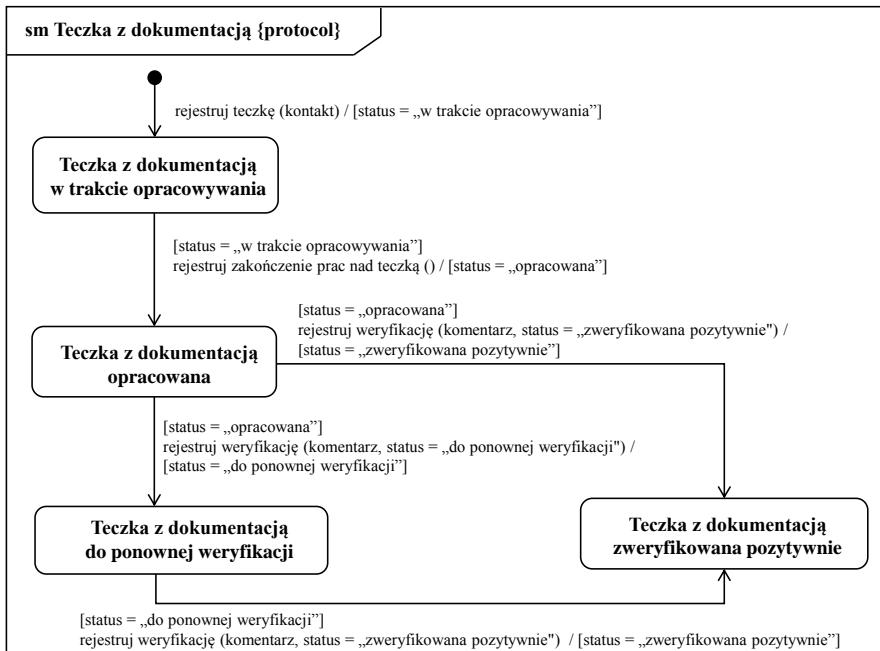
*[weryfikacja negatywna] / rejestruj weryfikację (komentarz,  
status="do ponownej weryfikacji"),*

a w przypadku protokołowej maszyny stanowej:

*[status="opracowana"] rejestruj weryfikację (komentarz, status="do  
ponownej weryfikacji") / [status="do ponownej weryfikacji"].*

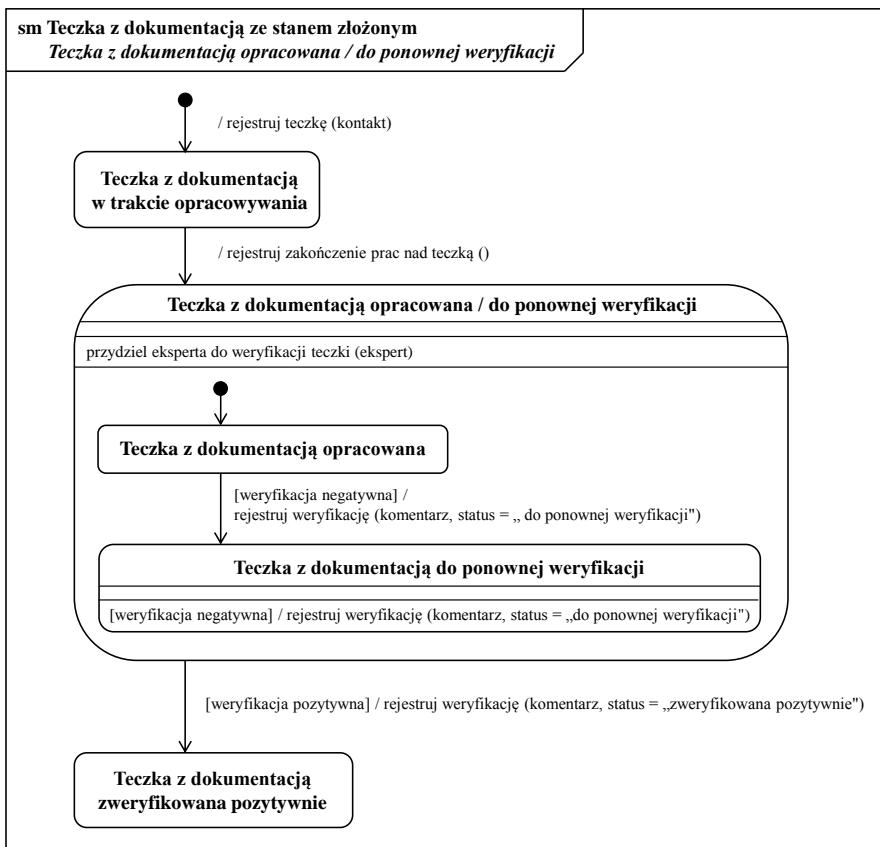
Maszyna stanowa zachowania dla klasy *Teczka z dokumentacją* została przedstawiona na Rys. 2-39.

Na Rys. 2-40 przedstawiono protokołową maszynę stanową dla tej samej klasy.

Rys. 2-39 Maszyna stanowa zachowania dla klasy *Teczka z dokumentacją*Rys. 2-40 Protokołowa maszyna stanowa dla klasy *Teczka z dokumentacją*

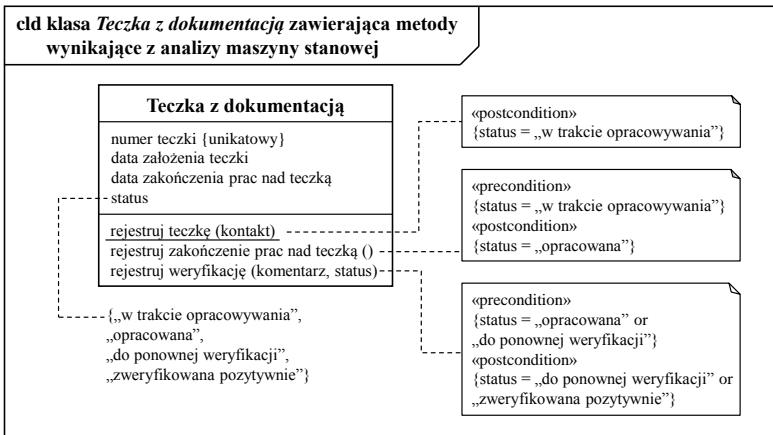
Ze względu na to, że w tekście wymagań nie zamieszczono informacji o warunkach usuwania kontaktów/opinii/teczek z dokumentacją, do żadnego z obu zamieszczonych diagramów nie został wprowadzony stan końcowy.

Ponieważ stany *Teczka z dokumentacją opracowana* i *Teczka z dokumentacją do ponownej weryfikacji* (maszyna stanowa zachowania z Rys. 2-39) posiadają takie samo zdarzenie wewnętrzne *przydziel eksperta do weryfikacji teczki (ekspert)*, a także ze względu na to, że rejestrowanie weryfikacji pozytywnej może być wykonane wyłącznie wówczas, gdy teczka występuje w jednym z obu wymienionych stanów, można rozważyć wprowadzenie stanu złożonego sekwencyjnego *Teczka z dokumentacją opracowana / do ponownej weryfikacji*, który obejmowałby oba stany, jak na Rys. 2-41.



Rys. 2-41 Maszyna stanowa zachowania dla klasy *Teczka z dokumentacją ze stanem złożonym Teczka z dokumentacją opracowana / do ponownej weryfikacji*

W trakcie analizy przedstawionych maszyn stanowych zidentyfikowano operacje, które należałyby uwzględnić w schemacie pojęciowym. Odpowiedni fragment tego schematu został przedstawiony na Rys. 2-42.



Rys. 2-42 Fragment schematu pojęciowego, zawierający klasę *Teczka z dokumentacją*, uwzględniający zmiany wynikające z analizy obu rodzajów maszyn stanowych

**Zad. 2:** Dla funkcjonalności *Wystaw opinię o kontakcie* skonstruuj diagram aktywności zgodny ze scenariuszem, który został umieszczony w podrozdziale *Analiza funkcjonalna* (Tab. 2-2).

Diagramy aktywności są wykorzystywane do określenia kolejnych kroków w trakcie realizowania funkcji systemu o różnym stopniu złożoności (np. przypadków użycia), bez określania, które klasy czy podsystemy są odpowiedzialne za zrealizowanie danej aktywności, a taką informację można przekazać za pomocą np. regionów dla diagramów aktywności i/lub wykorzystując diagramy interakcji.

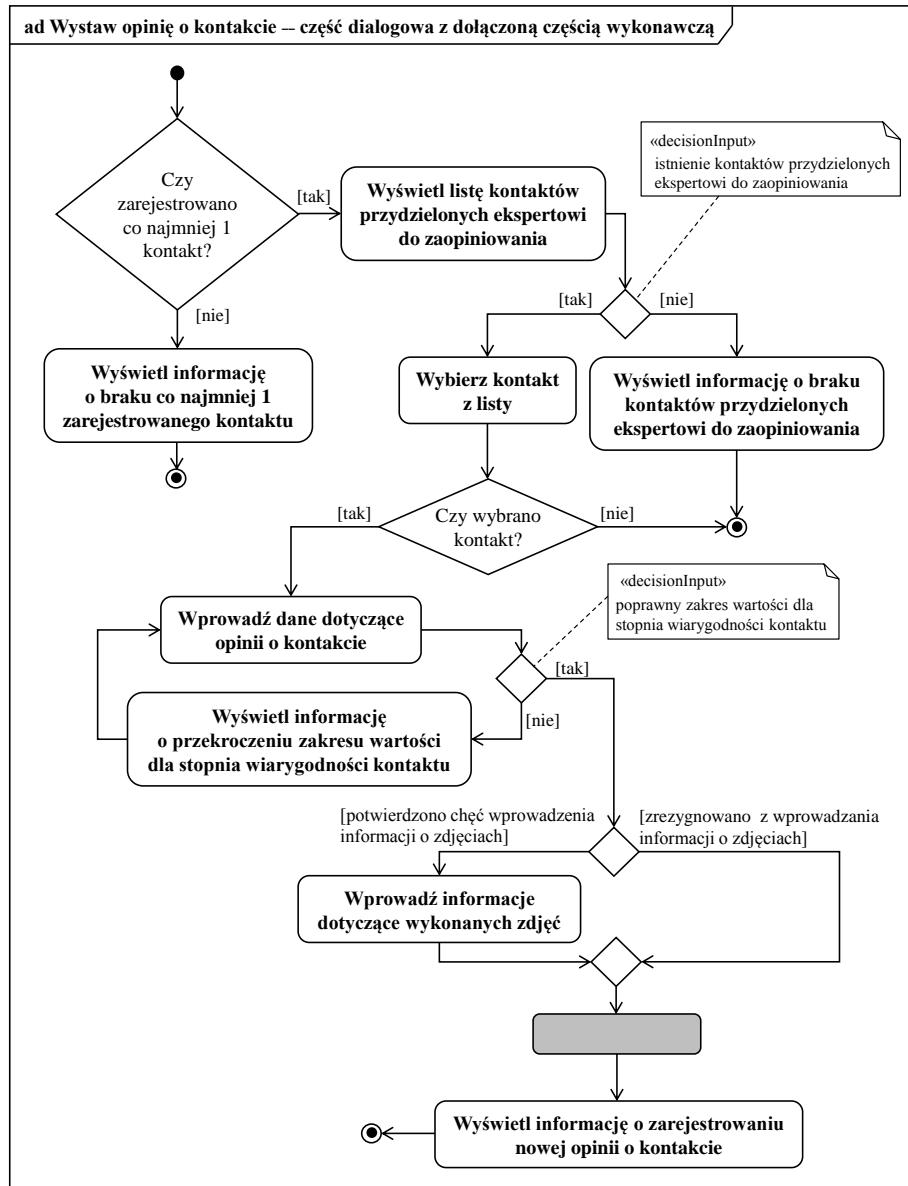
Całość realizacji przypadku użycia można podzielić na dwie części:

- część dialogową, której zadaniem jest komunikowanie się z aktorem w celu np. pobierania danych wejściowych (niezbędnych do zrealizowania przypadku) czy też informowanie o wynikach, błędach itp.;
- część wykonawczą, która jest odpowiedzialna za zrealizowanie zadania, np. zmiany własności, zrealizowanie zapytania itp.

Część dialogową przypadków zazwyczaj specyfikuje się za pomocą scenariuszy, które są zapisywane w języku naturalnym. Ponieważ może to sprawiać problem szczególnie dla scenariuszy złożonych (np. takich z dużą liczbą warunków czy iteracji), do scenariuszy dołączane są często diagramy aktywności.

*Uwaga:* Nie zawsze całość realizacji przypadku udaje się podzielić na dwie rozłączne części, np. część wykonawcza może poprzedzać komunikat końcowy przypisany do części dialogowej.

Przykładową realizację dla scenariusza sporzązonego w punkcie 2.2.3 (Tab. 2-2) ilustruje diagram aktywności zawarty na Rys. 2-43. Przedstawiono na nim część dialogową z dołączoną częścią wykonawczą, oznaczoną w postaci „szarej” aktywności. Jak widać, część wykonawcza poprzedza komunikat końcowy.



Rys. 2-43 Diagram aktywności dla przypadku użycia *Wystaw opinię o kontakcie* – część dialogowa z dołączoną częścią wykonawczą

Dla przypomnienia: Scenariusz przedstawiony w Tab. 2-2 nie uwzględnia sytuacji, w której przypadek użycia *Wystaw opinię o kontakcie* jest wywoływany z przypadku *Przeglądaj aktualne przydziały do prac*.

Ponieważ w przypadku rombu decyzyjnego możliwe są różne notacje służące do zamodelowania tej samej sytuacji, dlatego dla celów dydaktycznych na diagramie wykorzystano każdą z nich. Mimo że decyzja o wyborze notacji należy do analityka tworzącego model, to dobrze jest zachować jednolitą formę (tzn. nie wykorzystywać różnych notacji na tym samym diagramie), dzięki czemu całość będzie bardziej uporządkowana.

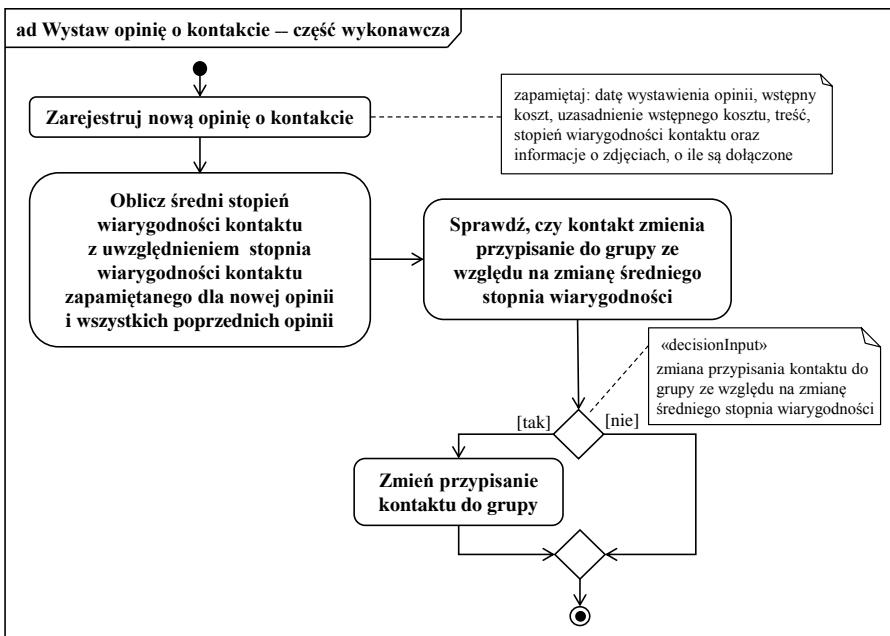
Część wykonawczą realizacji omawianego przypadku użycia można przedstawić za pomocą uproszczonego algorytmu zapisanego w postaci pseudokodu:

```

begin
    zarejestruj nową opinię o kontakcie (kontakt, ekspert, wstępny
    koszt, uzasadnienie wstępnego kosztu, treść, stopień wiarygodności
    kontaktu, zdjęcie [*]);
    średni stopień wiarygodności kontaktu := obl. średni stopień
    wiarygodności(); /* z uwzględnieniem wszystkich posiadanych opinii
    o kontakcie */
    zmiana przypisania kontaktu do grupy := sprawdź czy kontakt zmienia
    przypisanie do grupy ();
    if (zmiana przypisania kontaktu do grupy)
        then zmień przypisanie kontaktu do grupy()
end

```

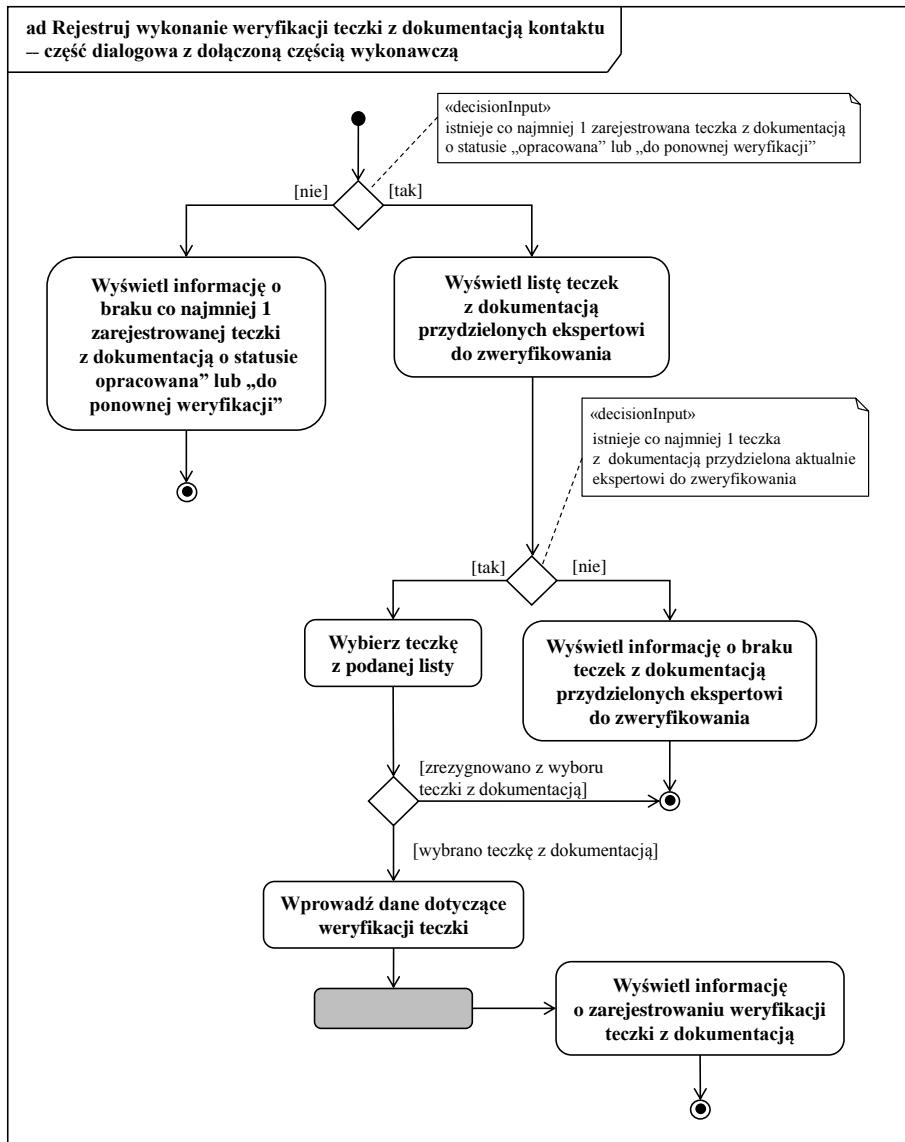
Na Rys. 2-44 przedstawiono diagram aktywności dla części wykonawczej odpowiedni dla algorytmu podanego powyżej.



Rys. 2-44 Diagram aktywności dla przypadku użycia *Wystaw opinię o kontakcie – część wykonawcza*

**Zad. 3:** Dla funkcjonalności *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu* skonstruj diagram aktywności zgodny ze scenariuszem, który został umieszczony w podrozdziale *Analiza funkcjonalna* (pkt 2.2.3, Tab. 2-1).

Przykładową realizację scenariusza sporzązonego dla wybranej funkcjonalności ilustruje diagram aktywności zaprezentowany na Rys. 2-45.



Rys. 2-45 Diagram aktywności dla przypadku użycia *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu* – część dialogowa z dołączoną częścią wykonawczą

Na diagramie przedstawiono część dialogową z dołączoną częścią wykonawczą, oznaczoną w postaci „szarej” aktywności. Jak widać, część wykonawcza poprzedza komunikat końcowy.

*Dla przypomnienia:* Scenariusz przedstawiony w Tab. 2-1 (pkt 2.2.3) nie uwzględnia sytuacji, w której przypadek użycia *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu* jest wywoływany z przypadku *Przeglądaj aktualne przydziały do prac*.

*Uwagi:*

- Obiekt klasy *Weryfikacja teczki* powstaje w momencie przydzielenia teczki z dokumentacją do weryfikacji danemu ekspertowi przez pracownika fundacji (przypadek użycia *Wybierz eksperta do weryfikacji teczki z dokumentacją kontaktu*).
- W trakcie tworzenia obiektu klasy *Weryfikacja teczki* zostają zainicjalizowane atrybuty: *data przydzielenia do weryfikacji* datą systemową, a *data weryfikacji, komentarz* oraz atrybut *czy pozytywna* wartością *null*. Dla trzech ostatnich atrybutów, ich wartości zostaną ustalone dopiero w trakcie wykonywania metody *rejestruj weryfikację teczki z dokumentacją kontaktu* (*komentarz, czy pozytywna*).
- Dla weryfikacji pozytywnej (zakończonej sukcesem) status teczki należy zmienić na „zweryfikowana pozytywnie”, dla weryfikacji negatywnej trzeba zmienić go na „do ponownej weryfikacji”.

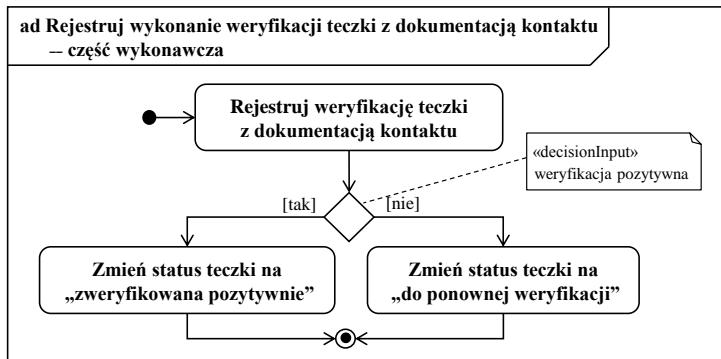
Część wykonawczą realizacji omawianego przypadku użycia można przedstawić za pomocą uproszczonego algorytmu zapisanego w postaci pseudokodu:

```

begin
    rejestruj weryfikację teczki z dokumentacją kontaktu
    (komentarz, czy pozytywna);
    if (czy pozytywna)
        then status teczki := „zweryfikowana pozytywnie”
        else status teczki := „do ponownej weryfikacji”
end

```

Na Rys. 2-46 przedstawiono diagram aktywności dla część wykonawczej przypadku użycia *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu*, odpowiedni dla algorytmu podanego powyżej.



Rys. 2-46 Diagram aktywności dla przypadku użycia *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu* – część wykonawcza

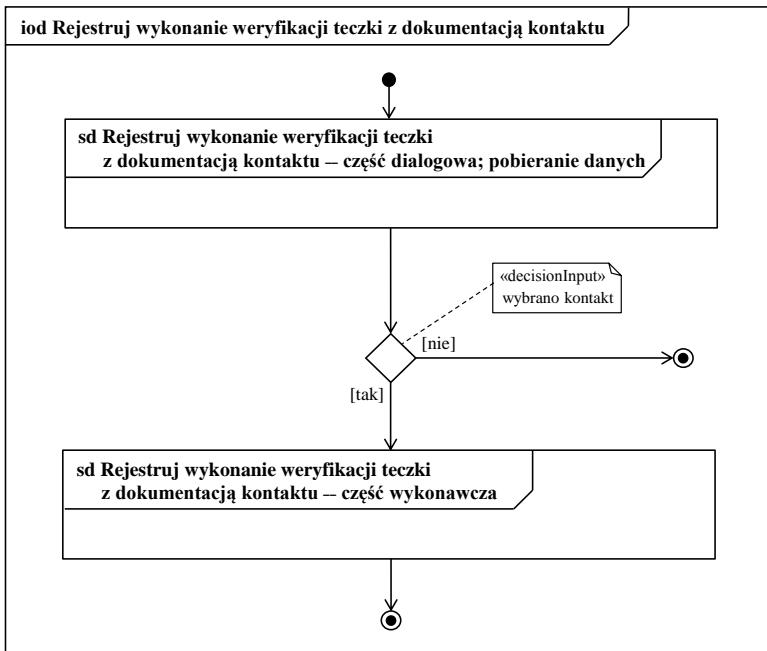
**Zad. 4:** Sporządź diagram interakcji dla wybranego przypadku użycia.

Poniżej zaprezentowano diagramy interakcji dla przypadku użycia *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu* – scenariusz dla tego przypadku znajduje się w Tab. 2-1.

Jako pierwszy został zaprezentowany *diagram przeglądu interakcji* (ang. *Interaction Overview Diagram*), określający sposób realizowania przypadku użycia traktowanego jako całość, wykorzystywany do naszkicowania (zarysowania) przepływu sterowania wewnątrz grupy logicznie powiązanych diagramów. Diagram ten przedstawiono na Rys. 2-47.

Warto zauważyć, że w trakcie realizowania części dialogowej, w celu ewaluowania warunku początkowego (*co najmniej 1 teczka z dokumentacją o statusie „opracowana” lub „do ponownej weryfikacji” została zarejestrowana*), należy wykonać operację na ekstensji klasy *Teczka z dokumentacją*, zwracającą liczbę teczek o statusie „opracowana” lub „do ponownej weryfikacji”. Trzeba również zwrócić się do obiektu klasy *Ekspert*, aby pobrać listę przypisanych do niego teczek z dokumentacją kontaktu do weryfikacji, a potem wyświetlić informacje o tych teczkach (o ile w ogóle ma jakieś przydzielone). Dzięki temu ekspert będzie mógł dokonać wyboru tej teczki, dla której zamierza zarejestrować przeprowadzenie weryfikacji jej zawartości.

W części wykonawczej należy wykonać rejestrację danych weryfikacji (*data weryfikacji, komentarz, czy pozytywna*) oraz operację na obiekcie klasy *Teczka z dokumentacją*, zmieniającą status teczki w zależności od wartości atrybutu *czy pozytywna*, umieszczonego w klasie *Weryfikacja teczki*.



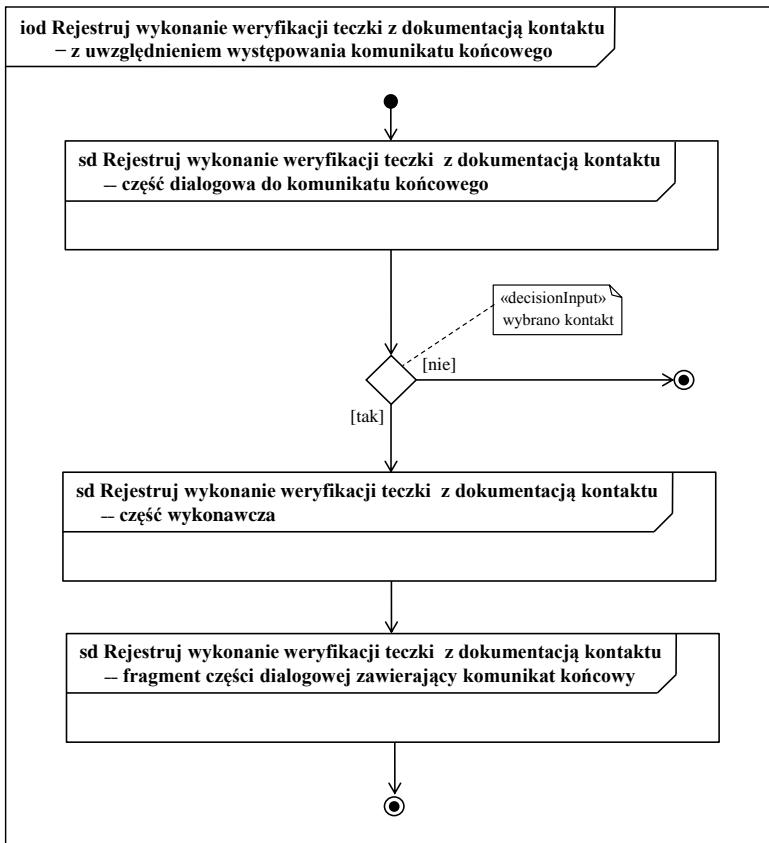
Rys. 2-47 Diagram przeglądu interakcji dla przypadku użycia *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu*

Po części wykonawczej trzeba zrealizować ostatnią aktywność z częścią dialogową, czyli wyświetlić komunikat końcowy. Diagram przeglądu interakcji dla przypadku użycia *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu* – z uwzględnieniem komunikatu końcowego, przedstawiono na Rys. 2-48.

Ze względu na różnice występujące na schematach pojęciowym (Rys. 2-17) i projektowym (Rys. 2-26), wynikające z konieczności zamiany klasy asocjacji *Weryfikacja teczki* (schemat pojęciowy) na klasę pośredniczącą (schemat projektowy), analizę interakcji przedstawiono oddzielnie dla obu rodzajów diagramów.

*Uwagi dotyczące prezentowanych diagramów sekwencji – dla schematów pojęciowego i projektowego:*

- Diagramy przedstawiono na dwa sposoby: pierwszy, bardziej ogólny – bez oznaczania czasu wykonywania komunikatu i drugi – z jego oznaczeniem;
- W celu zachowania przejrzystości i zapewnienia lepszej percepkcji diagramów, wykorzystano przywołanie wystąpień za pośrednictwem operatora *ref*, a następnie przedstawiono diagramy sekwencji, których dotyczyły odpowiednie przywołania.



Rys. 2-48 Diagram przeglądu interakcji dla przypadku użycia *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu* – uwzględnieniem komunikatu końcowego

Należy pamiętać, że „pośrednikiem” w interakcji aktora z instancjami klas przechowującymi informacje niezbędne do wykonania przypadku użycia jest interfejs użytkownika (GUI – ang. *Graphical User Interface*). Do zadań GUI nalezy:

- transformacja informacji opisujących aktora *Ekspert* w języku dziedziny problemowej (np. imię, nazwisko) na dane pozwalające na identyfikację w systemie skojarzonego z nim obiektu (np. w środowisku obiektowym na referencję do odpowiedniego obiektu klasy *Ekspert*);
- pośredniczenie w pozyskiwaniu od aktora informacji związanych z: wyborem odpowiedniej teczki, z pobraniem treści komentarza oraz wyniku weryfikacji.

Klasę *Ekspert* zidentyfikowano jako klasę graniczną, co zostało odpowiednio oznaczone na poniższych diagramach.

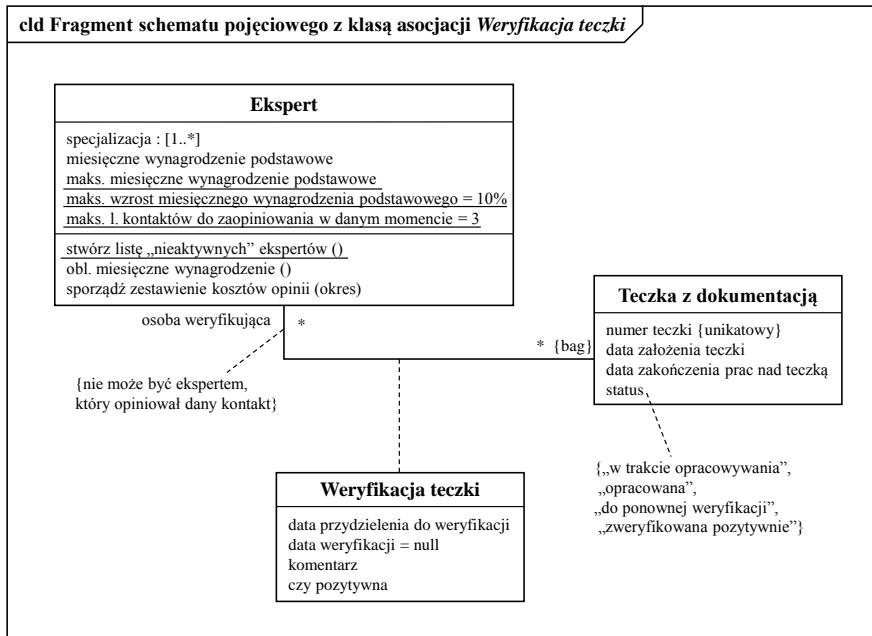
Dla przypomnienia: Klasę graniczną są wykorzystywane do zapewnienia komunikacji otoczenia systemu z jego wnętrzem i dlatego każdy przypadek użycia,

każda para aktor-scenariusz powinna być rozważana w aspekcie identyfikacji klas granicznych.

### Schemat pojęciowy

Poniżej przedstawiono analizę interakcji dla fragmentu schematu pojęciowego, który został zaprezentowany na Rys. 2-49.

Na diagramie pominięto komentarze oraz ograniczenia nieistotne dla prowadzonych rozważań.



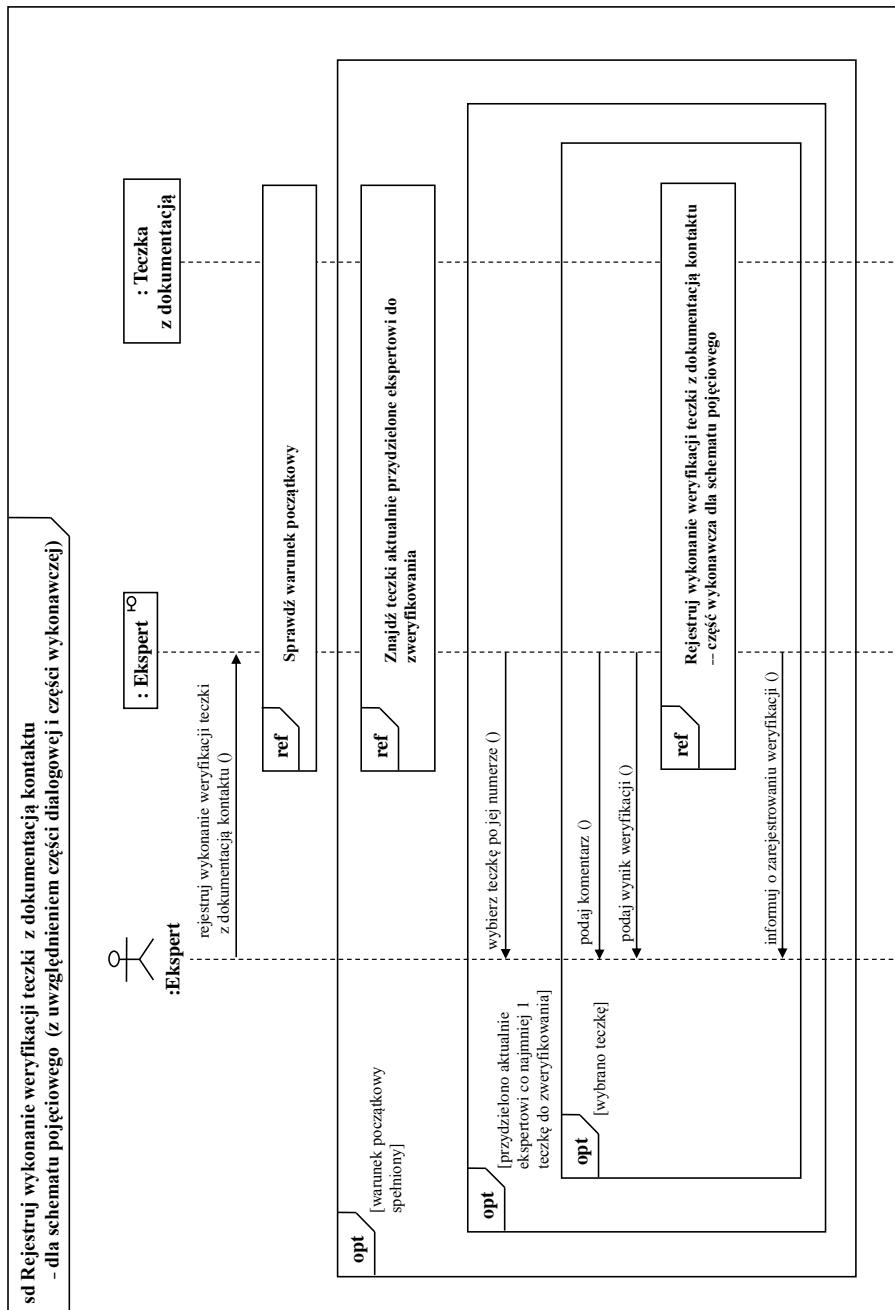
Rys. 2-49 Fragment schematu pojęciowego z klasą asocjacji *Weryfikacja teczki*

### Diagramy sekwencji bez oznaczania czasu wykonywania komunikatu

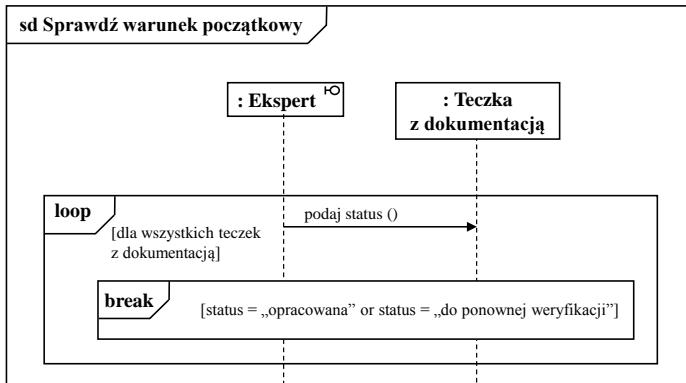
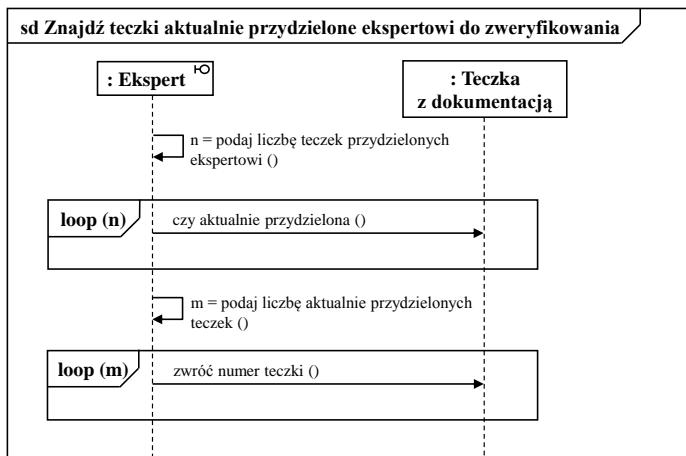
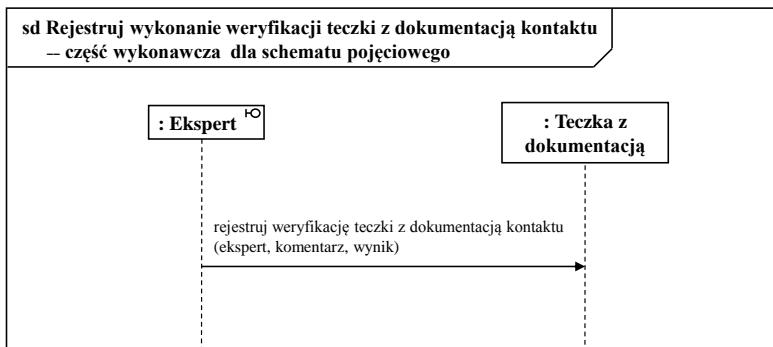
Pierwszy z diagramów sekwencji (Rys. 2-50) opisuje część dialogową i wykonawczą przypadku użycia *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu*. Diagram zawiera trzy przywołania wystąpień za pośrednictwem operatora *ref*: *Sprawdź warunek początkowy*, *Znajdź teczki aktualnie przydzielone ekspertowi do zweryfikowania* oraz *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu – część wykonawcza dla schematu pojęciowego*. Kolejne diagramy, których dotyczą przywołania, przedstawiono na Rys. 2-51, Rys. 2-52, Rys. 2-53.

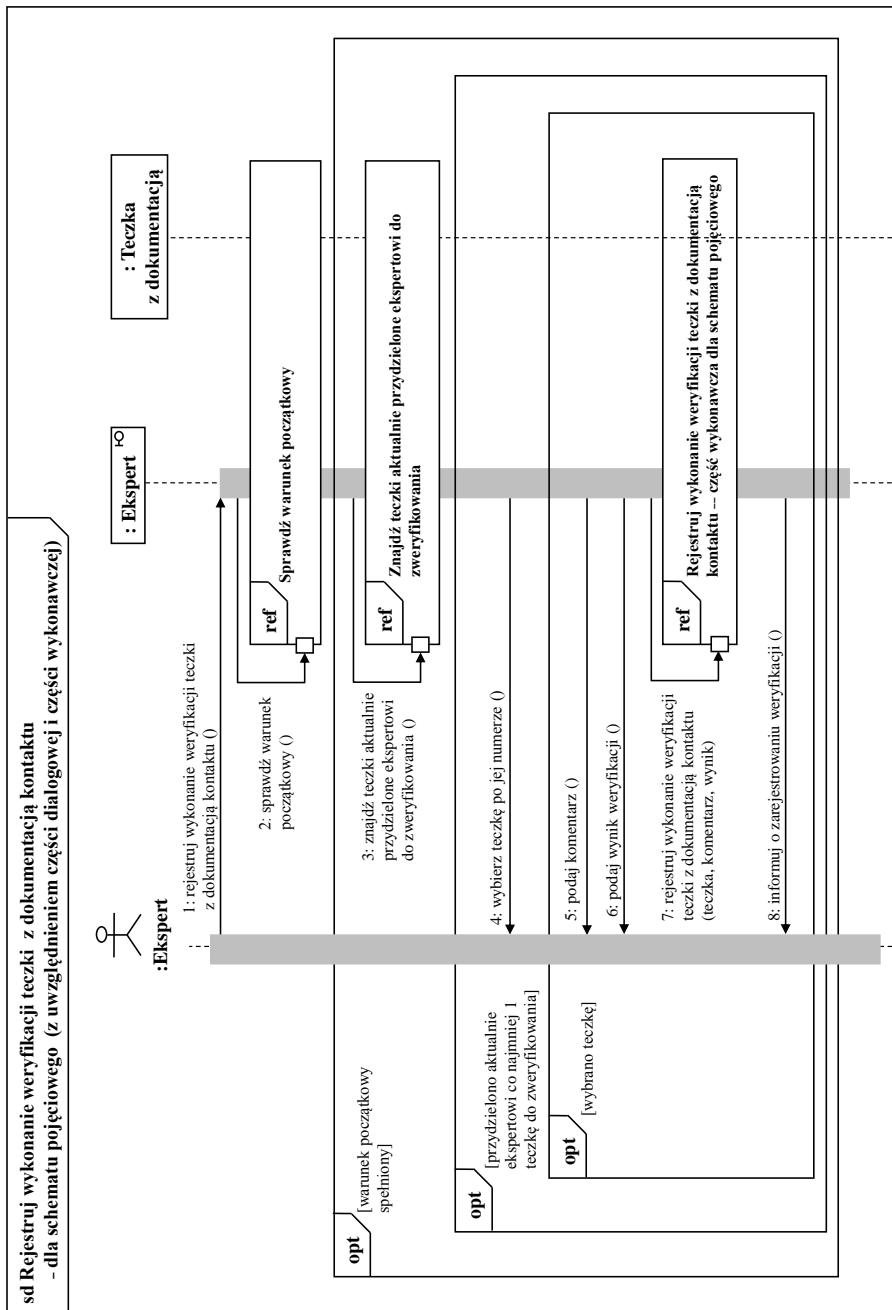
### Diagramy sekwencji z oznaczonym czasem wykonywania komunikatu

Część dialogową i wykonawczą opisuje diagram sekwencji przedstawiony na Rys. 2-54. Kolejne diagramy, których dotyczą przywołania na nim umieszczone, przedstawiono na Rys. 2-55, Rys. 2-56 i Rys. 2-57.

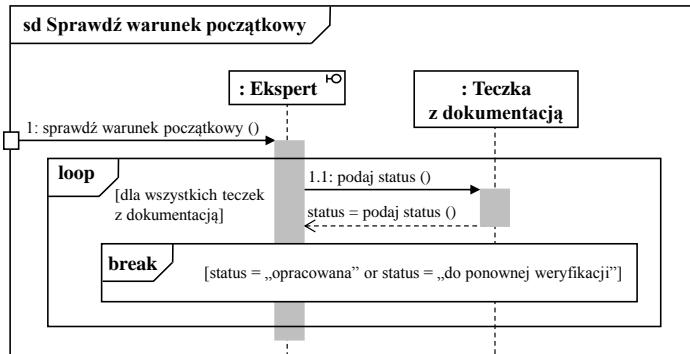


Rys. 2-50 Diagram sekwencji dla schematu pojęciowego – bez oznaczenia czasu wykonywania komunikatu

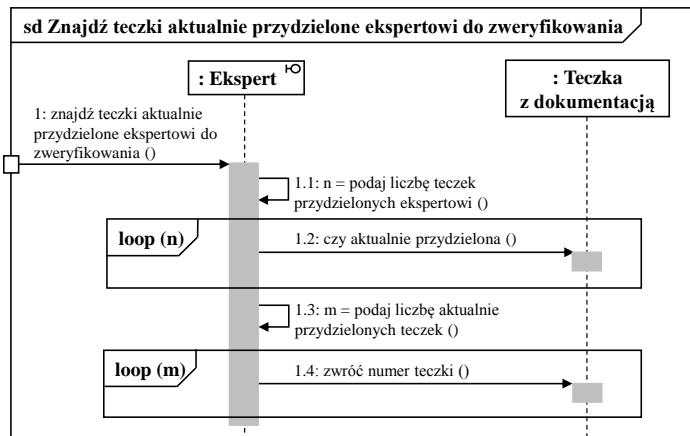
Rys. 2-51 Diagram sekwencji *Sprawdź warunek początkowy* – dla schematu pojęciowegoRys. 2-52 Diagram sekwencji *Znajdź teczki aktualnie przydzielone ekspertowi do zweryfikowania* – dla schematu pojęciowegoRys. 2-53 Diagram sekwencji *Rejestruj weryfikację teczki z dokumentacją kontaktu – część wykonawcza* dla schematu pojęciowego



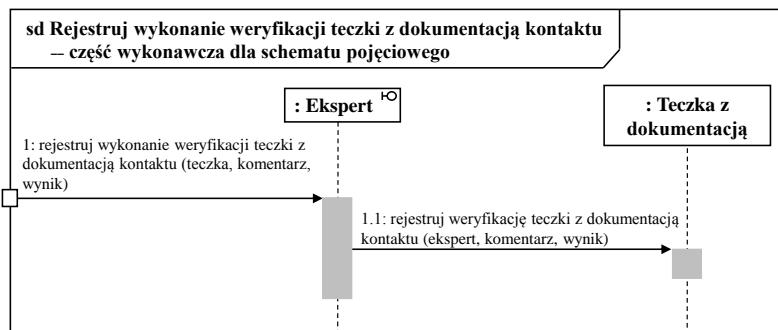
Rys. 2-54 Diagram sekwencji dla schematu pojęciowego z oznaczonym czasem wykonywania komunikatu



Rys. 2-55 Diagram sekwencji *Sprawdź warunek początkowy* – dla schematu pojęciowego, z oznaczonym czasem wykonywania komunikatu



Rys. 2-56 Diagram sekwencji *Znajdź teczki aktualnie przydzielone ekspertowi do zweryfikowania* – dla schematu pojęciowego, z oznaczonym czasem wykonywania komunikatu



Rys. 2-57 Diagram sekwencji *Rejestruj weryfikację teczki z dokumentacją kontaktu* – część wykonawcza dla schematu pojęciowego, z oznaczonym czasem wykonywania komunikatu

## Schemat projektowy

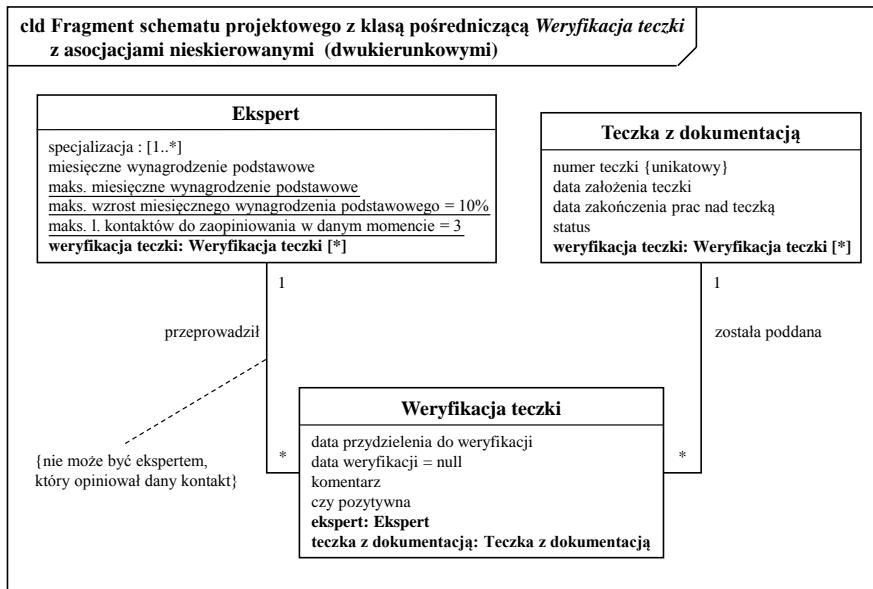
Na kolejnych rysunkach zaprezentowano diagramy interakcji dla części dialogowej oraz części wykonawczej na schemacie projektowym w dwóch wariantach:

- z wykorzystaniem asocjacji dwukierunkowych;
- z wykorzystaniem asocjacji jednokierunkowych (skierowanych).

### Schemat projektowy z asocjacjami dwukierunkowymi

Poniższe rozważania dotyczą fragmentu schematu projektowego bez asocjacji skierowanych, który został zaprezentowany na Rys. 2-58.

Na diagramie pominieto komentarze, ograniczenia i referencje, nieistotne dla prowadzonych rozważań.



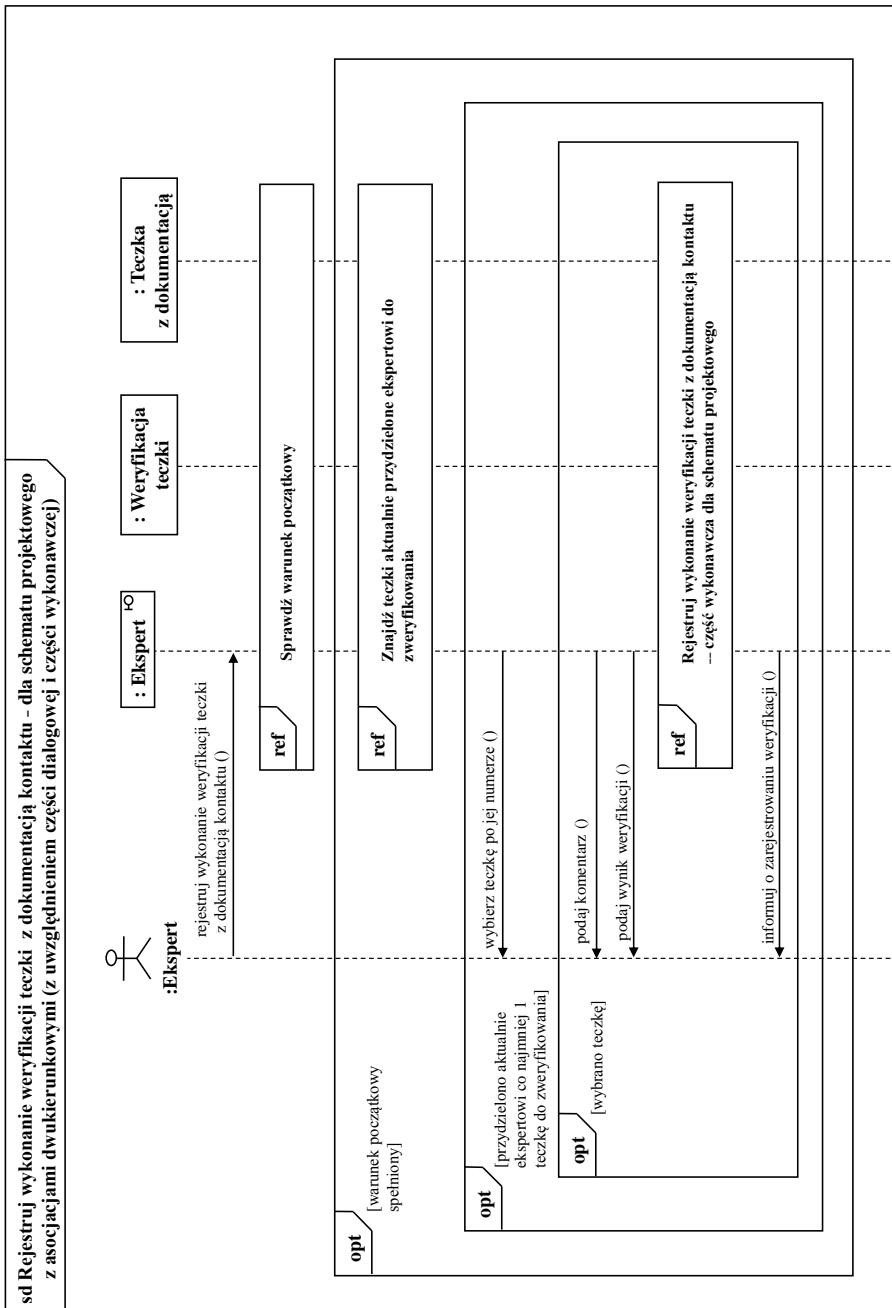
Rys. 2-58 Fragment schematu projektowego z klasą pośredniczącą *Weryfikacja teczki* – z asocjacjami dwukierunkowymi

### Diagramy sekwencji bez oznaczania czasu wykonywania komunikatu

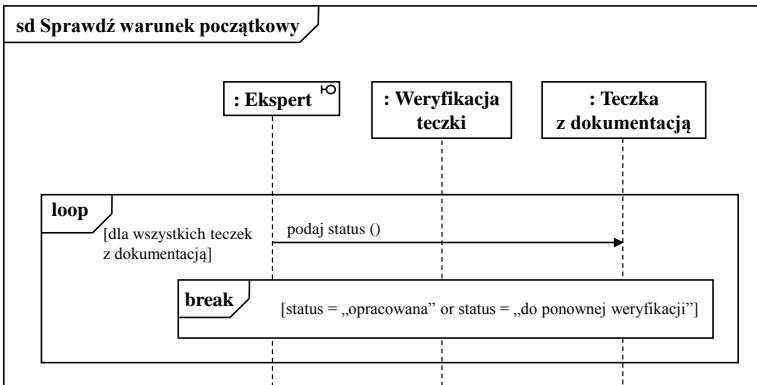
W przypadku diagramu sekwencji dla powyższego schematu, całość (część dialogową i wykonawczą) zaprezentowano na jednym diagramie (Rys. 2-59). Zawiera on przywołania wystąpień za pośrednictwem operatora *ref*. Kolejne diagramy, których dotyczą przywołania, przedstawiono na Rys. 2-60, Rys. 2-61 i Rys. 2-62.

### Diagramy sekwencji z oznaczonym czasem wykonywania komunikatu

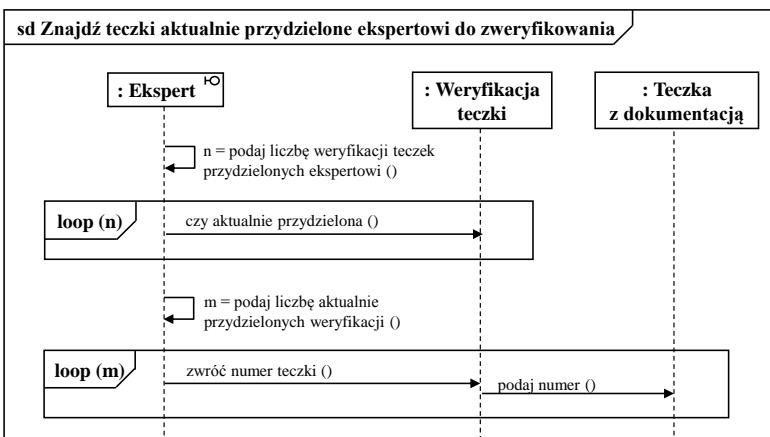
Część dialogową i wykonawczą opisuje diagram sekwencji przedstawiony na Rys. 2-63. Kolejne diagramy, których dotyczą przywołania na nim umieszczone, przedstawiono na Rys. 2-64, Rys. 2-65 i Rys. 2-66.



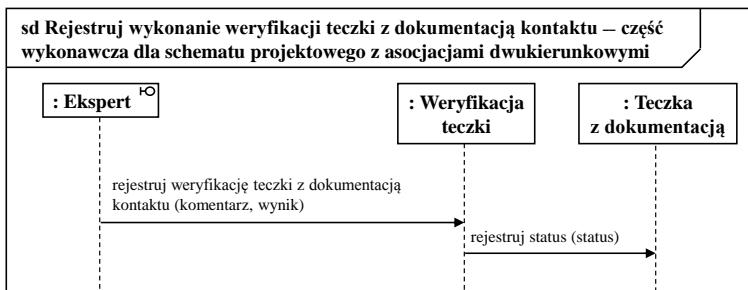
Rys. 2-59 Diagram sekwencji – dla schematu projektowego z asocjacjami dwukierunkowymi, bez oznaczenia czasu realizacji komunikatu



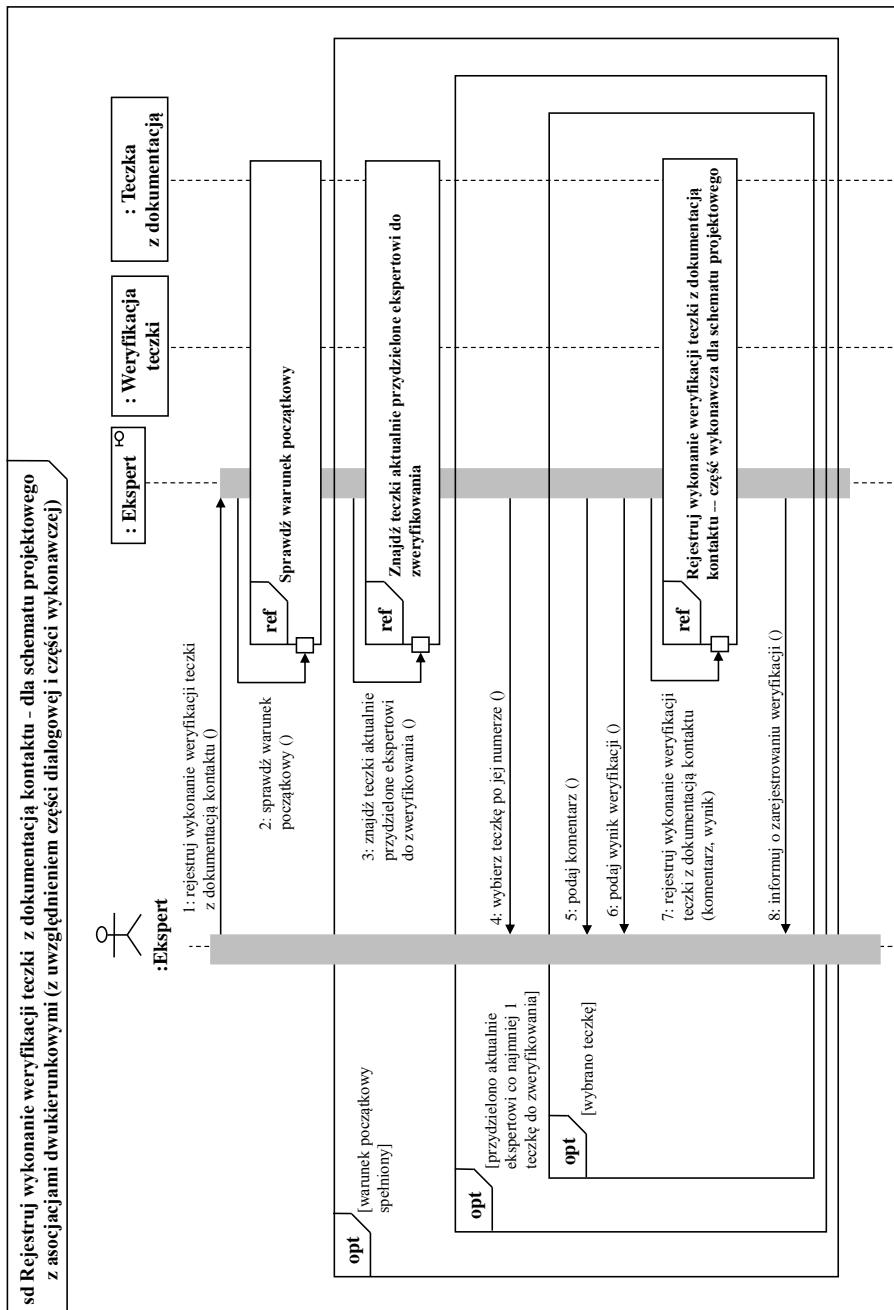
Rys. 2-60 Diagram sekwencji *Sprawdź warunek początkowy* dla schematu projektowego z asocjacjami dwukierunkowymi, bez oznaczenia czasu realizacji komunikatu



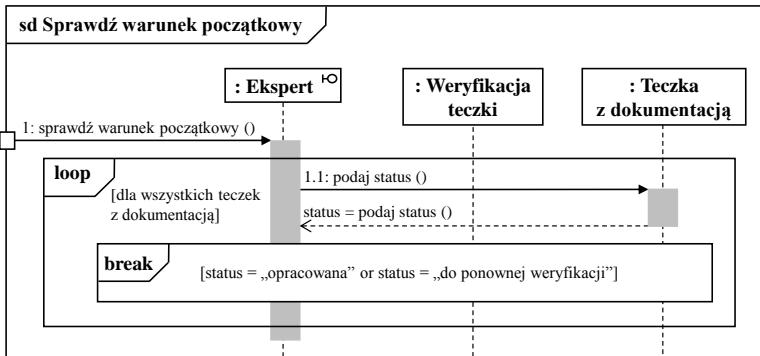
Rys. 2-61 Diagram sekwencji *Znajdź teczki aktualnie przydzielone ekspertowi do zweryfikowania* dla schematu projektowego z asocjacjami dwukierunkowymi, bez oznaczenia czasu realizacji komunikatu



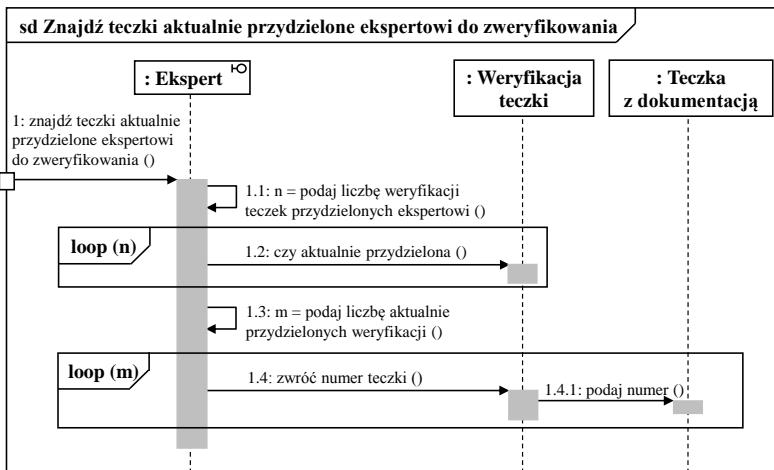
Rys. 2-62 Diagram sekwencji *Rejestruj weryfikację teczki z dokumentacją kontaktu – część wykonawcza* dla schematu projektowego z asocjacjami dwukierunkowymi, bez oznaczenia czasu realizacji komunikatu



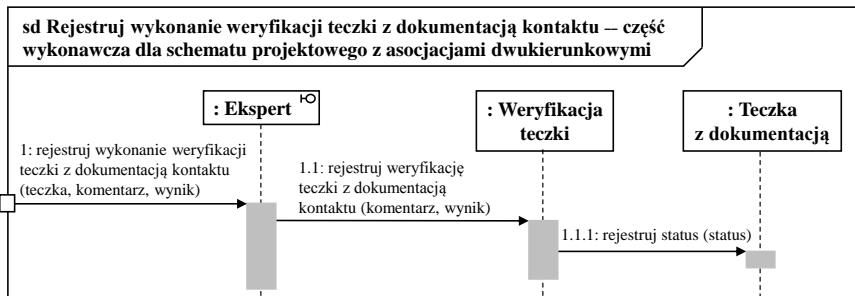
Rys. 2-63 Diagram sekwencji – dla schematu projektowego z asocjacjami dwukierunkowymi, z oznaczeniem czasu wykonywania komunikatu



Rys. 2-64 Diagram sekwencji *Sprawdź warunek początkowy* dla schematu projektowego z asocjacjami dwukierunkowymi, z oznaczeniem czasu wykonywania komunikatu

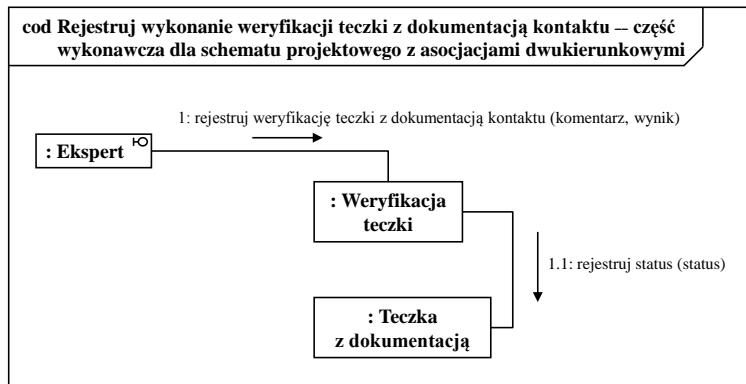


Rys. 2-65 Diagram sekwencji *Znajdź teczki aktualnie przydzielone ekspertowi do zweryfikowania* dla schematu projektowego z asocjacjami dwukierunkowymi, z oznaczeniem czasu wykonywania komunikatu



Rys. 2-66 Diagram sekwencji *Rejestruj weryfikację teczki z dokumentacją kontaktu – część wykonawcza* dla schematu projektowego z asocjacjami dwukierunkowymi, z oznaczeniem czasu wykonywania komunikatu

Na Rys. 2-67 przedstawiono diagram współpracy/ komunikacji (*cod – ang. Collaboration/ Communication Diagram*), izomorficzny z diagramami sekwencji na Rys. 2-62 i Rys. 2-66.



Rys. 2-67 Diagram komunikacji zgodny z diagramami sekwencji z Rys. 2-62 i Rys. 2-66

### Schemat projektowy z asocjacjami jednokierunkowymi (skierowane)

W drugim rozwiążaniu przyjęto, że zamiast asocjacji dwukierunkowych powstały w efekcie zamiany klasy asocjacji *Weryfikacja teczki* na klasę pośredniczącą, będą implementowane asocjacje skierowane (jednokierunkowe), w sposób zaprezentowany na Rys. 2-68.

Dla przejrzystości rysunku pominięto metody umieszczone w klasie *Ekspert* (nie mają one wpływu na przeprowadzane rozważania).



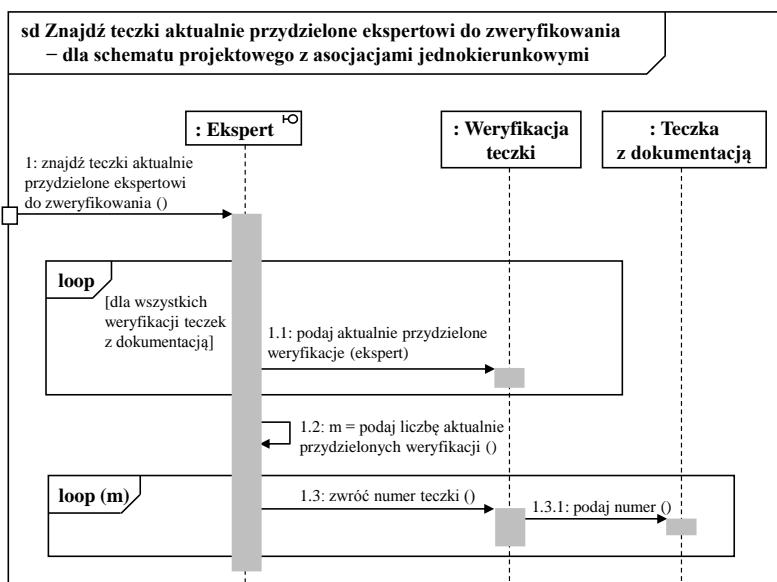
Rys. 2-68 Fragment schematu projektowego z klasą pośredniczącą *Weryfikacja teczki* (rozwiązanie z asocjacjami skierowanymi)

Podczas sprawdzania warunku początkowego metoda *sprawdź warunek początkowy()* nie odwołuje się do obiektów klasy *Weryfikacja teczki*, dlatego odpowiedni diagram sekwencji (Rys. 2-64) nie uległ zmianie.

Przeszukiwanie teczek z dokumentacją oraz weryfikacji jest realizowana przez metodę *znajdź teczki aktualnie przydzielone ekspertowi do zweryfikowania()*, dlatego ma znaczenie, czy w odpowiednim fragmencie struktury wykorzystano asocjacje jednokierunkowe czy dwukierunkowe.

Część wykonawcza obejmuje wyłącznie rejestrację weryfikacji teczki oraz zmianę statusu teczki. Wszystkie potrzebne obiekty zostały wcześniej odnalezione (metoda *Znajdź teczki aktualnie przydzielone ekspertowi do zweryfikowania*), dlatego diagram sekwencji *Rejestruj weryfikację teczki z dokumentacją kontaktu* nie uległ zmianie.

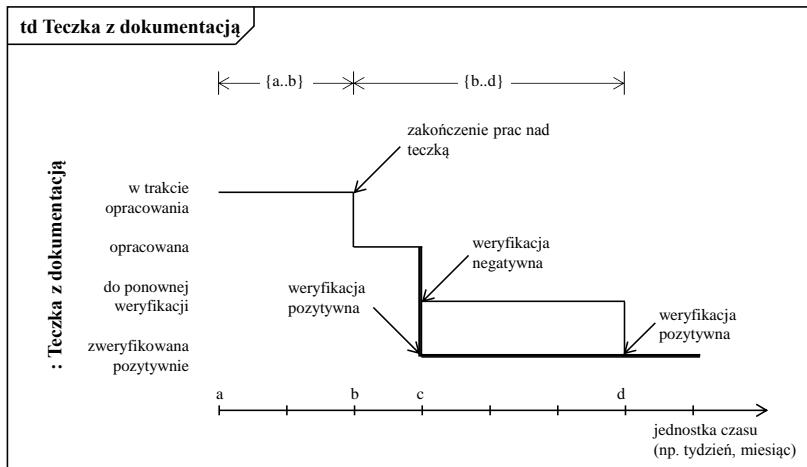
W związku z powyższym, dla asocjacji jednokierunkowych zmianie uległ jedynie diagram sekwencji *Znajdź teczki aktualnie przydzielone ekspertowi do zweryfikowania* (Rys. 2-65). Zmodyfikowany diagram przedstawiono na Rys. 2-69.



Rys. 2-69 Diagram sekwencji *Znajdź teczki aktualnie przydzielone ekspertowi do zweryfikowania* (dla schematu projektowego z asocjacjami skierowanymi), z oznaczeniem czasu wykonywania komunikatu

**Zad. 5:** Przedstaw diagram następstwa stanów dla obiektów klasy *Teczka z dokumentacją*.

*Diagram następstwa stanów* (ang. *timing diagram*) dla obiektów klasy *Teczka z dokumentacją* przedstawiono na Rys. 2-70. Diagram następstwa służy do ilustrowania zmian w czasie dla wartości atrybutów obiektów, a także dla stanów obiektów (najbardziej typowe zastosowanie).



Rys. 2-70 Diagram następstwa stanów dla obiektów klasy *Teczka z dokumentacją*

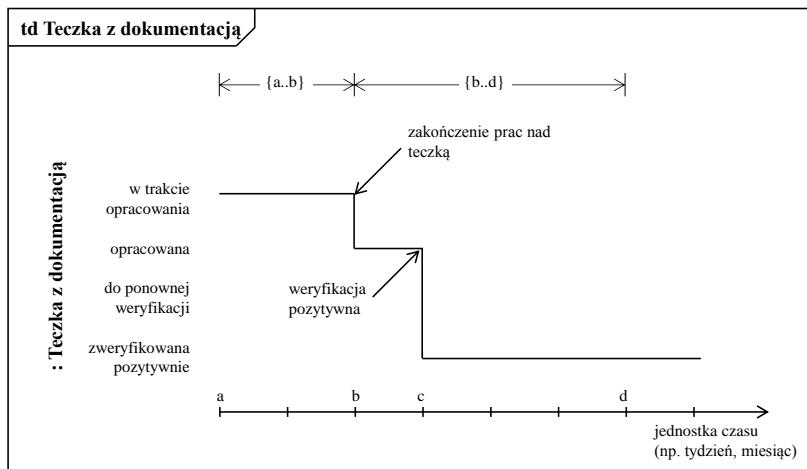
W celu zilustrowania użyteczności diagramów następstwa stanów, na Rys. 2-70 sformułowano następujące przykładowe ograniczenia:

- Prace nad teczką nie powinny trwać dłużej niż  $b-a$  jednostek czasu (po tym czasie teczka zmienia stan z „w trakcie opracowania” na „opracowana”). Można założyć, że po upływie  $b-a$  jednostek czasu będzie się to odbywało automatycznie i po tym okresie nie wolno już będzie dokładać nowych dokumentów do teczki. *Dla przypomnienia:* W systemie nie przechowuje się informacji o dokumentach przechowywanych w teczce;
- od momentu zakończenia prac nad teczką (teczka „opracowana”) do uzyskania pozytywnej weryfikacji jej zawartości (teczka „zweryfikowana pozytywnie”) nie może minąć więcej niż  $d-b$  jednostek czasu. Można założyć, że po tym czasie weryfikacja teczki nie może mieć miejsca.

Weryfikacja teczki powinna odbyć się w dowolnym momencie/momentach, między „ $b$ ” a „ $d$ ”, oznaczonym przykładowo na osi czasu jako „ $c$ ”, powinna odbyć się weryfikacja teczki. Jeśli weryfikacja będzie pozytywna, w tym momencie stan teczki zmieni się na „zweryfikowana pozytywnie” (teczka pozostaje w tym stanie aż do końca życia obiektu), co zostało oznaczone pogrubioną linią), a w przypadku weryfikacji negatywnej – na stan „do ponownej weryfikacji”, przy czym w tym stanie teczka może się znajdować nie dłużej niż do momentu „ $d$ ”.

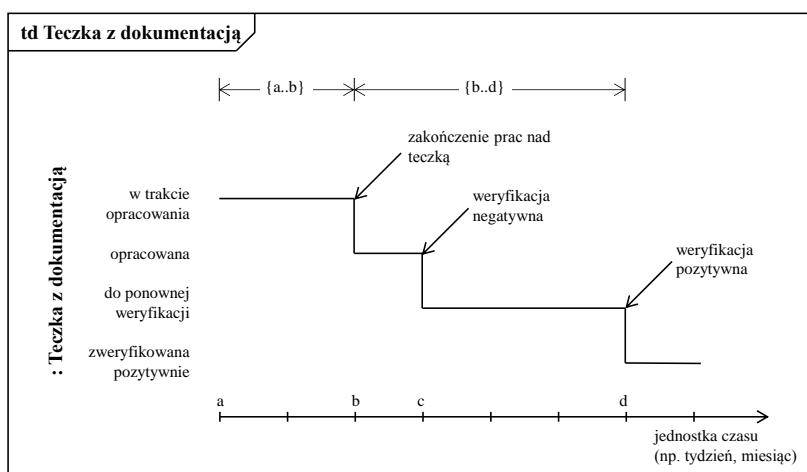
Dla przejrzystości i lepszej percepacji diagramu z Rys. 2-70 został on podzielony na dwa diagramy (Rys. 2-71, Rys. 2-72), odrębnie dla pozytywnej i negatywnej weryfikacji dokonanej w dowolnym momencie między „ $b$ ” a „ $d$ ”, oznaczonym przykładowo na osi czasu jako „ $c$ ”.

Diagram następstwa stanów dla sytuacji, w której pierwsza weryfikacja, dokonana między momentami „ $b$ ” a „ $d$ ”, jest pozytywna, przedstawiono na Rys. 2-71.



Rys. 2-71 Diagram następstwa stanów dla obiektów klasy *Teczka z dokumentacją* – gdy piersza weryfikacja dokonana między momentami „b” a „d” jest pozytywna

W przypadku, gdy pierwsza weryfikacja jest negatywna, obiekt klasy *Teczka z dokumentacją* przechodzi ze stanu „opracowana” do stanu „do ponownej weryfikacji” i pozostaje w nim tak długo, aż nie nastąpi pozytywna weryfikacja (każda kolejna weryfikacja negatywna nie powoduje zmiany tego stanu). Weryfikacja pozytywna musi nastąpić nie później niż w momencie „d”, co ilustruje Rys. 2-72.



Rys. 2-72 Diagram następstwa stanów dla obiektów klasy *Teczka z dokumentacją* – gdy pierwsza weryfikacja dokonana między momentami *b* a *d* jest negatywna

## *Komentarz do ograniczeń związanych z upływem czasu:*

W związku z ograniczeniem *weryfikacja pozytywna musi nastąpić nie później niż w momencie „d”* należy rozważyć, co będzie się działo z teczka, gdy nie zostanie

ona pozytywnie zweryfikowana w dopuszczalnym terminie. W takiej sytuacji można byłoby zaproponować wprowadzenie dla niej dodatkowego stanu, np. „odłożona w oczekiwaniu na wznowienie procesu weryfikacji”.

Przedstawiony diagram (Rys. 2-70) mógłby ulec modyfikacjom po uzyskaniu od przyszłego użytkownika odpowiedzi na następujące pytania:

- Czy zakończenie prac nad teczką oznacza automatycznie zablokowanie możliwości dołączania do teczki nowych dokumentów? Co stanie się w sytuacji, gdy nie nastąpi zakończenie prac nad teczką w dopuszczalnym terminie?
- Co należy zrobić w przypadku, gdy upłynie czas przeznaczony na weryfikację, a żadna weryfikacja nie zostanie przeprowadzona bądź będzie to ostatecznie weryfikacja negatywna?

#### **2.4.2 Omówienie zmian na schemacie pojęciowym i schemacie projektowym wynikających z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań**

Przeprowadzona analiza dynamiczna nie obejmowała całości schematu pojęciowego/projektowego, a jedynie ten jego fragment, który związany jest z analizowaniem maszyny stanowej dla klasy *Teczka z dokumentacją* oraz z realizowaniem przypadku użycia *Rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu*. W wyniku analizy uznano, że do schematów pojęciowego oraz projektowego należy wprowadzić wymienione poniżej operacje (Tab. 2-3).

Tab. 2-3 Zmiany na schematach pojęciowym i projektowym wynikające z analizy dynamicznej

Nazwa klasy	Metoda
<b>Analiza przeprowadzona na schemacie pojęciowym</b>	
<i>Teczka z dokumentacją</i>	<ul style="list-style-type: none"> <li>- <i>rejestruj teczkę (kontakt)</i></li> <li>- <i>rejestruj zakończenie prac nad teczką ()</i></li> <li>- <i>rejestruj weryfikację (komentarz, status)</i></li> <li>- <i>podaj status ()</i></li> <li>- <i>czy aktualnie przydzielona ()</i></li> <li>- <i>zwróć numer teczki ()</i></li> <li>- <i>rejestruj weryfikację teczki z dokumentacją kontaktu (ekspert, komentarz, wynik)</i></li> </ul>
<i>Ekspert</i>	<ul style="list-style-type: none"> <li>- <i>rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu ()</i></li> <li>- <i>sprawdź warunek początkowy ()</i></li> </ul>

	<ul style="list-style-type: none"> <li>- znajdź teczki aktualnie przydzielone ekspertowi do zweryfikowania ()</li> <li>- podaj liczbę teczek przydzielonych ekspertowi ()</li> <li>- podaj liczbę aktualnie przydzielonych teczek ()</li> <li>- rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu (teczka, komentarz, wynik)</li> </ul>
<b>Analiza przeprowadzona na schemacie projektowym – z asocjacjami nieskierowanymi</b>	
Teczka z dokumentacją	<ul style="list-style-type: none"> <li>- podaj status ()</li> <li>- podaj numer ()</li> <li>- rejestruj status (status)</li> </ul>
Weryfikacja teczki	<ul style="list-style-type: none"> <li>- czy aktualnie przydzielona ()</li> <li>- zwróć numer teczki ()</li> <li>- rejestruj weryfikację teczki z dokumentacją kontaktu (komentarz, wynik)</li> </ul>
Ekspert	<ul style="list-style-type: none"> <li>- rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu ()</li> <li>- sprawdź warunek początkowy ()</li> <li>- znajdź teczki aktualnie przydzielone ekspertowi do zweryfikowania ()</li> <li>- podaj liczbę weryfikacji teczek przydzielonych ekspertowi ()</li> <li>- podaj liczbę aktualnie przydzielonych weryfikacji ()</li> <li>- rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu (teczka, komentarz, wynik)</li> </ul>
<b>Analiza przeprowadzona na schemacie projektowym – z asocjacjami skierowanymi</b>	
Teczka z dokumentacją	<ul style="list-style-type: none"> <li>- podaj status ()</li> <li>- podaj numer ()</li> <li>- rejestruj status (status)</li> </ul>
Weryfikacja teczki	<ul style="list-style-type: none"> <li>- <u>podaj aktualnie przydzielone weryfikacje (ekspert)</u></li> <li>- zwróć numer teczki ()</li> <li>- rejestruj weryfikację teczki z dokumentacją kontaktu (komentarz, wynik)</li> </ul>
Ekspert	<ul style="list-style-type: none"> <li>- rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu ()</li> </ul>

	<ul style="list-style-type: none"> <li>- sprawdź warunek początkowy ()</li> <li>- znajdź teczki aktualnie przydzielone ekspertowi do zweryfikowania ()</li> <li>- podaj liczbę aktualnie przydzielonych weryfikacji ()</li> <li>- rejestruj wykonanie weryfikacji teczki z dokumentacją kontaktu (teczka, komentarz, wynik)</li> </ul>
--	--

## 2.5 Zadania do samodzielnej pracy

Poniżej przedstawiono listę pytań i zadań do samodzielnego rozwiązywania w oparciu o schemat pojęciowy z Rys. 2-17.

1. Dla funkcjonalności *Przeglądaj aktualne przydziały do prac* napisz scenariusz oraz zaproponuj podział na podprzypadki.
2. Podaj przykład *operacji, metody i komunikatu*.
3. Podaj definicję klasy abstrakcyjnej. Czy klasa abstrakcyjna może występować w dowolnym miejscu hierarchii klas? Odpowiedź uzasadnij odpowiednimi przykładami.
4. Objasnij różnicę między dziedziczeniem nierożłącznym a dynamicznym. W jakich sytuacjach powinno się każde z nich stosować? Czy któryś z tych dwóch rodzajów dziedziczenia wystąpił na schemacie pojęciowym? Jeśli tak, uzasadnij jego wykorzystanie.
5. Co oznacza znak elipsy (inaczej znak opuszczenia, ang. *ellipsis*) umieszczony na diagramie? Objasnij, czym różni się wykorzystanie tego symbolu od ograniczenia *{incomplete}*. Podaj odpowiedni przykład.
6. Wyjaśni różnicę między przeciążaniem a przesłanianiem w oparciu o przykłady. Czy przeciążanie jest rodzajem polimorfizmu metod?
7. Podaj przykład ograniczenia, które odnosi się do dwóch asocjacji. Do jakiej metody i w jakiej klasie włożyłbyś implementację tego ograniczenia?
8. Wyjaśni pojęcie stereotypu. Podaj odpowiedni przykład.
9. Objasnij różnicę pomiędzy komentarzem a ograniczeniem. Podaj odpowiednie przykłady.
10. Wskaż asocjację wiele-do-wielu (z niesymetrycznymi licznosciami dla obu końców asocjacji ze zdefiniowaną klasą lub atrybutem asocjacji). Na jakim etapie cyklu życia systemu klas asocjacji powinno się zamienić na konstrukcję zawierającą klasę pośredniczącą? Narysuj odpowiedni przykład (z atrybutami i metodami) na podstawie wskazanego uprzednio fragmentu schematu pojęciowego.
11. Zaproponuj model relacyjny dla fragmentu schematu pojęciowego, zawierającego dwie klasy połączone asocjacją z klasą asocjacji.

12. Dla klasy *Opinia* (ze schematu pojęciowego) wskaż kilka stanów (co najmniej dwa stany, nie licząc stanów początkowego i końcowego), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów, zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.
13. Dla funkcjonalności *Przeglądaj aktualne przydziały do prac* skonstruuj diagram aktywności, zgodnie z napisanym wcześniej scenariuszem. Wyodrębnij część dialogową i część wykonawczą.
14. Sporządź diagram interakcji dla przypadku użycia *Wystaw opinię o kontakcie*. Narysuj diagram sekwencji oraz odpowiadający mu diagram komunikacji.

## 2.6 Podsumowanie zadania

W Tab. 2-4 przedstawiono ogólną charakterystykę omawianego zadania. Szczegółową charakterystykę zadania dla grup tematycznych: *Wprowadzenie do przedmiotu*, *Analiza funkcjonalna*, *Analiza strukturalna*, *Analiza dynamiczna* zamieszczono odpowiednio w tabelach Tab. 2-5 – Tab. 2-8.

Tab. 2-4 Ogólna charakterystyka zadania

<b>Ogólna charakterystyka zadania</b>	
<i>Nazwa</i>	Fundacja „UFO”
<i>Dziedzina problemowa</i>	system można wykorzystać w każdej dziedzinie, w której kataloguje się różnego rodzaju zjawiska i wydarzenia
<i>Cel</i>	ułatwienie weryfikowania i katalogowania kontaktów ludzi z niezidentyfikowanymi obiektami latającymi
<i>Zakres odpowiedzialności</i>	zarządzanie zgłoszonymi kontaktami, uczestnikami kontaktów, ekspertami weryfikującymi wiarygodność kontaktów, rejestrowanie opinii ekspertów o wiarygodności kontaktów oraz wyliczanie wynagrodzenia ekspertów współpracujących z fundacją
<i>Stopień trudności</i>	średnio trudne
<i>Rozmiar</i>	10 klas
<i>Użyteczność w grupach tematycznych</i>	
<i>Grupa tematyczna</i>	Użyteczność w grupie tematycznej Użyteczność na etapie zgodnym z modelem efektywnego nauczania

<i>Wprowadzenie do przedmiotu</i>	niska	<i>Przygotowanie do nauki: niska Nauka: niska Podsumowanie dotychczasowych działań: niska Ewaluacja: niska</i>
<i>Analiza funkcjonalna</i>	średnia	<i>Przygotowanie do nauki: średnia Nauka: wysoka Podsumowanie dotychczasowych działań: średnia Ewaluacja: średnia</i>
<i>Analiza strukturalna</i>	średnia	<i>Przygotowanie do nauki: niska Nauka: średnia Podsumowanie dotychczasowych działań: średnia Ewaluacja: średnia</i>
<i>Analiza dynamiczna</i>	średnia	<i>Przygotowanie do nauki: niska Nauka: wysoka Podsumowanie dotychczasowych działań: średnia Ewaluacja: średnia</i>

Tab. 2-5 Szczegółowa charakterystyka zadania dla grupy tematycznej *Wprowadzenie do przedmiotu*

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>	<i>Wprowadzenie do przedmiotu</i>	
<b>Stopień trudności grupy w zadaniu</b>	łatwe	
<b>Poruszane tematy</b>	<i>analiza a projektowanie, mechanizmy rozszerzalności, dekompozycja diagramów, reguły nazewnictwa</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>analiza a projektowanie</i>	łatwe	<i>model pojęciowy, model projektowy</i>
<i>mechanizmy rozszerzalności</i>	łatwe	<i>stereotyp, ograniczenie, komentarz</i>

<i>dekompozycja diagramów</i>	łatwe	podział modelu przypadków użycia na diagramy, podział przypadku użycia na podprzypadki
<i>reguły nazewnictwa</i>	łatwe	omówienie reguł nazewnictwa dla aktorów, przypadków użycia, klas, atrybutów, metod i asocjacji

Tab. 2-6 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza funkcjonalna*

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>	<i>Analiza funkcjonalna</i>	
<b>Stopień trudności grupy w zadaniu</b>	średnio trudne	
<b>Poruszane tematy</b>	<i>wymagania na system, aktorzy, przypadki użycia, scenariusze, dekompozycja modelu/diagramu przypadków użycia, mechanizmy rozszerzalności</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>wymagania na system</i>	łatwe	<i>analiza wymagań na system – wymagania użytkownika przedstawiono w postaci pierwotnej oraz z wprowadzonymi licznymi komentarzami wyjaśniającymi proces konstruowania schematu pojęciowego</i>
<i>aktorzy</i>	łatwe	<i>aktor abstrakcyjny, aktor Podsystem czasu, hierarchia dla aktorów</i>
<i>przypadki użycia</i>	łatwe	<i>relacje między przypadkami użycia</i>
<i>scenariusze</i>	łatwe	<i>zależność głównego przepływu zdarzeń i alternatywnych przepływów zdarzeń od warunków początkowych</i>

<i>dekompozycja modelu/diagramu przypadków użycia</i>	średnio trudne	<i>poziomy abstrakcji w modelu przypadków użycia, dekompozycja pozioma, dekompozycja pionowa – trzy notacje dla aktora będącego przypadkiem użycia</i>
<i>mechanizmy rozszerzalności</i>	łatwe	ograniczenia nałożone na relacje między przypadkami użycia

Tab. 2-7 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza strukturalna*

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>	<i>Analiza strukturalna</i>	
<b>Stopień trudności grupy w zadaniu</b>	średnio trudne	
<b>Poruszane tematy</b>	<i>klasa a obiekt, atrybuty klasy, metody klasy, struktury generalizacji-specjalizacji, klasa abstrakcyjna a klasa konkretna, asocjacje, analiza wartości, diagramy struktur złożonych, przejście na schemat projektowy, przejście na schemat relacyjny, mechanizmy rozszerzalności</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>atributy klasy</i>	łatwe	<i>atribut złożony, opcjonalny, powtarzalny, pochodny, klasowy</i> różne sposoby modelowania zbioru wartości atrybutu
<i>metody klasy</i>	łatwe	<i>metoda zaimplementowana, obiektowa, klasowa, przesłanianie metod, polimorfizm metod</i>
<i>struktury generalizacji-specjalizacji</i>	średnio trudne	<i>rodzaje dziedziczenia – dziedziczenie dynamiczne, wieloaspektowe, nierozłączne, typu elipsa</i>
<i>klasa abstrakcyjna a klasa konkretna</i>	łatwe	<i>klasa abstrakcyjna, klasa konkretna</i>

<i>asocjacje</i>	łatwe	<i>nazwa asocjacji a rola asocjacji, kompozycja, asocjacja kwalifikowana</i>
<i>analiza wartości</i>	łatwe	<i>wartość pochodna – rozważania dotyczące poprawienia efektywności obliczeń atrybutu pochodnego, wartość początkowa – inicjalizowanie atrybutów wartością null, wartość graniczna</i>
<i>diagramy struktur złożonych</i>	łatwe	<i>wykorzystanie diagramu dla modelowania atrybutu złożonego</i>
<i>przejście na schemat projektowy</i>	łatwe	<i>obejście dla dziedziczenia klas – obejście dziedziczenia wieloaspektowego, nierożlaczego, dynamicznego</i>
<i>przejście na schemat relacyjny</i>	łatwe	<i>obejście dla dziedziczenia klas – różne sposoby przekształcania hierarchii</i>
<i>mechanizmy rozszerzalności</i>	łatwe	<i>ograniczenie, komentarz</i>

Tab. 2-8 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza dynamiczna*

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>	<i>Analiza dynamiczna</i>	
<b>Stopień trudności grupy w zadaniu</b>	średnio trudne	
<b>Poruszane tematy</b>	<i>diagramy aktywności, diagramy stanów, diagramy interakcji, podsumowanie analizy dynamicznej</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>diagramy aktywności</i>	średnio trudne	podział na część dialogową i część wykonawczą prezentacja części wykonawczej w postaci pseudokodu

		<i>romb decyzyjny – trzy notacje do modelowania rombu decyzyjnego</i>
<i>diagramy stanów</i>	średnio trudne	<i>maszyna stanowa behawioralna a maszyna protokołowa</i> <i>stan złożony sekwencyjny</i>
<i>diagramy interakcji</i>	trudne	<i>podział na część dialogową i część wykonawczą</i> <i>diagramy przeglądu interakcji</i> <i>diagramy sekwencji – dla schematu pojęciowego i schematu projektowego (z asocjacjami jednokierunkowymi i dwukierunkowymi)</i> <i>diagramy sekwencji – bez oznaczania czasu wykonywania komunikatu i z jego oznaczeniem</i> <i>klasy graniczne</i> <i>przywoływanie wystąpień interakcji</i> <i>fragmenty wyodrębnione na diagramach sekwencji – operatory loop, break, opt</i> <i>diagramy współpracy/kolaboracji izomorficzne z diagramami sekwencji</i> <i>diagram następstwa stanów – dla obiektów wybranej klas</i>
<i>podsumowanie analizy dynamicznej</i>	łatwe	modyfikacje fragmentu schematu pojęciowego/projektowego o dodatkowe metody

## Spis treści rozdziału

Rozdział 3 Wytwórnia porcelany stołowej .....	142
3.1 Tekst wymagań.....	142
3.2 Analiza funkcjonalna .....	144
3.2.1 Diagram kontekstowy .....	144
3.2.2 Model przypadków użycia.....	144
3.2.3 Scenariusz dla wybranego przypadku użycia .....	145
3.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzących przez osoby nauczane/uczące się .....	147
3.3 Analiza strukturalna.....	148
3.3.1 Schemat pojęciowy – rozwiązywanie pierwsze .....	148
3.3.2 Analiza elementów pochodnych wprowadzonych do schematu pojęciowego – rozwiązywanie pierwsze .....	151
3.3.3 Analiza wartości początkowych .....	152
3.3.4 Analiza wartości granicznych wprowadzonych do schematu pojęciowego – rozwiązywanie pierwsze .....	155
3.3.5 Schemat pojęciowy – rozwiązywania drugie i trzecie.....	159
3.3.6 Analiza elementów pochodnych wprowadzonych do schematu pojęciowego – rozwiązywanie drugie i trzecie.....	161
3.3.7 Zadania podstawowe .....	163
3.3.8 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzących przez osoby nauczane/uczące się .....	169
3.4 Analiza dynamiczna .....	170
3.4.1 Zadania podstawowe .....	170
3.4.2 Omówienie zmian na schemacie pojęciowym, wynikających z przeprowadzonej analizy dynamicznej i z poprzednich rozważań.....	172
3.5 Zadania do samodzielnej pracy .....	175
3.6 Podsumowanie zadania.....	176

## Rozdział 3

# Wytwórnia porcelany stołowej

Ewa Stemposz, Alina Stasiecka

### 3.1 Tekst wymagań

Wytwórnia porcelany stołowej wykonuje ręcznie malowane serwisy naczyniowe, a projektowany system ma ułatwiać jej pracownikom ewidencjonowanie zamówień na serwisy oraz określanie sposobów i terminów ich realizacji.

1. Wytwórnia zajmuje się wykonywaniem serwisów obiadowych oraz serwisów do kawy. Serwisy składają się z pojedynczych naczyń (np. talerz mały, filiżanka), dla których określona jest unikatowa nazwa, kolor, kształt, rozmiar (przykładowo średnica lub długość i szerokość), pojemność (tylko dla niektórych naczyń) oraz cena. W skład serwisu może wchodzić więcej niż jeden egzemplarz danego naczynia.
2. Serwis określany jest przez unikatową nazwę, liczbę osób, na jaką jest przewidziany, nazwę wzoru (namalowanego na nim) oraz czas potrzebny na wykonanie.
3. Cena serwisu zależy od tego, czy jest to serwis typowy czy kompletowany na zamówienie. Dla serwisu typowego cena jest sumą cen jego elementów składowych w dniu składania zamówienia. Z kolei dla serwisów kompletowanych na zamówienie cenę (uzyskaną jak powyżej) mnoży się przez pewien współczynnik, tzw. mnożnik ceny – jednakowy dla wszystkich zamówień tego rodzaju. Należy pamiętać aktualną wartość mnożnika.
4. Jedno zamówienie klienta może dotyczyć wielu egzemplarzy więcej niż jednego z dwóch wyżej wymienionych rodzajów serwisów. Dla każdego zamówienia pamiętany jest jego unikatowy numer, nazwa klienta oraz trzy daty: złożenia, planowanej realizacji oraz rzeczywistej realizacji. Ma być pamiętałe także, czy zamówienie zostało wykonane w planowanym terminie. W przypadku serwisów typowych klient nie ma możliwości wyboru ani wzorów, ani liczby i rodzajów naczyń wchodzących w ich skład.
5. Wśród rzemieślników zatrudnionych w wytwórni wyróżniamy m.in. pracowników zajmujących się malowaniem wzorów na porcelanie, tzw. malarzy wzorów. Podział rzemieślników jest kompletny.
6. Malarz musi umieć malować co najmniej jeden ze wzorów aktualnie wykorzystywanych w wytwórni. Z kolei, aby wzór mógł być wykorzystywany, musi być zatrudnionych co najmniej trzech malarzy, którzy potrafią go malować.

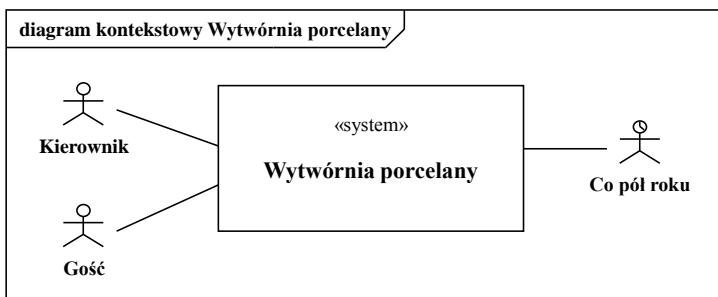
7. W danym momencie malarz może mieć maksymalnie dwóch uczniów, których uczy malowania jednego wzoru, a uczeń ma przydzielonego jednego nauczyciela. Nauka malowania wzoru nie może trwać dłużej niż dwa miesiące, a po jej zakończeniu uczeń obowiązkowo powinien przystąpić do egzaminu, którego pozytywne zaliczenie uprawnia go do samodzielnego malowania serwisów z tym wzorem. Ocena z egzaminu ma być przechowywana w postaci liczby w skali od 2 do 5, a pozytywne zaliczenie egzaminu oznacza uzyskanie oceny co najmniej 3. Jeśli uczeń nie zda egzaminu, to może jeszcze dwukrotnie ponowić próbę poprawy, każdą w odstępie co najmniej dwóch tygodni.
8. Uczeń może kolejno poznawać nowe wzory u tego samego malarza bądź u innych (w danym momencie może to być tylko jeden wzór). Oczywiście każdy z malarzy może poszerzać dalej swoje umiejętności, występując w roli ucznia.
9. Egzemplarz serwisu może być malowany przez co najwyżej dwóch malarzy. Chcemy przechowywać informacje o tym, kto malował dany egzemplarz serwisu, ale bez znajomości tego, kto malował poszczególne jego części. W danym momencie malarz może być przydzielony do malowania tylko jednego serwisu. Dla usprawnienia przydziału prac chcemy pamiętać: od kiedy do kiedy malarz był zajęty realizowaniem zamówienia.
10. Wypłata miesięczna malarza stanowi sumę pensji oraz premii podstawowej, której wysokość jest uzależniona od liczby serwisów pomalowanych przez malarza w miesiącu poprzedzającym wypłatę. Malarz może też otrzymać dodatkową premię, jeśli w poprzednim miesiącu miał ucznia/ucznów (w wysokości jednakowej na każdego z uczniów, niezależnie od tego, jakiego dnia miesiąca rozpoczęli naukę). Z kolei, wypłata ucznia jest równa takiemu ułamkowi określonej kwoty, jednakowej dla wszystkich uczniów, jaką część miesiąca poświęcił on na naukę.
11. System powinien m.in. wspomagać realizowanie poniższych zadań:
  - 11.1 prezentowanie oferty;
  - 11.2 znalezienie aktualnie wolnego malarza potrafiącego malować dany wzór;
  - 11.3 przydzielenie malarza do malowania serwisu;
  - 11.4 ustalenie harmonogramu prac związanych z malowaniem serwisów;
  - 11.5 wyliczenie wysokości premii dodatkowej dla malarza (za posiadanie uczniów w poprzednim miesiącu);
  - 11.6 wyliczenie miesięcznej wypłaty malarza;
  - 11.7 określenie, ilu uczniów miał malarz w miesiącu poprzedzającym wypłatę;
  - 11.8 wyszukanie klienta, który w zadanym okresie zapłacił najwięcej za zamówione serwisy (tzw. najlepszy klient);
  - 11.9 wyliczenie czasu, jaki w ostatnim miesiącu uczeń poświęcił na naukę;
  - 11.10 ustalenie liczby zamówień zrealizowanych po planowanym terminie (automatycznie, co pół roku).

## 3.2 Analiza funkcjonalna

### 3.2.1 Diagram kontekstowy

*Polecenie:* Zbuduj diagram kontekstowy w oparciu o ostatni punkt tekstu wymagań, tzn. punkt 11.

Diagram kontekstowy, przedstawiający aktorów wybranych dla systemu zgodnie z poleceniem, został zaprezentowany na Rys. 3-1.



Rys. 3-1 Diagram kontekstowy

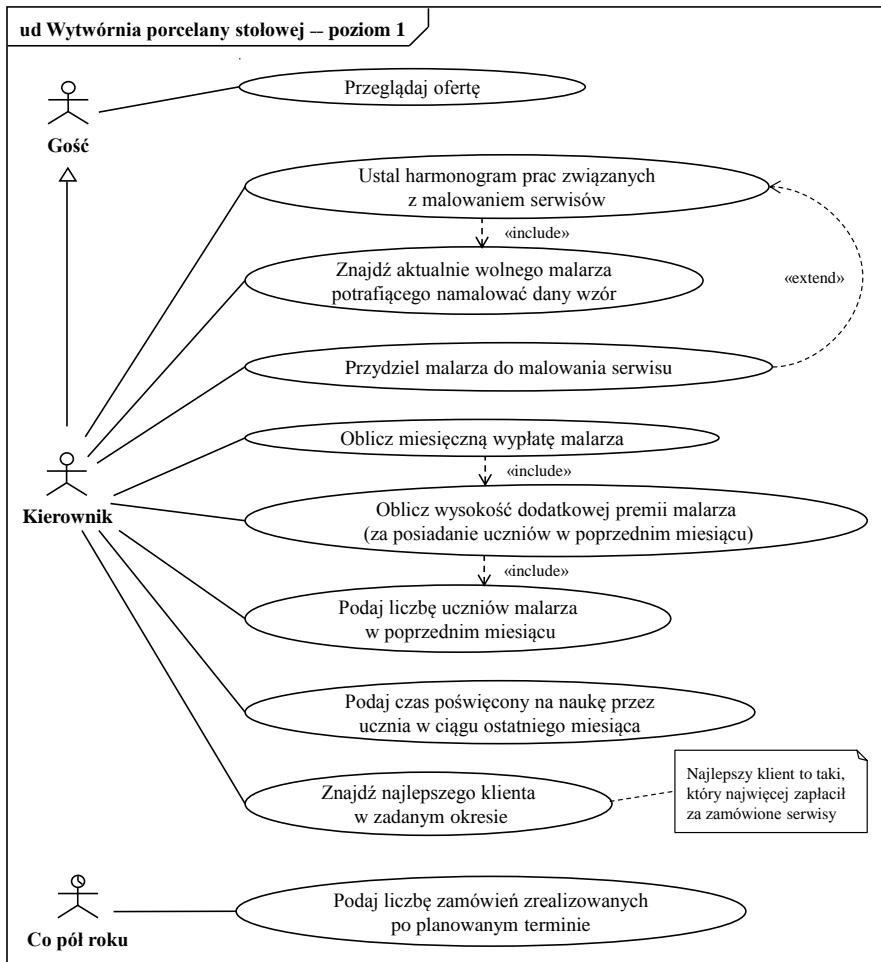
Należy zwrócić uwagę na fakt, że nie wszyscy aktorzy zostali bezpośrednio wymienieni w ww. punkcie tekstu wymagań. Ich wyboru dokonano na podstawie znajomości dziedziny problemowej.

### 3.2.2 Model przypadków użycia

*Polecenie:* Zbuduj model przypadków użycia, uwzględniając wyłącznie funkcjonalność sugerowaną w ostatnim punkcie tekstu wymagań, tzn. w punkcie 11.

*Uwaga:* Model należy skonstruować z perspektywy aktorów z otoczenia systemu, traktowanego jako całość (najwyższy poziom abstrakcji modelu). Należy uwzględnić hierarchię aktorów i relacje pomiędzy przypadkami (o ile mają/mogłyby mieć miejsce).

Diagram przypadków użycia [35, 40, 41, 46], dla funkcjonalności z punktu 11. tekstu wymagań, został przedstawiony na Rys. 3-2.



Rys. 3-2 Diagram przypadków użycia uwzględniający wyłącznie funkcjonalność sugerowaną w punkcie 11. tekstu wymagań

### 3.2.3 Scenariusz dla wybranego przypadku użycia

*Polecenie:* Dla funkcjonalności *Przydziel malarza do malowania serwisu* napisz scenariusz.

Przykładowy scenariusz dla wybranego przypadku użycia został przedstawiony w Tab. 3-1. Należy zwrócić uwagę, że wybrany przypadek użycia może być uruchomiony przez aktora *Kierownik* lub przypadek *Ustal harmonogram prac związkanych z malowaniem serwisu*. W zaproponowanym scenariuszu skoncentrowano się na możliwie najbardziej uproszczonym dialogu „aktor-system”, uwzględniając przepływy zdarzeń związane wyłącznie z aktorem *Kierownik*.

Tab. 3-1 Scenariusz dla przypadku użycia Przydziel malarza do malowania serwisu

<b>Nazwa przypadku użycia</b>	Przydziel malarza do malowania serwisu
<b>Warunek początkowy</b>	W systemie jest zarejestrowany co najmniej jeden serwis, co najmniej jeden malarz wzorów i co najmniej jeden wzór.
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Aktor <i>Kierownik</i> uruchamia przypadek użycia.</li> <li>2. System odpytuje o nazwę serwisu, do malowania którego ma być przydzielony malarz wzorów. Aktor wprowadza nazwę.<sup>1</sup></li> <li>3. System wyświetla listę aktualnie wolnych malarzy potrafiących malować wzór wykorzystany w danym serwisie. Aktor wybiera malarza.</li> <li>4. System prosi o potwierdzenie przypisania malarza do malowania serwisu. Aktor potwierdza.</li> <li>5. System informuje o przydzieleniu malarza do malowania serwisu.</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	<ol style="list-style-type: none"> <li>2a. Format wprowadzonych danych nie jest poprawny, system informuje o błędzie i powraca do punktu 2.</li> <li>2b. Serwis o podanej nazwie nie został zarejestrowany, system informuje o błędzie i odpytuje aktora o chęć kontynuacji wyboru dla innego serwisu.             <ol style="list-style-type: none"> <li>2ba. Aktor potwierdza chęć kontynuacji, system powraca do pkt. 2.</li> <li>2bb. Aktor rezygnuje z kontynuacji, system kończy przypadek użycia.</li> </ol> </li> <li>2c. Do malowania serwisu przydzielono już tylu malarzy, na ile pozwala zawarte w tekście wymagań ograniczenie (aktualnie do malowania serwisu wolno przydzielić co najwyżej dwóch malarzy), system informuje o błędzie i kończy przypadek użycia.</li> <li>3a. W danym momencie nie ma wolnych malarzy,</li> </ol>

<sup>1</sup> Można by zaproponować drugi wariant tego scenariusza, uwzględniający bardziej popularny sposób wyboru – z zaprezentowanej przez system listy.

	<p>którzy potrafią malować wzór wykorzystany w serwisie, system informuje o tym aktora i kończy przypadek użycia.</p> <p>4a. Aktor nie potwierdza przydzielenia wybranego malarza do malowania serwisu, system kończy przypadek użycia.</p>
<b>Zakończenie</b>	W dowolnym momencie.
<b>Warunek końcowy</b>	O ile warunki zostały spełnione – rejestracja faktu przydzielenia malarza do malowania serwisu.

### 3.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

- Aktorzy mieli być wybrani na podstawie tekstu wymagań i znajomości dziedziny problemowej, której dane wymagania dotyczą. Podane rozwiązanie jest przykładowe, tym niemniej nie powinni się tu znaleźć aktorzy tacy, jak np. *Podsystem obliczający premię malarza*, będący częścią składową systemu, a nie bytem z jego otoczenia – tylko takie byty mogą zostać uwzględnione w procesie identyfikacji aktorów na najwyższym poziomie abstrakcji dla modelu przypadków użycia. Jedynym wyjątkiem jest aktor nazwany *Podsystem czasu*.
- Częstym błędem jest niewłaściwy dobór nazw dla przypadków, jak np. *Liczba uczniów*. Nazwa przypadku powinna sugerować rodzaj zadania zleconego systemowi przez aktora, a tak nie jest dla tego przypadku, ponieważ jego nazwa sugeruje strukturę danych, a nie polecenie. Niepoprawne jest też umieszczenie na diagramie przypadku użycia *Maluj wzór*. Zgodnie z tekstem wymagań malowanie wzoru odbywa się poza systemem i bez wsparcia z jego strony.
- Błądem jest również niewłaściwy wybór relacji, które mogłyby zachodzić pomiędzy dwoma przypadkami. Na przykład, wydaje się być lepszym rozwiązaniem wprowadzenie relacji «*include*», łączącej przypadek *Oblicz miesięczną wyplatę malarza* (przypadek główny) z przypadkiem pobocznym *Oblicz wysokość dodatkowej premii malarza (za posiadanie uczniów w poprzednim miesiącu)* niż wykorzystanie w tym miejscu relacji «*extend*», ponieważ wynagrodzenie miesięczne obejmuje dwa składniki: pensję i premię dodatkową (która może być potencjalnie równa 0).

## 3.3 Analiza strukturalna

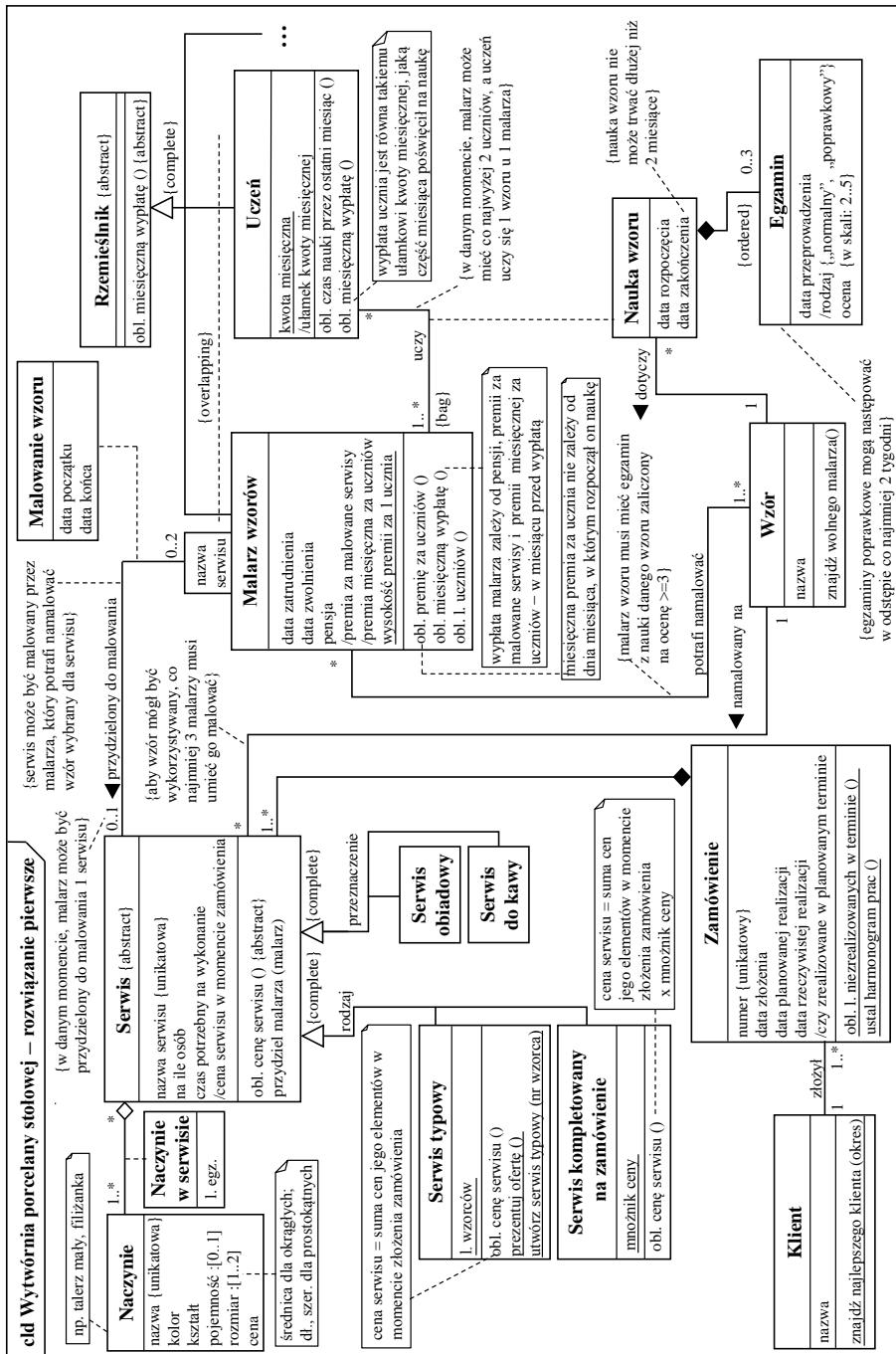
### 3.3.1 Schemat pojęciowy – rozwiązańe pierwsze

*Polecenie:* Dla podanego tekstu wymagań skonstruuj schemat pojęciowy.

Schemat pojęciowy [36, 41, 42, 45] przedstawiono na Rys. 3-3.

*Uwagi ogólne do schematu pojęciowego:*

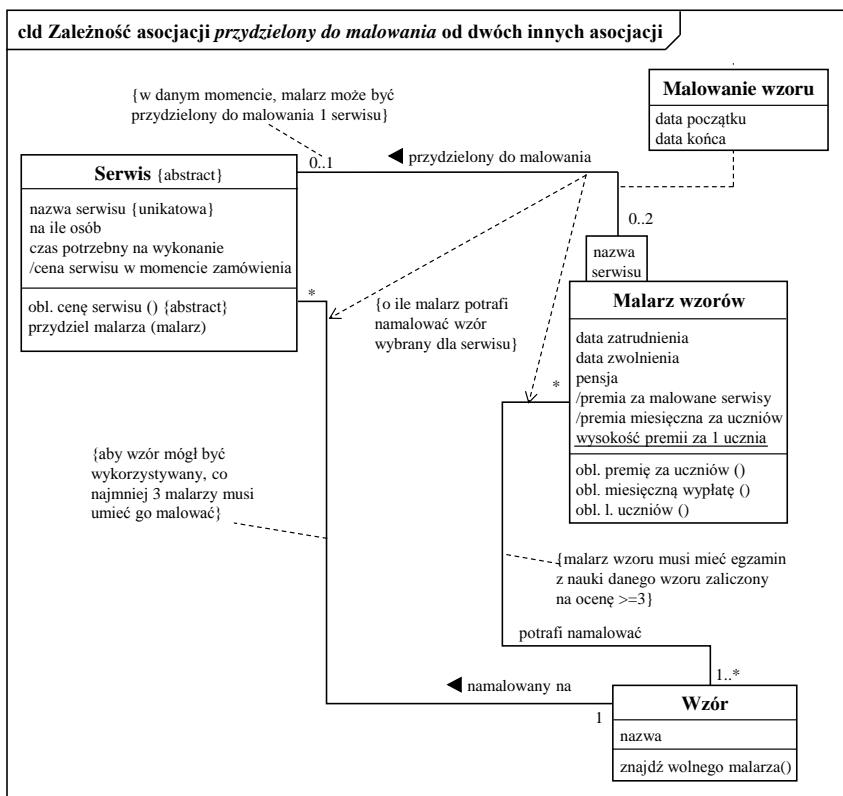
- Na podstawie analizy tekstu wymagań można założyć, że dla serwisu kompletowanego na zamówienie, klient w zupełnie dowolny sposób wybiera wzór, a także rodzaje i ilości naczyń, przeciwnie niż dla serwisu typowego, dla których w systemie powinny być zapamiętane ich wzorce niepodlegające sprzedaży, ale stanowiące podstawę zarówno do prezentowania oferty firmy, jak i do realizowania zamówień. Dlatego też zaproponowano kilka rozwiązań modelujących przechowywanie informacji o wzorach serwisów typowych. Pierwsze z nich zostało przedstawione na Rys. 3-3, a dwa pozostałe, nazywane dalej rozwiązaniami drugim i trzecim, zostały omówione w punkcie 3.3.5 (odpowiednie schematy pojęciowe umieszczone na Rys. 3-9 i Rys. 3-10).
- Analizując treść tekstu wymagań, zawsze należy rozważyć, jakie byty z dziedziny problemowej są reprezentowane przez jeden obiekt danej klasy. W tym przypadku, odnosi się to przede wszystkim do klas: *Serwis* i *Naczynie*. W rozwiążaniu pierwszym i drugim podklasy klasy *Serwis* przechowują informacje o egzemplarzach, natomiast w rozwiążaniu trzecim informacje o kategoriach serwisów. Z kolei, dla klasy *Naczynie*, dla wszystkich rozwiązań przyjęto, że jeden obiekt tej klasy modeluje kategorię naczyń.
- W rozwiążaniu pierwszym przedstawionym na Rys. 3-3 nie istnieją w systemie egzemplarze nie na sprzedaż, a opis typowych serwisów zapisany jest bezpośrednio w metodzie tworzącej nowy obiekt klasy *Serwis typowy*. Liczba zdefiniowanych wzorców stanowi wartość atrybutu klasowego *l. wzorców*. Zarejestrowanie typowego serwisu objętego zamówieniem klienta polega w tym rozwiążaniu na utworzeniu obiektu klasy *Serwis typowy* poprzez wywołanie metody z parametrem *nr wzorca – utwórz serwis typowy (nr wzorca)*. Numer ten można ustalić, wykorzystując metodę klasową *prezentuj ofertę()*. Wadą takiego podejścia jest, że każda zmiana dowolnego z już istniejących wzorców (lub dodanie nowego wzorca) wymagałyby zmiany kodu metody tworzącej nowy obiekt klasy *Serwis typowy* i dodatkowo kodu metody klasowej *prezentuj ofertę()*. Dla potrzeb generowania oferty, w trakcie realizowania przypadku użycia *Prezentuj ofertę*, mogłyby być tworzone tymczasowe obiekty przechowujące informacje o serwisach typowych wzorcowych, które byłyby usuwane tuż przed zakończeniem przypadku. W takim rozwiążaniu każdorazowa zmiana kodu metody tworzącej nowy obiekt klasy *Serwis typowy* nie wymagałaby jednoczesnej zmiany kodu metody *prezentuj ofertę()*.



Rys. 3-3 Schemat pojęciowy – rozwiążanie pierwsze

### *Uwagi dodatkowe do schematu pojęciowego:*

- Pomimo że w tekście wymagań nie umieszczoneo explicite informacji o konieczności archiwizowania danych dla malarzy, przyjęto założenie, że informacje o nich będą przechowywane również po zakończeniu przez nich pracy, aby zapobiec utracie informacji związanych z malowaniem serwisów, a zatem konieczne jest rejestrowanie daty rozpoczęcia i zakończenia przez nich pracy.
  - Ograniczenie *{serwis może być malowany przez malarza, który potrafi namalować wzór wybrany dla serwisu}* nałożone na asocjację *przydzielony do malowania* opisuje sytuację, w której ta asocjacja zależy od dwóch innych: *namalowany na* (przypisującą konkretny wzór do malowanego serwisu) i *potrafi namalować* (przechowującą informacje o malarzach potrafiących namalować dany wzór), co ilustruje Rys. 3-4.



Rys. 3-4 Fragment schematu pojęciowego zawierający zależność asocjacji *przydzielony do malowania* od dwóch innych asocjacji

- Na schemacie pojęciowym w nazwach atrybutów i metod przyjęto powszechnie stosowane skróty: *egz.* (egzemplarz), *l.* (liczba), *maks.* (maksymalny), *min.*

(minimalny), *obl.* (oblicz). Stosowanie skrótów jest dopuszczalne, o ile są one jednoznacznie interpretowane przez wszystkie osoby korzystające z danego schematu.

### **3.3.2 Analiza elementów pochodnych wprowadzonych do schematu pojęciowego – rozwiązańe pierwsze**

*Dla przypomnienia:* Wprowadzanie do schematu pojęciowego informacji redundantnej w postaci elementu pochodnego (czyli takiego, którego wartość można w każdym momencie wyliczyć lub wyprowadzić w oparciu o elementy bazowe) jest uzasadnione co najwyżej w przypadku, gdy proces wyliczenia jest kosztowny obliczeniowo, rzadko (lub nigdy) wartość ta ulega zmianom, ale jest często wykorzystywana w trakcie innych obliczeń. Zawsze należy więc rozważyć potrzebę wprowadzenia elementu pochodnego do schematu pojęciowego, natomiast należy przy tym zwrócić uwagę na to, czy po upływie pewnego czasu element ten pozostanie pochodny, to znaczy, czy zachowa swoją pierwotną wartość dla aktualnego wyliczenia. Ma to znaczenie dla potrzeb archiwizowania danych.

W trakcie przeprowadzonej analizy zidentyfikowano kilka atrybutów pochodnych, które omówiono szczegółowo poniżej.

#### **1. Atrybut pochodny: *cena serwisu* w klasie *Serwis***

Cena serwisu, zgodnie z tekstem wymagań, jest sumą cen naczyń (dla serwisu na zamówienie jest dodatkowo mnożona przez pewien współczynnik), a wartość zamówienia jest sumą cen serwisów obejmowanych przez to zamówienie. Należy rozważyć, co będzie się działało w momencie, gdy zostaną zmienione albo ceny naczyń, wchodzących w skład zamówionych serwisów, albo mnożnik ceny. Cena serwisu, wyliczona w momencie zmiany wartości elementów bazowych, będzie różniła się od tej ceny, za jaką serwis został sprzedany, a więc atrybut ten nie może być oznaczony jako pochodny. Analogicznie, wyliczona wartość zamówienia będzie różniła się od wartości obliczonej na podstawie cen naczyń uwzględnionych przy składaniu zamówienia. Dla rozwiązania pierwszego, ze względu na fakt, że obiekt klasy *Serwis* przechowuje informacje o jednym egzemplarzu serwisu, a nie o całej kategorii serwisów, atrybut *cena serwisu* w momencie złożenia zamówienia dobrze jest umieścić w klasie *Serwis*. W sytuacji, gdy obiekt klasy *Serwis* modeluje kategorię serwisów, atrybut cena serwisu powinien być umieszczony w klasie asocjacji pomiędzy klasami *Serwis* i *Zamówienie*. W obu przypadkach atrybut ten nie może być atrybutem pochodnym.

#### **2. Atrybut pochodny: *rodzaj* w klasie *Egzamin***

Zgodnie z tekstem wymagań i ogólną wiedzą o nauczaniu, w pierwszym terminie odbywa się zawsze egzamin normalny, a w kolejnych – egzaminy poprawkowe. Atrybut *rodzaj* (egzaminu) został oznaczony więc jako element pochodny, ponieważ przechowuje informację, którą można uzyskać na przykład

na podstawie porównania dat wszystkich egzaminów będących elementami konkretnej nauki wzoru i jak widać, jest wyliczany na podstawie danych niepodlegających zmianom w przyszłości. Obliczenie tej wartości nie jest kosztowne, w związku z tym można nie sugerować potrzeby jej przechowywania. Z drugiej strony, przechowywanie mogłoby być jednak uzasadnione, np. ułatwieniem przeszukiwania zbioru egzaminów przy sporządzaniu statystyk – dlatego jednak podjęto decyzję o pozostawieniu go na schemacie pojęciowym. Atrybut ten przyjmuje wartość „normalny” w trakcie tworzenia obiektu przechowującego informacje o pierwszym z egzaminów, przeciwnie niż przy tworzeniu obiektów dla egzaminów poprawkowych, gdzie będzie inicjalizowany wartością „poprawkowy”.

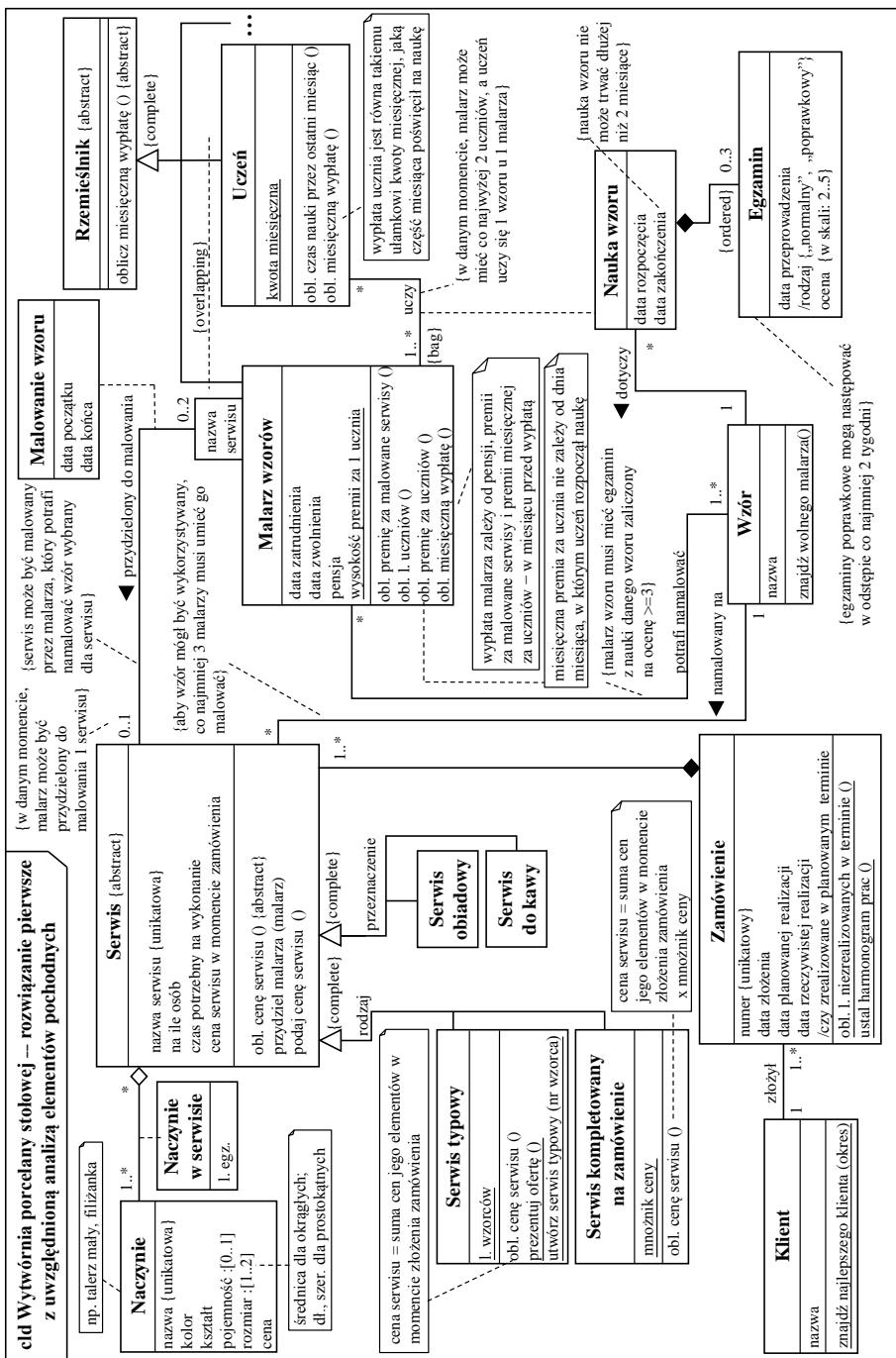
3. Atrybuty pochodne: *premia miesięczna za uczniów i premia za malowane serwisy* w klasie *Malarz wzorów*, *ułamek kwoty miesięcznej* w klasie *Uczeń* oraz atrybut *czy zrealizowane w planowanym terminie* w klasie *Zamówienie*

Wymienione powyżej atrybuty nie są wyliczane na podstawie danych mogących ulegać zmianom w przyszłości. Należy rozważyć, czy wprowadzanie do diagramu informacji redundantnej, w postaci elementów pochodnych, jest w ich przypadku uzasadnione. Przykładowo, atrybut *czy zrealizowane w planowanym terminie* w klasie *Zamówienie* jest wyliczany na podstawie porównania dwóch dat, a więc nie jest to kosztowne obliczeniowo. Można jednak sądzić, że funkcja korzystająca z tego atrybutu byłaby często wywoływaną. Dlatego też podjęto decyzję o pozostawieniu go w klasie *Zamówienie*. Atrybut ten mógłby być inicjalizowany wartością „true” przy rejestraniu nowego zamówienia, natomiast jego wartość finalna byłaby ustalona w momencie rejestrowania zakończenia prac nad realizacją zamówienia. W przypadku pozostałych atrybutów pochodnych, jak: *premia miesięczna za uczniów*, *premia za malowane serwisy* w klasie *Malarz wzorów*, *ułamek kwoty miesięcznej* w klasie *Uczeń* nie wydaje się, aby należało przechowywać ich wartości, gdyż zgodnie z tekstem wymagań nie jest planowane archiwizowanie wypłat czy premii.

Podsumowując, atrybut pochodny oznacza, że wartość wyliczona przez metodę ma być dodatkowo przechowywana. Możliwe są więc dwa rozwiązania: albo wprowadza się atrybuty pochodne i wyliczające je metody (które są z reguły pomijane na tym etapie analizy na diagramie, bo i tak wiadomo, że muszą się ostatecznie na nim pojawić), albo umieszczane są tylko metody wyliczające potrzebne wartości, bez sugerowania ich dodatkowego przechowywania. Schemat pojęciowy, po modyfikacji uwzględniającej analizę elementów pochodnych, został przedstawiony na Rys. 3-5.

### 3.3.3 Analiza wartości początkowych

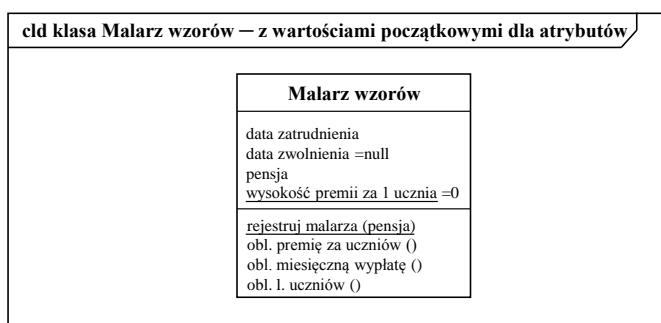
*Dla przypomnienia:* Wartość początkowa atrybutu jest to wartość, jaką przyjmuje ten atrybut w trakcie tworzenia nowego obiektu.



Rys. 3-5 Schemat pojęciowy po analizie elementów pochodnych – rozwiązanie pierwsze

Możliwa jest sytuacja, kiedy na skutek braku inicjalizacji wartość początkowa będzie poprawna formalnie (tzn. format danych będzie zgodny z typem atrybutu), a błędna semantycznie. Na przykład, atrybut *data zwolnienia* (z Rys. 3-5) – jeśli nie zostanie zainicjalizowany explicite w momencie tworzenia obiektu, może przyjąć wartość oznaczającą, że malarz będzie zwolniony już w chwili przyjęcia do pracy. Aby uniknąć sytuacji, w której przypadkowa wartość atrybutu może spowodować błędne działanie systemu, należy przeprowadzić analizę wartości początkowych i podjąć decyzję, które atrybuty muszą być zainicjalizowane w trakcie tworzenia obiektu i jakimi wartościami. Powyższe rozważania dotyczą także atrybutów klasowych.

Przykład klasy z zainicjalizowanymi atrybutami, obiektowym *data zwolnienia* i klasowym wysokość premii za 1 ucznia, przedstawiono na Rys. 3-6.

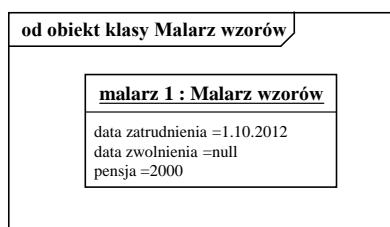


Rys. 3-6 Klasa *Malarz wzorów* z zainicjalizowanymi atrybutami

Zgodnie z Rys. 3-6, komunikat tworzący nowy obiekt klasy *Malarz wzorów* (nazwany *malarz 1*), zrealizowany w dniu 1.10.2012, miałby postać:

*malarz 1 := Malarz wzorów.rejestruj malarza (2000).*

Diagram z Rys. 3-7 pokazuje strukturę obiektu klasy *Malarz wzorów*, utworzonego za pośrednictwem tego komunikatu.



Rys. 3-7 Obiekt klasy *Malarz wzorów*

*Uwagi dodatkowe:*

- Atrybut *data zatrudnienia* jest inicjalizowany datą systemową, ale gdyby miało być inaczej, datę zatrudnienia należałoby przekazać na liście argumentów metody tworzącej nowy obiekt klasy *Malarz wzorów*.

- Decyzja, jak implementować wartość początkową *null* dla atrybutu *data zwolnienia*, jest podejmowana na etapie projektowania.
- Atrybuty klasowe nie są przechowywane w obiektach, dlatego też na diagramie obiektowym nie umieszczono atrybutu wysokość premii za 1 ucznia.

Na kolejnych schematach będą umieszczane tylko niektóre wartości początkowe, wynikające bezpośrednio z analizy tekstu wymagań.

### **3.3.4 Analiza wartości granicznych wprowadzonych do schematu pojęciowego – rozwiązańe pierwsze**

Po utworzeniu schematu pojęciowego dobrze jest przyjrzeć się wprowadzonym do niego wartościom granicznym i zastanowić się, czy wartości te nie będą, z różnych powodów, ulegać zmianom na przestrzeni czasu. Jeśli tak, to przyszły użytkownik systemu powinien mieć możliwość zrealizowania tych zmian, a co za tym idzie nie mogą one być zaimplementowane w kodzie systemu, np. w postaci literałów zarówno w tekstach ograniczeń, jak też w specyfikowaniu minimalnych i maksymalnych wartości dla liczności asocjacji.

Poniżej przedstawiono rozważania dotyczące przykładowych wartości granicznych, które zostały wprowadzone do schematu pojęciowego w rozwiązańu pierwszym z Rys. 3-5 (po analizie elementów pochodnych).

#### 1. Wartości graniczne zawarte w tekstuach ograniczeń

Pierwszą rozważaną wartością graniczną jest *1 serwis*, którą umieszczono w tekście ograniczenia *{w danym momencie malarz może być przydzielony do malowania 1 serwisu}*, nałożonym na asocjację *przydzielony do malowania* łączącą klasy *Malarz wzorów* i *Serwis*. Okres malowania wzoru na serwisie jest odmierzany za pomocą dat, bez określania godzin. Hipotetycznie można więc wyobrazić sobie sytuację, gdy przy dużej liczbie zamówień wyniknie potrzeba przydzielenia malarza do więcej niż jednego serwisu danego dnia (przykładowo, rano skończy malować jeden serwis i mógłby później rozpoczęć malowanie drugiego, co aktualnie nie jest możliwe do zrealizowania z powodu rozważanego ograniczenia). Dobra więc byłoby wprowadzić do schematu pojęciowego dodatkowy atrybut, który będzie przechowywał aktualnie wymaganą wartość graniczną, a dostarczenie skojarzonej z nim metody pozwoli na zmianę jego wartości bez konieczności ponownego kompilowania kodu. W omawianym przypadku powinien to być atrybut klasowy maks. l. serwisów (z ew. dodatkowym komentarzem o treści: *malowanych przez malarza w danym momencie*), umieszczony w klasie *Malarz wzorów*. Atrybut powinien być zainicjalizowany wartością *1*. Także tekst ograniczenia ulegnie zmianie po wprowadzeniu tego atrybutu na np. *{w danym momencie, malarz może być przydzielony do malowania <=maks. l. serwisów}*.

Wprowadzenie komentarza objaśniającego znaczenie atrybutu nie byłoby konieczne, gdyby potrzebną informację można było od razu zawsze w nazwie

atrybutu. Jest to rozwiązanie lepsze niż dołączenie komentarza, ale znaczaco utrudnia konstruowanie diagramu przy stosunkowo ograniczonej powierzchni kartki przeznaczonej na ten cel.

Analogiczne rozważania dotyczą pozostacych wartości granicznych zawartych w ograniczeniach. Dla przejrzystości: Wartości graniczne, ograniczenia, w których zostały zawarte oraz niezbędne modyfikacje schematu pojęciowego, pozwalające w przyszłości na zmianę wartości granicznych, zaprezentowano w Tab. 3-2.

Tab. 3-2 Wartości graniczne zawarte w ograniczeniach

Wartość graniczna	Ograniczenie	Zmiana w schemacie pojęciowym
3 malarzy	{aby wzór mógł być wykorzystany, co najmniej 3 malarzy musi umieć go malować}  nałożone na rolę asocjacji <i>namalowany na łączącej klasy Wzór i Serwis</i>	atrybut klasowy: <u>min. l. malarzy=3</u> w klasie <i>Wzór</i> , ew. z komentarzem <i>umiejących malować dany wzór</i>  zmienione ograniczenie: <i>{aby wzór mógł być wykorzystany, musi umieć go malować nie mniej malarzy, niż to określa min. l. malarzy}</i>
3	{malarz wzoru musi mieć egzamin z nauki danego wzoru zaliczony na ocenę >=3}	atrybut klasowy: <u>ocena_zaliczająca_egzamin=3</u> w klasie <i>Egzamin</i>  zmienione ograniczenie: <i>{malarz wzoru musi mieć egzamin z nauki danego wzoru zaliczony na ocenę &gt;=ocena_zaliczająca_egzamin}</i>
2 uczniów 1 wzór 1 malarz	{w danym momencie. malarz może mieć co najwyżej 2 uczniów, a uczeń uczy się 1 wzoru u 1 malarza}  nałożone na role asocjacji <i>uczy łączącej klasy Malarz wzorów i Uczeń</i>	atrybut klasowy: <u>maks. l. uczniów=2</u> w klasie <i>Malarz wzorów</i> , ew. z komentarzem <i>w danym momencie</i>  atrybut klasowy: <u>maks. l. wzorów=1</u> w klasie <i>Uczeń</i> , ew. z komentarzem <i>w danym momencie</i>  atrybut klasowy: <u>maks. l. malarzy=1</u> w klasie <i>Wzór</i> , ew. z komentarzem <i>w danym momencie uczących wzoru 1 ucznia</i>  zmienione ograniczenie: <i>{w danym momencie, malarz może mieć co najwyżej maks. l. uczniów, a uczeń uczy się nie więcej niż maks. l. wzorów, każdego u nie więcej niż maks. l. malarzy}</i>

2 miesiące	{nauka wzoru nie może trwać dłużej niż 2 miesiące}  nałożone na atrybut <i>data zakończenia</i> w klasie <i>Nauka wzoru</i>	atrybut klasowy: <u>maks. czas trwania</u> =2miesiące w klasie <i>Nauka wzoru</i>  zmienione ograniczenie: {data zakończenia - data rozpoczęcia<= maks. czas trwania}
2 tygodnie	{egzaminy poprawkowe mogą następować w odstępie co najmniej 2 tygodni}  warunek na utworzenie nowego obiektu klasy <i>Egzamin</i>	atrybut klasowy: <u>min. odstęp między poprawkami</u> =2tyg  zmienione ograniczenie: {odstęp między poprawkami}>= min. odstęp między poprawkami
2..5	{w skali 2..5}  nałożone na zbiór wartości atrybutu <i>ocena</i> w klasie <i>Egzamin</i>	atrybuty klasowe: <u>min. ocena</u> =2, <u>maks. ocena</u> =5 w klasie <i>Egzamin</i>  zmienione ograniczenie: {w skali min. ocena .. maks. ocena}

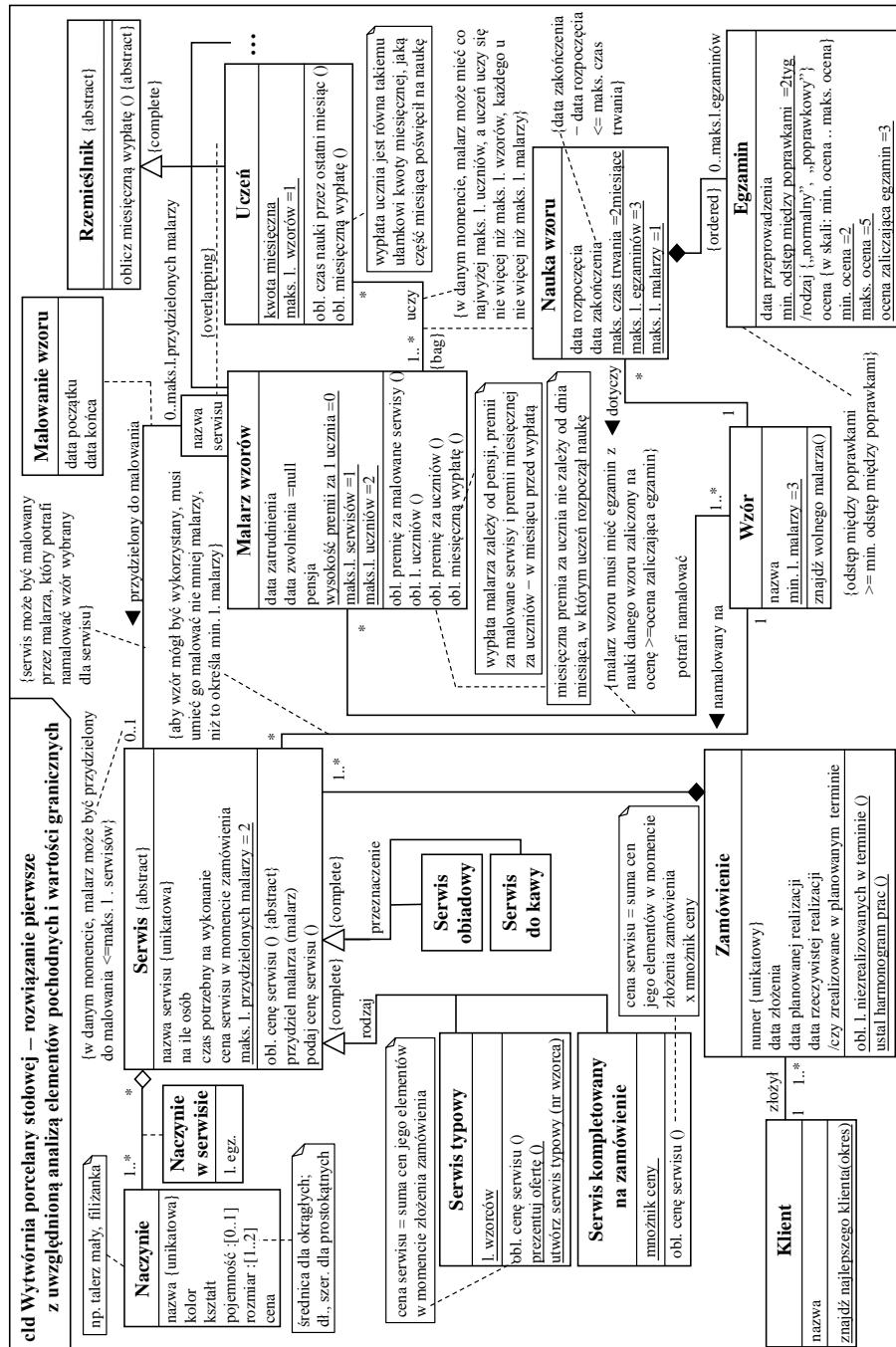
## 2. Wartości graniczne w oznaczeniu liczności asocjacji

Wartości graniczne wystąpiły także w oznaczeniu liczności: 0..2 dla liczności asocjacji *zatrudniony* przy łączającej klasy *Malarz wzorów* i *Serwis* oraz 0..3 dla kompozycji łączającej klasy *Nauka wzoru* i *Egzamin*.

W pierwszym przypadku, po zaimplementowaniu systemu, nie będzie wolno przydzielić więcej niż dwóch malarzy do malowania serwisu. Jednak przyjmując założenie, że serwis kompletnie na zamówienie może zawierać dużo elementów, a klient nalega na szybki termin realizacji, dobrze jest posiadać możliwość zatrudnienia przy serwisie więcej niż dwóch malarzy, gdy sytuacja tego wymaga. Warto więc wprowadzić do schematu atrybut klasowy maks. l. przydzielonych malarzy=2 w klasie *Serwis* oraz zmianę liczności z 0..2 na \*, z ograniczeniem {nie więcej niż maks. l. przydzielonych malarzy} (lub też zmianę liczności na 0..maks. l. przydzielonych malarzy, wtedy nie będzie potrzebny dodatkowy tekst ograniczenia).

W drugim przypadku, jeżeli w przyszłości polityka wytwórnii będzie wymagała zmiany (zmniejszenia, zwiększenia) liczby przystępów do egzaminu, to aby było to możliwe bez rekompilacji kodu, do schematu pojęciowego należy wprowadzić atrybut klasowy maks. l. egzaminów=3, umieszczony w klasie *Nauka wzoru* oraz zmianę liczności z 0..3 na \* z ograniczeniem {nie więcej niż maks. liczba egzaminów} (lub też zmianę liczności na 0..maks. l. egzaminów).

Kolejna wersja schematu pojęciowego, po modyfikacji uwzględniającej analizę wartości granicznych, została przedstawiona na Rys. 3-8. Dalsze rozważania będą prowadzone w oparciu o ten właśnie schemat pojęciowy.



Rys. 3-8 Schemat pojęciowy dla rozwiązywania pierwszego (po analizie wartości granicznych)

### 3.3.5 Schemat pojęciowy – rozwiązań drugie i trzecie

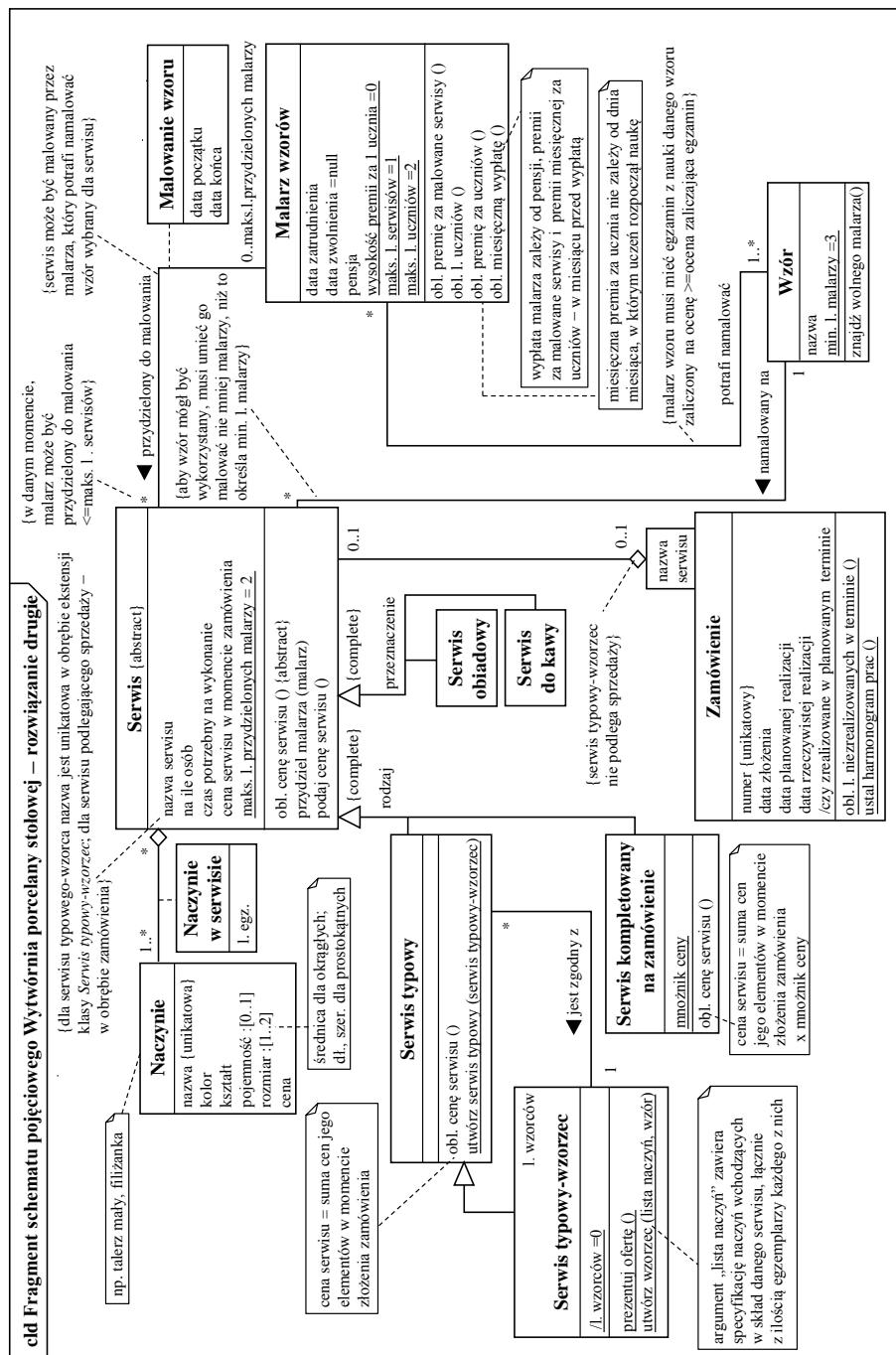
*Dla przypomnienia:* „W rozwiązyaniu pierwszym przedstawionym na Rys. 3-3 nie istnieją w systemie egzemplarze nie na sprzedaż, a opis typowych serwisów zapisany jest bezpośrednio w metodzie tworzącej nowy obiekt klasy *Serwis typowy*. Liczba zdefiniowanych wzorców stanowi wartość atrybutu klasowego *l\_wzorców*. Zarejestrowanie typowego serwisu objętego zamówieniem klienta polega w tym rozwiązyaniu na utworzeniu obiektu klasy *Serwis typowy* poprzez wywołanie metody z parametrem *nr wzorca* – utwórz serwis typowy (nr wzorca). Numer ten można ustalić, wykorzystując metodę klasową prezentuj oferte(). Wadą takiego podejścia jest, że każda zmiana dowolnego z już istniejących wzorców (lub dodanie nowego wzorca) wymagałyby zmiany kodu metody tworzącej nowy obiekt klasy *Serwis typowy* i dodatkowo kodu klasowej prezentuj oferte().”

Z powodu nieelastyczności powyższego rozwiązania zaproponowano dwa inne, przedstawione odpowiednio na Rys. 3-9 i Rys. 3-10. Dla przejrzystości rysunków pominięto na schematach pojęciowych klasy nieistotne dla przeprowadzanych rozważań; są to klasy: *Rzemieślnik*, *Uczeń*, *Nauka wzoru*, *Egzamin*, *Klient* oraz związane z nimi asocjacje, ograniczenia i komentarze.

#### Rozwiązanie drugie (Rys. 3-9)

W rozwiązyaniu drugim, będącym modyfikacją rozwiązyania pierwszego, zasadnicza różnica polega na wprowadzeniu klasy *Serwis-typowy wzorzec*. Dzięki temu zmiana własności już istniejącego serwisu typowego wzorcowego (czy też jego usunięcie) albo dodanie nowego wzorca nie wymaga zmiany konstruktora i ponownej komplikacji kodu. Klasa *Serwis typowy* ma tu podkласę *Serwis typowy-wzorzec*, w ekstensji której przechowywane są obiekty modelujące wzorcowe egzemplarze serwisów typowych, niepodlegające sprzedaży, w przeciwieństwie do serwisów modelowanych przez wystąpienia bezpośrednie klasy *Serwis typowy*. W związku z powyższym, asocjacja między klasami *Serwis* i *Zamówienie* musi być agregacją, a nie kompozycją, ponieważ nie każdy serwis podlega sprzedaży.

Trzeba też umieścić dodatkowe ograniczenie na rolę asocjacji *{serwis typowy-wzorzec nie podlega sprzedaży}*. Użyteczne wydaje się być również wprowadzenie ograniczenia dotyczącego unikatowości atrybutu *nazwa* umieszczonego w klasie *Serwis*: *{dla serwisu typowego-wzorca nazwa jest unikatowa w obrębie ekstensji klasy Serwis typowy-wzorzec; dla serwisu podlegającego sprzedaży – w obrębie zamówienia}*. W efekcie wprowadzenia ograniczenia na unikatowość nazwy, jako asocjację kwalifikowaną zamodelowano agregację między klasami *Serwis* i *Zamówienie*. Warto też zwrócić uwagę na sygnatury metod tworzących obiekty klas: *Serwis typowy* i *Serwis typowy-wzorzec*.



Rys. 3-9 Fragment schematu pojęciowego dla rozwiązywanie drugiego

### Rozwiązań trzecie (Rys. 3-10)

W rozwiążaniu trzecim, w związku z tym, że jeden obiekt klasy *Serwis* przechowuje informacje o całej kategorii serwisów, do schematu pojęciowego została wprowadzona klasa *Egzemplarz serwisu*, której wystąpienia bezpośrednie przechowują informacje o egzemplarzach każdego z serwisów w danym zamówieniu, co jest spowodowane koniecznością posiadania informacji o tym, które konkretnie egzemplarze zostały pomalowane przez danego malarza – ta informacja jest potrzebna do wyliczania jego miesięcznej wypłaty.

### **3.3.6 Analiza elementów pochodnych wprowadzonych do schematu pojęciowego – rozwiązanie drugie i trzecie**

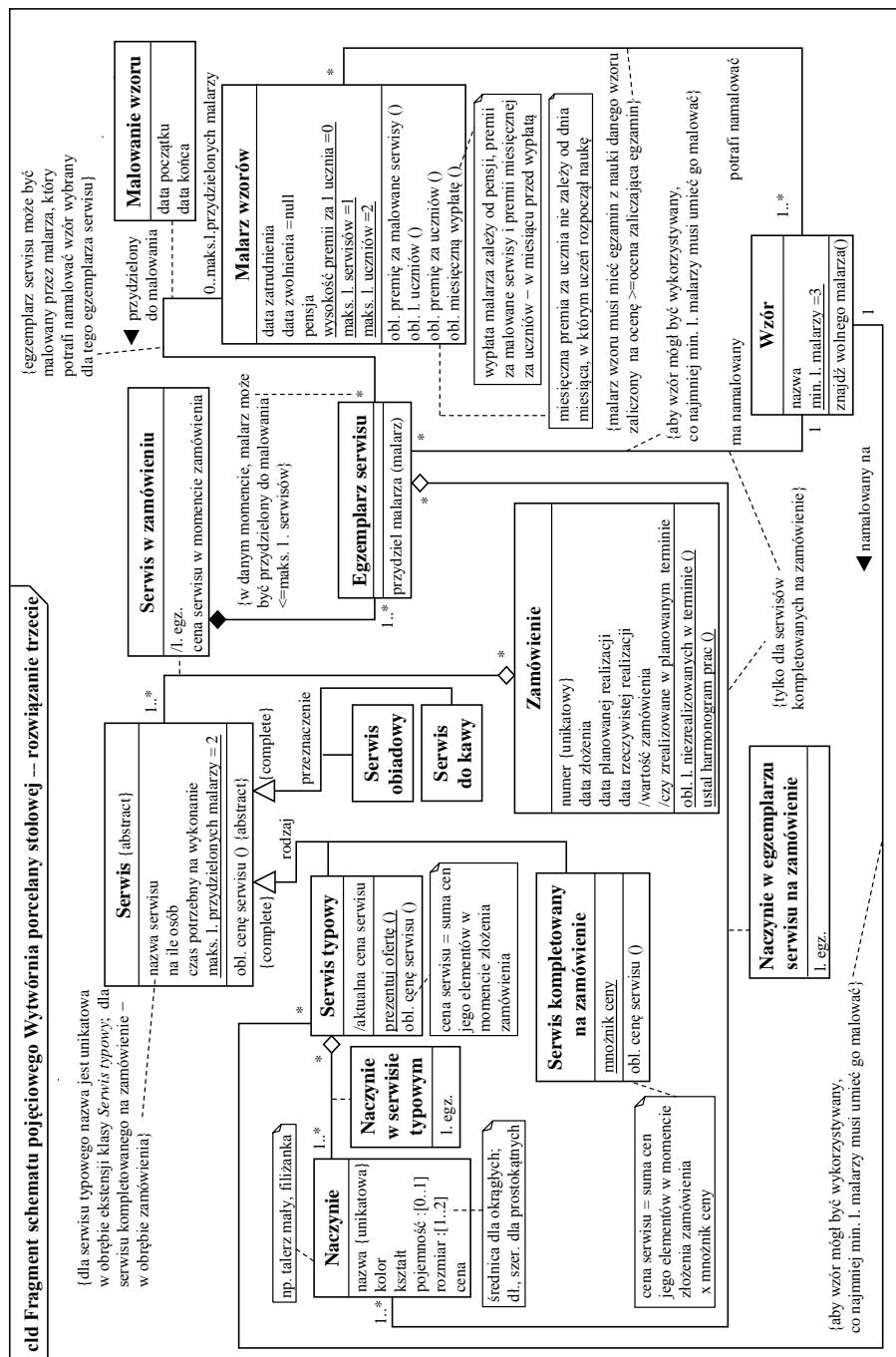
Dla rozwiązań: drugiego i trzeciego należy przeprowadzić analizę tych elementów pochodnych, które uległy zmianie w stosunku do rozwiązań poprzednich lub zostały dopiero teraz wprowadzone. Analizę wartości granicznych pominięto, gdyż proponowane rozwiązania drugie i trzecie nie wprowadziły, w stosunku do rozwiązania pierwszego, żadnych nowych wartości granicznych.

### Rozwiązań drugie (Rys. 3-9)

Rozwiązań to wprowadziło, w stosunku do rozwiązania pierwszego, tylko jeden nowy atrybut pochodny *l. wzorców* w klasie *Serwis typowy-wzorzec*. *L. wzorców* jest atrybutem pochodnym, klasowym, zainicjalizowanym wartością 0, oznaczającym aktualną licznosć ekstensji klasy *Serwis typowy-wzorzec*. Ponieważ nie jest on kosztowny obliczeniowo i nie wydaje się, by był często wykorzystywany w trakcie realizowania zadań systemu, nie widać oczywistej potrzeby przechowywania jego wartości. Można wprowadzić do klasy *Serwis typowy-wzorzec* metodę klasową *obl. l. wzorców()*, przy jednoczesnym usunięciu z niej atrybutu *L. wzorców* lub zostawić atrybut i zmniejszać lub zwiększać jego wartość przy tworzeniu/usuwaniu wzorców z systemu. Ze względu na niewielką zmianę schematu pojęciowego nie będziemy prezentować nowej wersji diagramu.

### Rozwiązań trzecie (Rys. 3-10)

Rozwiązań trzecie wprowadziło, w stosunku do poprzednich, następujące atrybuty pochodne: *aktualna cena serwisu* w klasie *Serwis typowy*, *l. egz.* w klasie *Serwis w zamówieniu* oraz *wartość zamówienia* w klasie *Zamówienie*. Pierwszy z wymienionych atrybutów dotyczy aktualnej ceny serwisu. Można przyjąć, że nie ma potrzeby przechowywania ceny, której wyliczenie nie jest kosztowne, a jej aktualna wartość jest potrzebna tylko w chwili, gdy rejestrowane jest zamówienie. Ceny archiwalne, za jakie zostały sprzedane serwisy, przechowywane są w klasie *Serwis w zamówieniu* (atrribut *cena serwisu w momencie zamówienia*). Tak więc w klasach *Serwis* i *Serwis typowy* pozostałą



Rys. 3-10 Fragment schematu pojęciowego dla rozwiązania trzeciego

jedynie metody *obl. cena serwisu()*, z których ta w nadklasie jest metodą abstrakcyjną.

W przypadku pozostałych atrybutów pochodnych również nie widać potrzeby przechowywania ich wartości.

L. egz. w klasie *Serwis w zamówieniu* jest to atrybut o wartości równej liczności roli asocjacji. Chociaż na przestrzeni czasu jego wartość nie ulega zmianie, to wyliczenie nie jest kosztowne. Do klasy *Serwis w zamówieniu* wystarczy więc jedynie wprowadzić metodę *obl. l. egz. ()*.

Kolejny z wymienionych powyżej atrybutów pochodnych, *wartość zamówienia* w klasie *Zamówienie*, jest atrybutem o wartości wyliczanej na podstawie cen serwisów wchodzących w skład danego zamówienia, które nie ulegają zmianom na przestrzeni czasu. Wyliczenie tej wartości nie jest kosztowne, więc wydaje się, że wystarczy jedynie wprowadzić do klasy *Zamówienie* metodę *obl. wartość zamówienia ()*.

Fragment schematu pojęciowego dla rozwiązania trzeciego, po analizie elementów pochodnych, został przedstawiony na Rys. 3-11.

### 3.3.7 Zadania podstawowe

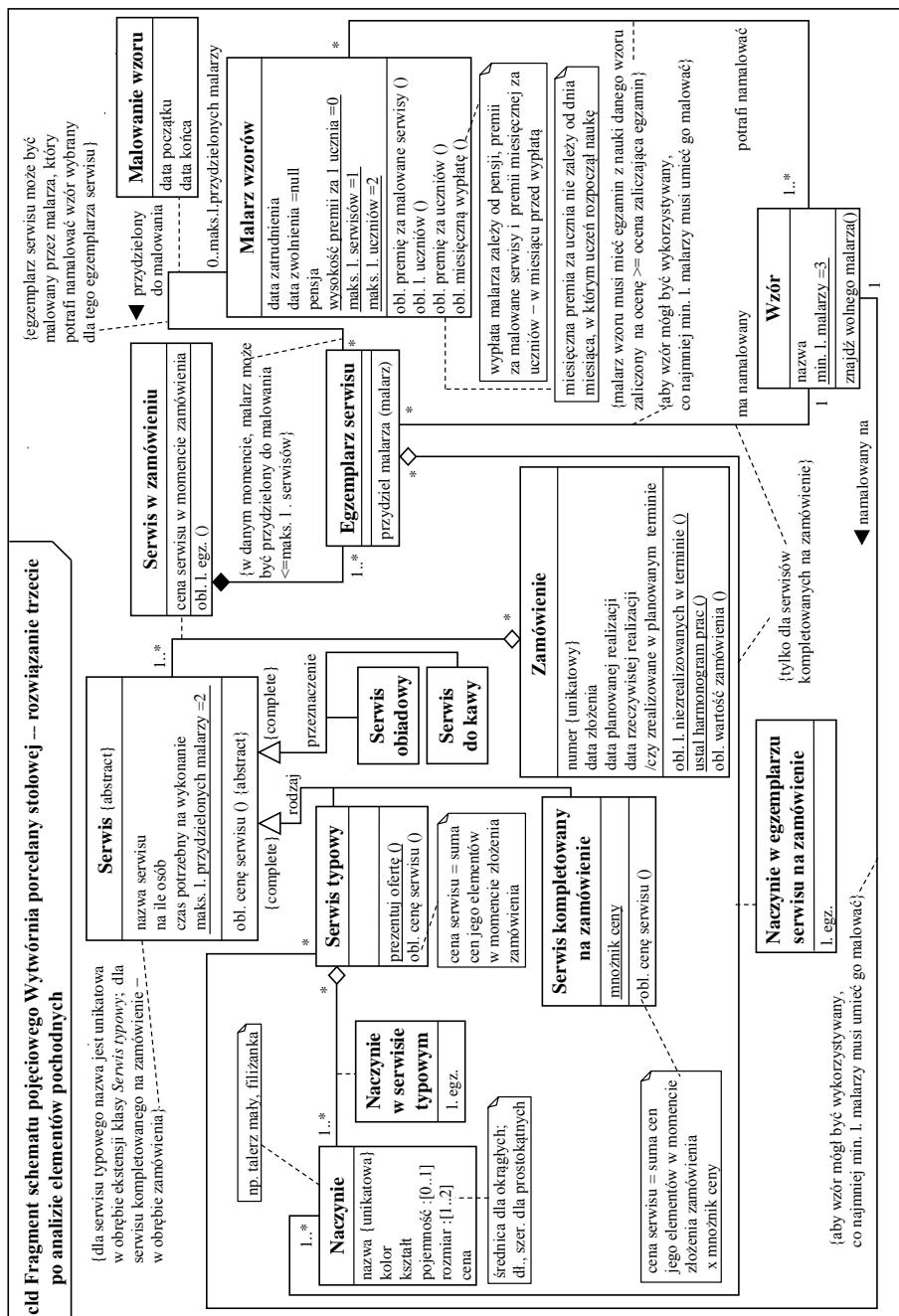
Zaprezentowane poniżej rozwiązania zadań odnoszą się do pierwszego z omawianych rozwiązań, przedstawionego na Rys. 3-8. Jeżeli jednak pozostałe rozwiązania zmieniałyby treść podanej odpowiedzi lub też wymagałyby rozszerzenia objaśnienia problemu, zostało to dodatkowo uwzględnione.

**Zad. 1:** Podkreśl te rodzaje dziedziczenia (rozłączne, nierozłączne, kompletne, niekompletne, jednokrotne, wielokrotne, jednoaspektowe, wieloaspektowe, dynamiczne, elipsa), które uważaś za potrzebne do wykorzystania w schemacie pojęciowym. Dla każdego wykorzystanego podziału podaj zastosowane rodzaje dziedziczenia. *Uwaga:* Domyslnie dziedziczenie jest rozłączne i niekompletne.

- hierarchia dla serwisów: dziedziczenie wieloaspektowe (aspekty: *rodzaj* i *przeznaczenie*), rozłączne (oba aspekty), kompletne (oba aspekty), jednokrotne (oba aspekty)
- hierarchia dla rzemieślników: dziedziczenie kompletne, nierozłączne (podklasy: *Malarz wzorów* i *Uczeń* – przecięcie ekstensji tych podklas nie jest zbiorem pustym), jednokrotne, jednoaspektowe, elipsa

W rozwiązaniu drugim, przedstawionym na Rys. 3-9, pojawiła się dodatkowa hierarchia dla serwisów typowych.

- hierarchia dla serwisów typowych: dziedziczenie niekompletne, rozłączne, jednokrotne, jednoaspektowe



Rys. 3-11 Fragment schematu pojęciowego dla rozwiązań trzeciego – po analizie elementów pochodnych

**Zad. 2:** Jakie rodzaje metod (abstrakcyjne, zaimplementowane) może zawierać klasa abstrakcyjna, a jakie klasa konkretna? Podaj przykłady.

Klasa abstrakcyjna może zawierać oba rodzaje metod, a klasa konkretna tylko metody zaimplementowane.

- metoda abstrakcyjna w klasie abstrakcyjnej: *obl. cenę serwisu ()* w klasie *Serwis*
- metoda zaimplementowana w klasie abstrakcyjnej: *podaj cenę serwisu ()* w klasie *Serwis*
- metoda zaimplementowana w klasie konkretnej: *obl. cenę serwisu ()* w klasie *Serwis typowy*

**Zad. 3:** Podaj przykłady metody klasowej, metody obiektowej, atrybutu klasowego i atrybutu obiektowego.

- metoda obiektowa: *obl. cenę serwisu ()* w klasie *Serwis*
- metoda klasowa: *znajdź najlepszego klienta (okres)* – operuje na ekstensji klasy *Klient*
- atrybut obiektowy: *nazwa serwisu* w klasie *Serwis*
- atrybut klasowy: *mnożnik ceny* w klasie *Serwis kompletowany na zamówienie*

**Zad. 4:** Wskaź przykład zastosowania polimorfizmu metod.

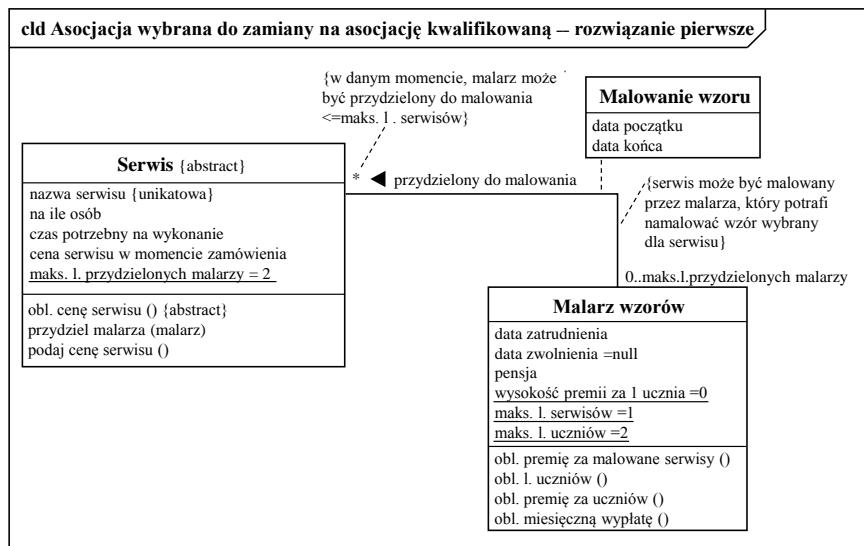
Metody polimorficzne to np. metody: *obl. cenę serwisu ()* w klasach hierarchii zbudowanej dla serwisów lub *obl. miesięczną wyplatę ()* w hierarchii dla rzemieślników.

**Zad. 5:** Asocjację posiadającą niesymetryczne liczności oraz atrybut (lub klasę asocjacji) zamień na asocjację kwalifikowaną. Uzasadnij, dlaczego poddałeś zamianie właśnie tę asocjację.

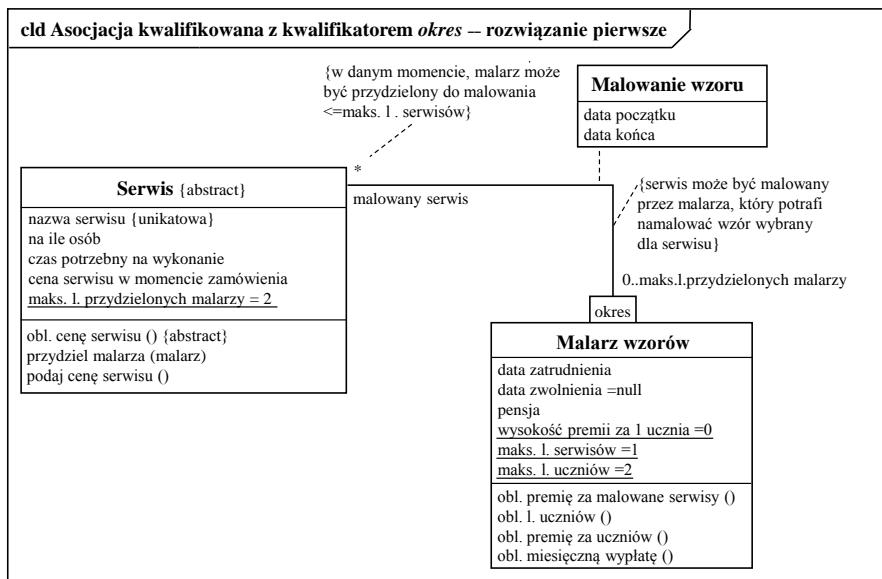
Na schemacie pojęciowym dla rozwiązania pierwszego (Rys. 3-8) umieszczono asocjację kwalifikowaną pomiędzy klasami *Serwis* i *Malarz wzorów*, która wskazuje, że kluczem do przeszukiwania zbioru serwisów malowanych przez danego malarza może być *nazwa serwisu*. Kwalifikatorem mógłby być również atrybut *okres*, umożliwiający znalezienie wszystkich serwisów malowanych przez danego malarza w zadany okresie. Należy zwrócić uwagę na fakt, że zbiór serwisów malowanych przez danego malarza w zadany okresie jest podzbiorem zbioru serwisów malowanych na przestrzeni czasu przez tego malarza.

Rys. 3-12 przedstawia fragment diagramu klas dla rozwiązania pierwszego przed zamianą asocjacji „zwykłej” na asocjację kwalifikowaną.

Wykorzystanie asocjacji kwalifikowanej z kwalifikatorem *okres* ilustruje Rys. 3-13. Dla przejrzystości rozważać nazwę asocjacji *przydzielony do malowania* zastąpiono rolą asocjacji o nazwie *malowany serwis*.



Rys. 3-12 Fragment schematu pojęciowego z Rys. 3-8 sprzed zamiany asocjacji „zwykłej” na asocjację kwalifikowaną

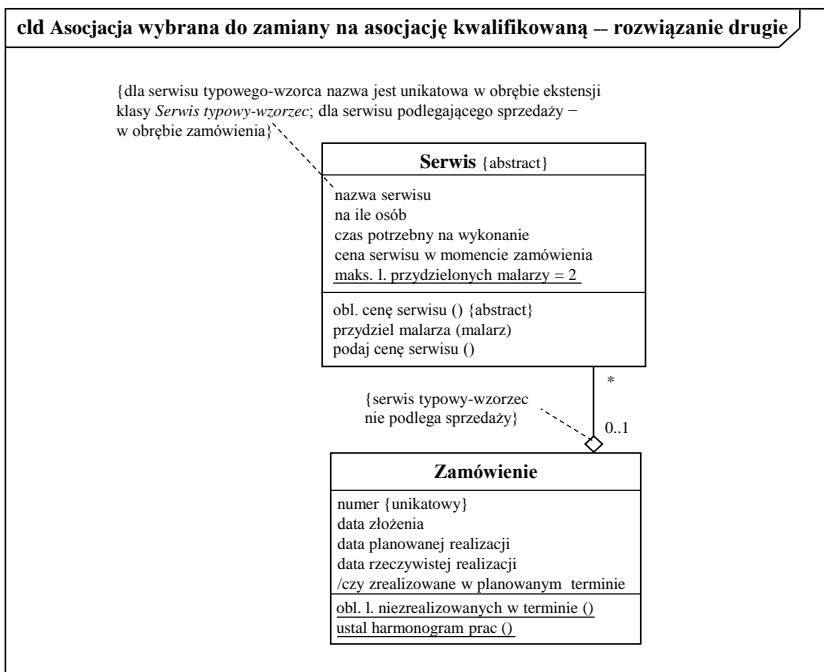


Rys. 3-13 Fragment schematu pojęciowego z Rys. 3-8 po zamianie asocjacji „zwykłej” na asocjację kwalifikowaną z kwalifikatorem *okres*

W przypadku rozwiązania drugiego, przedstawionego na Rys. 3-9, *nazwa serwisu* pozostaje atrybutem unikatowym, ale nałożone zostały na nią dodatkowe ograniczenia: *dla serwisu typowego-wzorca nazwa jest unikatowa w obrębie*

*ekstensji klasy Serwis typowy-wzorzec; dla serwisu podlegającego sprzedaży – w obrębie zamówienia} (serwis podlegający sprzedaży – modelowany przez obiekt klasy Serwis typowy lub klasy Serwis kompletowany na zamówienie). Atrybut nazwa serwisu mógłby pozostać kwalifikatorem roli malowany serwis asocjacji pomiędzy klasami Serwis i Malarz wzorów, trzeba jednak pamiętać, że w przypadku serwisów podlegających sprzedaży mógłby wskazywać na więcej niż jeden serwis o tej samej nazwie malowany przez danego malarza; oczywiście ten zbiór byłby tylko podzbiorem zbioru serwisów malowanych przez danego malarza na przestrzeni czasu.*

Drugie z ograniczeń nałożonych na wartość atrybutu nazwa (*{dla serwisu podlegającego sprzedaży nazwa jest unikatowa w obrębie zamówienia}*) skutkuje tym, że asocjacją kwalifikowaną mogłaby być asocjacja między klasami Serwis i Zamówienie, jak to przedstawia fragment schematu pojęciowego dla rozwiązania drugiego sprzed zamiany asocjacji „zwykłej” na asocjację kwalifikowaną (Rys. 3-14).



Rys. 3-14 Fragment schematu pojęciowego z Rys. 3-9 sprzed zamiany asocjacji „zwykłej” na asocjację kwalifikowaną

Na schemacie pojęciowym dla rozwiązania trzeciego, zaprezentowanym na Rys. 3-11 nie umieszczone w ogóle asocjacji kwalifikowanej. Do zamiany na asocjację kwalifikowaną można by było wybrać asocjację między klasami Serwis i Zamówienie, analogicznie jak dla rozwiązania drugiego. Odpowiednie fragmenty schematów pojęciowych wyglądałyby identycznie.

**Zad. 6:** Wskaż asocjację, która jest agregacją lub kompozycją (lub może kwalifikować się do zamiany na takową). Objaśnij, dlaczego zdecydowałeś się na wybór agregacji/kompozycji, a nie „zwykłej” asocjacji? Ponadto, dlaczego wybrałeś agregację, a nie kompozycję (czy też odwrotnie)?

Wszystkie zastosowane agregacje/kompozycje lepiej niż zwykła asocjacja wskazują na związki typu część-całość występujące między obiektami odpowiednich klas.

Na schematach pojęciowych dla wszystkich trzech rozwiązań występuje kompozycja łącząca klasę *Nauka wzoru* z klasą *Egzamin*. Egzaminy są częścią składową nauki wzoru i bez niej nie mają racji istnienia. Usunięcie obiektu klasy *Nauka wzoru* powinno skutkować usunięciem z systemu wszystkich związanych z nim obiektów klasy *Egzamin*. Tak więc powyższe przesłanki w sposób oczywisty wskazują na potrzebę wykorzystania kompozycji do modelowania relacji zachodzącej pomiędzy obiektami tych klas.

Na schemacie pojęciowym, dla rozwiązania pierwszego i drugiego, umieszczonego agregację pomiędzy klasami *Serwis* i *Naczynie*, gdzie obiekty klasy *Serwis* traktowane są jako całość (agregat) w relacji zachodzącej pomiędzy serwisem a naczyniami wchodząymi w jego skład, traktowanymi jako części. Wykorzystano agregację, ponieważ obiekty klasy *Naczynie* mogą samodzielnie istnieć w systemie i nie będą z niego usuwane łącznie z usuwaniem obiektów klasy *Serwis*.

#### Rozwiązanie pierwsze

Do schematu wprowadzono kompozycję łączącą klasę *Zamówienie* (całość) z klasą *Serwis* (część). Zastosowano kompozycję, gdyż w tym rozwiązaniu obiekty klasy *Serwis* są wprowadzone do systemu dopiero po utworzeniu obiektu klasy *Zamówienie*, w trakcie rejestrowania zamówienia i będą usuwane z systemu łącznie z obiektami klasy *Zamówienie*.

#### Rozwiązanie drugie

Schemat pojęciowy zawiera agregację pomiędzy klasami *Zamówienie* (całość) i *Serwis* (część), a nie kompozycję (jak w rozwiązaniu pierwszym), ponieważ obiekt klasy *Serwis typowy-wzorzec*, będący wystąpieniem pośrednim klasy *Serwis*, istnieje w systemie „samodzielnie”, tzn. bez powiązania z obiektem klasy *Zamówienie* (nie wszystkie serwisy podlegają sprzedaży, zgodnie z ograniczeniem {serwisy typowe – wzorce nie podlegają sprzedaży} nałożonym na rolę asocjacji).

#### Rozwiązanie trzecie

W tym rozwiązaniu zastosowano kompozycję pomiędzy klasami *Serwis w zamówieniu* (całość) i *Egzemplarz serwisu* (część), co jest oczywiste i nie wymaga specjalnego uzasadniania. Oprócz kompozycji, diagram zawiera trzy agregacje pomiędzy klasami: *Naczynie* (część) i *Serwis typowy* (całość), *Naczynie* (część) i *Egzemplarz serwisu* (całość) oraz *Serwis* (część) i *Zamówienie* (całość). We wszystkich trzech przypadkach części składowe, odpowiednio naczynia czy serwisy, mogą istnieć w systemie bez obiektu-całości,

ponieważ jeden obiekt zarówno klasy *Serwis*, jak i klasy *Naczynie* przechowuje informacje o całej kategorii serwisów czy naczyń, może więc wchodzić w skład wielu całości (obiekty klas *Serwis typowy*, *Zamówienie* czy też *Egzemplarz serwisu*).

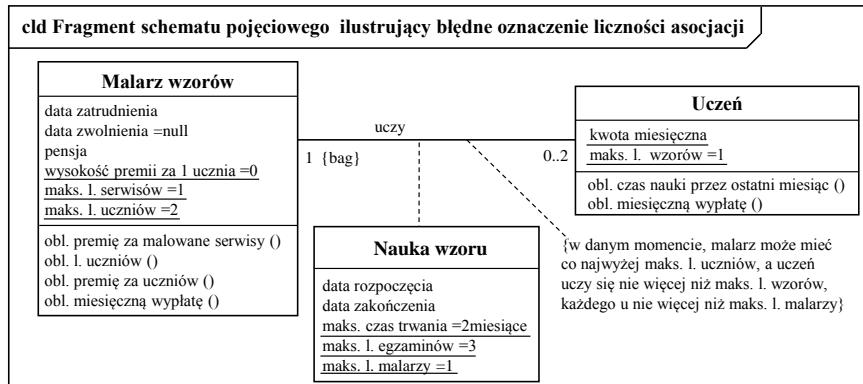
**Zad. 7:** Wskaż przykładowe ograniczenie. Na jaki element modelu zostało ono nałożone? Czy jest to ograniczenie statyczne czy dynamiczne? Zaproponuj, jaka metoda i w jakiej klasie mogłaby być odpowiedzialna za jego zapewnienie.

Jako przykład można podać ograniczenie *{w danym momencie malarz może mieć nie więcej uczniów, niż to określa maks. l. uczniów, a uczeń uczy się maks. l. wzorów}*. Jest to ograniczenie dynamiczne (ponieważ ważny jest poprzedni stan zbiorów powiązań dla obu łączonych obiektów), które zostało nałożone na role asocjacji między klasami *Malarz wzorów* i *Uczeń*; mogłoby być implementowane np. w metodzie obiektowej *Przypisz nauczyciela (nauczyciel)* w klasie *Uczeń*.

### 3.3.8 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

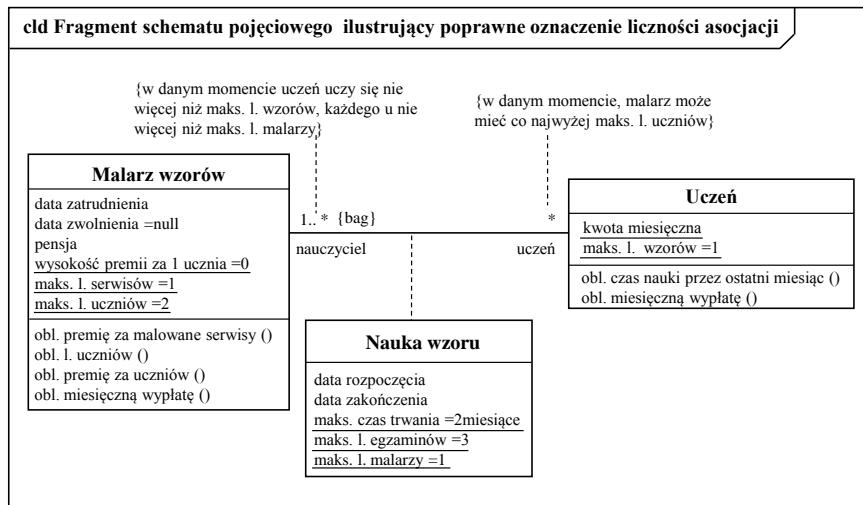
- Częstym błędem jest nieumieszczenie nazwy nadklasy w nazwie podklasy: np. klasa *Typowy* zamiast *Serwis typowy* czy też klasa *Na zamówienie* zamiast *Serwis kompletowany na zamówienie*, co może utrudniać rozumienie, o jaki rodzaj bytów chodzi (np. w kodzie źródłowym).
- Często też do schematów są wprowadzane metody, dla których nie określono listy argumentów, mimo że w tekście wymagań explicite taka potrzeba została sformułowana – np. metoda *znajdź najlepszego klienta ()* (klasa *Klient*) wymaga podania argumentu *okres*, zgodnie z punktem 11.8 tekstu wymagań.
- *Dla przypomnienia:* Jeśli dwie klasy zostały połączone asocjacją, to nie należy wprowadzać do żadnej z nich skojarzonej z tą asocjacją dodatkowej informacji. Przykładowo, jeśli klasy *Serwis* i *Wzór* zostały połączone asocjacją modelującą wzór, który został namalowany na serwisie, to nie należy umieszczać w klasie *Serwis* atrybutu *nazwa wzoru* (ani atrybutu powtarzalnego *serwis* w klasie *Wzór*). Nie można też wprowadzać atrybutów zamiast asocjacji na etapie analizy – w analizie asocjacja ma pierwszeństwo przed atrybutem.
- Ponadto, konstruując schemat pojęciowy, należy uwzględniać archiwizowanie danych jako efekt upływu czasu. Przykładowo, na przestrzeni czasu malarz wzorów mógł nauczać wielu uczniów, ale tylko dwóch w danym momencie, dlatego należy wprowadzić do schematu ograniczenie dynamiczne – a nie statyczne – na rolę *uczeń* dla asocjacji *uczy*. Z kolei, ograniczenie *{bag}*, które też zostało wprowadzone dla tej asocjacji, wynika z następującego fragmentu tekstu wymagań: „Uczeń może kolejno poznawać nowe wzory u tego samego malarza bądź u innych ...”, co oznacza możliwość istnienia wielu powiązań między tym samym malarzem i tym samym uczniem.

Rozwiążanie z błędami zostało przedstawione na Rys. 3-15.



Rys. 3-15 Błędne oznaczenie liczności asocjacji *uczy*

Rozwiążanie poprawne zostało zaprezentowane na Rys. 3-16, przy czym dla uwidocznienia wyjaśnień, zamiast nazwy asocjacji, wprowadzono jej role.



Rys. 3-16 Poprawne oznaczenie liczności asocjacji *uczy*

## 3.4 Analiza dynamiczna

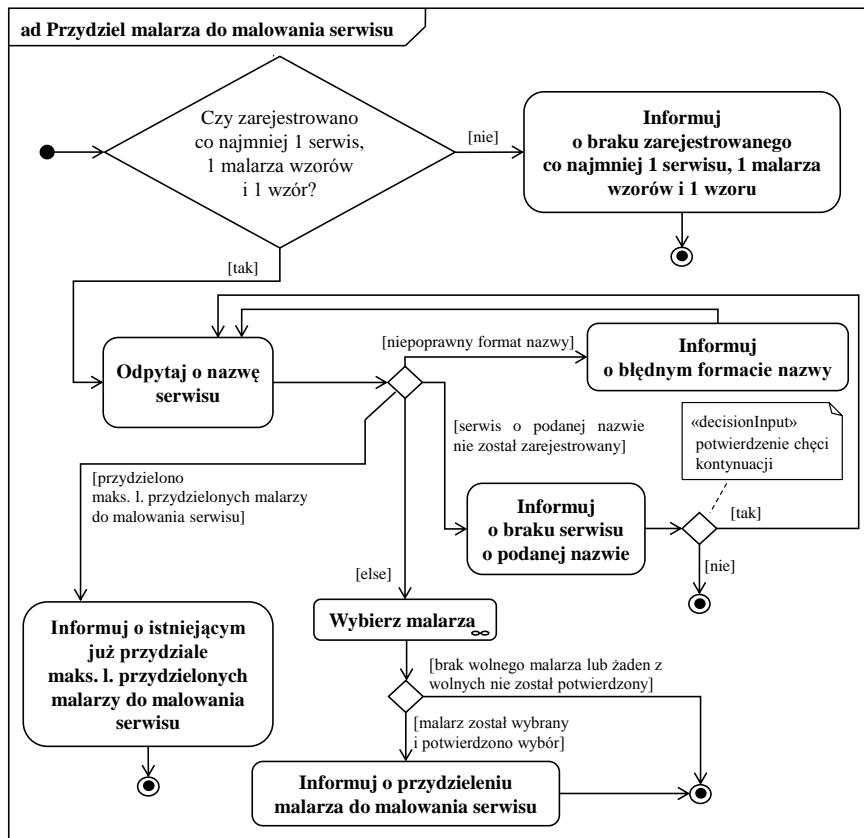
### 3.4.1 Zadania podstawowe

Poniższe przykładowe rozwiązania zadań odnoszą się do pierwszego z rozwiązań, dla schematu pojęciowego przedstawionego na Rys. 3-8.

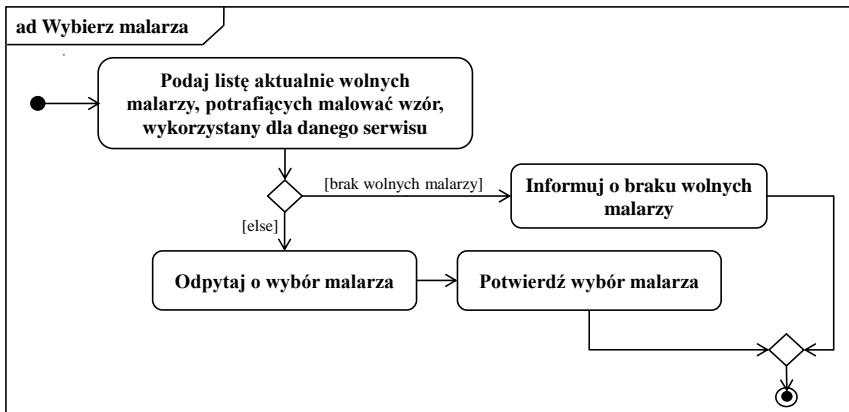
**Zad. 1:** Dla funkcjonalności *Przydziel malarza do malowania serwisu* skonstruuj diagram aktywności zgodny ze scenariuszem umieszczonym w punkcie 3.2 (*Analiza funkcjonalna*).

*Uwaga do scenariusza:* W przypadku rozwiązania drugiego, w sytuacji, gdy przypadek jest wywoływany przez aktora *Kierownik*, należałoby w pierwszym kroku odnaleźć odpowiednie zamówienie, np. odpytując o klienta i wybierając z listy jego zamówienia, a dopiero potem wyszukać serwis po nazwie, która jest unikatowa w obrębie jednego zamówienia. Dla rozwiązania trzeciego, jak w rozwiązaniu drugim, najpierw należy odnaleźć zamówienie, a następnie wyszukać egzemplarz serwisu.

Przykładową realizację dla scenariusza, sporzązonego w trakcie przeprowadzania analizy funkcjonalnej, ilustruje diagram aktywności [40, 41] zawarty na Rys. 3-17. Diagram ten zawiera aktywność złożoną *Wybierz malarza*, której strukturę przedstawiono na Rys. 3-18.



Rys. 3-17 Diagram aktywności dla przypadku użycia *Przydziel malarza do malowania serwisu* z wykorzystaniem aktywności złożonej *Wybierz malarza*



Rys. 3-18 Diagram aktywności dla aktywności *Wybierz malarza* pokazujący strukturę aktywności złożonej z Rys. 3-17

Dla przypomnienia: UML dopuszcza wykorzystanie kilku notacji pozwalających na modelowanie rombu decyzyjnego. Zostały one zaprezentowane na Rys. 3-17. Wybór rodzaju notacji jest subiektywny i zależy od decyzji analityka tworzącego model.

### 3.4.2 Omówienie zmian na schemacie pojęciowym, wynikających z przeprowadzonej analizy dynamicznej i z poprzednich rozważań

Poniższe rozważania odnoszą się do pierwszego z prezentowanych rozwiązań dla schematu pojęciowego, przedstawionego na Rys. 3-8. Analogiczne rozważania można przeprowadzić dla pozostałych rozwiązań (drugiego i trzeciego).

#### 1. Wyniki analizy dynamicznej

W efekcie przeprowadzonej analizy dynamicznej w strukturze systemu należy umieścić (o ile nie zostały dotąd umieszczone) metody takie, jak:

klasa *Malarz*:

- metoda obiektowa *przydziel malarza do malowania serwisu (serwis)*
- metoda obiektowa *podaj l. serwisów aktualnie przydzielonych malarzowi do malowania ()*
- metoda klasowa *wybierz malarza (serwis)*
- metoda klasowa *podaj listę aktualnie wolnych malarzy (wzór)*
- metoda klasowa *podaj listę malarzy potrafiących malować wzór (wzór)*

klasa *Serwis*:

- metoda obiektowa *przydziel malarza do malowania (malarz)*
- metoda obiektowa *obl. l. aktualnie przypisanych malarzy do malowania()*
- metoda klasowa *znajdź serwis (nazwa)*

2. Metody wynikłe z różnych sposobów tworzenia obiektów przechowujących informacje o serwisach

Z przeprowadzonych rozważań wynika kilka spostrzeżeń dotyczących metod tworzących serwisy. Metoda tworząca serwis typowy różni się listą argumentów od metody tworzącej serwis kompletowany na zamówienie:

klasa *Serwis typowy*:

*utwórz serwis (zamówienie, nr wzorca)*

gdzie *zamówienie* oznacza konkretny obiekt klasy *Zamówienie*, do którego ma być przypisany serwis, a *nr wzorca* pozwala na utworzenie obiektu klasy *Serwis typowy*, zgodnego ze wzorcem identyfikowanym za pomocą numeru

Zamiast podanej metody można zaproponować dwie inne:

*utwórz serwis (nr wzorca)*

*przypisz do zamówienia (zamówienie)*

klasa *Serwis kompletowany na zamówienie*:

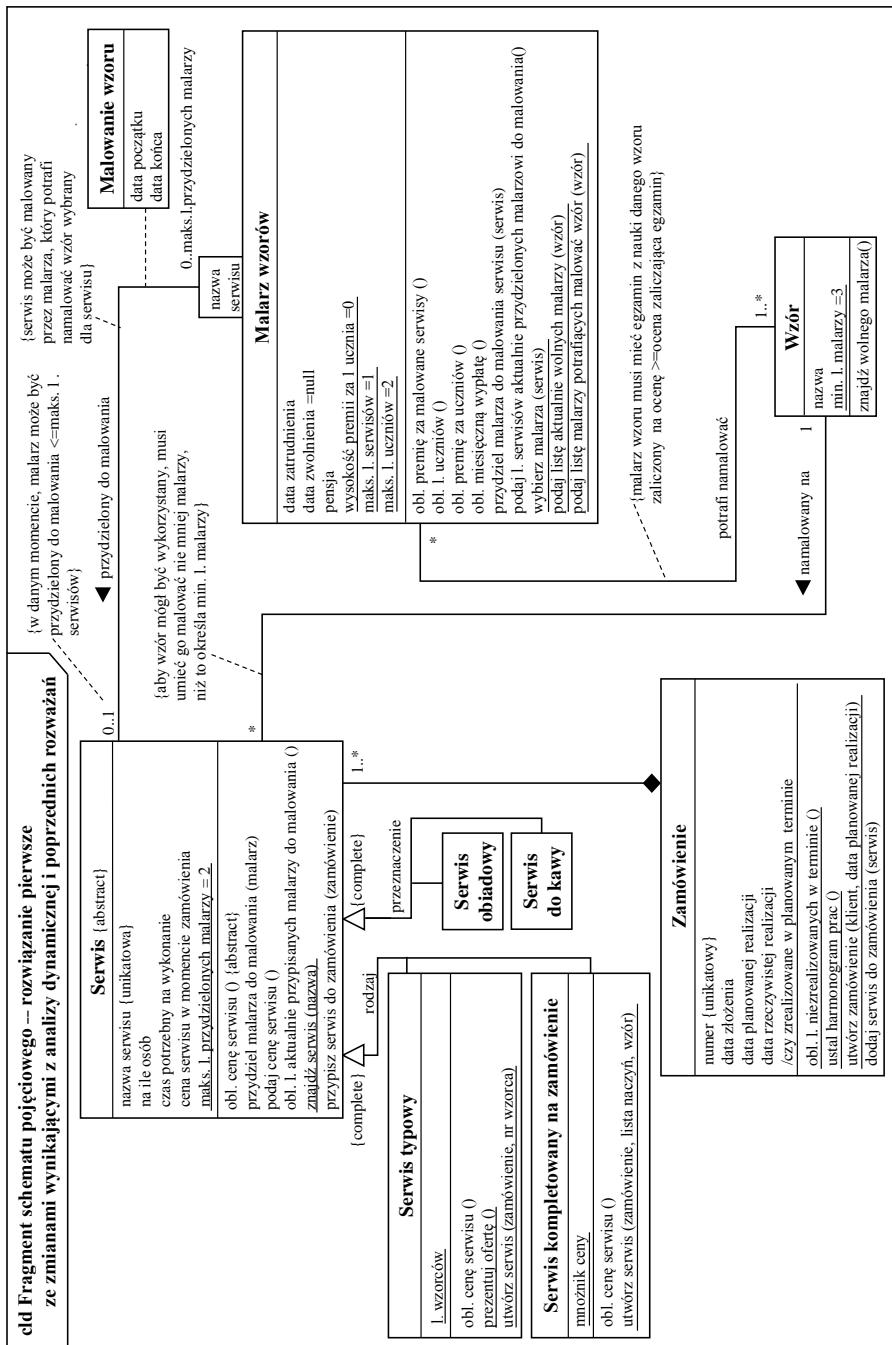
*utwórz serwis (zamówienie, lista naczyń, wzór)*

gdzie *zamówienie* jest konkretnym obiektem, a aktor musi podać listę naczyń wchodzących w skład serwisu oraz wzór, jaki chce mieć namalowany na zamawianym serwisie wraz z wyszczególnieniem listy egzemplarzy każdego z naczyń; dla serwisu typowego wzorzec serwisu, a więc rodzaj naczyń i ilość egzemplarzy wchodzących w jego skład oraz malowany wzór są z góry określone i nie podlegają wyborowi

Utworzenie nowego serwisu (typowego lub kompletowanego na zamówienie) musi być poprzedzone stworzeniem obiektu klasy *Zamówienie*. Tak więc w tej klasie powinna pojawić się metoda tworząca nowy obiekt *utwórz zamówienie (klient, data planowanej realizacji)*.

Aby mieć możliwość dołączenia serwisu utworzonego po zarejestrowaniu zamówienia, należy wprowadzić do schematu pojęciowego następujące metody obiektowe: *przypisz serwis do zamówienia (zamówienie)* w klasie *Serwis* oraz *dodaj serwis do zamówienia (serwis)* w klasie *Zamówienie*.

Na Rys. 3-19 przedstawiono fragment schematu pojęciowego wraz z ww. zmianami.



Rys. 3-19 Fragment schematu pojęciowego dla rozwiązywanie pierwszego z wprowadzonymi zmianami wynikającymi z przeprowadzonej analizy dynamicznej i z poprzednich rozważań

### 3.5 Zadania do samodzielnej pracy

Poniżej przedstawiono listę pytań i zadań do samodzielnego rozwiązania. Odpowiedzi należy uzasadniać w oparciu o tekst wymagań zamieszczony w 3.1.

1. Dla przypadku użycia związanego z obliczeniem miesięcznej wypłaty malarza zaproponuj scenariusz.
2. Dla przypadku użycia związanego z obliczeniem miesięcznej wypłaty malarza zaproponuj podział tego przypadku na podprzypadki.
3. Objaśnij różnicę między dziedziczeniem wielokrotnym a nierożłącznym. Czy któryś z tych dwóch rodzajów dziedziczenia wystąpił na schemacie pojęciowym? Jeśli tak, uzasadnij jego wykorzystanie. Zaproponuj wykorzystanie drugiego z wymienionych rodzajów dziedziczenia.
4. Podaj przykład metody abstrakcyjnej. W którym miejscu w hierarchii klas może być ona umieszczona? Czy może być metodą klasową?
5. Wyjaśnij korzyści wynikające z zastosowania polimorfizmu metod w oparciu o przykłady. Dla podanych przykładów zaproponuj rozwiązania alternatywne, w których nie wykorzystano polimorfizmu metod.
6. Podaj przykład ograniczenia, które odnosi się do dwóch asocjacji. Do jakiej metody i w jakiej klasie włożyłbyś implementację tego ograniczenia?
7. Skonstruuj diagram aktywności bazujący na scenariuszu dla przypadku użycia związanego z obliczeniem miesięcznej wypłaty malarza, wprowadzając aktywność złożoną.
8. Dla wybranego przypadku użycia narysuj taki fragment diagramu interakcji, który przedstawia interakcje pomiędzy obiektami co najmniej dwóch różnych klas.
9. Dla dowolnej klasy ze schematu pojęciowego dla rozwiązania pierwszego wskaż dwa stany (wyłączając stan początkowy i stan końcowy), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.
10. Ze schematu pojęciowego dla rozwiązania pierwszego wybierz klasę posiadającą przynajmniej dwie podklasy (w razie potrzeby dorysuj brakującą podklassę). Zamodeluj wybrany fragment diagramu w systemie relacyjnym na dwa sposoby.

### 3.6 Podsumowanie zadania

W Tab. 3-3 przedstawiono ogólną charakterystykę omawianego zadania. Szczegółową charakterystykę zadania dla grup tematycznych: *Wprowadzenie do przedmiotu*, *Analiza funkcjonalna*, *Analiza strukturalna*, *Analiza dynamiczna* zamieszczono odpowiednio w tabelach Tab. 3-4 – Tab. 3-7.

Tab. 3-3 Ogólna charakterystyka zadania

<b>Ogólna charakterystyka zadania</b>		
<i>Nazwa</i>	Wytwórnia porcelany stołowej	
<i>Dziedzina problemowa</i>	system może znaleźć zastosowanie w firmach produkcyjnych zajmujących się wytwarzaniem zastawy stołowej lub zestawów dowolnych przedmiotów użytkowych/ ozdobnych	
<i>Cel</i>	ułatwienie realizacji zamówień, obliczanie wynagrodzenia malarzy serwisów oraz promowanie firmy w Internecie	
<i>Zakres odpowiedzialności</i>	prezentowanie oferty, ustalenie harmonogramu prac związanych z malowaniem serwisów, wyliczenie miesięcznej wypłaty i premii dodatkowej malarza, wyszukanie najlepszego klienta, wyliczenie czasu, jaki w ostatnim miesiącu uczeń poświęcił na naukę, ustalenie liczby zamówień zrealizowanych po planowanym terminie	
<i>Stopień trudności</i>	trudne	
<i>Rozmiar</i>	13 klas	
<i>Użyteczność w grupach tematycznych</i>		
<i>Grupa tematyczna</i>	<i>Użyteczność w grupie tematycznej</i>	<i>Użyteczność na etapie zgodnym z modelem efektywnego nauczania</i>
<i>Wprowadzenie do przedmiotu</i>	niska	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> niska <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> niska
<i>Analiza funkcjonalna</i>	niska	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> niska <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> niska
<i>Analiza</i>	wysoka	<i>Przygotowanie do nauki:</i> niska

<i>strukturalna</i>		<i>Nauka:</i> bardzo wysoka <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> średnia
<i>Analiza dynamiczna</i>	niska	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> średnia <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> niska

Tab. 3-4 Szczegółowa charakterystyka zadania dla grupy tematycznej *Wprowadzenie do przedmiotu*

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>	<i>Wprowadzenie do przedmiotu</i>	
<b>Stopień trudności grupy w zadaniu</b>	średnio trudne	
<b>Poruszane tematy</b>	<i>klasyfikatory a instancje, mechanizmy rozszerzalności, reguły nazewnictwa</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>klasyfikatory a instancje</i>	trudne	<i>klasyfikator, instancja klasyfikatora – objaśnienie różnicy na przykładach</i>
<i>mechanizmy rozszerzalności</i>	łatwe	<i>stereotyp, ograniczenie, komentarz</i>
<i>reguły nazewnictwa</i>	łatwe	reguły nazewnictwa dla aktorów, przypadków użycia, klas, atrybutów, metod i asocjacji

Tab. 3-5 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza funkcjonalna*

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>	<i>Analiza funkcjonalna</i>	
<b>Stopień trudności grupy w zadaniu</b>	łatwe	
<b>Poruszane tematy</b>	<i>aktorzy, przypadki użycia, scenariusze, mechanizmy rozszerzalności</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>aktorzy</i>	łatwe	<i>aktor Podsystem czasu, hierarchia dla aktorów</i>
<i>przypadki użycia</i>	łatwe	relacje pomiędzy przypadkami użycia «include» i «extend»
<i>scenariusze</i>	łatwe	
<i>mechanizmy rozszerzalności</i>	łatwe	<i>stereotyp, komentarz</i>

Tab. 3-6 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza strukturalna*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Analiza strukturalna</i>
<b>Stopień trudności grupy w zadaniu</b>	trudne
<b>Poruszane tematy</b>	<i>klasa a obiekt, atrybuty klasy, metody klasy, struktury generalizacji-specjalizacji, klasa abstrakcyjna a klasa konkretna, asocjacje, analiza wartości, mechanizmy rozszerzalności</i>

<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>klasa a obiekt</i>	trudne	objaśnienie na przykładach różnicy między klasyfikatorem a instancją tego klasyfikatora, różne sposoby prezentowania i reprezentowania typowych egzemplarzy stanowiących ofertę firmy
<i>atrybuty klasy</i>	łatwe	<i>atrybut opcjonalny, powtarzalny, pochodny, klasowy</i>
<i>metody klasy</i>	łatwe	<i>metoda klasowa, metoda abstrakcyjna, metoda zaimplementowana, polimorfizm metod</i>
<i>struktury generalizacji-specjalizacji</i>	średnio trudne	<i>rodzaje dziedziczenia – wieloaspektowe, nierożłączne, kompletne, niekompletne, typu elipsa</i>
<i>asocjacje</i>	łatwe	<i>nazwa asocjacji a rola asocjacji, klasa asocjacji, asocjacja kwalifikowana, agregacja, kompozycja</i>
<i>analiza wartości</i>	łatwe	<i>wartość pochodna, wartość początkowa, wartość graniczna – szczegółowa analiza</i>
<i>mechanizmy rozszerzalności</i>	łatwe	<i>ograniczenie, komentarz ograniczenia nałożone na asocjacje</i>

Tab. 3-7 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza dynamiczna*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Analiza dynamiczna</i>
<b>Stopień trudności grupy w zadaniu</b>	łatwe
<b>Poruszane tematy</b>	<i>diagramy aktywności, podsumowanie analizy dynamicznej</i>

<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>diagramy aktywności</i>	łatwe	<i>aktywność złożona, romb decyzyjny – trzy notacje do modelowania rombu decyzyjnego</i>
<i>podsumowanie analizy dynamicznej</i>	łatwe	modyfikacja schematu pojęciowego o dodatkowe metody

## Spis treści rozdziału

Rozdział 4 Dom aukcyjny „ <i>Stare auta</i> ”.....	182
4.1 Tekst wymagań.....	182
4.2 Analiza funkcjonalna .....	183
4.2.1 Diagram kontekstowy .....	183
4.2.2 Model przypadków użycia.....	184
4.2.3 Scenariusze dla wybranych przypadków użycia łącznie z ich podziałem na podprzypadki.....	186
4.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	191
4.3 Analiza strukturalna.....	193
4.3.1 Schemat pojęciowy .....	193
4.3.2 Analiza elementów pochodnych .....	193
4.3.3 Analiza wartości początkowych .....	195
4.3.4 Analiza wartości granicznych.....	198
4.3.5 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji .....	198
4.3.6 Zadania podstawowe .....	201
4.3.7 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	204
4.3.8 Schemat relacyjny dla fragmentu schematu pojęciowego .....	204
4.4 Analiza dynamiczna .....	206
4.4.1 Zadania podstawowe .....	206
4.4.2 Omówienie zmian na schemacie projektowym wynikających z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań.....	210
4.5 Zadania do samodzielnej pracy .....	211
4.6 Podsumowanie zadania.....	212

## Rozdział 4

# Dom aukcyjny „*Stare auta*”

Ewa Stemposz, Andrzej Jodłowski

### 4.1 Tekst wymagań

Dom aukcyjny planuje budowę serwisu internetowego umożliwiającego przeprowadzanie transakcji kupna/sprzedaży starych aut za pośrednictwem Internetu.

1. Zostały wyróżnione dwie grupy potencjalnych użytkowników serwisu: osoby sprzedające oraz osoby kupujące stare auta. Dany użytkownik może wystąpić w obu tych rolach.
2. Dla każdej z osób mają być przechowywane następujące informacje: dane osobowe, login, hasło oraz email. Ma być pamiętane, czy dane osobowe zostały już potwierdzone za pośrednictwem poczty.
3. Pojedyncza aukcja (dotycząca wyłącznie jednego auta) ma być opisywana przez następujące informacje: unikatowa nazwa, opis, dane osoby sprzedającej auto oraz obie daty – rozpoczęcia i zakończenia aukcji.
4. Aukcje dzielą się na dwa rodzaje: aukcja typu *Licytacja* i aukcja typu *Kup-teraz*. Dana aukcja może być zarówno typu *Licytacja*, jak i typu *Kup-teraz*.
5. Dla aukcji typu *Licytacja* ma być przechowywana cena początkowa (od jakiej rozpocznie się licytacja) oraz opcjonalnie cena minimalna (za jaką osoba sprzedająca zgodzi się sprzedać auto).
6. Ma być przechowywana historia ofert składanych przez wszystkie osoby biorące udział w danej licytacji – dla każdej osoby ma być przechowywana wyłącznie ostatnia oferta. Każda kolejna oferta nie może być niższa niż cena początkowa określona przez osobę sprzedającą i powinna być wyższa od oferty poprzedniej. Ponadto, w ramach aukcji typu *Licytacja* każdemu kupującemu przypisywany jest unikatowy identyfikator.
7. Jeśli na aukcji typu *Licytacja* wylicytowana cena jest niższa niż minimalna, to osoba sprzedająca może auto nie sprzedać. Jeśli wylicytowana cena jest wyższa niż cena minimalna, kupujący musi auto kupić.
8. Dla aukcji typu *Kup-teraz* ma być przechowywana cena, za którą auto można kupić od razu, bez licytowania. Osoby z niepotwierdzonymi danymi osobowymi nie mogą brać udziału w tego rodzaju aukcjach.

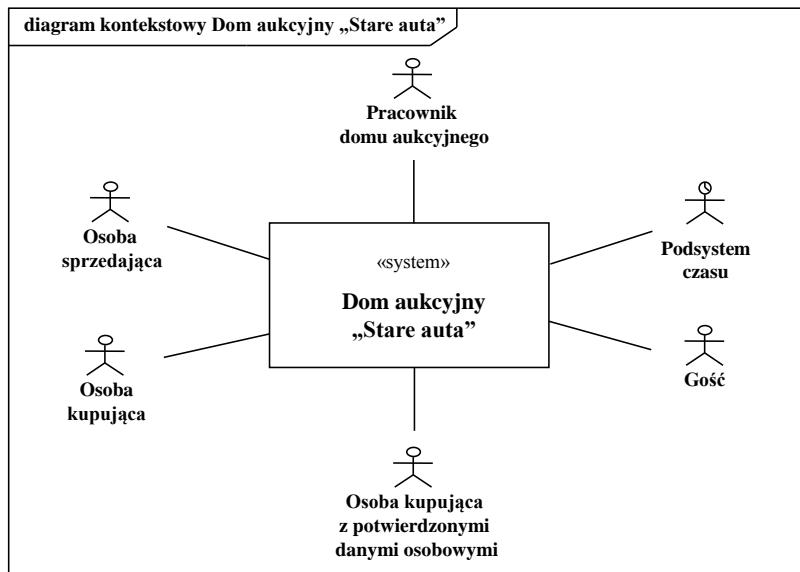
9. Dla każdego auta mają być przechowywane następujące informacje: rodzaj, marka, rok produkcji, wiek (aktualnie minimum 20 lat), przebieg (w km), stan techniczny, czy auto jest sprawne (czy też uszkodzone), lista dodatkowego wyposażenia (o ile je posiada), dodatkowe informacje (ewentualnie) oraz zdjęcie.
10. Auta dzielone są według dwóch kryteriów: pierwsze kryterium stanowi przedział wiekowy (20-30 lat itd.), a drugie – sprawność auta (auto uszkodzone lub auto sprawne). Oba podziały są rozłączne i kompletne. Dla aut uszkodzonych ma być przechowywana lista uszkodzeń.
11. Auto niesprzedane na jednej aukcji (bo np. nie zgłosił się żaden kupujący), może być wystawione do sprzedaży na innej – przez tego samego sprzedającego.
12. Po pięciu latach od zakończenia ostatniej aukcji dla danego auta, mają być usuwane informacje o nim i o wszystkich aukcjach, na jakich było wystawiane.
13. System powinien wspomagać m.in. realizację poniższych funkcjonalności:
  - 13.1 podawanie opisu aut (informacje, takie jak określono w punkcie 9) – dla aut uszkodzonych ma być dodatkowo dołączana lista uszkodzeń – każdy użytkownik, także niezarejestrowany (gość);
  - 13.2 przeglądanie aktualnie trwających aukcji (każdy użytkownik, w tym gość), z możliwością złożenia oferty na licytacji lub udziału w aukcji typu *Kup-teraz* – osoba kupująca;
  - 13.3 utworzenie aukcji, z ewentualnym wprowadzeniem opisu auta wystawianego na sprzedaż (dla aut wystawianych po raz pierwszy) – osoba sprzedająca;
  - 13.4 złożenie oferty na licytacji – osoba kupująca;
  - 13.5 zakup auta dla aukcji typu *Kup-teraz* – osoba kupującą;
  - 13.6 podanie listy aut, które nie zostały sprzedane na aukcjach przeprowadzanych w zadanym okresie – pracownik domu aukcyjnego.

## 4.2 Analiza funkcjonalna

### 4.2.1 Diagram kontekstowy

*Polecenie:* Zbuduj diagram kontekstowy w oparciu o tekst wymagań zamieszczony w punkcie 4.1 .

Diagram kontekstowy, przedstawiający aktorów wybranych dla systemu *Dom aukcyjny „Stare auta”*, w oparciu o analizę tekstu wymagań, został zaprezentowany na Rys. 4-1.

Rys. 4-1 Diagram kontekstowy dla systemu *Dom aukcyjny „Stare auta”*

Na diagramie umieszczone aktorów wskazanych *explicite* w tekście wymagań (m.in. w punkcie 13 tekstu wymagań).

Aktor *Osoba kupująca* oznacza tu zarówno osobę z potwierdzonymi danymi osobowymi, jak i z niepotwierdzonymi.

## 4.2.2 Model przypadków użycia

*Polecenie:* Zbuduj model przypadków użycia, uwzględniając między innymi funkcjonalność sugerowaną w ostatnim punkcie tekstu wymagań (tzn. w punkcie 13).

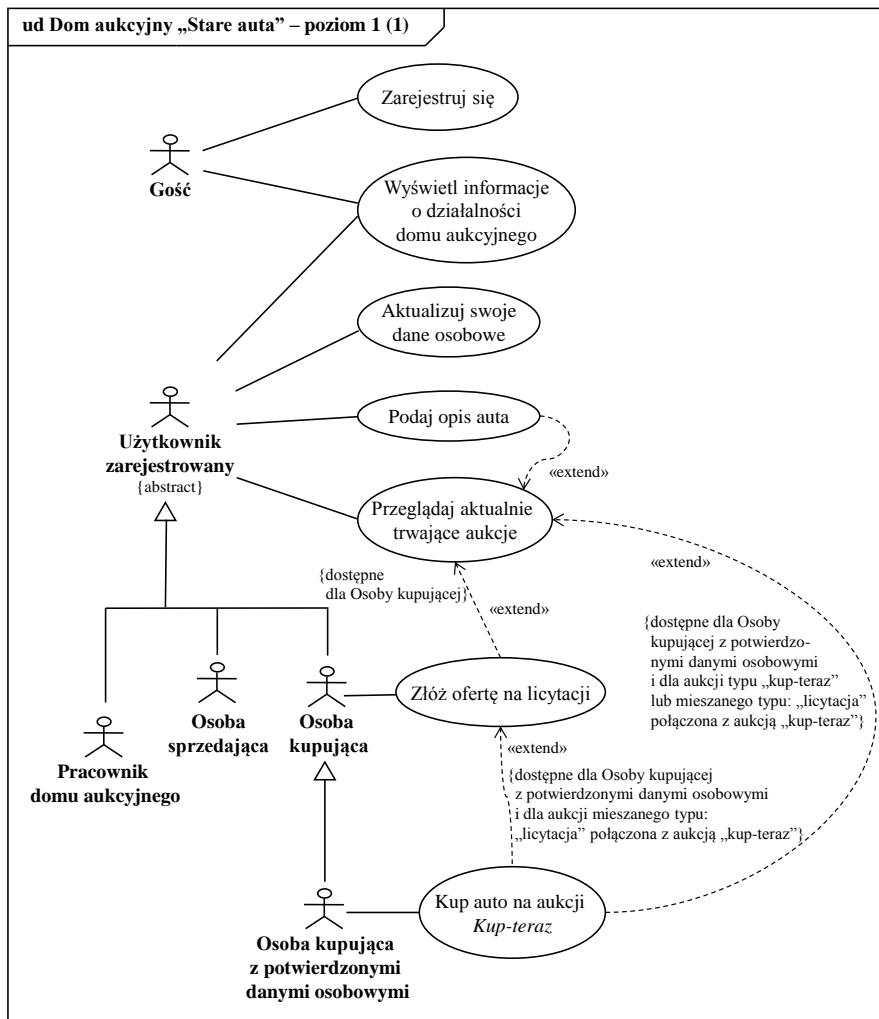
*Uwaga:* Model przypadków użycia należy skonstruować z perspektywy aktorów z otoczenia systemu (najwyższy poziom abstrakcji modelu), biorąc pod uwagę możliwą ewolucję systemu w przyszłości. Należy uwzględnić hierarchie aktorów i relacje pomiędzy przypadkami (o ile mają/mogłyby mieć miejsce).

Model przypadków użycia [36, 41, 42, 45] przedstawiono za pomocą dwóch diagramów, dzięki czemu jest bardziej czytelny. Na tym etapie analizy pominięto przypadki użycia typu CRUD, służące wypełnieniu bazy danych, aby system mógł zacząć pracować.

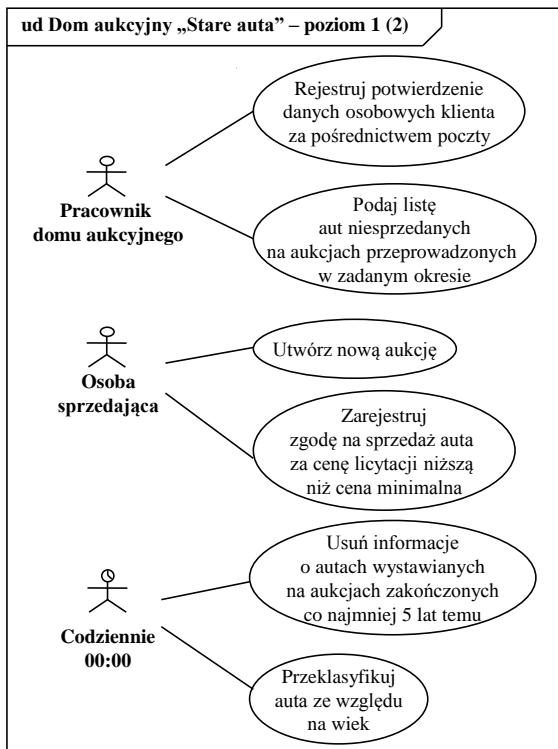
Rys. 4-2 (diagram pierwszy) zawiera aktora *Gość* i hierarchię aktorów dla użytkowników zarejestrowanych, wraz z przypadkami użycia dziedzicznymi w wyniku istnienia tej hierarchii. Aktor *Użytkownik zarejestrowany* oznaczać będzie

dowolną osobę korzystającą z systemu *Dom aukcyjny „Stare auta”*, oprócz użytkownika niezarejestrowanego (gościa).

Diagram drugi (Rys. 4-3) obejmuje przypadki użycia przypisane następującym aktorom: *Osoba sprzedająca*, *Pracownik domu aukcyjnego*, *Codziennie 00:00*. Warto zwrócić uwagę na fakt, że aktor *Podsystem czasu* umieszczony na diagramie kontekstowym jest pojęciem ogólnym, które oznaczać może więcej niż jednego aktora z mniej lub bardziej szczegółowym określeniem momentu wywołania danego przypadku, i na dalszym etapie analizy należy to sprecyzować.



Rys. 4-2 Model przypadków użycia (diagram pierwszy)



Rys. 4-3 Model przypadków użycia (diagram drugi)

#### 4.2.3 Scenariusze dla wybranych przypadków użycia łącznie z ich podziałem na podprzypadki

*Polecenie:* Dla przypadku użycia *Kup auto na aukcji Kup-teraz* napisz scenariusz.

Przykładowy scenariusz przypadku użycia został przedstawiony w Tab. 4-1.

Tab. 4-1 Scenariusz dla przypadku użycia *Kup auto na aukcji Kup-teraz*

<b>Nazwa przypadku użycia</b>	Kup auto na aukcji <i>Kup-teraz</i>
<b>Warunek początkowy</b>	Co najmniej jedna aukcja <i>Kup-teraz</i> została zarejestrowana.
<b>Główny przepływ zdarzeń</b>	1. Przypadek użycia jest wywoływany przez aktora <i>Osoba kupująca z potwierdzonymi danymi</i>

	<p><i>osobowymi</i> albo przypadek użycia <i>Złoż ofertę na licytacji</i>, albo przypadek użycia <i>Przeglądaj aktualnie trwające aukcje</i>.</p> <ol style="list-style-type: none"> <li>2. System wyświetla formularz z kryteriami wyszukiwania aut na aukcjach <i>Kup-teraz</i>. Aktor wprowadza kryteria.</li> <li>3. System wyświetla listę aukcji spełniających zadane kryteria i umożliwia wybór jednej z nich. Aktor wybiera aukcję.</li> <li>4. System wyświetla dane aukcji oraz wystawionego na niej auta i prosi o potwierdzenie zakupu auta na aukcji. Aktor potwierdza.</li> <li>5. System wyświetla komunikat o zakupie auta na aukcji <i>Kup-teraz</i>.</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	<p>2a. Aktorem jest przypadek użycia <i>Złoż ofertę na licytacji</i> lub przypadek użycia <i>Przeglądaj aktualnie trwające aukcje</i>, aukcja jest albo typu kup-teraz, albo mieszana, osoba kupująca ma potwierdzone dane osobowe i jeśli aukcja jest mieszana, to aktualnie najwyższa oferta jest mniejsza niż cena kup-teraz. System przechodzi do punktu 4.</p> <p>2aa. Aktorem jest przypadek użycia <i>Złoż ofertę na licytacji</i> lub przypadek użycia <i>Przeglądaj aktualnie trwające aukcje</i>, aukcja jest typu licytacja, system informuje o niemożności zakupu kup-teraz na aukcji typu licytacja i kończy przypadek użycia.</p> <p>2ab. Aktorem jest przypadek użycia <i>Złoż ofertę na licytacji</i> lub przypadek użycia <i>Przeglądaj aktualnie trwające aukcje</i>, aukcja jest typu kup-teraz albo mieszana, a osoba kupująca nie ma potwierdzonych danych osobowych, system informuje o niemożności zakupu auta z powodu braku potwierdzonych danych osobowych i kończy przypadek użycia.</p> <p>2ac. Aktorem jest przypadek użycia <i>Złoż ofertę na licytacji</i> lub przypadek użycia <i>Przeglądaj aktualnie trwające aukcje</i>, aukcja jest typu mieszana, a <i>aktualnie najwyższa oferta</i> jest większa lub równa <i>cenie kup-teraz</i>. System wyświetla informację o niemożności zakupu</p>

	<p>auta w cenie kup-teraz i kończy przypadek użycia.</p> <p>2b. Osoba kupująca rezygnuje z wprowadzenia kryteriów wyszukiwania aut na aukcjach <i>Kup-teraz</i>. System kończy przypadek użycia.</p> <p>3a. Lista aukcji spełniających zadane kryteria jest pusta. System wyświetla odpowiedni komunikat i pyta aktora, czy kontynuować wybór aukcji w oparciu o inne kryteria.</p> <p>3aa. Osoba kupująca decyduje się na zmianę kryteriów wyszukiwania. System wraca do punktu 2.</p> <p>3ab. Osoba kupująca rezygnuje z kontynuacji wyszukiwania. System kończy przypadek użycia.</p> <p>3b. Osoba kupująca rezygnuje z wyboru aukcji z listy wyszukanych aukcji, system kończy przypadek użycia.</p> <p>4a. Osoba kupująca rezygnuje z zakupu auta, system kończy przypadek użycia.</p>
<b>Zakończenie</b>	Zgodnie ze scenariuszem.
<b>Warunek końcowy</b>	O ile zostały wypełnione warunki wyspecyfikowane w treści scenariusza, zostaje zarejestrowana sprzedaż auta na aukcji <i>Kup-teraz</i> .

*Uwaga 1.* Możliwe byłoby sformułowanie innego warunku początkowego dla scenariusza w Tab. 4-1 w postaci następującej: „została zarejestrowana co najmniej jedna aukcja *Kup-teraz*, trwająca aktualnie”.

*Uwaga 2.* Inna wersja sekcji *Alternatywne przepływy zdarzeń* dla scenariusza umieszczonego w Tab. 4-1 mogłaby zostać określona jak w Tab. 4-2.

Tab. 4-2 Druga wersja sekcji *Alternatywne przepływy zdarzeń* dla scenariusza z Tab. 4-1

<b>Alternatywne przepływy zdarzeń</b>	<p>2a. Aktorem jest przypadek użycia <i>Złoż ofertę na licytacji</i> lub przypadek użycia <i>Przeglądaj aktualnie trwające aukcje</i>.</p> <p>2aa Aukcja jest albo typu kup-teraz albo mieszana, osoba kupująca ma potwierdzone dane osobowe i jeśli aukcja jest mieszana, to aktualnie</p>
---------------------------------------	---

	<p>najwyższa oferta jest mniejsza niż cena kup-teraz. System przechodzi do punktu 4.</p> <p>2ab. Aktorem jest przypadek użycia <i>Złóż ofertę na licytacji</i> lub przypadek użycia <i>Przeglądaj aktualnie trwające aukcje</i>, aukcja jest typu licytacja, system informuje o niemożności zakupu kup-teraz na aukcji typu licytacja i kończy przypadek użycia.</p> <p>2ac. Aktorem jest przypadek użycia <i>Złóż ofertę na licytacji</i> lub przypadek użycia <i>Przeglądaj aktualnie trwające aukcje</i>, aukcja jest typu kup-teraz albo mieszana, a osoba kupująca nie ma potwierdzonych danych osobowych, system informuje o niemożności zakupu auta z powodu braku potwierdzonych danych osobowych i kończy przypadek użycia.</p> <p>2ad. Aktorem jest przypadek użycia <i>Złóż ofertę na licytacji</i> lub przypadek użycia <i>Przeglądaj aktualnie trwające aukcje</i>, aukcja jest typu mieszana, a <i>aktualnie najwyższa oferta</i> jest większa lub równa <i>cenie kup-teraz</i>. System wyświetla informację o niemożności zakupu auta w cenie kup-teraz i kończy przypadek użycia.</p> <p>2b. Osoba kupująca rezygnuje z wprowadzenia kryteriów wyszukiwania aut na aukcjach <i>Kup-teraz</i>. System kończy przypadek użycia.</p> <p>3a. Lista aukcji spełniających zadane kryteria jest pusta. System wyświetla odpowiedni komunikat i pyta aktora, czy kontynuować wybór aukcji w oparciu o inne kryteria.</p> <p>3aa. Osoba kupująca decyduje się na zmianę kryteriów wyszukiwania. System wraca do punktu 2.</p> <p>3ab. Osoba kupująca rezygnuje z kontynuowania wyszukiwania. System kończy przypadek użycia.</p> <p>3b. Osoba kupująca rezygnuje z wyboru aukcji z listy wyszukanych aukcji, system kończy przypadek użycia.</p> <p>4a. Osoba kupująca rezygnuje z zakupu auta, system kończy przypadek użycia.</p>
--	--

*Uwaga 3.* Podaj opis auta jest jednym z przypadków użycia, dlatego punkt 4 scenariusza mógłby zawierać wywołanie podprzepływu nazwanego *Podaj opis auta*. W takiej sytuacji należałoby dołączyć scenariusz tego podprzepływu do scenariusza zawartego w Tab. 4-1.

Dalsza analiza będzie prowadzona dla scenariusza określonego jak w Tab. 4-1.

*Polecenie:* Dla przypadku użycia *Utwórz nową aukcję*, z ewentualnym wprowadzeniem opisu auta (punkt 13.3 tekstu wymagań):

- napisz scenariusz;
- zaproponuj podział tego przypadku na podprzypadki.

Przykładowy scenariusz wybranego przypadku użycia został określony w Tab. 4-3.

Tab. 4-3 Scenariusz dla przypadku użycia *Utwórz nową aukcję*

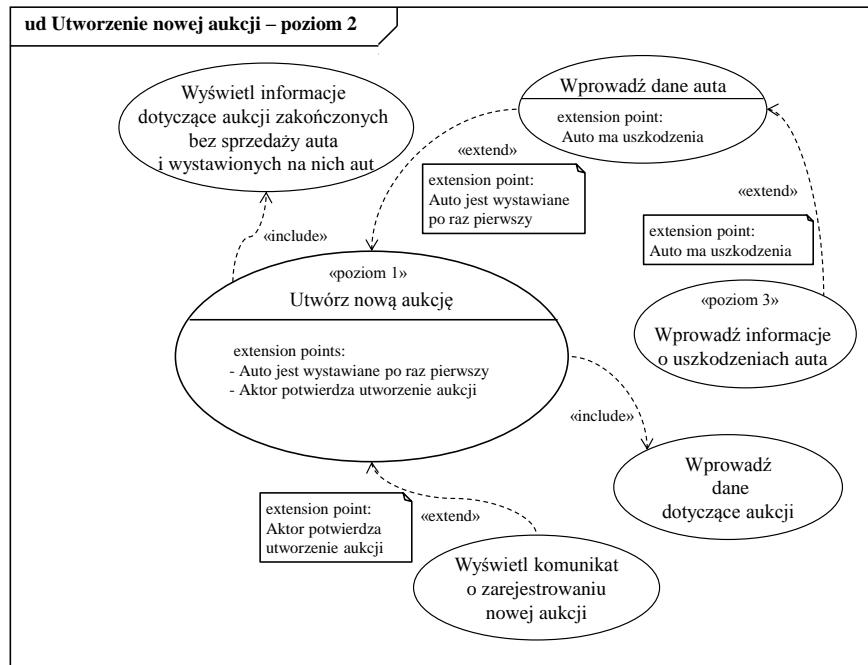
<b>Nazwa przypadku użycia</b>	Utwórz nową aukcję
<b>Warunek początkowy</b>	Brak
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Przypadek użycia jest wywoływany przez aktora <i>Osoba sprzedająca</i>.</li> <li>2. System pyta o rodzaj aukcji. Aktor wybiera rodzaj.</li> <li>3. System prosi o wprowadzenie danych aukcji. Aktor wprowadza dane: nazwę, opis, datę rozpoczęcia i datę zakończenia (z możliwym wykorzystaniem kalendarza), opis, a dodatkowo dla licytacji – cenę początkową i opcjonalnie cenę minimalną, zaś dla aukcji <i>Kup-teraz</i> – cenę zakupu.</li> <li>4. System wyświetla listę aukcji aktora wraz z danymi niesprzedanych aut i prosi o wybór jednego z nich. Aktor wybiera auto z listy niesprzedanych aut.</li> <li>5. System prosi o zatwierdzenie danych nowej aukcji. Aktor potwierdza dane.</li> <li>6. System wyświetla komunikat o zarejestrowaniu nowej aukcji.</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	<ol style="list-style-type: none"> <li>3a. Aktor wprowadził niepoprawne dane, np. nazwę, która nie jest unikatowa, datę zakończenia wcześniejszą niż data rozpoczęcia, cenę minimalną mniejszą niż cena początkowa itp. System wyświetla komunikat o błędzie i wraca do punktu 3.</li> <li>4a. Aktor zrezygnował z wyboru auta z listy aut</li> </ol>

	<p>niesprzedanych, system przechodzi do punktu 4c1.</p> <p>4b. W systemie brak aukcji aktora, które zostały zakończone bez sprzedaży auta, system przechodzi do punktu 4c1.</p> <p>4c1. System wyświetla formularz rejestracyjny dla nowego auta. Aktor wprowadza dane auta: rodzaj, markę, rok produkcji, przebieg, stan techniczny, opcjonalnie dodatkowe wyposażenie i dodatkowe informacje, zdjęcie.</p> <p>4c2. System odpytuje, czy auto jest sprawne. Aktor potwierdza</p> <p>4c2a. Auto jest uszkodzone, system prosi o wprowadzenie listy uszkodzeń. Aktor wprowadza dane.</p> <p>5a. Aktor nie potwierdza danych aukcji, system kończy przypadek użycia.</p>
<b>Zakończenie</b>	Zgodnie ze scenariuszem.
<b>Warunek końcowy</b>	Zostaje zarejestrowana nowa aukcja i ewentualnie nowe auto.

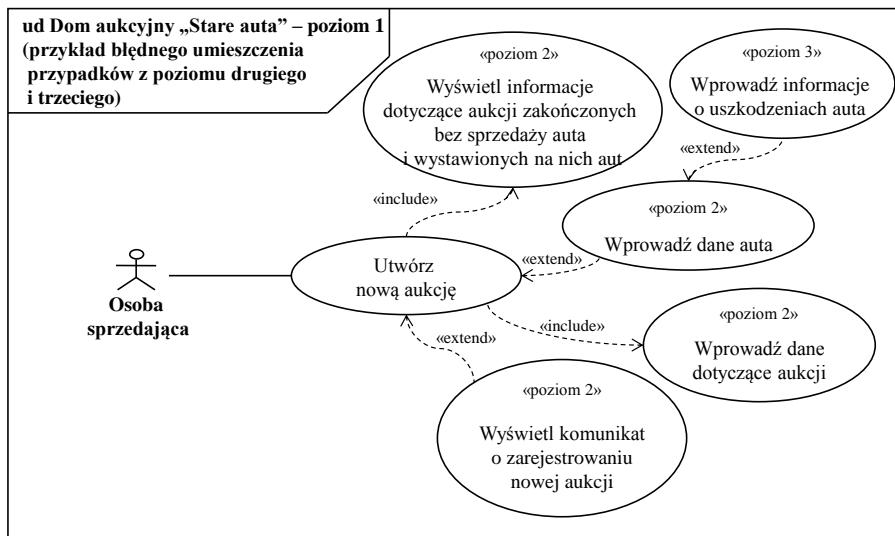
Diagram z przykładowym podziałem przypadku *Utwórz nową aukcję* zilustrowano na Rys. 4-4.

#### 4.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

- Nazwa aktora *Dom aukcyjny* jest zbyt ogólna i nie sugeruje potencjalnego użytkownika, ale nazwę dla tworzonego systemu, bądź też nazwę instytucji, dla której system jest tworzony. Bardziej odpowiednią nazwą byłaby *Pracownik domu aukcyjnego*.
- Niejasne (lub w ogóle błędne) nazwy przypadków, jak np.: *Dodatkowe dane*, *Lista uszkodzeń*. Są to nazwy danych, a nie zadań, jakie ma wykonać system na polecenie aktora.
- Błądem jest umieszczenie na diagramie, z wyższego poziomu abstrakcji modelu, przypadków z poziomu niższego, np. umieszczenie wraz z przypadkami użycia z poziomu pierwszego, czyli tymi wywoływanymi bezpośrednio przez aktorów z otoczenia systemu, podprzypadków powstacych z podziału przypadku *Utwórz nową aukcję*, jak na Rys. 4-5.



Rys. 4-4 Diagram z przykładowym podziałem przypadku *Utwórz nową aukcję*



Rys. 4-5 Przykład błędnego umieszczenia przypadków z niższego poziomu abstrakcji

- Zdarza się błędne przypisywanie aktorom przypadków użycia, np. przypadek użycia *Utwórz listę aut niesprzedanych na aukcjach przeprowadzonych w zadanym okresie* nie może być usługą wywoływaną przez aktora *Podsystem czasu*, ponieważ:
  - (1) w tekście wymagań wyraźnie jest przypisany do niej aktor *Pracownik domu aukcyjnego*;
  - (2) „zadany okres” sugeruje wprowadzanie/określenie tego okresu przez aktora.
- Błędny wybór aktora (lub błędna nazwa aktora), np. *Podsystem rozliczania aukcji*, który nie jest aktorem z otoczenia systemu, chyba że byłby odrębnym systemem, ale wtedy nie mógłby być podsystemem.

## 4.3 Analiza strukturalna

### 4.3.1 Schemat pojęciowy

*Polecenie:* Dla podanego tekstu wymagań skonstruuj schemat pojęciowy (nie redukując liczności).

Schemat pojęciowy [35, 41, 43] dla systemu *Dom aukcyjny „Stare auta”* przedstawiono na Rys. 4-6.

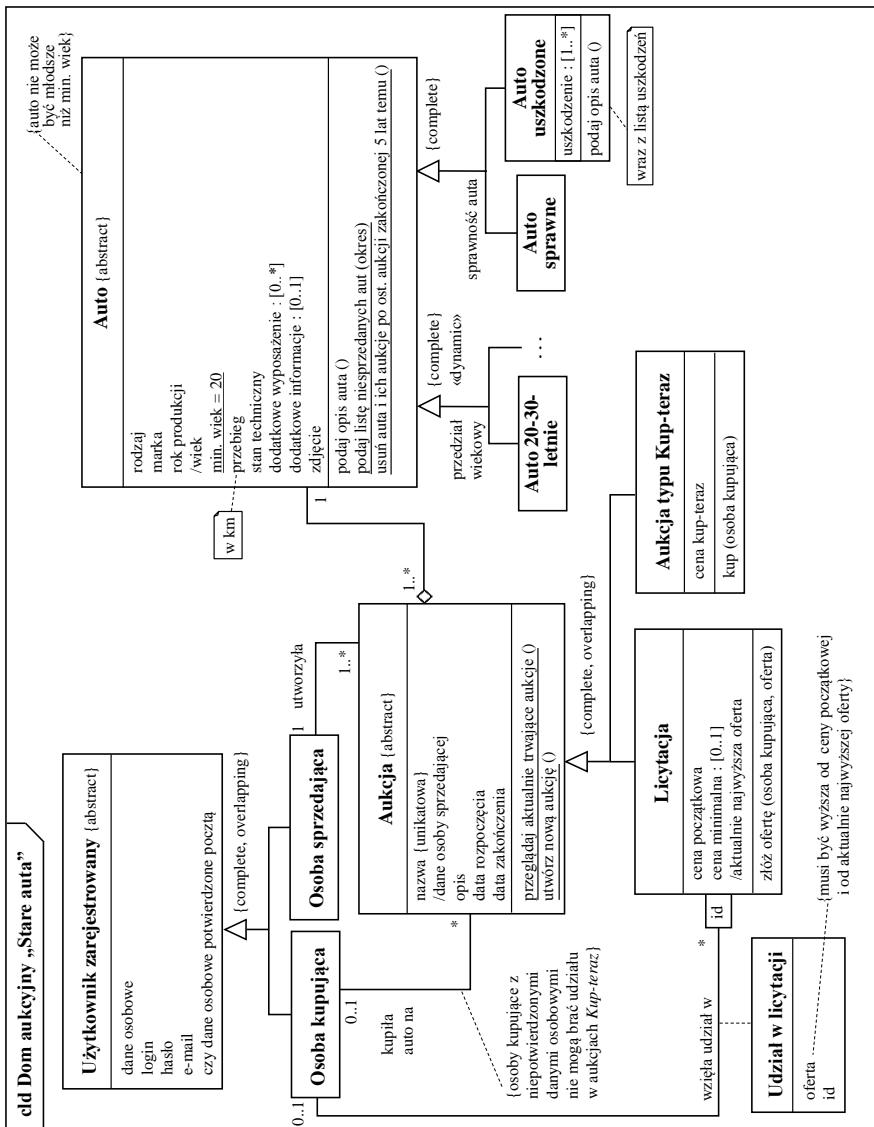
### 4.3.2 Analiza elementów pochodnych

Na schemacie pojęciowym na Rys. 4-6 można zidentyfikować trzy atrybuty pochodne *dane osoby sprzedającej* w klasie *Aukcja*, *aktualnie najwyższa oferta* w klasie *Licytacja* oraz *wiek* w klasie *Auto*.

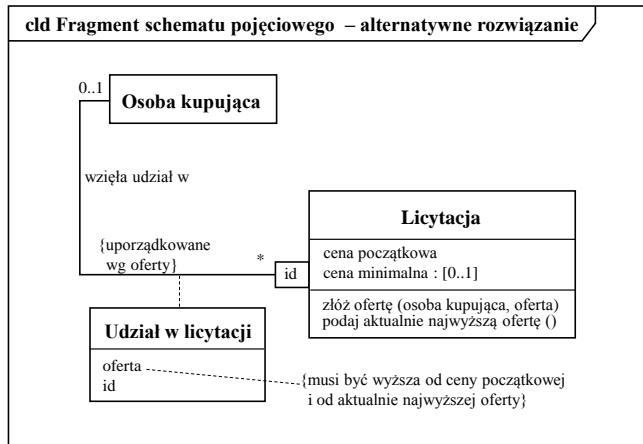
Atrybut *dane osoby sprzedającej* w klasie *Aukcja* zawiera dane personalne osoby sprzedającej. Ponieważ każda aukcja jest powiązana z osobą, która ją utworzyła w celu sprzedaży auta, to łatwo jest odzyskać jej dane osobowe, nie wydaje się więc być potrzebne przechowywanie kopii tych danych w obiektach *Aukcja*. Do schematu pojęciowego zostanie zatem wprowadzona w klasie *Aukcja* tylko metoda *podaj dane osoby sprzedającej()*.

Atrybut *aktualnie najwyższa oferta* w klasie *Licytacja* jest wyliczany na podstawie najwyższej oferty złożonej na danej licytacji. Wartość tego atrybutu ulega zmianie wraz ze złożeniem każdej nowej oferty na licytacji i jest niezbędna do prawidłowego przeprowadzenia procesu licytacji. W czasie „między ofertami”, tj. od momentu przyjęcia poprzednio zaakceptowanej oferty aż do momentu przyjęcia nowej oferty, wartość atrybutu nie ulega zmianie. Wydaje się, że wartość atrybutu *aktualnie najwyższa oferta* warto jest dodatkowo przechowywać i dlatego powinien on pozostać atrybutem pochodnym. Możliwe jest także inne rozwiązanie. Gdyby uporządkować zbiór ofert złożonych w ramach danej aukcji według ich wysokości

lub według kolejności ich składania i wprowadzić metodę *podaj aktualnie najwyższą ofertę* (), zwracającą wartość najwyższej oferty, to wówczas dostęp do tej informacji byłby bardzo szybki i można by zrezygnować z przechowywania wyliczonej wartości pochodnej. Odpowiedni fragment diagramu klas przedstawia Rys. 4-7. To alternatywne rozwiązanie nie będzie jednak uwzględniane w dalszej analizie.



Rys. 4-6 Schemat pojęciowy dla systemu *Dom aukcyjny „Stare auto”*



Rys. 4-7 Fragment schematu pojęciowego bez sugestii dodatkowego przechowywania wartości pochodnej – alternatywne rozwiązanie

Atrybut *wiek* w klasie *Auto* jest wyliczany na podstawie roku produkcji auta. Wartość atrybutu zmienia się rzadko, na początku lub przy końcu roku i jest wykorzystywana tylko do klasyfikacji aut. Nie wydaje się zatem celowe dodatkowe przechowywanie wartości atrybutu pochodnego *wiek* w obiektach *Auto*. Do schematu pojęciowego powinna być wprowadzona do klasy *Auto* metoda *podaj wiek()*. Mimo że oba rozwiązania nie wydają się ani kosztowne obliczeniowo, ani implementacyjnie złożone, to dzięki wykorzystaniu zaproponowanej metody automatyczna aktualizacja klasyfikacji aut będzie łatwiejsza w implementacji.

Schemat pojęciowy uwzględniający omówioną analizę elementów pochodnych został przedstawiony na Rys. 4-8.

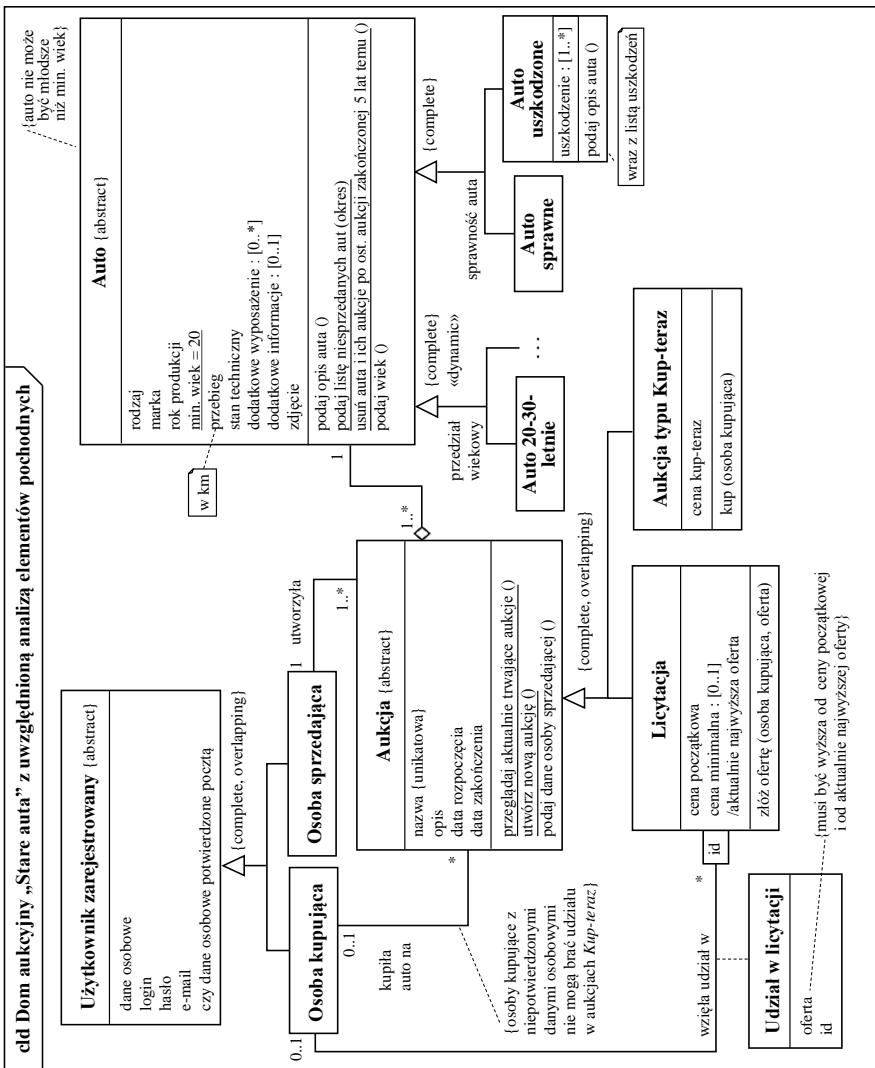
### 4.3.3 Analiza wartości początkowych

Analizując schemat pojęciowy z Rys. 4-8, zawsze warto rozważyć atrybuty, dla których można określić ich domyślne wartości początkowe, a także metody z parametrami domyślnymi.

Na Rys. 4-8 nie ma metod z określonymi parametrami domyślnymi. Gdyby dla funkcjonalności opisanej w punkcie 13.6 tekstu wymagań okresem domyślnym zostało przyjęte pięć ostatnich lat, to specyfikacja metody w klasie *Auto* podającej listę aut niesprzedanych w zadanym okresie mogłaby być następująca:

podaj listę niesprzedanych aut (okres = pięć ostatnich lat).

Lista atrybutów z określonymi wartościami początkowymi została przedstawiona w Tab. 4-4. Schemat pojęciowy uwzględniający zaproponowane w Tab. 4-4 wartości początkowe został zaprezentowany na Rys. 4-9.

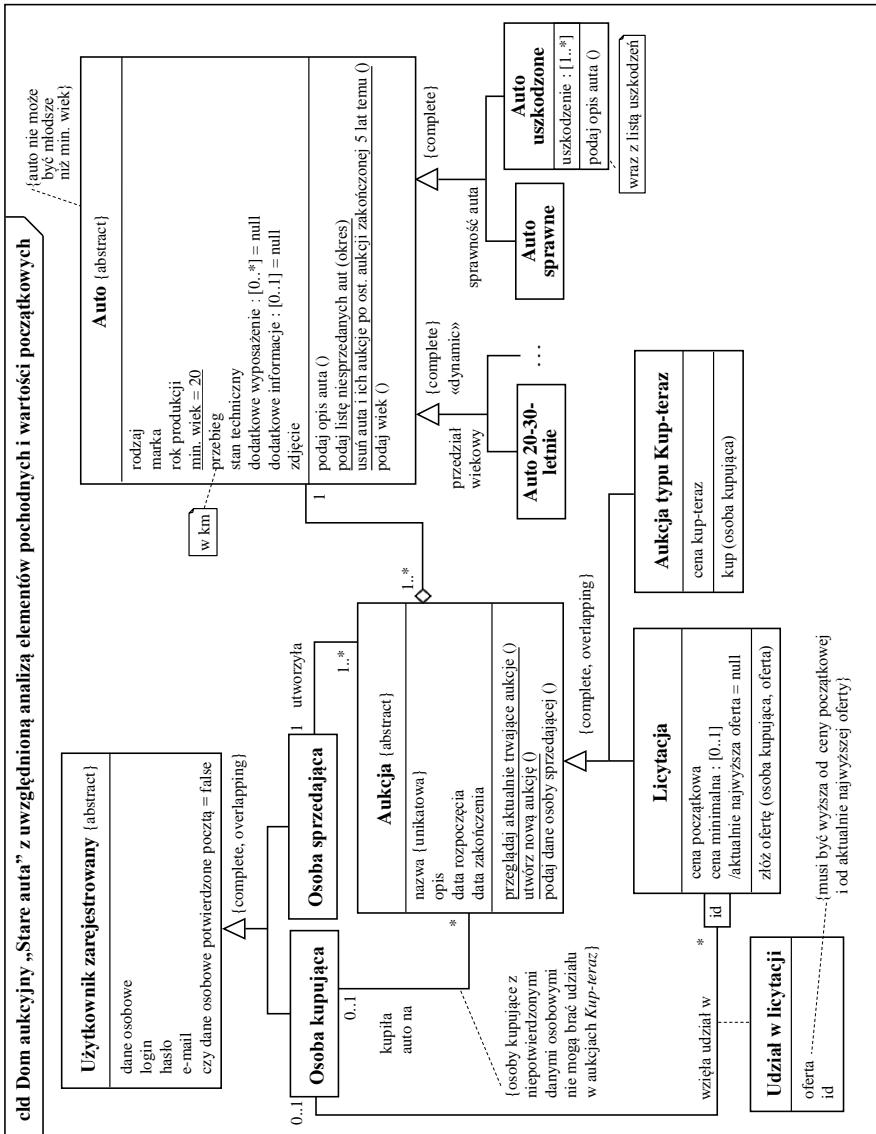


Rys. 4-8 Schemat pojęciowy z uwzględnieniem analizy elementów pochodnych

Tab. 4-4 Lista atrybutów z określonymi wartościami początkowymi

Klasa	Atrybut	Wartość początkowa (domyślna)
Auto	dodatkowe wyposażenie	null
	dodatkowe informacje	null

<i>Użytkownik zarejestrowany</i>	<i>czy dane osobowe potwierdzone pocztą</i>	<i>false</i>
<i>Licytacja</i>	<i>/aktualnie najwyższa oferta</i>	<i>null</i>



Rys. 4-9 Schemat pojęciowy z uwzględnieniem analizy elementów pochodnych i wartości początkowych

#### 4.3.4 Analiza wartości granicznych

Na schemacie pojęciowym na Rys. 4-9 można zidentyfikować kilka wartości granicznych.

W klasie *Auto* znajduje się metoda *usuń auta i ich aukcje po ost. aukcji zakończonej 5 lat temu ()*. Może się zdarzyć w przyszłości, że np. z powodu dużej liczby aukcji i aut, zaistnieje potrzeba wcześniejszego usuwania „starych” aukcji oraz aut z tych aukcji. Wydaje się zasadne wprowadzenie w klasie *Auto* dodatkowego atrybutu *maks. okres przechowywania bez wystawiania na aukcjach = 5* wraz ze zmianą specyfikacji metody *usuń auta historyczne i ich aukcje ()*, przyjmując, że aukcje historyczne to aukcje dotyczące auta, którego ostatnia aukcja zakończyła się więcej lat temu niż *maks. okres przechowywania bez wystawiania na aukcjach*.

Możliwe jest także inne rozwiązanie, polegające na wprowadzeniu metody z parametrem *okres usuń auta i ich aukcje z aukcji historycznych (okres)* w klasie *Auto* (w tym przypadku atrybut *maks. okres przechowywania bez wystawiania na aukcjach* oraz metoda *usuń auta i ich aukcje po ost. aukcji zakończonej 5 lat temu ()* byłyby zbędne).

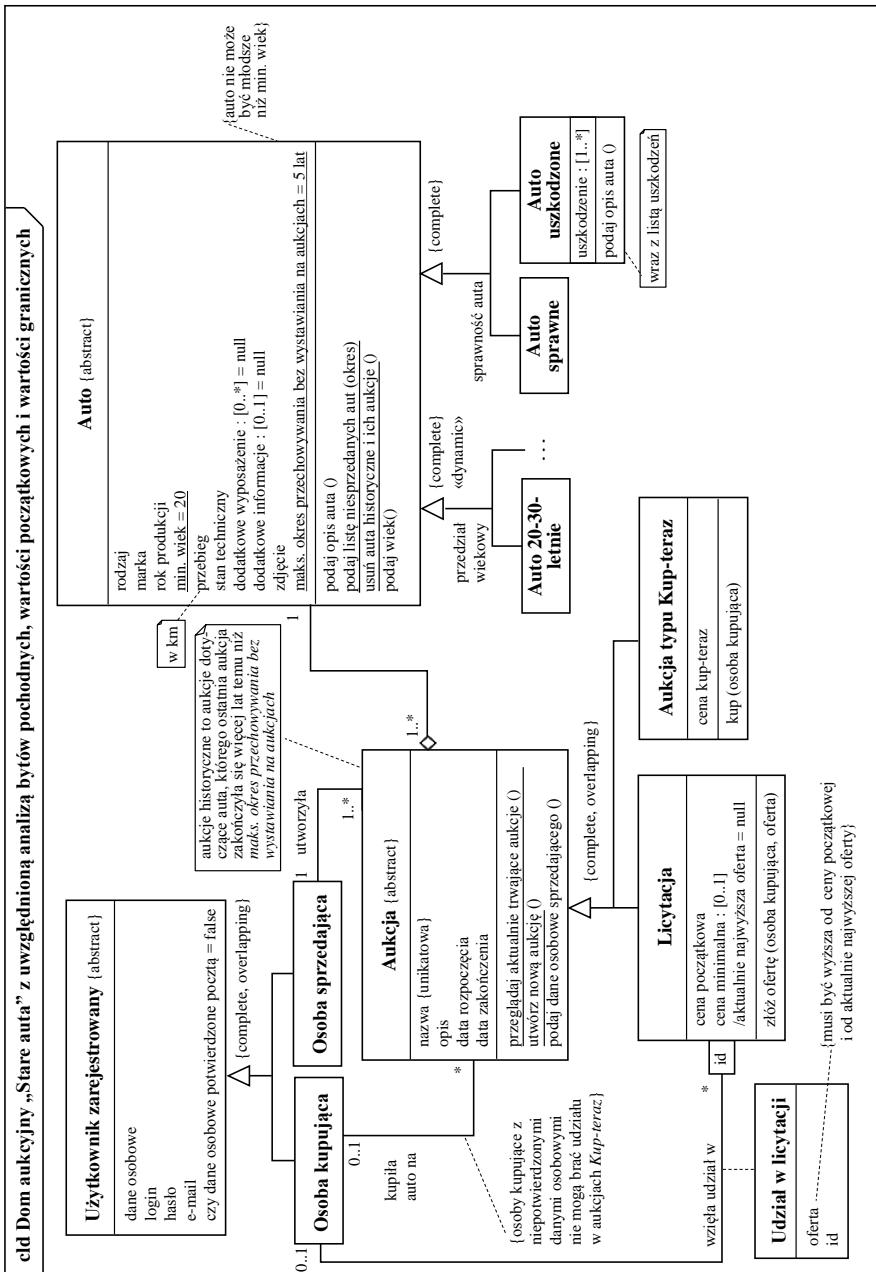
W klasie *Auto* jest także umieszczony atrybut *min. wiek = 20* (atrribut klasowy z zainicjowaną wartością), który zawiera wartość graniczną niezbędną do weryfikacji ograniczenia *auto nie może być młodsze niż min. wiek*. W przypadku potrzeby ustalenia nowej wartości minimalnego wieku aut, które mogą być wystawiane na aukcjach, w tej sytuacji będzie to można wykonać w bardzo łatwy sposób.

Inna wartość graniczna jest związana z przedziałem wiekowym aut, wykorzystywanym dla podziału aut ze względu na kategorię wiekową. Ponieważ jednak trudno przewidzieć możliwe różne sposoby podziału, ten element zostanie w dalszej analizie w tym zadaniu pominięty.

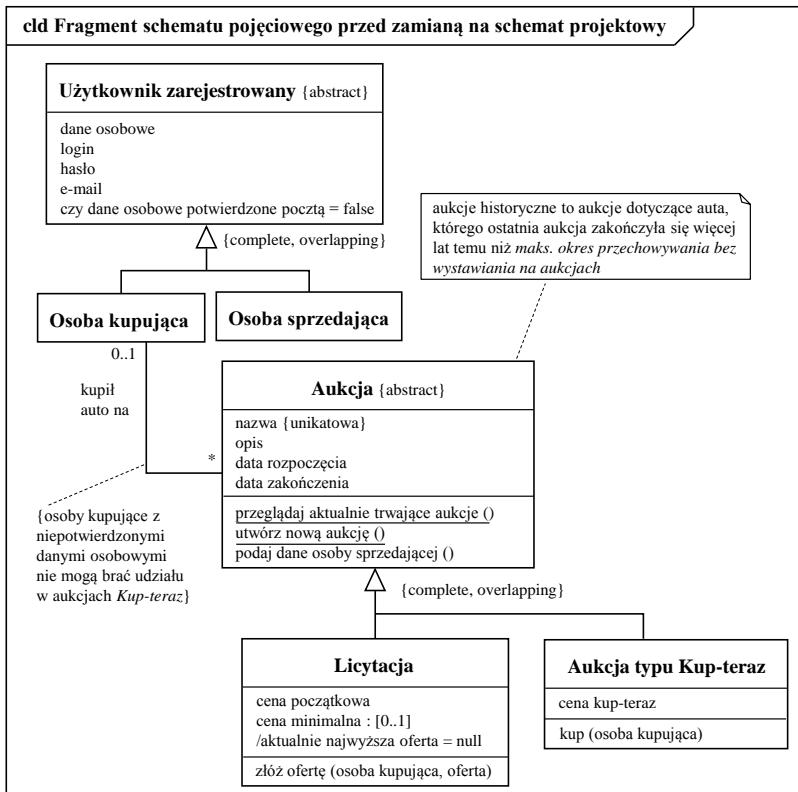
Schemat pojęciowy po zmianach uwzględniających analizę elementów pochodnych, wartości początkowych i wartości granicznych został przedstawiony na Rys. 4-10.

#### 4.3.5 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji

Na Rys. 4-11 przedstawiony został fragment schematu pojęciowego, niezbędny dla zamodelowania w popularnym języku obiektowym (np. w Javie) [37, 40] metody *kup (osoba kupująca)* w klasie *Aukcja typu Kup-teraz*, który zawiera hierarchię użytkowników zarejestrowanych, hierarchię aukcji oraz asocjację między klasą *Osoba kupująca* a klasą *Aukcja*. Metoda ta będzie modelowania z wykorzystaniem diagramu sekwencji w Zad. 2 w rozdziale związanym z analizą dynamiczną.



Rys. 4-10 Schemat pojęciowy po uwzględnieniu elementów pochodnych, wartości początkowych i wartości granicznych

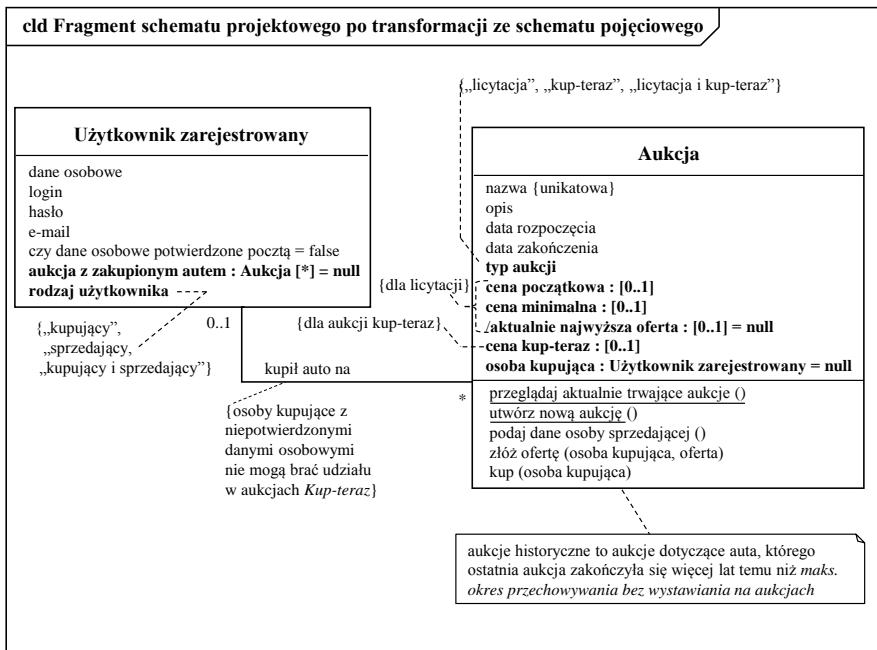


Rys. 4-11 Schemat pojęciowy sprzed zamiany na schemat projektowy

Zarówno w hierarchii użytkowników zarejestrowanych, jak i w hierarchii aukcji zostało określone dziedziczenie nierożłączne. W tych obu przypadkach najłatwiejszym sposobem realizacji hierarchii dziedziczenia wydaje się, że jest „zwinięcie” podklas do nadklasy, dla osób głównie z powodu braku własności zdefiniowanych w podklasach w ogóle, a dla aukcji – ponieważ zdefiniowano różne atrybuty, a także (dla osób i aukcji) ze względu na nieobecność metod polimorficznych.

Na tym etapie rozważań, z powodu niedostatku szczegółowych informacji o funkcjonalności systemu, trudno jest rozstrzygać o kierunkach nawigowania asocjacji, choć wydaje się, że asocjacja między klasą *Osoba\_kupujaca* a klasą *Aukcja* będzie wymagać nawigacji dwukierunkowej (potrzebne jest uzyskiwanie informacji, kto kupił auto na aukcji i prawdopodobnie także informacji, w jakich aukcjach uczestniczyła pewna osoba kupująca oraz jakie auta zostały na nich zakupione). Realizacja asocjacji będzie wymagała wprowadzenia dodatkowych atrybutów zawierających referencje do obiektów klas z przeciwej strony asocjacji: atrybutu *aukcja z zakupionym autem : Aukcja* [\*] w klasie *Uzytkownik\_zarejestrowany* oraz atrybutu *osoba kupująca : Uzytkownik\_zarejestrowany* w klasie *Aukcja*.

Schemat projektowy dla wskazanego na Rys. 4-11 fragmentu schematu pojęciowego został przedstawiony na Rys. 4-12. Atrybuty o charakterze dyskryminatorów oraz referencyjne, wprowadzone w trakcie transformacji, zostały wytluszczone dla większej czytelności diagramu.



Rys. 4-12 Fragment schematu projektowego po transformacji ze schematu pojęciowego

### 4.3.6 Zadania podstawowe

Zadania omówione w tym rozdziale są realizowane w oparciu o schemat pojęciowy z Rys. 4-10.

**Zad. 1:** Podkreś te rodzaje dziedziczenia (rozłączne, nierozłączne, kompletne, niekompletne, jednokrotne, wielokrotne, jednoaspektowe, wieloaspektowe, dynamiczne, elipsa), które uważaś za potrzebne do wykorzystania na schemacie pojęciowym dla systemu *Dom aukcyjny „Stare auta”*. Dla każdego podziału klas podaj, jakie rodzaje dziedziczenia zostały wykorzystane.

- Podział użytkowników zarejestrowanych: dziedziczenie nierozłączne, kompletne, jednokrotne, jednoaspektowe;
- podział aukcji: dziedziczenie nierozłączne, kompletne, jednokrotne, jednoaspektowe;
- podział aut: dziedziczenie wieloaspektowe (aspekty: *przedzial wiekowy*, *sprawność auta*), rozłączne (oba aspekty), kompletne (oba aspekty),

jednokrotne (oba aspekty), dynamiczne (aspekt *przedział wiekowy*), elipsa (aspekt *przedział wiekowy*).

**Zad. 2:** Jakie rodzaje metod (abstrakcyjne, zaimplementowane) może zawierać klasa abstrakcyjna, a jakie klasa konkretna? Podaj przykłady.

Klasa abstrakcyjna może zawierać oba rodzaje metod, a klasa konkretna tylko metody zaimplementowane.

- Metoda abstrakcyjna w klasie abstrakcyjnej: na diagramie nie ma metody abstrakcyjnej, ale można by umieścić taką metodę *podaj opis auta ()* w klasie abstrakcyjnej *Auto* (wtedy należałoby także umieścić odpowiednią metodę zaimplementowaną *podaj opis auta ()* w klasie *Auto sprawne*;
- metoda zaimplementowana w klasie abstrakcyjnej: *podaj listę niesprzedanych aut (okres)* w klasie *Auto*;
- metoda zaimplementowana w klasie konkretnej: *kup (osoba kupująca)* w klasie *Aukcja Kup-teraz*.

**Zad. 3:** Podaj przykłady metody klasowej, metody obiektowej, atrybutu klasowego i atrybutu obiektowego.

- Metoda obiektowa: *podaj opis auta ()* w klasie *Auto*;
- metoda klasowa: *podaj listę niesprzedanych aut (okres)* – operuje na ekstensji klasy *Auto*;
- atrybut obiektowy: *dane osobowe* w klasie *Użytkownik zarejestrowany*;
- atrybut klasowy: *min. wiek = 20* w klasie *Auto* (atrybut z wartością początkową).

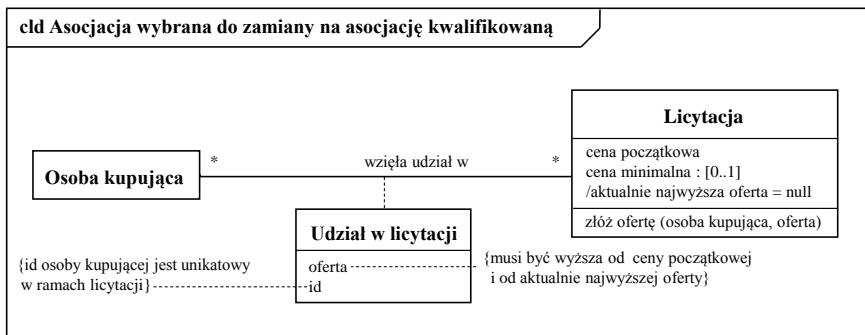
**Zad. 4:** Wskaź przykład zastosowania polimorfizmu metod.

Metody polimorficzne to metody *podaj opis auta ()* w klasach hierarchii zbudowanej dla aut (w aspekcie *sprawność auta*). Jest to przykład przesłaniania metod.

**Zad. 5:** Asocjację posiadającą niesymetryczne liczności oraz atrybut (lub klasę asocjacji) zamień na asocjację kwalifikowaną. Uzasadnij, dlaczego poddałeś zamianie właśnie tę asocjację.

Na diagramie klas umieszczono asocjację kwalifikowaną łączącą klasy *Osoba kupująca* i *Licytacja*, która wskazuje, że kluczem do przeszukiwania zbioru osób kupujących, przypisanych do danej licytacji, może być identyfikator osoby, unikatowy w obrębie licytacji. W tekście wymagań umieszczono następującą informację „... w ramach aukcji typu *Licytacja* każdemu *kupującemu* przypisywany jest unikatowy identyfikator ...”. Identyfikator ten mógłby być więc kwalifikatorem asocjacji kwalifikowanej. Taki identyfikator nie ma sensu w przypadku aukcji „*kup-teraz*”, gdyż z taką aukcją związana jest co najwyżej jedna osoba kupująca. Dlatego identyfikator zastosowano wyłącznie w przypadku asocjacji klasy *Osoba kupująca* z klasą *Licytacja*. W przypadku asocjacji kwalifikowanej liczność roli asocjacji po stronie przeciwej do kwalifikatora wskazuje „silę” kwalifikatora - klucza. Po

stronie kwalifikatora liczność pozostaje bez zmiany. Rys. 4-13 przedstawia fragment diagramu klas sprzed zamiany asocjacji „zwykłej” na asocjację kwalifikowaną.



Rys. 4-13 Fragment diagramu klas bez wykorzystania asocjacji kwalifikowanej

**Zad. 6:** Wskaź asocjację, która jest agregacją lub kompozycją (lub może kwalifikować się do zamiany na takową). Objaśni, dlaczego zdecydowałeś się na wybór agregacji/kompozycji, a nie „zwykłej” asocjacji? Ponadto, dlaczego wybrałeś agregację, a nie kompozycję (czy też odwrotnie)?

Na diagramie agregacja występuje pomiędzy klasami: *Aukcja* i *Auto*. *Auto* jest tutaj potraktowane jak element składowy aukcji. Wykorzystano agregację, a nie kompozycję, ponieważ w tym przypadku cykl życiowy części nie zawiera się w cyklu życiowym całości, tzn. usunięcie z systemu obiektu klasy *Aukcja* nie będzie skutkowało usunięciem obiektu klasy *Auto*, związanego z daną aukcją. Konkretnie auto może być elementem składowym wielu aukcji.

**Zad. 7:** Wskaź przykładowe ograniczenie. Na jaki element modelu zostało ono nałożone? Czy jest to ograniczenie statyczne czy dynamiczne? Zaproponuj, jaka metoda i w jakiej klasie mogłaby być odpowiedzialna za jego zapewnienie.

Jako przykład można podać ograniczenie na wartość atrybutu *oferta* umieszczonego w klasie asocjacji *Udziały\_w\_licytacji*: *{musi być wyższa od ceny początkowej i od aktualnie najwyższej oferty}*.

W rzeczywistości są to dwa ograniczenia: pierwsze to *{musi być wyższa od ceny początkowej}* – ograniczenie statyczne (nie interesuje nas poprzednia wartość atrybutu *oferta*) i drugie – *{musi być wyższa od aktualnie najwyższej oferty}* – także ograniczenie statyczne, ponieważ trzeba porównać wartość atrybutu *oferta*, zaproponowaną przez określona *Osobę kupującą*, z wartością atrybutu *aktualnie najwyższa oferta* (jest to inny byt, dlatego nie jest to ograniczenie dynamiczne). Wartość wprowadzanej nowej oferty musi spełniać obydwa ograniczenia.

Ograniczenia powinny zostać zaimplementowane w metodzie *złoż\_ofertę (osoba\_kupująca, oferta)* w klasie *Licitacja* – w metodzie, dzięki której osoba kupująca wprowadza swoją ofertę kupna auta.

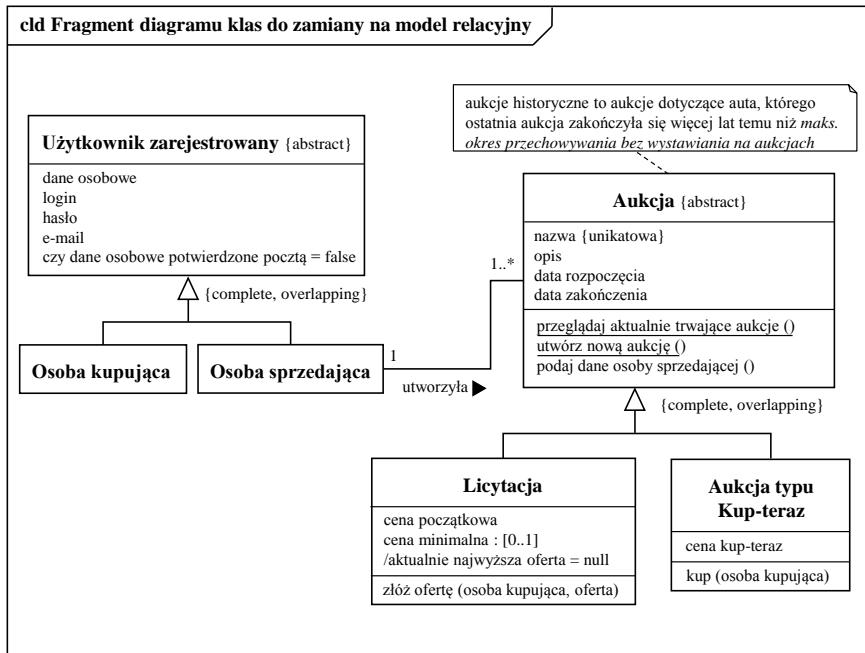
#### 4.3.7 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

- KLASY: Błądem jest wprowadzanie na tym etapie do schematu pojęciowego klas, które reprezentują wyłącznie aktorów systemu, np. *Pracownik domu aukcyjnego*.
- Nazwy klas muszą jednoznacznie określać, co jest obiektem danej klasy, tak więc błędem jest umieszczenie klasy o nazwie *20-30*. Co miałoby być obiektem tej klasy? Dobrze dobrana nazwa to np. *Auto 20-30 letnie*. Podobnie z nazwami *Uszkodzone*, *Sprawne* czy klasą *Kupiec* zamiast *Osoba kupującą*; należy bazować na słownictwie z dziedziny problemowej, którym posługiwano się w tekście wymagań.
- ATRYBUTY: Należy oznaczać rodzaje atrybutów, czyli np. w klasie *Auto* powinien pojawić się atrybut klasowy *min. wiek* – z jego wartością porównuje się wiek auta, które chcemy wystawić na aukcji. „20 lat” jest wartością początkową tego atrybutu. W klasie *Aukcja* umieszczono atrybut *dane osoby sprzedającej auto*, którego wartość jest kopią informacji przechowywanej w klasie *Osoba sprzedająca* i jest dostępna w klasie *Aukcja* dzięki asocjacji łączącej te klasy. Atrybut ten więc jest atrybutem pochodnym i tak należałoby go oznaczyć na diagramie. Podobnie, historia ofert jest przechowywana za pomocą atrybutów klasy asocjacji (*Udział w licytacji*). *Aktualnie najwyższa oferta* w klasie *Licytacja* jest więc atrybutem pochodnym – jego wartość jest równa ostatniej i zarazem najwyższej ofercie złożonej w czasie licytacji. Atrybut *czy dane osobowe potwierdzone* nie jest atrybutem opcjonalnym, tylko dwuwartościowym.
- METODY: metody należy umieszczać w tych klasach, których obiektów dotyczą, a nie w klasach modelujących aktorów, wywołujących te metody. Ponadto należy pamiętać o specyfikowaniu list argumentów dla metod, np. argument *okres* dla metody *podaj listę niesprzedanych aut (okres)* w klasie *Auto*.
- HIERARCHIE: w hierarchii *Auto* – w aspekcie *przedział wiekowy*, należało oznaczyć dziedziczenie dynamiczne (*auto*, o którym przechowujemy informacje, może się „zestarzeć” i może zmienić się jego przynależność do podklasy).
- ASOCJACJE: powinny mieć nazwy (frazy czasownikowe) i oznaczone liczności. Nazwę klasy asocjacji oraz roli asocjacji (podobnie jak „zwykłej” klasy) określamy, wykorzystując frazy rzeczownikowe, a nie czasownikowe.

#### 4.3.8 Schemat relacyjny dla fragmentu schematu pojęciowego

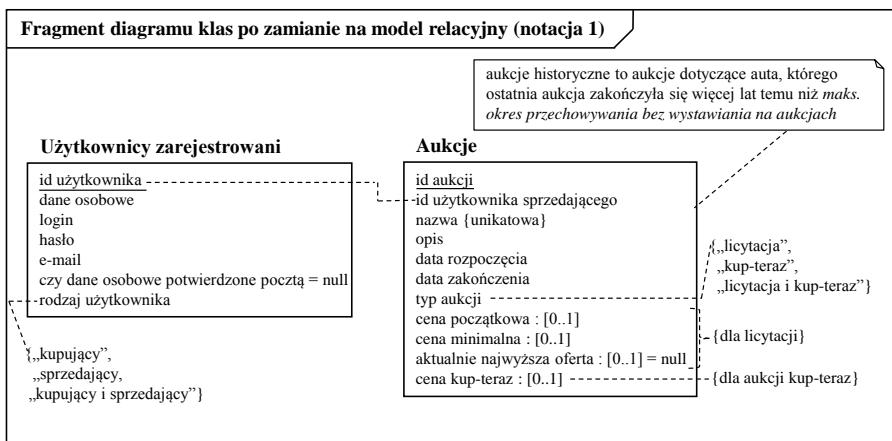
*Polecenie:* Fragment schematu pojęciowego, zawierający przynajmniej trzy klasy, związek dziedziczenia oraz asocjację, przedstaw w modelu relacyjnym.

Fragment schematu pojęciowego, podlegający zamianie na model relacyjny [44, 47], przedstawiono na Rys. 4-14.



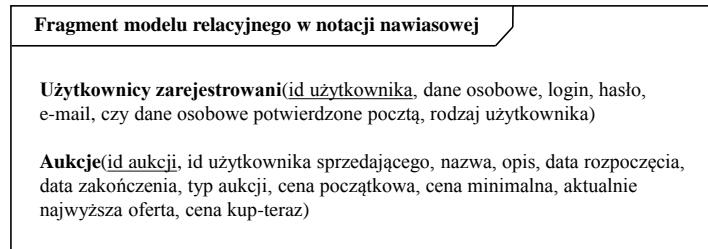
Rys. 4-14 Fragment diagramu klas podlegający zamianie na model relacyjny

Rys. 4-15 zawiera model relacyjny tego fragmentu diagramu klas w notacji 1 [41].



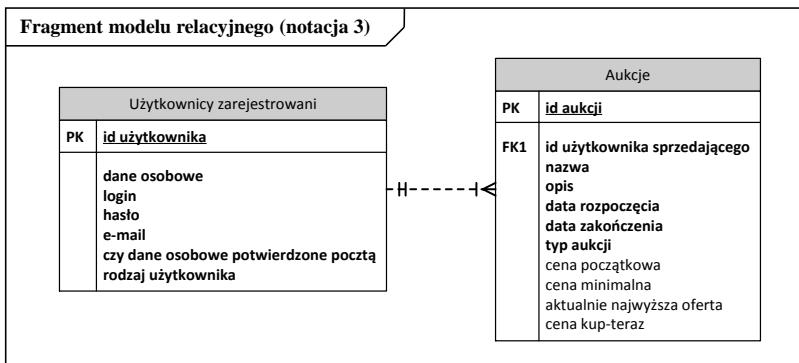
Rys. 4-15 Model relacyjny fragmentu diagramu klas przedstawionego na Rys. 4-14 (notacja 1)

Na Rys. 4-16 przedstawiono model relacyjny tego fragmentu diagramu klas w notacji nawiasowej.



Rys. 4-16 Model relacyjny fragmentu diagramu klas przedstawionego na Rys. 4-14 (notacja nawiasowa)

Natomiast na Rys. 4-17 zaprezentowano model relacyjny rozważanego fragmentu klas w notacji *Crow's Foot*.



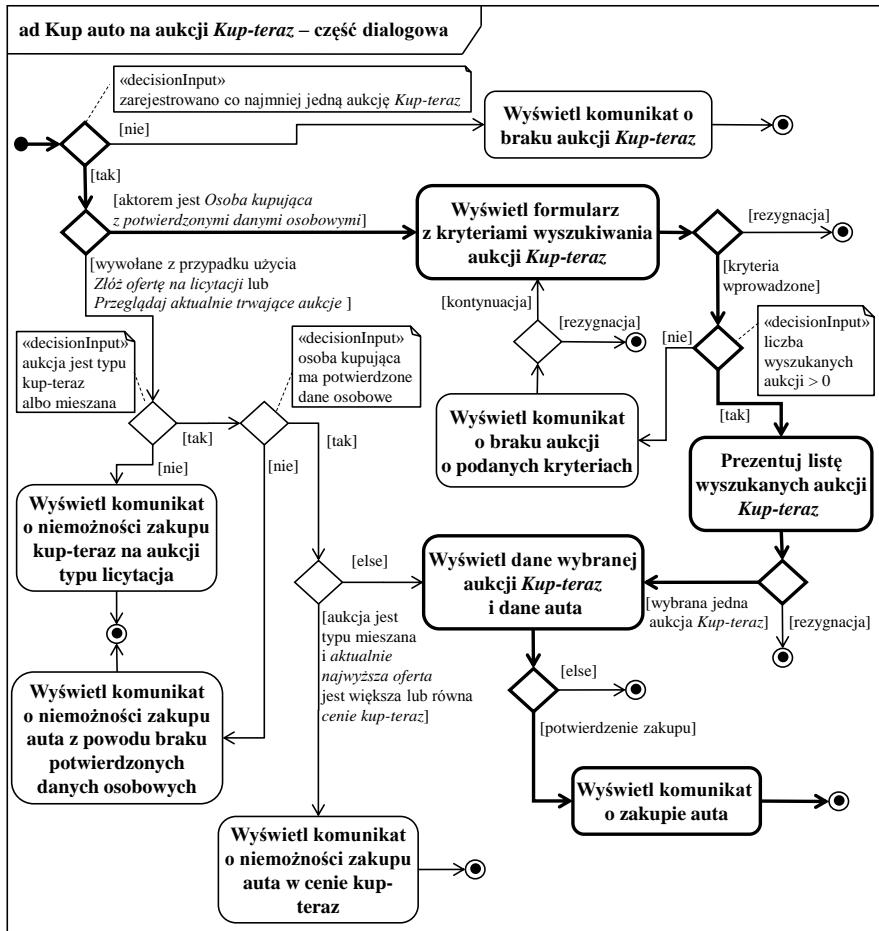
Rys. 4-17 Model relacyjny fragmentu diagramu klas przedstawionego na Rys. 4-14 w notacji *Crow's Foot*

## 4.4 Analiza dynamiczna

### 4.4.1 Zadania podstawowe

**Zad. 1:** Dla przypadku użycia *Kup auto na aukcji Kup-teraz* skonstruj diagram aktywności zgodny ze scenariuszem z Tab. 4-1.

Rys. 4-18 przedstawia diagram aktywności z przykładową realizacją scenariusza z Tab. 4-1. Dla lepszej czytelności przebieg głównego scenariusza został wyróżniony poprzez narysowanie go grubszą kreską.

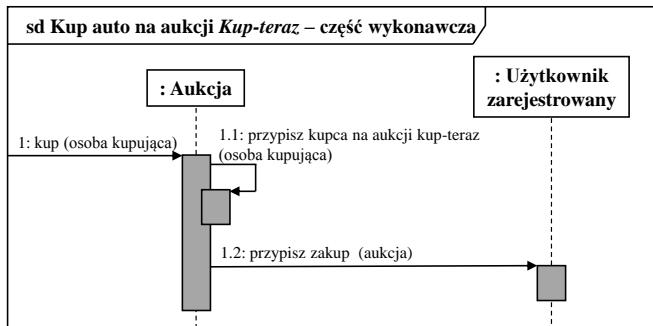


Rys. 4-18 Diagram aktywności dla przypadku użycia Kup auto na aukcji Kup-teraz

**Zad. 2:** Sporządź diagram interakcji dla funkcjonalności Kup auto na aukcji Kup-teraz.

Scenariusz dla przypadku użycia Kup auto na aukcji Kup-teraz został określony w Tab. 4-1.

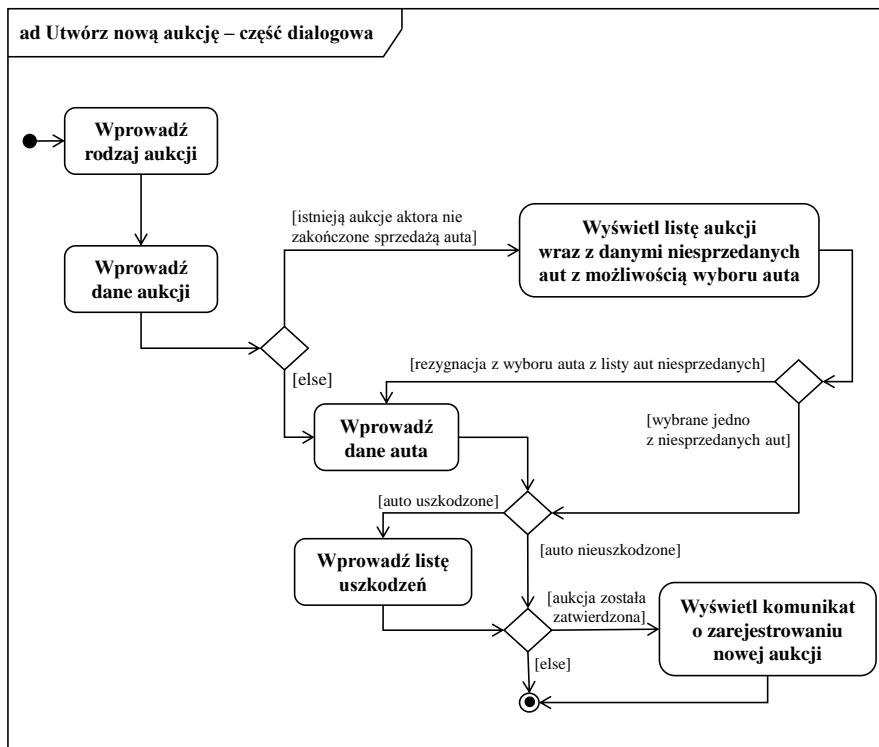
Dla opisu części wykonawczej przypadku użycia Kup auto na aukcji Kup-teraz przyjęte zostało założenie, że aukcja typu Kup-teraz została już wcześniej wybrana przez kupującego, którego dane osobowe zostały potwierdzone (zgodnie ze scenariuszem i diagramem aktywności) przez część dialogową przypadku użycia (za pośrednictwem GUI). Rys. 4-19 przedstawia odpowiedni diagram sekwencji, który został zamodelowany, bazując na fragmencie schematu projektowego z rozdziału 4.3.5 (Rys. 4-12).



Rys. 4-19 Diagram sekwencji dla przypadku użycia *Kup auto na aukcji *Kup-teraz* (część wykonawcza)*

**Zad. 3:** Dla przypadku użycia *Utwórz nową aukcję* skonstruuj diagram aktywności zgodny ze scenariuszem z Tab. 4-3.

Diagram aktywności zawarty na Rys. 4-20 ilustruje przykładową realizację dla scenariusza znajdującego się w Tab. 4-3.

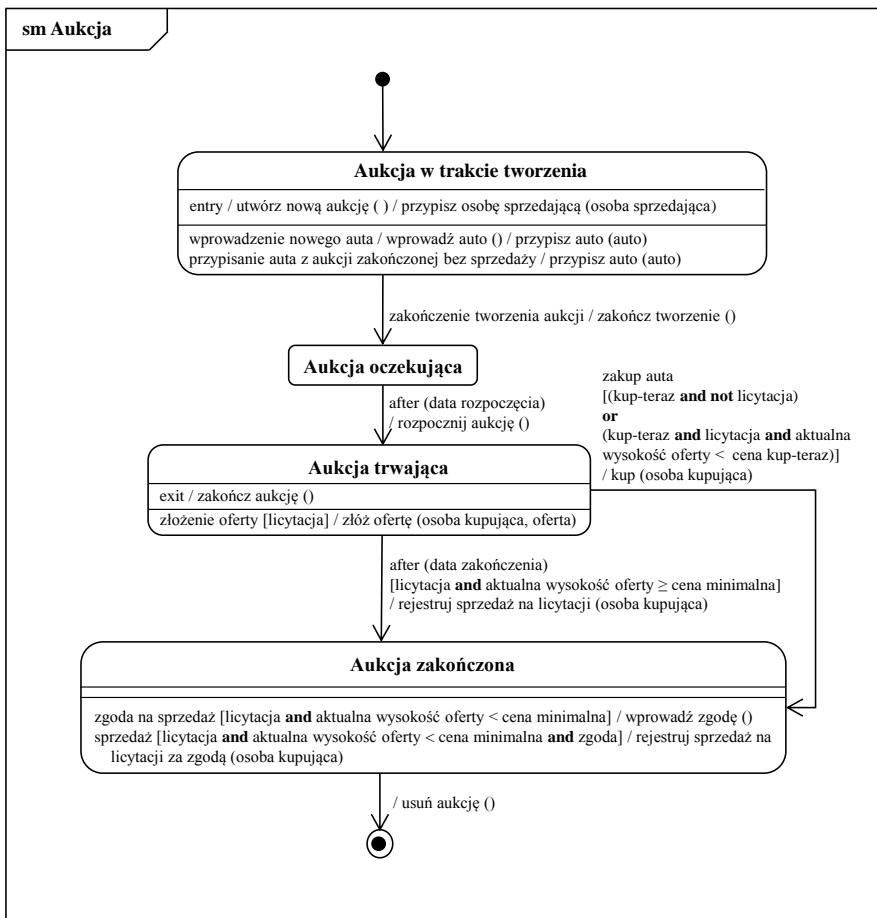


Rys. 4-20 Diagram aktywności dla przypadku użycia *Utwórz nową aukcję*

**Zad. 4:** Dla klasy *Aukcja* ze schematu projektowego z Rys. 4-12 wskaż kilka stanów (co najmniej dwa stany, nie licząc stanów początkowego i końcowego), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów, zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

*Uwaga:* W tym zadaniu nie jest wymagana cała maszyna stanowa, wystarczy zaprezentować dwa stany wraz ze specyfikacją przejścia.

Maszyna stanowa zachowania dla klasy *Aukcja* została przedstawiona na Rys. 4-21.



Rys. 4-21 Maszyna stanowa zachowania dla klasy *Aukcja*

#### **4.4.2 Omówienie zmian na schemacie projektowym wynikających z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań**

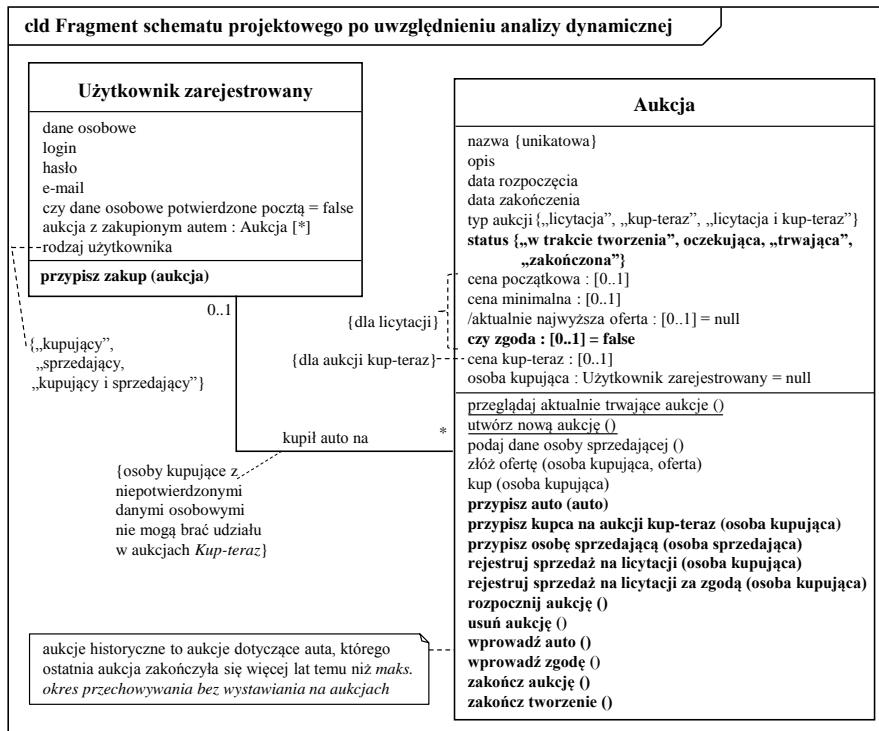
Po przeprowadzeniu analizy dynamicznej należy uzupełnić schemat projektowy o nowe elementy, które zostały ujęte na diagramach interakcji oraz diagramach stanów.

Lista zmian została zamieszczona w Tab. 4-5.

Tab. 4-5 Lista zmian na schemacie projektowym po analizie dynamicznej

Klasa	Zmiana
<i>Aukcja</i>	<p>dodatkowe metody:</p> <p><i>przypisz auto (auto)</i>  <i>przypisz kupca na aukcji kup-teraz (osoba kupująca)</i>  <i>przypisz osobę sprzedającą (osoba sprzedająca)</i>  <i>rejestruj sprzedaż na licytacji (osoba kupująca)</i>  <i>rejestruj sprzedaż na licytacji za zgodą (osoba kupująca)</i>  <i>rozpocznij aukcję ()</i>  <i>usuń aukcję ()</i>  <i>wprowadź auto ()</i>  <i>wprowadź zgodę ()</i>  <i>zakończ aukcję ()</i>  <i>zakończ tworzenie ()</i></p>
	<p>dodatkowe atrybuty:</p> <p><i>czy zgoda</i>  <i>status</i></p>
<i>Użytkownik zarejestrowany</i>	<p>dodatkowa metoda:</p> <p><i>przypisz zakup (aukcja)</i></p>

Na Rys. 4-22 przedstawiony został fragment schematu projektowego po uwzględnieniu listy zmian, tj. atrybutów i metod, które zostały umieszczone w tabeli Tab. 4-5.



Rys. 4-22 Fragment schematu projektowego po uwzględnieniu wniosków z analizy dynamicznej

## 4.5 Zadania do samodzielnnej pracy

- Podaj scenariusz dla wybranego (innego niż omówionego w analizie funkcjonalnej lub dynamicznej) przypadku i zaproponuj jego podział na podprzypadki.
- Co oznacza znak elipsy (inaczej znak opuszczenia, ang. *ellipsis*) umieszczony na diagramie? Objasnij, czym różni się wykorzystanie tego symbolu od ograniczenia *incomplete*. Podaj odpowiedni przykład w oparciu o schemat pojęciowy.
- Objasnij różnicę między klasą a ekstensją klasy. Podaj przykładowe ekstensje dla klas wchodzących w skład najbardziej rozwiniętej hierarchii na schemacie pojęciowym.
- Objasnij pojęcie przesłaniania. Wskaż przykład zastosowania na schemacie pojęciowym. Czy przesłanianie jest rodzajem polimorfizmu metod?
- Czy zasadę zamienialności można stosować do obiektów należących do różnych hierarchii klas? Odpowiedź uzasadnij. Jeśli odpowiedź jest na „tak”, podaj przykład oparty na schemacie pojęciowym.

6. Wskaż na schemacie pojęciowym asocjację wiele-do-wielu (z niesymetrycznymi licznosciami dla obu końców asocjacji ze zdefiniowaną klasą lub atrybutem asocjacji). Na jakim etapie cyklu życia systemu klasę asocjacji powinno się zamienić na konstrukcję zawierającą klasę pośredniczącą? Narysuj odpowiedni przykład (z atrybutami i metodami) na podstawie wskazanego uprzednio fragmentu diagramu.
7. Wskaż na schemacie pojęciowym fragment, gdzie mogłyby zostać wykorzystane role asocjacji. Narysuj odpowiedni przykład na podstawie wskazanego uprzednio fragmentu schematu.
8. Dla wybranego przypadku użycia systemu (innego niż omówionego szczegółowo w analizie funkcjonalnej lub dynamicznej) sformułuj algorytm i narysuj diagram aktywności, określając przynajmniej dwie aktywności, nie licząc początkowej i końcowej.
9. Dla przypadku użycia wybranego w zadaniu 8 narysuj diagram interakcji, zgodny ze sformułowanym algorytmem, który przedstawia interakcje pomiędzy obiektami co najmniej dwóch różnych klas.
10. Dla dowolnej klasy (innej niż omówionej szczegółowo w analizie dynamicznej) ze schematu pojęciowego wskaż dwa stany (wyłączając stan początkowy i stan końcowy), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów, zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

## 4.6 Podsumowanie zadania

W Tab. 4-6 przedstawiono ogólną charakterystykę omawianego zadania. Szczegółową charakterystykę zadania dla grup tematycznych: *Wprowadzenie do przedmiotu*, *Analiza funkcjonalna*, *Analiza strukturalna*, *Analiza dynamiczna* zamieszczono odpowiednio w tabelach Tab. 4-7, Tab. 4-8, Tab. 4-9 i Tab. 4-10.

Tab. 4-6 Ogólna charakterystyka zadania

<b>Ogólna charakterystyka zadania</b>	
<i>Nazwa</i>	Dom aukcyjny „ <i>Stare auta</i> ”
<i>Dziedzina problemowa</i>	system mógłby znaleźć zastosowanie w każdej dziedzinie, gdzie występuje potrzeba zarządzania różnego rodzaju aukcjami
<i>Cel</i>	zadaniem systemu jest ułatwienie działalności domu aukcyjnego „ <i>Stare auta</i> ” za pomocą serwisu internetowego, zwiększenie liczby klientów i oferowanych aukcji

<i>Zakres odpowiedzialności</i>	ewidencja osób sprzedających i kupujących, ewidencja aukcji i aut, tworzenie nowych aukcji, składanie ofert, generowanie prostych raportów (listy niesprzedanych aut), zarządzanie danymi historycznymi aukcji i aut	
<i>Stopień trudności</i>	średnio trudne	
<i>Rozmiar</i>	10 klas	
<i>Użyteczność w grupach tematycznych</i>		
<i>Grupa tematyczna</i>	<i>Użyteczność w grupie tematycznej</i>	<i>Użyteczność na etapie zgodnym z modelem efektywnego nauczania</i>
<i>Wprowadzenie do przedmiotu</i>	niska	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> niska <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> niska
<i>Analiza funkcjonalna</i>	wysoka	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> wysoka <i>Podsumowanie dotychczasowych działań:</i> wysoka <i>Ewaluacja:</i> wysoka
<i>Analiza strukturalna</i>	średnia	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> średnia <i>Podsumowanie dotychczasowych działań:</i> wysoka <i>Ewaluacja:</i> wysoka
<i>Analiza dynamiczna</i>	średnia	<i>Przygotowanie do nauki:</i> średnia <i>Nauka:</i> średnia <i>Podsumowanie dotychczasowych działań:</i> średnia <i>Ewaluacja:</i> wysoka

Tab. 4-7 Szczegółowa charakterystyka zadania dla grupy tematycznej *Wprowadzenie do przedmiotu*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Wprowadzenie do przedmiotu</i>
<b>Stopień trudności grupy w zadaniu</b>	łatwe

<b>Poruszane tematy</b>	<i>analiza a projektowanie, dekompozycja diagramów, mechanizmy rozszerzalności, reguły nazewnictwa, poprawa percepcji diagramu/diagramów</i>		
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>	
<i>analiza a projektowanie</i>	łatwe	<i>model pojęciowy, model projektowy</i>	
<i>mechanizmy rozszerzalności</i>	łatwe	<i>stereotyp, ograniczenie, komentarz</i>	
<i>dekompozycja diagramów</i>	łatwe	podział modelu przypadków użycia na diagramy, podział przypadku użycia na podprzypadki	
<i>reguły nazewnictwa</i>	łatwe	reguły nazewnictwa dla aktorów, przypadków użycia, klas, atrybutów, metod i asocjacji	
<i>poprawa percepcji diagramu/diagramów</i>	łatwe	wyróżniony przebieg główny na diagramie aktywności	

Tab. 4-8 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza funkcjonalna*

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>	<i>Analiza funkcjonalna</i>	
<b>Stopień trudności grupy w zadaniu</b>	średnio trudne	
<b>Poruszane tematy</b>	<i>aktorzy, przypadki użycia, scenariusze, dekompozycja modelu/diagramu przypadków użycia, mechanizmy rozszerzalności</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>

<i>aktorzy</i>	średnio trudne	<i>aktor abstrakcyjny, aktor Podsystem czasu, hierarchia dla aktorów, stosunkowo duża liczba aktorów</i>
<i>przypadki użycia</i>	średnio trudne	relacje pomiędzy przypadkami użycia «include» i «extend»
<i>scenariusze</i>	trudne	stosunkowo duża złożoność scenariuszy przypadków użycia
<i>dekompozycja modelu/diagramu przypadków użycia</i>	średnio trudne	<i>dekompozycja pozioma, dekompozycja pionowa, punkty/miejsca rozszerzeń (extension points)</i>
<i>mechanizmy rozszerzalności</i>	łatwe	ograniczenia nałożone na relacje między przypadkami użycia

Tab. 4-9 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza strukturalna*

<b>Szczegółowa charakterystyka zadania</b>			
<b>Grupa tematyczna</b>	<i>Analiza strukturalna</i>		
<b>Stopień trudności grupy w zadaniu</b>	łatwe		
<b>Poruszane tematy</b>	<i>atrybuty klasy, metody klasy, struktury generalizacji-specjalizacji, klasa abstrakcyjna a klasa konkretna, asocjacje, analiza wartości, przejście na schemat projektowy, przejście na schemat relacyjny, mechanizmy rozszerzalności</i>		
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>	
<i>atrybuty klasy</i>	łatwe	<i>atrybut opcjonalny, powtarzalny, pochodny, obiektowy, klasowy</i>	
<i>metody klasy</i>	łatwe	<i>metoda zaimplementowana, obiektowa, klasowa, przesłanianie metod, polimorfizm metod</i>	
<i>struktury generalizacji-specjalizacji</i>	średnio trudne	<i>dziedziczenie nierożłączne, dynamiczne, wieloaspektowe i elipsa</i>	
<i>klasa abstrakcyjna a klasa konkretna</i>	łatwe	<i>klasa abstrakcyjna, klasa konkretna</i>	

<i>asocjacje</i>	łatwe	<i>agregacja, asocjacja kwalifikowana</i>
<i>analiza wartości</i>	łatwe	<i>wartość pochodna, wartość początkowa (domyślna) atrybutów i argumentów metod, wartość graniczna</i>
<i>przejście na schemat projektowy</i>	łatwe	<i>obejście dziedziczenia nierożłącznego</i>
<i>przejście na schemat relacyjny</i>	łatwe	<i>obejście dziedziczenia nierożłącznego</i>
<i>mechanizmy rozszerzalności</i>	łatwe	<i>stereotyp («dynamic»), ograniczenie, komentarz</i>

Tab. 4-10 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza dynamiczna*

<b>Szczegółowa charakterystyka zadania</b>			
<b>Grupa tematyczna</b>	<i>Analiza dynamiczna</i>		
<b>Stopień trudności grupy w zadaniu</b>	średnio trudne		
<b>Poruszane tematy</b>	<i>diagramy aktywności, diagramy interakcji, diagramy stanów, podsumowanie analizy dynamicznej</i>		
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>	
<i>diagramy aktywności</i>	średnio trudne	stosunkowo duża złożoność diagramów aktywności, romb decyzyjny <i>branch</i> i <i>merge</i> , wyróżniony przebieg główny	
<i>diagramy interakcji</i>	łatwe	<i>diagram sekwencji</i>	
<i>diagramy stanów</i>	średnio trudne	akcja <i>entry</i> , akcja <i>exit</i> , zdarzenie <i>after</i>	
<i>podsumowanie analizy dynamicznej</i>	średnio trudne	modyfikacje schematu projektowego o dodatkowe atrybuty i metody	

## Spis treści rozdziału

Rozdział 5 System „Najlepsze gry” .....	218
5.1 Tekst wymagań.....	218
5.2 Analiza funkcjonalna .....	220
5.2.1 Diagram kontekstowy .....	220
5.2.2 Model przypadków użycia.....	220
5.2.3 Scenariusze dla wybranych przypadków użycia łącznie z ich podziałem na podprzypadki.....	222
5.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	225
5.3 Analiza strukturalna.....	227
5.3.1 Schemat pojęciowy .....	227
5.3.2 Analiza elementów pochodnych .....	229
5.3.3 Analiza wartości początkowych .....	230
5.3.4 Analiza wartości granicznych.....	230
5.3.5 Zadania podstawowe .....	232
5.3.6 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji .....	235
5.3.7 Schemat relacyjny dla fragmentu diagramu klas .....	240
5.4 Analiza dynamiczna .....	241
5.4.1 Zadania podstawowe .....	241
5.4.2 Omówienie zmian na schemacie pojęciowym i projektowym, wynikających z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań.....	243
5.5 Zadania do samodzielnej pracy .....	244
5.6 Podsumowanie zadania.....	245

# Rozdział 5

## System „Najlepsze gry”

Ewa Stemposz, Andrzej Jodłowski

### 5.1 Tekst wymagań

System „Najlepsze gry” ma na celu ułatwienie budowy gier dla różnego rodzaju graczy poprzez śledzenie statystyk związanych z wykorzystaniem gier przez użytkowników systemu, jak np. znajdowanie najbardziej i najmniej popularnych gier, znajdowanie najlepszych graczy (potencjalni testerzy dla przyszłych gier) itd.

1. Każda gra jest opisywana poprzez listę następujących własności: nazwę katalogową, nazwę potoczną (niektóre gry nie mają nazwy potocznej), stopień trudności (w skali 0-5) oraz minimalną i maksymalną liczbę współgraczy, uczestników rozgrywki (rozegranej gry). Przechowywana jest informacja o tym, czy gra jest grą zespołową. Jeśli jest, to dodatkowo przechowywana jest maksymalna wielkość zespołu. Ponadto, do każdej gry jest dodawany zbiór reguł (oddzielny opis dla każdej reguły w postaci tekstu). Nazwy gier (katalogowa i potoczna) są unikatowe.
2. Gry dzielą się na kobiece i męskie. Dla gier kobiecych znany jest wymagany stopień współpracy między współgraczami w trakcie rozgrywania gry (w skali 0-3) – panie lubią gry zespołowe. Dla gier męskich przechowywane są dwa współczynniki, z których pierwszy określa poziom przemocy, a drugi poziom akcji w grze (oba w skali 0-10). Jest możliwe, że gra nadaje się zarówno dla kobiet, jak i dla mężczyzn.
3. Ponadto, gry dzielą się w zależności od wieku osób, dla których zostały zaprojektowane, np. „dla dzieci do lat pięciu”, „dla dzieci od pięciu do siedmiu lat” itp. Podział na grupy wiekowe jest rozłączny i kompletny. W grach z grupy „dla dzieci do lat pięciu” może brać udział niewielka liczba współgraczy (aktualnie maksymalnie trzech). Gra, dla której poziom przemocy został określony przez liczbę większą od 0, nie może zostać zakwalifikowana jako gra dla osób poniżej lat piętnastu.
4. Każda gra wchodzi w skład co najmniej jednej kategorii gier. Kategoria jest opisywana przez unikatową nazwę, jak np.: gry rysunkowe, gry słowne, łamigłówki, gry strategiczne itp. Jest możliwe, że istnieje kategoria gier chwilowo niezawierająca żadnych gier. Dla każdej gry znana jest jej pozycja w rankingu popularności w ramach danej kategorii, ustalana na podstawie liczby przeprowadzonych rozgrywek w ostatnich 6 miesiącach.

5. Dla każdej rozgrywki znana jest data przeprowadzenia, godzina rozpoczęcia oraz lista współgraczy, których nie może być ani mniej, ani więcej, niż zostało to ustalone dla gry, w ramach której przeprowadzono rozgrywkę. Wiadomo, kto zwyciężył w rozgrywce oraz jaki wynik osiągnął. Wynik, tj. liczba punktów, jaką uzyskuje zwycięzca rozgrywki, jest wyliczany na podstawie tego, czy gra jest indywidualna czy zespołowa, czy jest grą kobiecą czy męską oraz do jakiej kategorii wiekowej należy.
6. W przypadku gier zespołowych, osób – członków zwycięskiego zespołu (z tym samym wynikiem) może być kilku – liczba członków zwycięskiego zespołu nie może być większa niż maksymalna wielkość zespołu określona dla tej gry. Informacje o zespołach nie są przechowywane. Wyniki osób, które przegrali w rozgrywce, nie są przechowywane. O ile gra zakończyła się remisem, wyniki również nie są przechowywane (bo nie ma zwycięzcy).
7. Wiek osób biorących udział w rozgrywce musi być zgodny z grupą wiekową, do której została przypisana gra. Także płeć każdego współgracza musi być zgodna z przeznaczeniem gry (dla kobiet czy dla mężczyzn, czy też dla obu płci). W obu przypadkach chodzi o wyeliminowanie tzw. „oszukanych zwycięstw”, np. gdy osoba dorosła próbuje zarejestrować zwycięski wynik w grze przeznaczonej dla dzieci.
8. Dla każdej osoby przechowane są informacje, takie jak: pseudonim (unikatowy), płeć, data urodzenia, kontakt. Dla osoby będącej autorem gier przechowywane są następujące dane: imię, nazwisko, pseudonim (unikatowy), płeć, data urodzenia, kontakt oraz liczba gier zarejestrowanych przez nią dotychczas w systemie. Autor może zarejestrować nową grę w systemie nie wcześniej niż po tygodniu od zarejestrowania poprzedniej.
9. System ma wspomóc w realizowaniu usług, jak na przykład:
  - 9.1 Podanie charakterystyki gry (z uwzględnieniem różnicy we własnościach gier kobiecych i męskich);
  - 9.2 Pobieranie gry (gry są rozgrywane poza systemem, w systemie są rejestrowane jedynie wyniki rozgrywek);
  - 9.3 Znajdowanie graczy „najlepszych w danej grze”, czyli takich, którzy uzyskali najwięcej punktów w grze;
  - 9.4 Znajdowanie graczy „najlepszych w ogóle” (z największą łączną liczbą punktów dla wszystkich gier);
  - 9.5 Znajdowanie najbardziej i najmniej popularnej gry (dla zadanego okresu), czyli gry z największą oraz gry z najmniejszą liczbą rozgrywek (w zadanym okresie);
  - 9.6 Ustalanie rankingu popularności gier w danej kategorii (w pierwszym dniu każdego miesiąca);
  - 9.7 Przypisywanie grze statusu o wartości „aktualnie w archiwum” w sytuacji, gdy gra zajmuje niską pozycję w rankingu popularności dla gier z danej kategorii (być może będzie wymagała przeróbek przed ponownym udostępnieniem użytkownikom);

- 9.8 Podawanie sumarycznego wyniku gracza w danej grze;  
 9.9 Udostępnianie gry z archiwum.

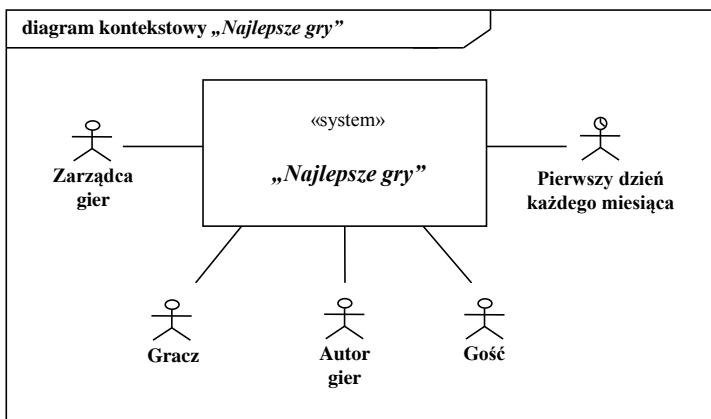
## 5.2 Analiza funkcjonalna

### 5.2.1 Diagram kontekstowy

*Polecenie:* Zbuduj diagram kontekstowy w oparciu o zamieszczony tekst wymagań.

Diagram kontekstowy, przedstawiający aktorów wybranych dla systemu „Najlepsze gry”, w oparciu o analizę tekstu wymagań) został zaprezentowany na Rys. 5-1.

Należy zwrócić uwagę na to, że nie wszyscy aktorzy zostali bezpośrednio wymienieni w tekście wymagań. Wyboru aktorów dokonał analityk, na podstawie znajomości dziedziny problemowej.



Rys. 5-1 Diagram kontekstowy

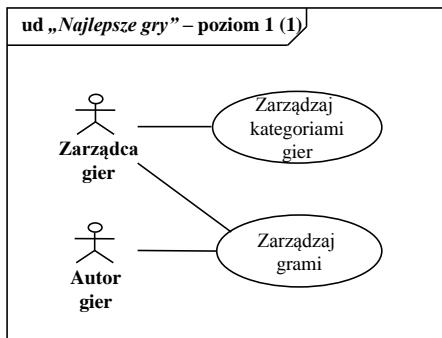
### 5.2.2 Model przypadków użycia

*Polecenie:* Zbuduj model przypadków użycia, uwzględniając głównie funkcjonalność sugerowaną w ostatnim punkcie tekstu wymagań (tj. w punkcie 9).

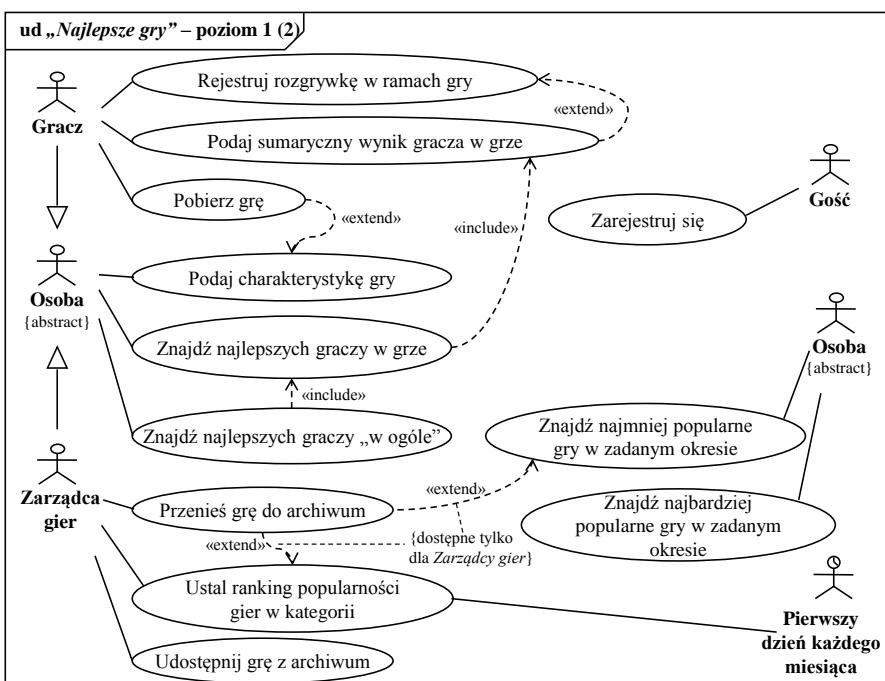
*Uwaga:* Model należy skonstruować z perspektywy aktorów z otoczenia systemu (najwyższy poziom abstrakcji modelu), biorąc pod uwagę możliwą ewolucję systemu w przyszłości. Należy uwzględnić hierarchie aktorów i relacje pomiędzy przypadkami (o ile mają/mogłyby mieć miejsce).

W celu zachowania przejrzystości model przypadków użycia [35, 41, 43, 46] został podzielony na dwa diagramy. W nagłówkach oznaczono numer poziomu abstrakcji modelu, a w nawiasach podano numer diagramu.

Diagram pierwszy (Rys. 5-2) prezentuje funkcjonalność typu *Zarządzaj danymi* (tzw. CRUD) dotyczącą danych, które opisują kategorie gier i gry. Pozostała funkcjonalność przedstawiona została na Rys. 5-3.



Rys. 5-2 Model przypadków użycia (diagram pierwszy)



Rys. 5-3 Model przypadków użycia (diagram drugi)

### 5.2.3 Scenariusze dla wybranych przypadków użycia łącznie z ich podziałem na podprzypadki

**Polecenie:** Napisz scenariusz dla przypadku użycia *Znajdź najlepszych graczy w grze*.

Przykładowy scenariusz ilustrujący interakcję aktora z systemem dla wybranego przypadku użycia został przedstawiony w Tab. 5-1.

Tab. 5-1 Scenariusz dla przypadku użycia *Znajdź najlepszych graczy w grze*

<b>Nazwa przypadku użycia</b>	Znajdź najlepszych graczy w grze
<b>Warunek początkowy</b>	W systemie musi być zarejestrowana co najmniej jedna rozgrywka.
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>Aktor, którym jest <i>Zarządcą gier</i> lub <i>Gracz</i>, uruchamia przypadek użycia.</li> <li>System odpytuje o nazwę katalogową gry. Aktor wprowadza nazwę.</li> <li>System wyświetla informacje o graczech, którzy uzyskali największą liczbę punktów w wybranej grze.</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	<ol style="list-style-type: none"> <li>Format nazwy jest błędny, system informuje o tym aktora i powraca do punktu 2.</li> <li>Gra o podanej nazwie katalogowej nie została odnaleziona. System informuje o tym i pyta, czy wprowadzić inną nazwę katalogową.</li> <li>Aktor decyduje się na wprowadzanie nazwy katalogowej.</li> </ol>
<b>Zakończenie</b>	W każdym momencie
<b>Warunek końcowy</b>	Brak

Zamiast wprowadzać nazwę katalogową gry (jak w scenariuszu z Tab. 5-1), można wyświetlić alfabetyczną listę gier według ich nazw katalogowych i/lub umożliwić przeszukiwanie po kategoriach i nazwach potocznych.

*Polecenie:* Dla przypadku użycia *Rejestruj rozgrywkę w ramach gry*:

- napisz scenariusz;
- zaproponuj podział tego przypadku na podprzypadki.

Przykładowy scenariusz dla wybranego przypadku użycia został przedstawiony w Tab. 5-2.

Tab. 5-2 Scenariusz dla przypadku użycia *Rejestruj rozgrywkę w ramach gry*

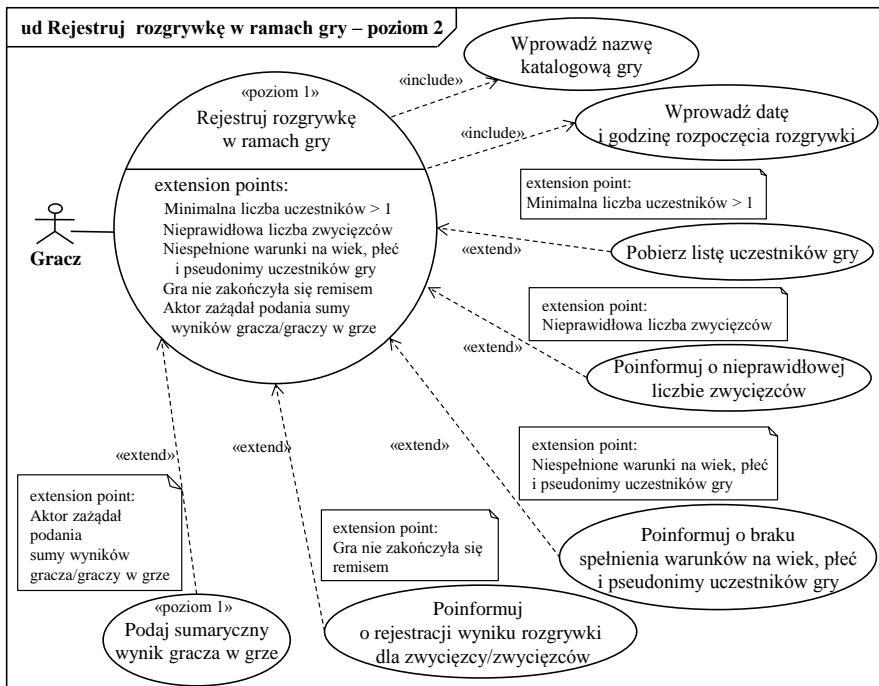
<b>Nazwa przypadku użycia</b>	Rejestruj rozgrywkę w ramach gry
<b>Warunek początkowy</b>	W systemie jest zarejestrowana co najmniej jedna gra i co najmniej jeden gracz.
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Przypadek użycia rozpoczyna aktor <i>Gracz</i>.</li> <li>2. System pyta o nazwę katalogową gry. Aktor wprowadza nazwę.</li> <li>3. System pyta o datę i godzinę rozpoczęcia rozgrywki oraz o liczbę uczestników gry. Aktor wprowadza dane.</li> <li>4. Dla każdego kolejnego gracza system odpytuje o pseudonimy uczestników gry i o to, czy zwyciężył. Aktor wprowadza odpowiednią informację.</li> <li>5. System informuje o zarejestrowaniu rozgrywki.</li> <li>6. System podaje sumaryczne wyniki gracza/graczy w danej grze.</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	<ol style="list-style-type: none"> <li>2a. Format nazwy jest błędny, system informuje o tym aktora i powraca do punktu 2.</li> <li>2b. Gra o podanej nazwie katalogowej nie została zarejestrowana, system informuje o tym aktora i pyta, czy podać listę wszystkich gier z podziałem na kategorie.             <ol style="list-style-type: none"> <li>2ba. Aktor rezygnuje z poszukiwania gry, co kończy przypadek użycia.</li> <li>2bb. System wyświetla listę gier z podziałem na kategorie. Aktor wybiera grę.</li> </ol> </li> <li>3a. Format daty/godziny jest nieprawidłowy, system informuje o tym aktora i powraca do punktu 3.</li> <li>3b. Rozgrywka jest już zarejestrowana (system rozpoznaje rozgrywkę po dacie, godzinie</li> </ol>

	<p>rozpoczęcia i przypisanych do rozgrywki uczestnikach), system informuje o tym aktora i kończy przypadek użycia.</p> <p>4a. Liczba współgraczy / wielkość zespołu (dla gry zespołowej) nie są zgodne z wartościami określonymi dla danej gry, system informuje o tym aktora i powraca do punktu 4.</p> <p>4b. Format pseudonimu któregoś z graczy jest nieprawidłowy, system informuje o tym aktora i powraca do punktu 4.</p> <p>4c. Gracz o podanym pseudonimie nie jest zarejestrowany w systemie, system informuje o błędzie i powraca do punktu 4.</p> <p>4d. Jeśli któryś z uczestników gry nie spełnia wymogów nałożonych na płeć i kategorię wiekową określoną dla danej gry, to system informuje o tym aktora i kończy przypadek użycia.</p> <p>4e. Liczba odpowiedzi „zwyciężyły” nie jest zgodna z wartościami określonymi dla danej gry, system informuje o tym aktora i pyta, czy kontynuować.</p> <p>4ea. Aktor potwierdza wolę kontynuowania, system powraca do punktu 4.</p> <p>4eb. Aktor rezygnuje, system kończy przypadek użycia.</p> <p>6a Aktor rezygnuje z podawania sumarycznych wyników gracza/graczy w grze, co kończy przypadek użycia.</p>
<b>Zakończenie</b>	W dowolnym momencie.
<b>Warunek końcowy</b>	<p>1. Jeśli warunki nakładane na grę zostały spełnione, to zostanie zapamiętana informacja o przeprowadzonej rozgrywce (data rozgrywki, godzina rozpoczęcia, lista uczestników gry).</p> <p>2. Jeśli gra nie zakończyła się remisem, to wynik zostanie zapisany na konto zwycięzcy/zwycięzców.</p>

Diagram z przykładowym podziałem przypadku *Rejestruj rozgrywkę w ramach gry* przedstawiono na Rys. 5-4.

Na diagramie można byłoby ująć również takie przypadki, jak np.: *Wypełnij ankietę o grze* czy *Wprowadź uwagi do autora gry*. W takiej sytuacji scenariusz powinien

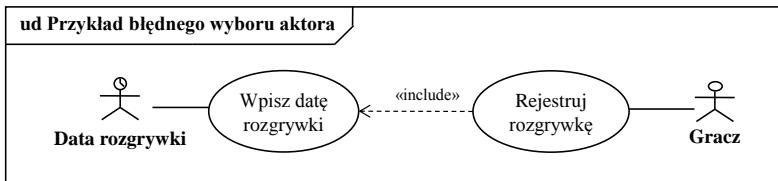
zostać zmieniony tak, aby uwzględnić wymienione przypadki użycia. Również w punkcie 6 można było wywołać w pętli (dla wszystkich uczestników rozgrywki) przypadek użycia *Podaj sumaryczny wynik gracza w grze*.



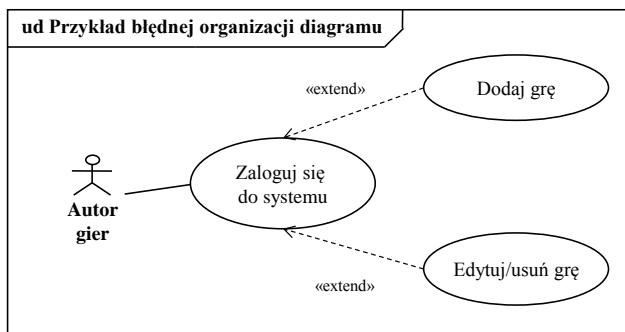
Rys. 5-4 Diagram z przykładowym podziałem przypadku *Rejestruj rozgrywkę w ramach gry*

#### 5.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

- Aktorzy tacy jak: *Podsystem gier*, *Podsystem statystyk* nie są bytami z otoczenia systemu.
- Nazwa przypadku nie jest nazwą zadania zlecanego systemowi, ale określa dane, np. nazwa taka jak *Ranking gier* – trudno się tu zorientować, jaki ma być *obserwowałny rezultat* dla tak nazwanego przypadku – ma być utworzony ranking czy też może tylko odczytany aktualny ranking.
- Niewłaściwy przydział usług dla aktorów, np. aktorowi *Podsystem czasu* przypisywano uprawnienia do przypadku związanego z wyszukiwaniem najmniej i najbardziej popularnych gier – a nie został określony moment, kiedy takie zadanie miałoby być realizowane. Również diagram z Rys. 5-5 wykazuje niezrozumienie istoty aktora *Podsystem czasu*.

Rys. 5-5 Diagram z błędny wykorzystaniem aktora typu *Podsystem czasu*

- Niewłaściwa organizacja diagramu związana z wprowadzeniem przypadku *Zaloguj się do systemu*. Domyślnie przyjmuje się, że zarówno autentykacja, jak i autoryzacja zostały już wykonane (zanim w ogóle wywołany został jakikolwiek przypadek użycia). Przykład niewłaściwej organizacji diagramu z wykorzystaniem zbędnego przypadku użycia *Zaloguj się do systemu* przedstawiono na Rys. 5-6.

Rys. 5-6 Niewłaściwa organizacja diagramu; niepotrzebny przypadek *Zaloguj się do systemu*

- Niezrozumienie sekcji *Warunek wstępny* w dokumentacji przypadku. W tej sekcji umieszczane są warunki nakładane na „świat przed wywołaniem przypadku”. Trudno jest do nich zaliczać warunki np. na wiek czy płeć graczy, ponieważ przed wywołaniem przypadku *Rejestruj rozgrywkę w ramach gry* i przed przeprowadzeniem dialogu aktor/system nie jest wiadomo, kto brał udział w rozgrywce.
- Zgodnie z tekstem wymagań gry są rozgrywane poza systemem. Ostatecznie można było przyjąć założenie, że każda z gier pobranych z systemu „Najlepsze gry”, po przeprowadzeniu rozgrywki, potrafi skomunikować się z systemem „Najlepsze gry”. W takim przypadku można było potraktować grę jako aktora dla tego systemu. Należało jednak zastanowić się nad sposobem przekazywania informacji o rozgrywkach przeprowadzonych w ramach gry do systemu, co powinno było zostać uwidocznione w scenariuszu dla przypadku *Rejestruj rozgrywkę w ramach gry*.

## 5.3 Analiza strukturalna

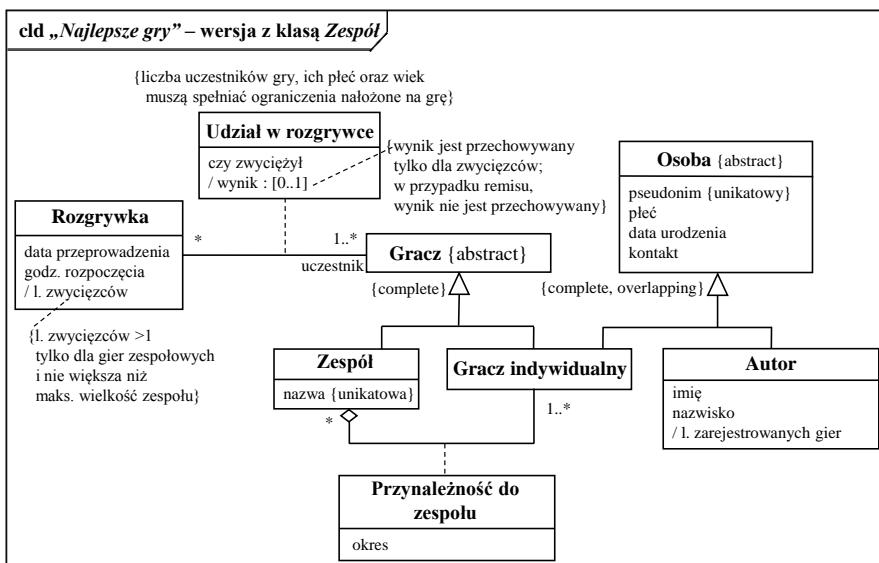
### 5.3.1 Schemat pojęciowy

*Polecenie:* Dla podanego tekstu wymagań skonstruuj schemat pojęciowy (nie redukując liczności).

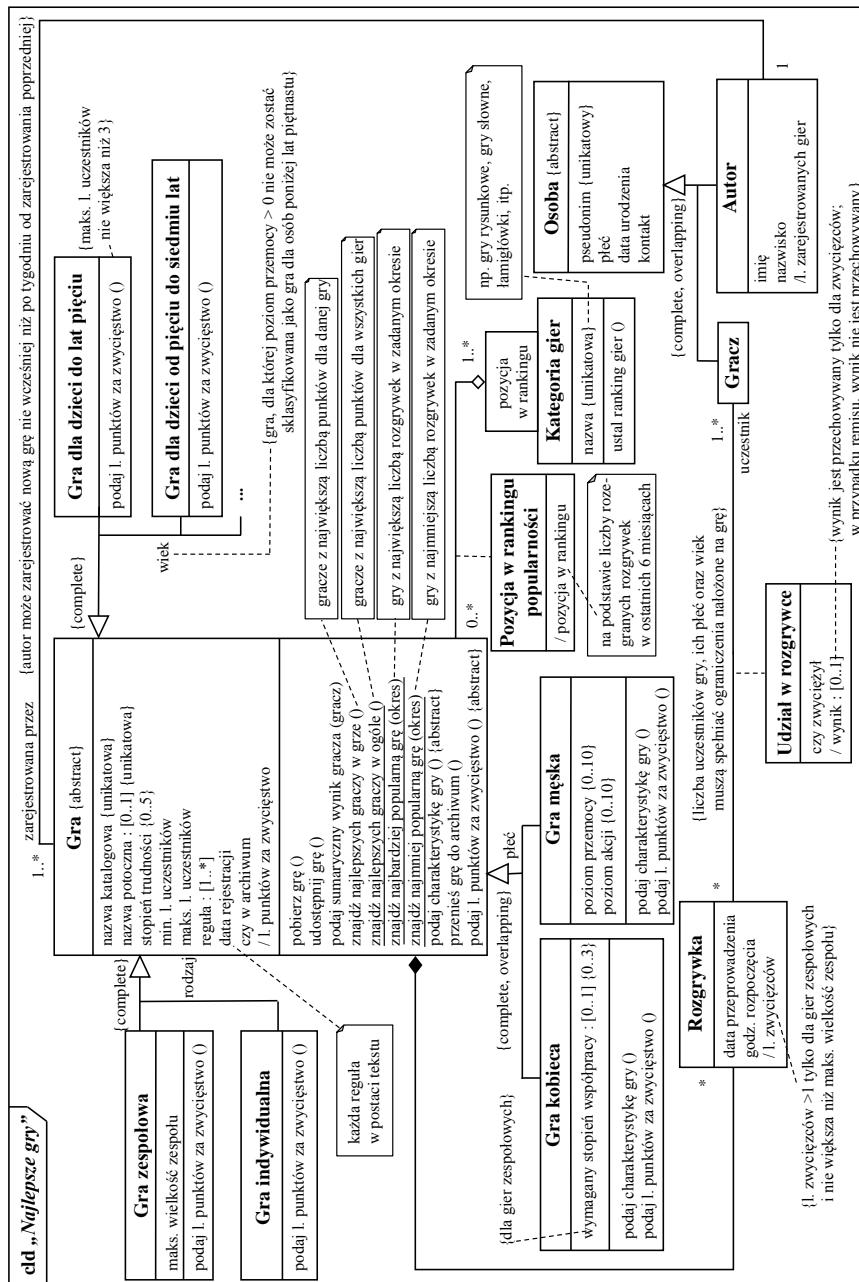
Schemat pojęciowy [36, 38, 40, 41] dla systemu „Najlepsze gry” przedstawiono na Rys. 5-8.

*Uwaga.* Puste sekcje atrybutów w specjalizacjach klasy *Gra* w aspekcie *wiek* zostały opuszczone ze względu na brak miejsca, chociaż nie jest to zalecane.

Na podstawie tekstu wymagań tylko zwycięzcy rozgrywki otrzymują punkty za udział w grze. W przypadku gry indywidualnej punkty za wygraną może uzyskać tylko jeden gracz, natomiast w przypadku gry zespołowej wszyscy uczestnicy zespołu zwycięskiego otrzymują taką samą liczbę punktów wyliczoną dla danej gry. Taki sposób przyznawania punktów nie wymaga pamiętania informacji o zespołach biorących udział rozgrywkach – jak na Rys. 5-8, ale wydaje się, że wprowadzenie dodatkowej informacji o zespołach – uczestnikach gry zwiększyłoby czytelność modelu pojęciowego. Dzięki temu byłaby także możliwa ewentualna klasyfikacja zespołów biorących udział w rozgrywkach. Alternatywne rozwiązanie zostało przedstawione na Rys. 5-7.



Rys. 5-7 Fragment schematu pojęciowego – wersja z klasą *Zespół*



Rys. 5-8 Schemat pojęciowy dla systemu „Najlepsze gry”

### 5.3.2 Analiza elementów pochodnych

Na schemacie pojęciowym z Rys. 5-8 znajdują się następujące atrybuty pochodne:

- *l. punktów za zwycięstwo* w klasie *Gra*;
- *wynik* w klasie asocjacji *Udział w rozgrywce*;
- *l. zwycięzców* w klasie *Rozgrywka*;
- *pozycja w rankingu* w klasie asocjacji *Pozycja w rankingu popularności*;
- oraz *l. zarejestrowanych gier* w klasie *Autor*.

Wartość atrybutu *l. punktów za zwycięstwo* w klasie *Gra* jest wyliczana przy pomocy metod polimorficznych *podaj l. punktów za zwycięstwo ()*. Ponieważ liczba punktów nie ulega zmianie po zarejestrowaniu gry i potencjalnie może być często wykorzystywana (np. w metodzie *podaj charakterystykę gry ()* lub w metodzie rejestrującej wyniki rozgrywki – brak jej na schemacie pojęciowym z Rys. 5-8), dlatego atrybut *l. punktów za zwycięstwo* warto zostawić jako atrybut pochodny.

Atrybut *wynik* w klasie asocjacji *Udział w rozgrywce* jest opcjonalny, gdyż jest przechowywany tylko dla uczestników rozgrywki, którzy zostali zwycięzcami (w przypadku remisu także nie przechowujemy wyniku). Jego wartość jest równa liczbie punktów za zwycięstwo, a ponieważ ta liczba jest przechowywana w klasie *Gra*, to atrybut ten jest (i pozostanie) atrybutem pochodnym.

Wartość atrybutu *l. zwycięzców* w klasie *Rozgrywka* określa liczbę zwycięzców rozgrywki. W przypadku gry indywidualnej zwycięzcą rozgrywki (niezakończonej remisem) może być tylko jeden gracz, natomiast dla gry zespołowej – tylu graczy, ilu wchodziło w skład zwycięskiego zespołu. Ponieważ tylko dla zwycięzców rozgrywki są przyznawane punkty za wygraną, to można by ewentualnie zdefiniować metodę zliczającą uczestników rozgrywki, którzy otrzymali za nią punkty (na podstawie atrybutu *czy zwyciężył* w klasie asocjacji *Udział w rozgrywce*). Taka metoda zwracałaby tę samą wartość od chwili zarejestrowania wyników rozgrywki, dlatego wydaje się, że pozostawienie atrybutu *l. zwycięzców* jako pochodnego uprości implementację operacji podawania informacji o liczbie zwycięzców danej rozgrywki.

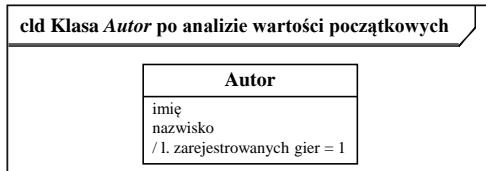
Atrybut *pozycja w rankingu* jest wyliczany pierwszego dnia każdego miesiąca na podstawie liczby rozegranych rozgrywek dla danej gry z ostatnich 6 miesięcy. Jego wartość może ulec zmianie dopiero na początku następnego miesiąca, po wykonaniu kolejnego rankingu popularności gier. Ponieważ aktualna pozycja gry w rankingu popularności może być często wykorzystywana przez graczy (np. podczas wyboru gry do pobrania), dlatego warto pozostawić ten atrybut jako pochodny.

Podobnie w przypadku atrybutu *l. zarejestrowanych gier*, atrybut ten może pozostać w tym momencie pochodny, dla przypomnienia, że na dalszym etapie należy rozważyć jego obsługę, biorąc pod uwagę to, że wartość atrybutu zmienia się tylko w określonym momencie, po zarejestrowaniu nowej gry przez danego autora.

Schemat pojęciowy po uwzględnieniu analizy elementów pochodnych nie uległ zmianie i pozostał jak na Rys. 5-8.

### 5.3.3 Analiza wartości początkowych

Na schemacie pojęciowym z Rys. 5-8 można określić wartość początkową równą 0 dla atrybutu *l. zarejestrowanych gier = 1* w klasie Autor (jak na Rys. 5-9). Natomiast nie ma żadnych metod z określonymi wartościami domyślnymi (początkowymi).



Rys. 5-9 Klasa *Autor* z uwzględnioną analizą wartości początkowych

### 5.3.4 Analiza wartości granicznych

Podczas dalszej analizy schematu pojęciowego (z Rys. 5-8) zostaną omówione występujące na nim ograniczenia oraz związane z nimi wartości graniczne. W szczególności należy rozważyć takie modyfikacje schematu pojęciowego, aby zmiana wartości granicznych nie powodowała konieczności zmiany w kodzie. Syntetyczne zestawienie wybranych wartości granicznych wraz z proponowanymi modyfikacjami schematu pojęciowego przedstawiono w Tab. 5-3.

Tab. 5-3 Lista wartości granicznych, ograniczeń i propozycje zmian w schemacie pojęciowym

0-5, stopień trudności	$\{0..5\}$ nałożone na atrybut <i>stopień trudności</i> w klasie <i>Gra</i>	dodatkowe atrybuty o zasięgu klasowym z zainicjalizowanymi wartościami w klasie <i>Gra</i> : <u>min. stopień trudności = 0</u> <u>maks. stopień trudności = 5</u>  zmiana ograniczenia do postaci <i>{w zakresie min.-maks.}</i>
tydzień, minimalny okres od rejestracji gry autora	<i>{autor może zarejestrować nową grę nie wcześniej niż po tygodniu od zarejestrowania poprzedniej}</i>  nałożone na asocjację <i>zarejestrowana przez</i> między klasą <i>Gra</i> a klasą <i>Autor</i>	dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością w klasie <i>Gra</i> : <u>min. okres = tydzień</u>  zmiana ograniczenia do postaci <i>{autor może zarejestrować nową grę nie wcześniej niż po upływie min. okresu od zarejestrowania poprzedniej}</i>
15 lat, minimalny	<i>{gra, dla której poziom</i>	dodatkowy atrybut o zasięgu

wiek gracza dla gier z poziomem przemocy > 0	$\{przemocy > 0 \text{ nie może zostać sklasyfikowana jako gra dla osób poniżej lat piętnastu}\}$  nałożone na obiekt podklasy klasy <i>Gra</i> w aspekcie <i>wiek</i>	klasowym z zainicjalizowaną wartością w klasie <i>Gra</i> : <u>min. wiek gracza dla gier z przemocą = 15 lat</u>
3, maksymalna liczba uczestników gry dla dzieci do lat pięciu	$\{maks. l. uczestników nie większa niż 3\}$  nałożone na obiekt klasy <i>Gra dla dzieci do lat pięciu</i>	zmienna ograniczenia do postaci $\{gra, dla której poziom przemocy > 0 \text{ nie może zostać sklasyfikowana jako gra dla osób poniżej min. wieku gracza dla gier z przemocą}\}$  dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością w klasie <i>Gra dla dzieci do lat pięciu</i> : <u>limit uczestników = 3</u>
0-3, wymagany stopień współpracy	ograniczenia $\{0..3\}$ , $\{dla gier zespołowych\}$  nałożone na atrybut opcjonalny <i>wymagany stopień współpracy</i> w klasie <i>Gra kobieca</i>	zmienna ograniczenia do postaci $\{maks. l. uczestników nie większa niż limit uczestników\}$  dodatkowe atrybuty o zasięgu klasowym z zainicjalizowanymi wartościami w klasie <i>Gra kobieca</i> : <u>min. stopień współpracy = 0</u> <u>maks. stopień współpracy = 3</u>
0-10, poziom przemocy	$\{0..10\}$  nałożone na atrybut <i>poziom przemocy</i> w klasie <i>Gra męska</i>	zmienna ograniczenia do postaci $\{wymagany poziom współpracy dla gier zespołowych jest w zakresie min.-maks. stopnia współpracy\}$  dodatkowe atrybuty o zasięgu klasowym z zainicjalizowanymi wartościami w klasie <i>Gra męska</i> : <u>min. poziom przemocy = 0</u> <u>maks. poziom przemocy = 10</u>
0-10, poziom akcji	$\{0..10\}$  nałożone na atrybut <i>poziom akcji</i> w klasie <i>Gra męska</i>	zmienna ograniczenia do postaci $\{poziom przemocy jest w zakresie min.-maks. poziomu przemocy\}$  dodatkowe atrybuty o zasięgu klasowym z zainicjalizowanymi wartościami w klasie <i>Gra męska</i> : <u>min. poziom akcji = 0</u> <u>maks. poziom akcji = 10</u>

		<i>{poziom akcji jest w zakresie min.- maks. poziomu akcji}</i>
--	--	---

Schemat pojęciowy uwzględniający proponowane zmiany po analizie wartości granicznych został zaprezentowany na Rys. 5-10.

### 5.3.5 Zadania podstawowe

Zadania omówione w tym rozdziale są realizowane w oparciu o schemat pojęciowy z Rys. 5-10.

**Zad. 1:** Podkreśl te rodzaje dziedziczenia (rozłączne, nierozłączne, kompletne, niekompletne, jednokrotne, wielokrotne, jednoaspektowe, wieloaspektowe, dynamiczne, elipsa), które uważasz za potrzebne do wykorzystania na diagramie klas. Dla każdego wykorzystanego podziału podaj zastosowane rodzaje dziedziczenia.

- Hierarchia osób: dziedziczenie nierozłączne, kompletne, jednokrotne, jednoaspektowe;
- hierarchia gier: dziedziczenie wieloaspektowe (trzy aspekty: *rodzaj*, *pleć*, *wiek*), rozłączne (aspekty *rodzaj* i *wiek*), nierozłączne (aspekt *pleć*), kompletne (wszystkie aspekty), jednokrotne (wszystkie aspekty), elipsa (aspekt: *wiek*).

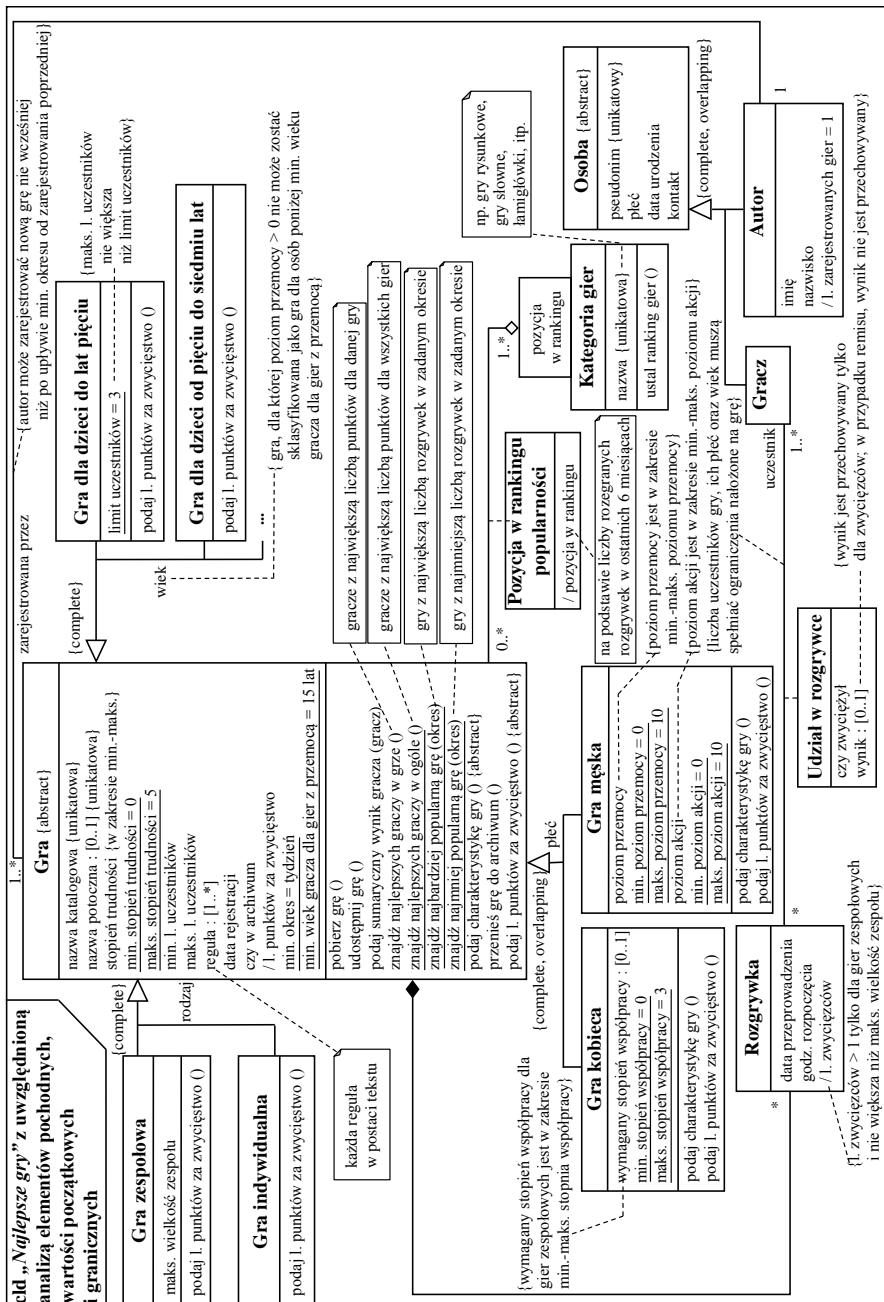
**Zad. 2:** Jaki rodzaje metod (abstrakcyjne, zaimplementowane) może zawierać klasa abstrakcyjna, a jakie klasa konkretna? Podaj przykłady.

Klasa abstrakcyjna może zawierać oba rodzaje metod, a klasa konkretna tylko metody zaimplementowane.

- Metoda abstrakcyjna w klasie abstrakcyjnej: *podaj charakterystykę gry ()* w klasie abstrakcyjnej *Gra*;
- metoda zaimplementowana w klasie abstrakcyjnej: *pobierz grę ()* w klasie abstrakcyjnej *Gra*;
- metoda zaimplementowana w klasie konkretnej: *podaj charakterystykę gry ()* w klasie konkretnej *Gra kobieca*.

**Zad. 3:** Podaj przykłady metody klasowej, metody obiektu, atrybutu klasowego i atrybutu obiektu.

- Metoda obiektu: *pobierz grę ()* w klasie *Gra*;
- metoda klasowa: *znajdź najlepszych graczy w ogóle ()* w klasie *Gra*;
- atrybut obiektu: *nazwa katalogowa* w klasie *Gra*;
- atrybut klasowy: *limit uczestników* w klasie *Gra dla dzieci do lat pięciu* z zainicjalizowaną wartością początkową równą 3.



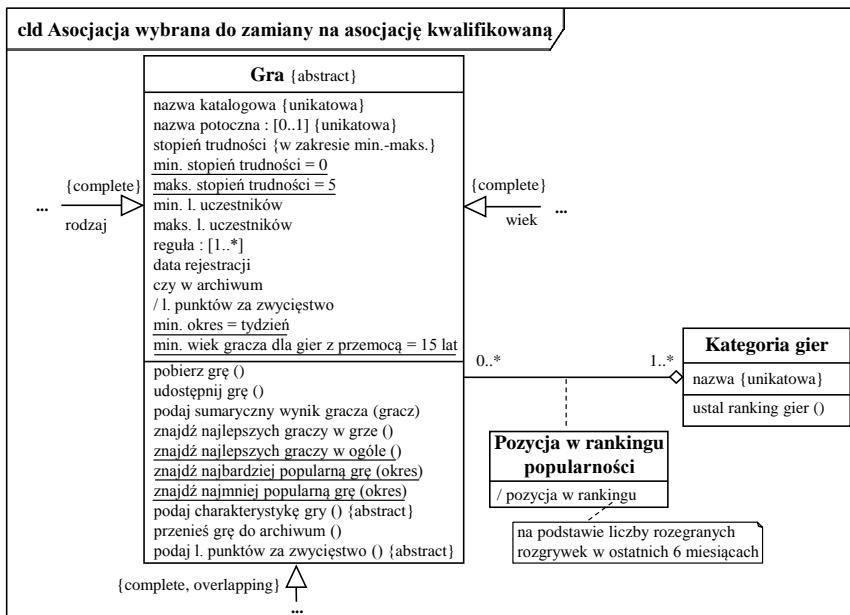
Rys. 5-10 Schemat pojęciowy dla systemu „Najlepsze gry” po uwzględnieniu analizy elementów pochodnych, wartości początkowych i wartości granicznych

**Zad. 4:** Wskaż przykład zastosowania polimorfizmu metod.

Metody polimorficzne to metody: *podaj charakterystykę gry ()* w klasach *Gra kobieta* i *Gra męska* oraz metody *podaj l. punktów za zwycięstwo ()* w podklasach klasy *Gra* – mają identyczne sygnatury i różne implementacje.

**Zad. 5:** Asocjację z diagramu klas, posiadającą niesymetryczne liczności oraz atrybut (lub klasę asocjacji), zamień na asocjację kwalifikowaną. Uzasadnij, dlaczego poddałeś zamianie właśnie tę asocjację.

Asocjacja kwalifikowana została od razu umieszczona na diagramie klas. Do budowy kwalifikatora posłużył atrybut *pozycja w rankingu*, dzięki któremu możliwe jest ustalenie gier (nawet więcej niż jednej), które w danej kategorii gier mają określoną pozycję w rankingu. Zbiór gier znajdujących się na danej pozycji w rankingu jest podzbiorem zbioru wszystkich gier przypisanych do danej kategorii. Odpowiedni fragment diagramu klas bez zastosowania asocjacji kwalifikowanej przedstawia Rys. 5-11.



Rys. 5-11 Fragment diagramu klas bez zastosowania asocjacji kwalifikowanej

**Zad. 6:** Wskaż na diagramie klas asocjację, która jest agregacją lub kompozycją (lub może kwalifikować się do zamiany na takową). Objasnij, dlaczego zdecydowałeś się na wybór agregacji/kompozycji, a nie „zwykłej” asocjacji? Ponadto, dlaczego wybrałeś agregację, a nie kompozycję (czy też odwrotnie)?

Na diagramie została umieszczona agregacja pomiędzy klasami *Gra* i *Kategoria gier*. Agregacja lepiej niż zwykła asocjacja oddaje związek pomiędzy tymi klasami,

związek typu całość-część (gra wchodzi w skład kategorii gier). Ponadto, wybrano agregację (a nie kompozycję), ponieważ cykl życiowy części nie zawiera się w cyklu życiowym całości – jeden obiekt klasy *Gra* może wchodzić w skład wielu obiektów klasy *Kategoria gier*.

**Zad. 7:** Wskaż na diagramie klas przykładowe ograniczenie odnoszące się do asocjacji. Zaproponuj, jaka metoda i w jakiej klasie mogłaby być odpowiedzialna za jego zapewnienie.

Jako przykład ograniczenia na asocjację można tu podać warunek nakładany na możliwość zarejestrowania nowej gry przez jej autora: *autor może zarejestrować nową grę nie wcześniej niż po upływie min. okresu od zarejestrowania poprzedniej*. Metoda rejestruj grę (dane gry, autor) w klasie *Gra* mogłaby być odpowiedzialna za jego zapewnienie. Jest to ograniczenie dynamiczne.

### 5.3.6 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji

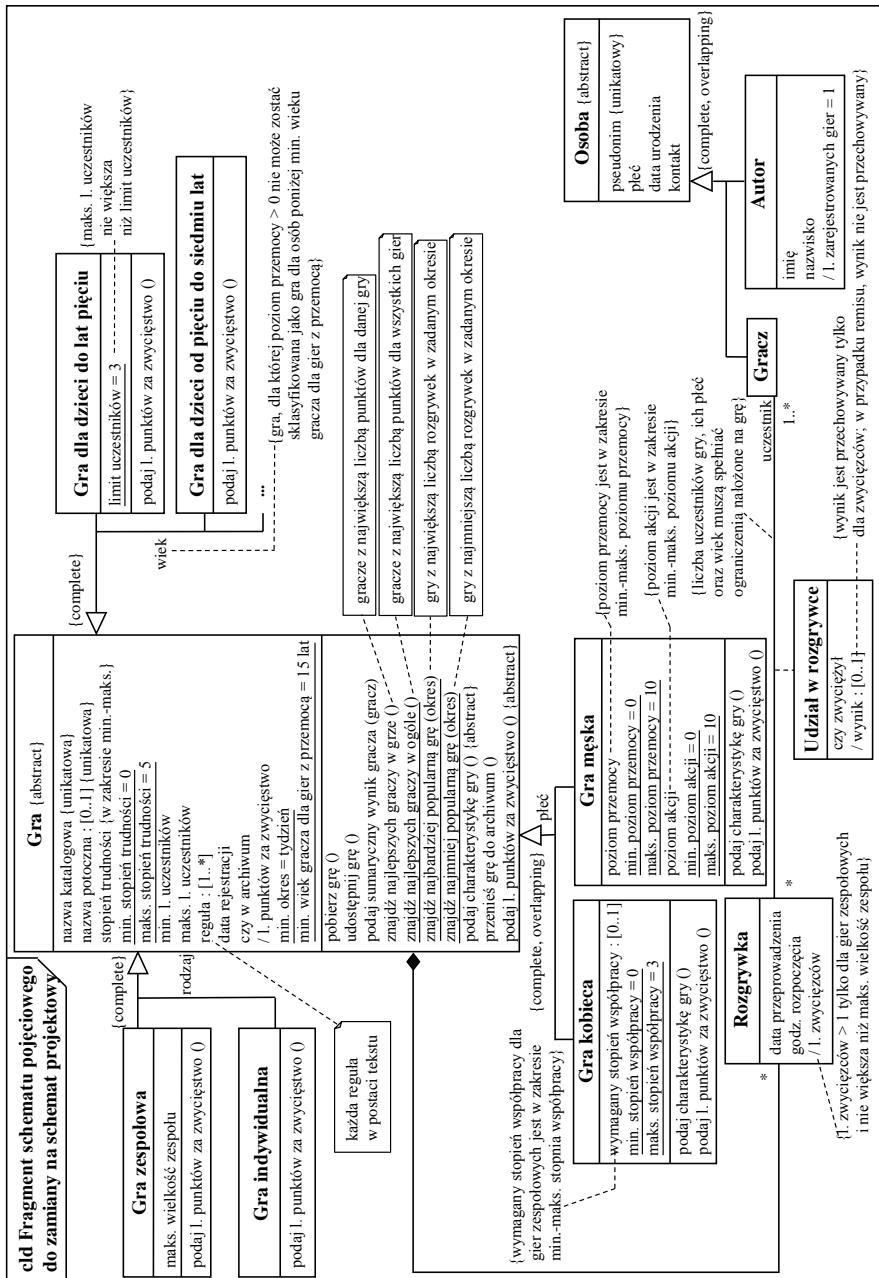
Rys. 5-12 ilustruje fragment schematu pojęciowego, niezbędny dla zamodelowania diagramu interakcji dla części wykonawczej przypadku użycia *Znajdź najlepszych graczy w grze ()*, przy założeniu implementacji w popularnym języku obiektowym (jak np. Java, C#) [37, 40].

Pierwszy krok transformacji fragmentu schematu pojęciowego polega na eliminacji dziedziczenia wieloaspektowego dla gier i dziedziczenia nierożłącznego dla osób.

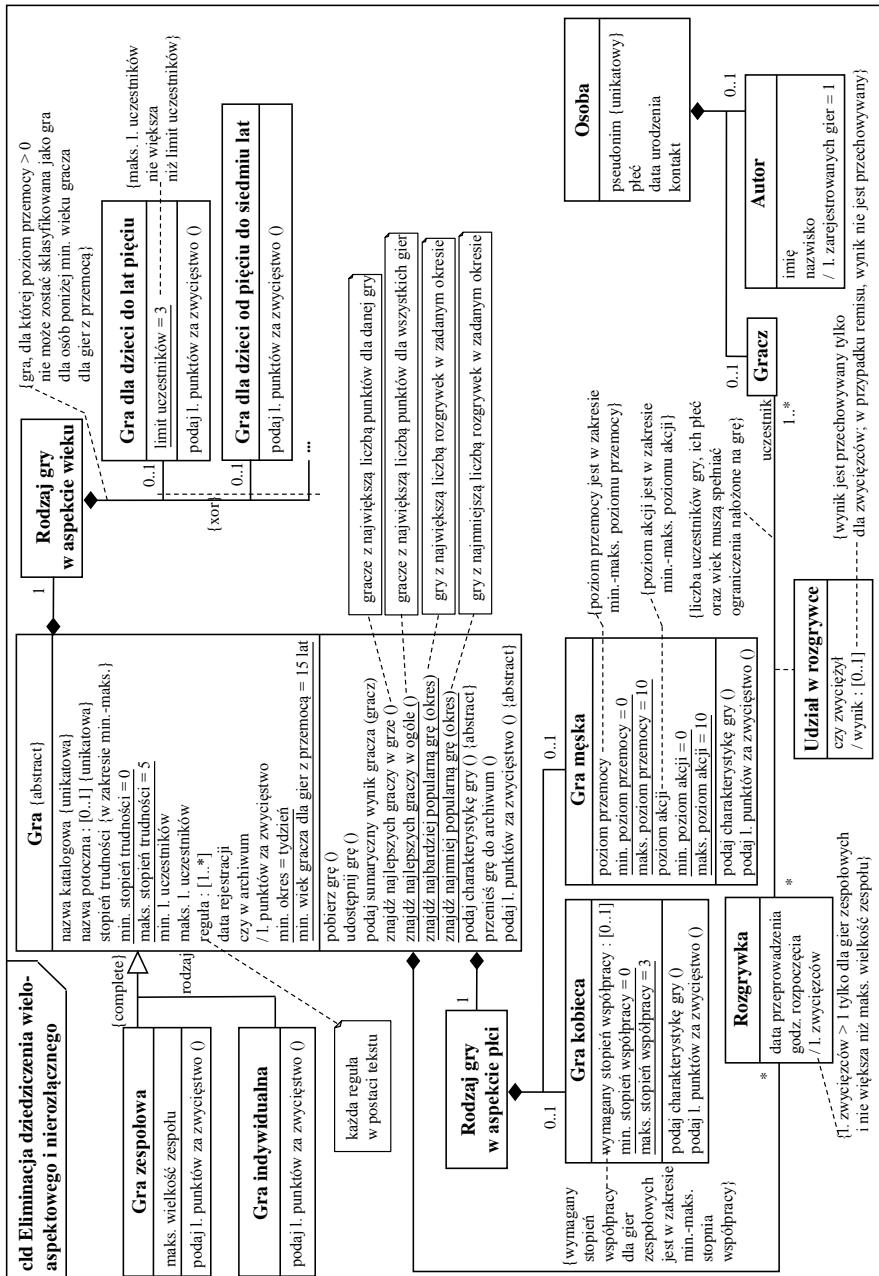
W rozwiązaniu zaprezentowanym na Rys. 5-13 hierarchia dziedziczenia dla gier w dwóch aspektach: *wiek* (z dziedziczeniem rozłącznym i elipsą) i *pleć* (z dziedziczeniem nierożłącznym) została zastąpiona za pomocą kompozycji, głównie ze względu na metody polimorficzne *podaj l. punktów za zwycięstwo ()*. Dla likwidowanych aspektów wprowadzone zostały klasy je reprezentujące *Rodzaj gry w aspekcie płci* i *Rodzaj gry w aspekcie wieku*. Trzeci aspekt – *rodzaj* jest łatwo implementowalny i umożliwia w prosty sposób wprowadzenie ewentualnych zmian dla rozwiązania alternatywnego, polegającego na dodaniu do schematu pojęciowego klasy *Zespół* (jak na Rys. 5-7), dlatego pozostał w niezmienionej postaci.

Druga hierarchia dziedziczenia określona na schemacie pojęciowym (z Rys. 5-12) dla osób, z dziedziczeniem nierożłącznym i asocjacjami prowadzącymi do podklas została zastąpiona także przez kompozycję. Na Rys. 5-14 zilustrowano alternatywną wersję dla realizacji tego fragmentu schematu pojęciowego. Podklasy *Gracz* i *Autor* zostały „zwinięte” do nadklasy *Osoba*; ta wersja nie została uwzględniona w dalszej analizie zadania, między innymi ze względu na rozmycie różnicy między graczem a autorem: niezbędne byłyby zmiany dla asocjacji, które dotychczas prowadziły do klas *Gracz* lub *Autor*, tak aby prowadziły do klasy *Osoba* oraz konieczne byłoby wprowadzenie dodatkowych ograniczeń dla asocjacji po zmianach, tak aby rozgrywki były powiązane z osobami (uczestnikami rozgrywki), którzy są graczami,

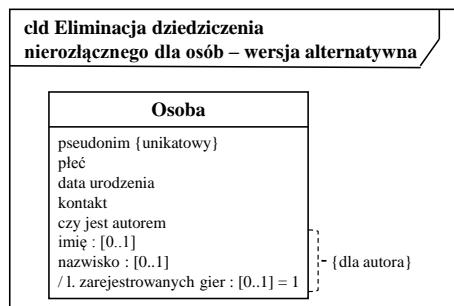
zaś asocjacja *zarejestrowana przez* mogła wiązać gry tylko z osobami, które są autorami.



Rys. 5-12 Fragment schematu pojęciowego do zamiany na schemat projektowy



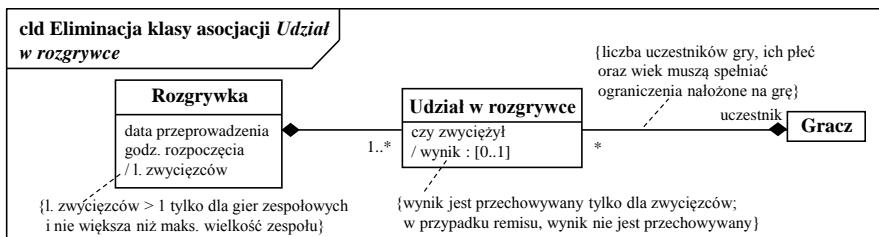
Rys. 5-13 Eliminacja dziedziczenia wieloaspektowego dla gier i nierożłącznego dla osób

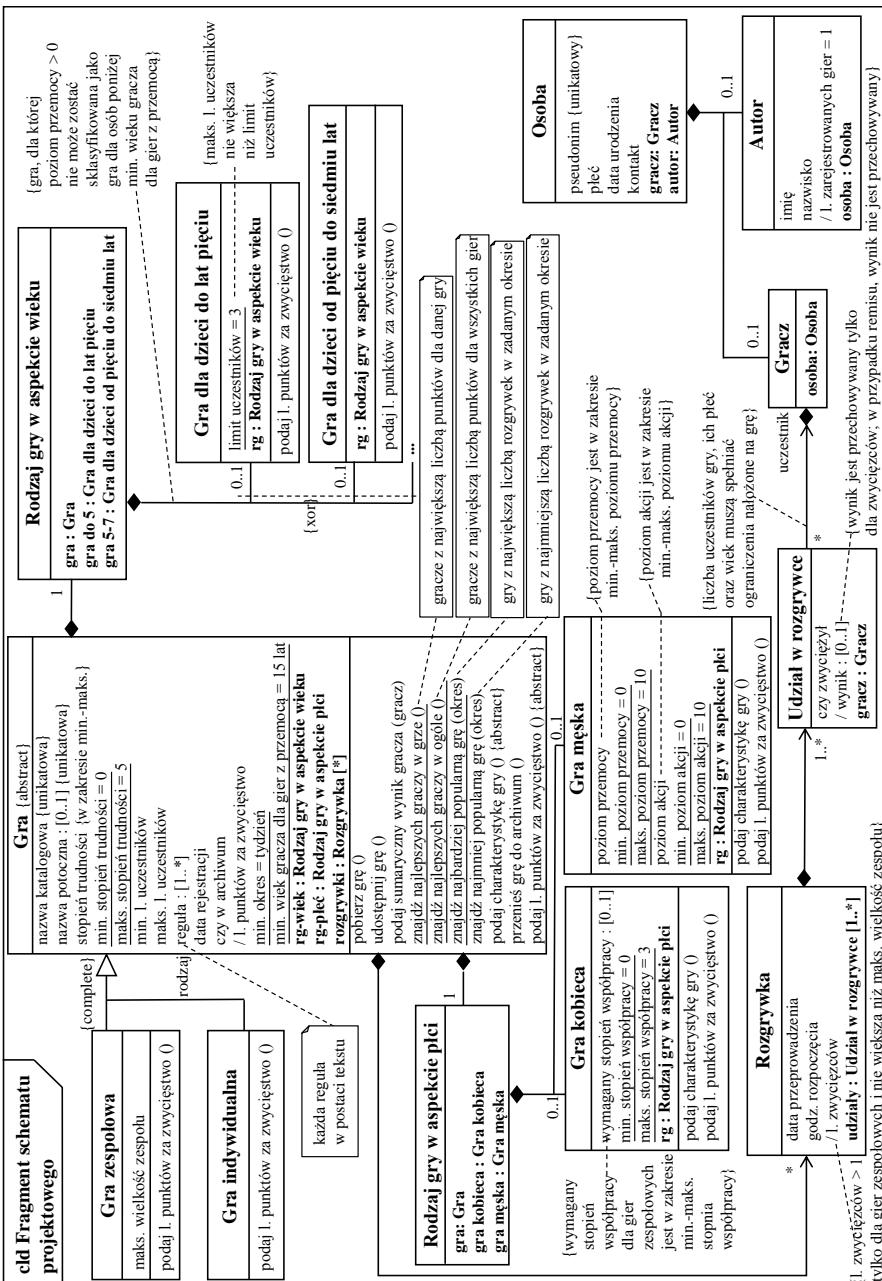


Rys. 5-14 Eliminacja dziedziczenia nierożłącznego dla osób – wersja alternatywna

W drugim kroku transformacji fragmentu schematu pojęciowego do schematu projektowego asocjację między klasą *Rozgrywka* a klasą *Gracz* z klasą asocjacji *Udział w rozgrywce* zastąpiono przy dwiema kompozycjami i klasą pośredniczącą *Udział w rozgrywce* (Rys. 5-15). Z tekstu wymagań, dla efektywnej implementacji metod w klasie *Gra*, wynika konieczność nawigacji tylko w jedną stronę, tj. od gry do jej rozgrywek oraz od rozgrywki do jej uczestników – graczy, dlatego kompozycja między klasą *Gra* a klasą *Rozgrywka* oraz kompozycje między klasą *Rozgrywka* a klasą *Udział w rozgrywce* oraz między klasą *Udział w rozgrywce* a klasą *Gracz* zostały przekształcone na asocjacje skierowane na schemacie projektowym. Należy jednak zaznaczyć, że tekst wymagań mówi o niewielkiej liczbie usług, dlatego w przypadku ewentualnego rozszerzenia funkcjonalności systemu możliwe, że potrzebna byłaby implementacja rozważanej asocjacji jako nawigowalnej w obu kierunkach.

Rys. 5-16 ilustruje ostateczny fragment schematu projektowego (po transformacji fragmentu schematu pojęciowego z Rys. 5-12), na którym wprowadzono realizację asocjacji w typowym języku programowania obiektowym za pomocą dodatkowych atrybutów. Dla zwiększenia czytelności diagramu, atrybuty przechowujące referencje do nawigowalnych ról asocjacji zostały wytluszczone.

Rys. 5-15 Eliminacja klasy asocjacji *Udział w rozgrywce*



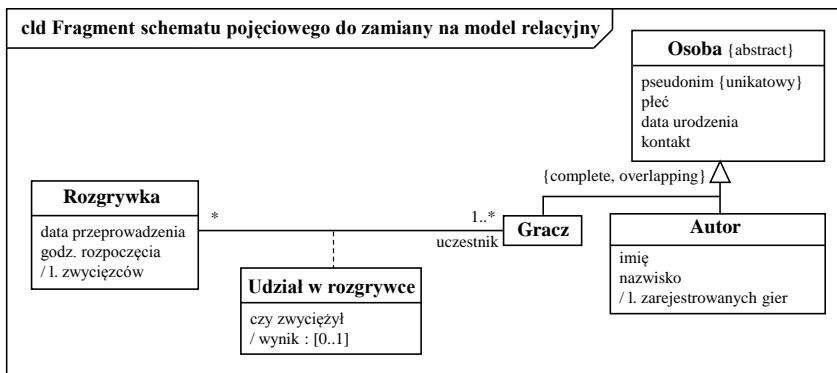
Rys. 5-16 Fragment schematu projektowego

### 5.3.7 Schemat relacyjny dla fragmentu diagramu klas

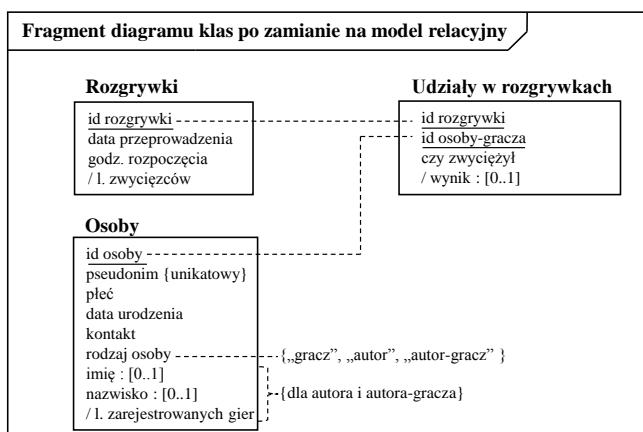
Rys. 5-17 przedstawia fragment diagramu klas, zawierający cztery klasy, związek dziedziczenia i asocjację z klasą asocjacji.

Rys. 5-18 ilustruje model relacyjny [44, 48] tego fragmentu diagramu klas w notacji 1 [41].

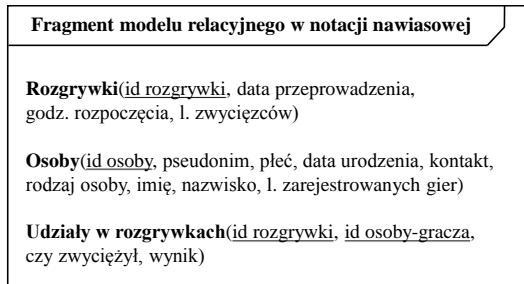
Rys. 5-19 zawiera model relacyjny w notacji nawiasowej, zaś Rys. 5-20 – w notacji *Crow's Foot*.



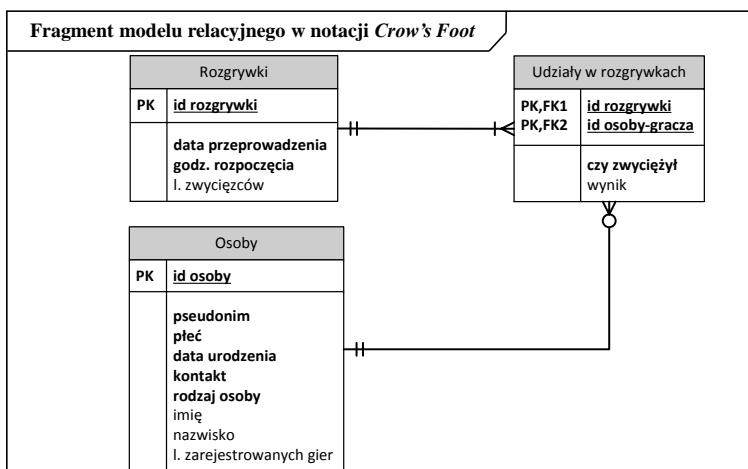
Rys. 5-17 Fragment diagramu klas podlegający zamianie na model relacyjny



Rys. 5-18 Model relacyjny fragmentu diagramu klas przedstawionego na Rys. 5-17 (w notacji 1)



Rys. 5-19 Model relacyjny fragmentu diagramu klas przedstawionego na Rys. 5-17 w notacji nawiasowej



Rys. 5-20 Model relacyjny fragmentu diagramu klas przedstawionego na Rys. 5-17 w notacji *Crow's Foot*

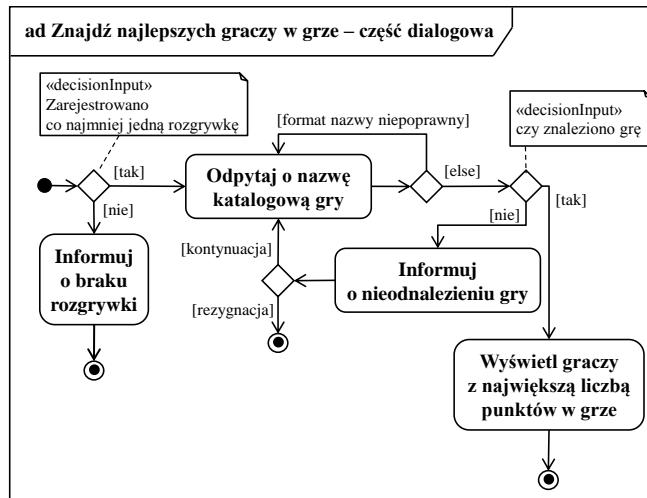
## 5.4 Analiza dynamiczna

### 5.4.1 Zadania podstawowe

Zadania omówione w tym rozdziale są realizowane w oparciu o schemat projektowy z Rys. 5-16.

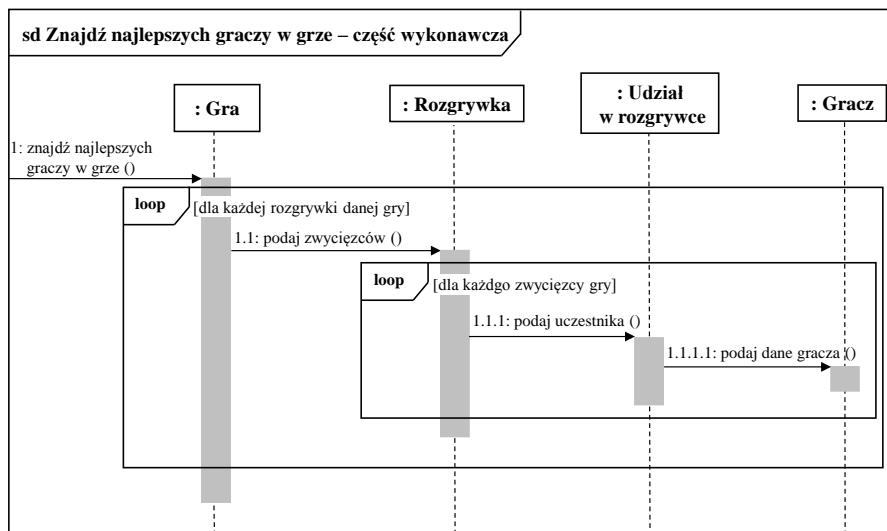
**Zad. 1:** Dla części dialogowej funkcjonalności *Znajdź najlepszych graczy w grze* skonstruuj diagram aktywności zgodny ze scenariuszem z Tab. 5-1.

Przykładową realizację dla scenariusza sporzązonego w zadaniu poprzednim ilustruje diagram aktywności zawarty na Rys. 5-21. Diagram przedstawia część dialogową realizacji przypadku użycia.

Rys. 5-21 Diagram aktywności dla przypadku użycia *Znajdź najlepszych graczy w grze*

**Zad. 2:** Sporządź diagram interakcji dla części wykonawczej przypadku użycia *Znajdź najlepszych graczy w grze*.

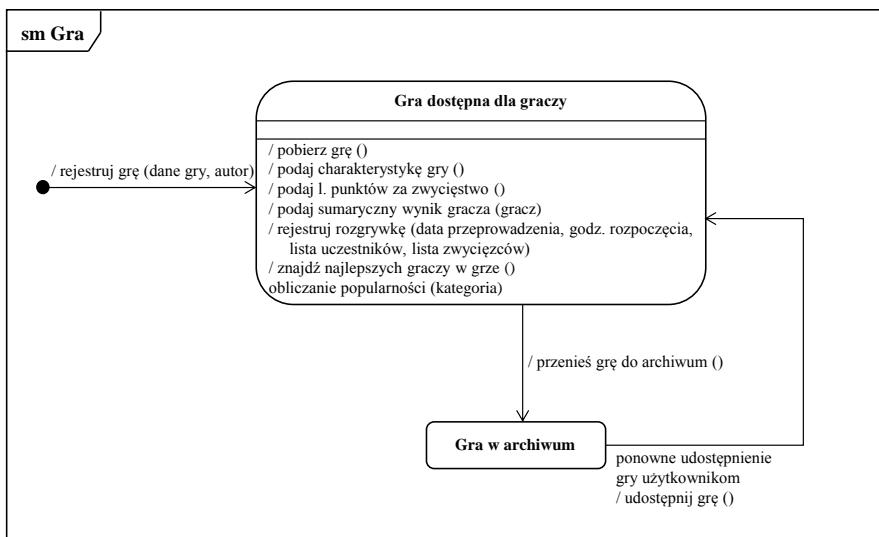
Przykładowy diagram sekwencji dla przypadku użycia *Znajdź najlepszych graczy w grze* ilustruje Rys. 5-22.

Rys. 5-22 Diagram sekwencji dla przypadku użycia *Znajdź najlepszych graczy w grze*

**Zad. 3:** Dla klasy *Gra* wskaż kilka stanów (co najmniej dwa stanów, nie licząc stanów początkowego i końcowego), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów, zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

*Uwaga:* W tym zadaniu nie jest wymagana cała maszyna stanowa, wystarczy zaprezentować dwa stany wraz ze specyfikacją przejścia.

Maszyna stanowa dla klasy *Gra* została przedstawiona na Rys. 5-23. *Uwaga:* na diagramie nie oznaczono stanu końcowego, ponieważ w tekście wymagań nie określono warunków dla usuwania obiektów klasy *Gra*.



Rys. 5-23 Maszyna stanowa dla klasy *Gra*

#### 5.4.2 Omówienie zmian na schemacie pojęciowym i projektowym, wynikających z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań

Po przeprowadzeniu analizy dynamicznej należy uzupełnić schemat pojęciowy i schemat projektowy nowe elementy, które zostały ujęte na diagramach interakcji oraz diagramach stanów.

Lista zmian do schematu pojęciowego została zamieszczona w Tab. 5-4, natomiast lista zmian do schematu projektowego została zamieszczona w Tab. 5-5

Tab. 5-4 Lista zmian na schemacie pojęciowym po analizie diagramu interakcji (Rys. 5-22) i maszyny stanowej (Rys. 5-23)

Klasa	Zmiana
Gra	dodatkowe metody: <i>oblicz popularność (kategoria)</i> <u>rejestruj grę (dane gry, autor)</u> <i>rejestruj rozgrywkę (data przeprowadzenia, godz. rozpoczęcia, lista uczestników, lista zwycięzców)</i>

Tab. 5-5 Lista zmian na schemacie projektowym po analizie diagramu interakcji (Rys. 5-22) i maszyny stanowej (Rys. 5-23)

Klasa	Zmiana
Gra	dodatkowe metody: <i>oblicz popularność (kategoria)</i> <u>rejestruj grę (dane gry, autor)</u> <i>rejestruj rozgrywkę (data przeprowadzenia, godz. rozpoczęcia, lista uczestników, lista zwycięzców)</i>
Gracz	dodatkowa metoda: <i>podaj dane gracza ()</i>
Rozgrywka	dodatkowa metoda: <i>podaj zwycięzców ()</i>
Udział w rozgrywce	dodatkowa metoda: <i>podaj uczestnika ()</i>

## 5.5 Zadania do samodzielnej pracy

- Objaśnij, dlaczego aktor *Podsystem czasu* jest traktowany inaczej niż inne podsystemy w modelu przypadków użycia? Czy aktor *Podsystem czasu*

- wystąpił w modelu przypadków użycia skonstruowanym na podstawie tekstu wymagań?
2. Objaśnij różnicę między dziedziczeniem *nierozłącznym* (ang. overlapping) a *dziedziczeniem wielokrotnym* (ang. multiple inheritance). Czy któryś z tych obu rodzajów dziedziczenia wystąpił na schemacie pojęciowym skonstruowanym na podstawie tekstu wymagań?
  3. Podaj definicję *klasy abstrakcyjnej*. Czy klasa abstrakcyjna może być zdefiniowana wyłącznie w korzeniu hierarchii klas? Czy klasa w korzeniu hierarchii klas zawsze musi być klasą abstrakcyjną? Oznacz klasy abstrakcyjne na schemacie pojęciowym skonstruowanym na podstawie tekstu wymagań.
  4. Objaśnij różnicę między *przeciążaniem* a *przesłanianiem metod*. Wskaż przykłady wykorzystania tych pojęć na schemacie pojęciowym skonstruowanym na podstawie tekstu wymagań. Czy przeciążanie jest rodzajem polimorfizmu metod?
  5. Czy *zasadę zamienialności* można stosować do obiektów klas tworzących agregację? Odpowiedź uzasadnij. Jeśli odpowiedź jest na „tak”, wskaż przykład jego zastosowania na schemacie pojęciowym skonstruowanym na podstawie tekstu wymagań.
  6. Objaśnij różnicę pomiędzy komentarzem a ograniczeniem. Podaj odpowiednie przykłady dla schematu pojęciowego skonstruowanego na podstawie tekstu wymagań.
  7. Dla schematu pojęciowego skonstruowanego na podstawie tekstu wymagań podaj przykład *ograniczenia statycznego* i *ograniczenia dynamicznego*. Uzasadnij swój wybór. Na jakie elementy modelu zostały nałożone te ograniczenia? Do jakiej metody i w jakiej klasie włożyłbyś ich implementację?
  8. Dla wybranego przypadku użycia systemu (innego niż omówionego szczegółowo w analizie funkcjonalnej lub dynamicznej) sformułuj algorytm i narysuj diagram aktywności, określając przynajmniej dwie aktywności, nie licząc początkowej i końcowej.
  9. Dla przypadku użycia wybranego w zadaniu 8 narysuj diagram interakcji, zgodny ze sformułowanym algorytmem, który przedstawia interakcje pomiędzy obiekttami co najmniej dwóch różnych klas.
  10. Dla dowolnej klasy (innej niż omówionej szczegółowo w analizie dynamicznej) ze schematu pojęciowego wskaż dwa stany (wyłączając stan początkowy i stan końcowy), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów, zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

## 5.6 Podsumowanie zadania

W Tab. 5-6 przedstawiono ogólną charakterystykę omawianego zadania. Szczegółową charakterystykę zadania dla grup tematycznych: *Wprowadzenie do przedmiotu*, *Analiza funkcjonalna*, *Analiza strukturalna*, *Analiza dynamiczna* zamieszczono odpowiednio w tabelach Tab. 5-7, Tab. 5-8, Tab. 5-9 i Tab. 5-10.

Tab. 5-6 Ogólna charakterystyka zadania

<b>Ogólna charakterystyka zadania</b>	
<i>Nazwa</i>	System „Najlepsze gry”
<i>Dziedzina problemowa</i>	system mógłby znaleźć zastosowanie jako wsparcie dla opracowywania różnego rodzaju gier
<i>Cel</i>	wprowadzenie systemu ułatwi poszukiwanie osób, które mogłyby pracować w roli testerów dla opracowywanych gier
<i>Zakres odpowiedzialności</i>	wieloaspektowa klasyfikacja gier, ewidencja graczy, ewidencja rozgrywek, zarządzanie dostępnością gier, tworzenie zestawień dotyczących wykorzystania (popularności) gier i umożliwiających wyszukanie najlepszych graczy
<i>Stopień trudności</i>	trudne
<i>Rozmiar</i>	12 klas
<i>Użyteczność w grupach tematycznych</i>	
<i>Grupa tematyczna</i>	Użyteczność w grupie tematycznej
<i>Wprowadzenie do przedmiotu</i>	niska <i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> niska <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> niska
<i>Analiza funkcjonalna</i>	średnia <i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> średnia <i>Podsumowanie dotychczasowych działań:</i> średnia <i>Ewaluacja:</i> średnia
<i>Analiza strukturalna</i>	wysoka <i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> wysoka <i>Podsumowanie dotychczasowych działań:</i> wysoka <i>Ewaluacja:</i> wysoka
<i>Analiza dynamiczna</i>	średnia <i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> średnia <i>Podsumowanie dotychczasowych działań:</i> średnia <i>Ewaluacja:</i> wysoka

Tab. 5-7 Szczegółowa charakterystyka zadania dla grupy tematycznej *Wprowadzenie do przedmiotu*

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>	<i>Wprowadzenie do przedmiotu</i>	
<b>Stopień trudności grupy w zadaniu</b>	łatwe	
<b>Poruszane tematy</b>	<i>analiza a projektowanie, dekompozycja diagramów, mechanizmy rozszerzalności, reguły nazewnictwa, poprawa percepcji diagramu/diagramów</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>analiza a projektowanie</i>	łatwe	<i>model pojęciowy, model projektowy</i>
<i>mechanizmy rozszerzalności</i>	łatwe	<i>stereotyp, ograniczenie, komentarz</i>
<i>dekompozycja diagramów</i>	łatwe	podział modelu przypadków użycia na diagramy
<i>reguły nazewnictwa</i>	łatwe	reguły nazewnictwa dla aktorów, przypadków użycia
<i>poprawa percepcji diagramu/diagramów</i>	łatwe	organizacja diagramu przypadków użycia

Tab. 5-8 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza funkcjonalna*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Analiza funkcjonalna</i>
<b>Stopień trudności grupy w zadaniu</b>	średnio trudne

<b>Poruszane tematy</b>	<i>wymagania na system, aktorzy, przypadki użycia, scenariusze, dekompozycja modelu/diagramu przypadków użycia, mechanizmy rozszerzalności</i>		
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>	
<i>wymagania na system</i>	średnio trudne	możliwość wprowadzenia dodatkowej informacji o zespołach	
<i>aktorzy</i>	średnio trudne	<i>aktor abstrakcyjny, aktor Podsystem czasu, hierarchia dla aktorów, stosunkowo duża liczba aktorów</i>	
<i>przypadki użycia</i>	średnio trudne	relacje pomiędzy przypadkami użycia «include» i «extend»	
<i>scenariusze</i>	średnio trudne		
<i>dekompozycja modelu/diagramu przypadków użycia</i>	średnio trudne	<i>dekompozycja pozioma, dekompozycja pionowa, punkty/miejsca rozszerzeń (extension points)</i>	
<i>mechanizmy rozszerzalności</i>	łatwe	ograniczenia nałożone na relacje między przypadkami użycia	

Tab. 5-9 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza strukturalna*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Analiza strukturalna</i>
<b>Stopień trudności grupy w zadaniu</b>	trudne
<b>Poruszane tematy</b>	<i>atrybuty klasy, metody klasy, struktury generalizacji-specjalizacji, klasa abstrakcyjna a klasa konkretna, asocjacje, analiza wartości, przejście na schemat projektowy, przejście na schemat relacyjny, mechanizmy rozszerzalności</i>

<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>atrybuty klasy</i>	średnio trudne	<i>atrybut opcjonalny, powtarzalny, pochodny, obiektowy, klasowy, stosunkowo duża liczba atrybutów pochodnych</i>
<i>metody klasy</i>	średnio trudne	<i>metoda abstrakcyjna, zaimplementowana, obiektowa, klasowa, polimorfizm metod</i>
<i>struktury generalizacji-specjalizacji</i>	trudne	dziedziczenie nierożłączne, wieloaspektowe i elipsa
<i>klasa abstrakcyjna a klasa konkretna</i>	łatwe	<i>klasa abstrakcyjna, klasa konkretna</i>
<i>asocjacje</i>	trudne	<i>rola asocjacji, klasa asocjacji, agregacja, kompozycja, asocjacja kwalifikowana</i>
<i>analiza wartości</i>	średnio trudne	<i>wartość pochodna, wartość początkowa (domyślna), wartość graniczna, stosunkowo duża liczba wartości granicznych</i>
<i>przejście na schemat projektowy</i>	trudne	obejście dziedziczenia nierożłącznego i dziedziczenia wieloaspektowego, zamiana asocjacji z klasą asocjacji, <i>asocjacja skierowana</i>
<i>przejście na schemat relacyjny</i>	średnio trudne	obejście dziedziczenia nierożłącznego, zamiana asocjacji z klasą asocjacji
<i>mechanizmy rozszerzalności</i>	trudne	<i>ograniczenie, komentarz, stosunkowo duża liczba komentarzy i ograniczeń na schemacie pojęciowym</i>

Tab. 5-10 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza dynamiczna*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Analiza dynamiczna</i>
<b>Stopień trudności grupy w zadaniu</b>	średnio trudne
<b>Poruszane tematy</b>	<i>diagramy aktywności, diagramy interakcji, diagramy stanów, podsumowanie analizy dynamicznej</i>

<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia w temacie</b>	<b>omawiane</b>
<i>diagramy aktywności</i>	łatwe	<i>romb decyzyjny branch i merge</i>	
<i>diagramy interakcji</i>	średnio trudne	zagnieżdżone fragmenty wyodrębnione <i>loop</i> na diagramach sekwencji	
<i>diagramy stanów</i>	łatwe		
<i>podsumowanie analizy dynamicznej</i>	łatwe	modyfikacje schematu pojęciowego i schematu projektowego o dodatkowe metody	

## Spis treści rozdziału

Rozdział 6 Muzeum.....	252
6.1 Tekst wymagań.....	252
6.2 Analiza funkcjonalna .....	253
6.2.1 Diagram kontekstowy .....	253
6.2.2 Model przypadków użycia.....	254
6.2.3 Scenariusze dla wybranych przypadków użycia łącznie z ich podziałem na podprzypadki.....	255
6.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	256
6.3 Analiza strukturalna.....	257
6.3.1 Schemat pojęciowy .....	257
6.3.2 Analiza elementów pochodnych .....	257
6.3.3 Analiza wartości początkowych .....	257
6.3.4 Analiza wartości granicznych.....	259
6.3.5 Zadania podstawowe .....	260
6.3.6 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	265
6.3.7 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji .....	266
6.3.8 Schemat relacyjny dla fragmentu diagramu klas .....	268
6.4 Analiza dynamiczna .....	270
6.4.1 Zadania podstawowe .....	270
6.4.2 Omówienie zmian na schemacie pojęciowym i schemacie projektowym, wynikających z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań .....	273
6.5 Zadania do samodzielnej pracy .....	273
6.6 Podsumowanie zadania.....	274

# Rozdział 6

## Muzeum

*Ewa Stemposz, Andrzej Jodłowski*

### 6.1 Tekst wymagań

Dyrekcja muzeum postanowiła wprowadzić komputerowy system wspierający ewidencjonowanie dzieł sztuki, wystaw oraz liczby gości muzeum – w celu wykazania się aktywnością przed Ministerstwem.

1. W systemie mają być przechowywane informacje o salach muzealnych, takie jak: numer i powierzchnia. W danym momencie sala może być zajęta przez wystawę, wolna lub w remoncie. Jeśli sala jest w remoncie, to wtedy trzeba pamiętać datę planowanego zakończenia remontu.
2. W muzeum organizowane są wystawy, dla których należy przechowywać datę rozpoczęcia i datę zakończenia, tytuł (niepowtarzający się dla wystaw odbywających się w tym samym czasie), od 1 do 10 słów kluczowych oraz liczbę widzów, którzy obejrzaeli tę wystawę. Każda wystawa ma swojego lokalnego kustosza, dla którego mają być przechowywane jego imię, nazwisko i kontakt. Dla wszystkich wystaw obowiązuje limit liczby wystawianych eksponatów.
3. Informacje o wystawach nie mają być usuwane po ich zakończeniu.
4. Wystawy dzielą się na indywidualne (o artyście, którego pracom została poświęcona ta wystawa, trzeba przechowywać następujące informacje: imię, nazwisko, kontakt) lub zbiorowe, dla których ma być pamiętany tekst opisujący ideę przewodnią wystawy. System ma umożliwiać przechowywanie informacji także o innych rodzajach wystaw (które są znane), ale nie zajmujemy się nimi w tym momencie.
5. Drugim kryterium podziału wystaw jest czas trwania wystawy. Wystawy dzielone są na stałe i czasowe (dla których okres trwania nie przekracza 2 lat). Dla wystaw czasowych, oprócz kustosza lokalnego, może być wyznaczony dodatkowy kustosz zewnętrzny, którego imię, nazwisko i kontakt mają być pamiętaane w systemie. Dla wystaw stałych przechowywana ma być elektroniczna wersja katalogu.
6. Wystawa zajmuje określone sale muzeum. W trakcie trwania wystawy sala może zawierać eksponaty z jednej tylko wystawy. Dla każdej sali pamiętamy, które eksponaty się w niej znajdują. Po zakończeniu wystawy ma być

przechowywana informacja, które sale były przez nią zajmowane, ale bez przechowywania informacji o rozmieszczeniu eksponatów.

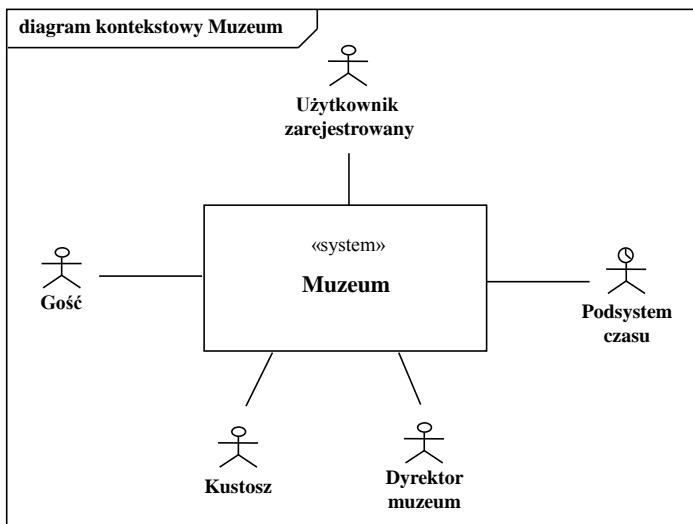
7. Eksponaty powinny być opisywane informacjami, takimi jak: nazwa, numer katalogowy, krótki opis oraz danymi jednego lub więcej artystów, którzy są ich autorami (imię, nazwisko, kontakt, ewentualnie pseudonim). W ramach wystawy każdemu eksponatowi zostaje przypisany unikatowy numer identyfikacyjny. Po zakończeniu wystawy ma być rejestrowana liczba osób, które obejrzały dany eksponat w okresie jej trwania. Osoby takie będą nazywane dalej widzami eksponatu.
8. Eksponaty można wypożyczać. Chcemy pamiętać, jaka instytucja i na jaki czas wypożyczyła dany eksponat. Jeden eksponat nie może być wypożyczony więcej niż 15 razy (aktualnie, ale to może ulegać zmianom w przyszłości). Nie wypożyczamy też eksponatu wcześniej niż pół roku po oddaniu.
9. W celu prowadzenia badań statystycznych, dla każdego eksponatu ma być znana liczba osób, które obejrzały dany eksponat, określana na podstawie łącznej liczby osób, które obejrzały dany eksponat w ramach wystaw, na których był on prezentowany.
10. Po zamknięciu wystawy stałej zajmowane przez nią sale są oddawane do remontu, natomiast po zamknięciu wystawy czasowej zajmowane sale są tylko zwalniane.
11. System ma wspomóc w realizowaniu usług, takich jak na przykład:
  - 11.1 przeglądanie kalendarium aktualnej działalności wystawienniczej;
  - 11.2 przeglądanie wystaw;
  - 11.3 podanie listy eksponatów dla danej wystawy;
  - 11.4 podanie liczby widzów eksponatu na wystawie (tzn. liczby osób, które obejrzały eksponat podczas wystawy);
  - 11.5 podanie liczby dotychczasowych widzów eksponatu;
  - 11.6 zamknięcie wystawy;
  - 11.7 zwolnienie sal zajmowanych przez wystawę;
  - 11.8 dla zadanego okresu znalezienie osoby, która najczęściej pełniła rolę kustosza wystaw organizowanych w tym czasie przez muzeum;
  - 11.9 podanie liczby widzów dla wystaw zakończonych w ubiegłym roku (automatycznie, zawsze pierwszego stycznia).

## 6.2 Analiza funkcjonalna

### 6.2.1 Diagram kontekstowy

*Polecenie:* Zbuduj diagram kontekstowy w oparciu o tekst wymagań zamieszczony w punkcie 6.1 .

Diagram kontekstowy dla systemu *Muzeum* został zilustrowany na Rys. 6-1. Aktor *Podsystem czasu* reprezentuje aktora/ów z najwyższego poziomu abstrakcji dla przypadków użycia wykonywanych automatycznie i związanych z upływem czasu; na dalszym etapie analizy funkcjonalnej zostanie on uszczegółowiony jako konkretny aktor *I stycznia*.



Rys. 6-1 Diagram kontekstowy

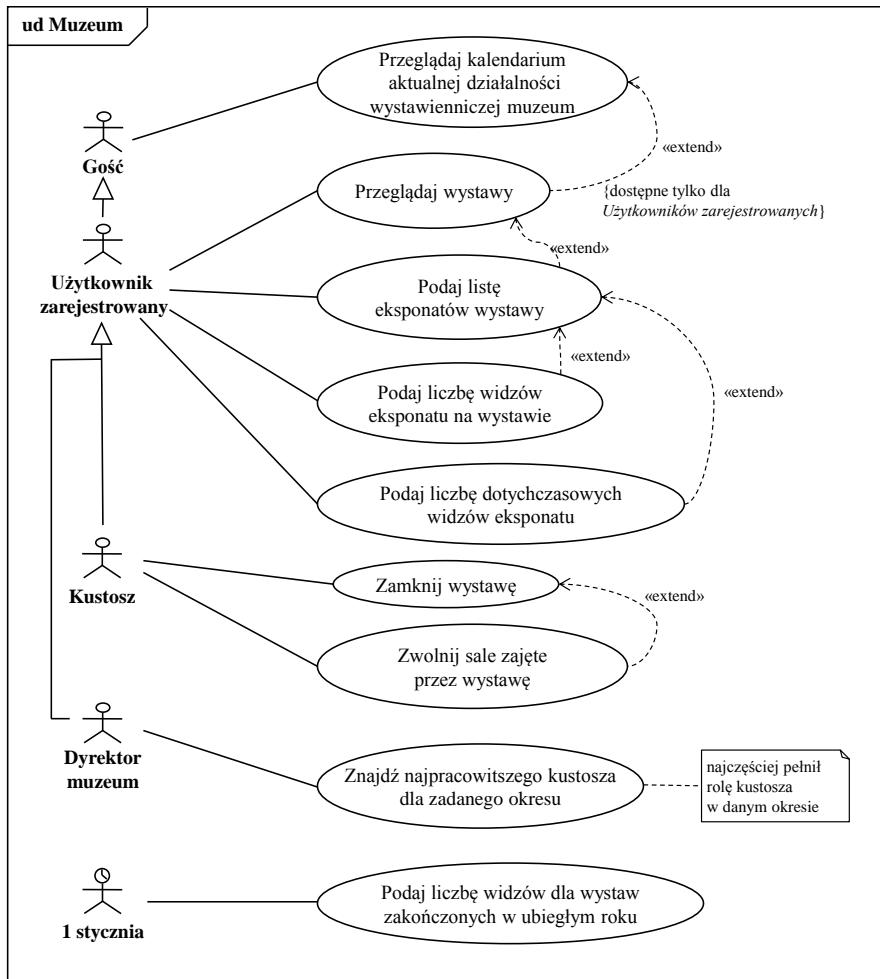
### 6.2.2 Model przypadków użycia

*Polecenie:* Zbuduj model przypadków użycia, uwzględniając wyłącznie funkcjonalność sugerowaną w ostatnim punkcie tekstu wymagań.

*Uwaga:* Model należy skonstruować z perspektywy aktorów z otoczenia systemu (najwyższy poziom abstrakcji modelu), biorąc pod uwagę możliwą ewolucję systemu w przyszłości. Należy uwzględnić hierarchie aktorów i relacje pomiędzy przypadkami (o ile mają/mogłyby mieć miejsce).

Model przypadków użycia [36, 38, 41, 46] został przedstawiony na Rys. 6-2.

Należy zwrócić uwagę na fakt, że nie wszystkie wykorzystane na diagramie relacje między przypadkami użycia zostały explicite wymienione w tekście wymagań, przykładowo: relacje «extend» między przypadkami użycia *Podaj listę eksponatów wystawy* i *Podaj liczbę dotychczasowych widzów eksponatu* oraz *Podaj liczbę widzów eksponatu na wystawie*. Wykorzystując wymienione relacje, przedstawiono propozycje opcjonalnego (warunkowego) rozszerzenia działania przypadku *Podaj listę eksponatów wystawy*.



Rys. 6-2 Model przypadków użycia

### 6.2.3 Scenariusze dla wybranych przypadków użycia łącznie z ich podziałem na podprzypadki

*Polecenie:* Dla funkcjonalności *Zwolnij sale zajęte przez wystawę* napisz scenariusz.

Przykładowy scenariusz dla wybranego przypadku użycia został przedstawiony w Tab. 6-1.

Scenariusz został napisany z punktu widzenia aktora *Kustosz*.

Tab. 6-1 Scenariusz dla przypadku użycia *Zwolnij sale zajęte przez wystawę*

<b>Nazwa przypadku użycia</b>	Zwolnij sale zajęte przez wystawę
<b>Warunek początkowy</b>	W systemie jest zarejestrowana co najmniej jedna wystawa.
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Aktor, którym jest tu <i>Kustosz</i>, uruchamia przypadek.</li> <li>2. System prosi o podanie tytułu wystawy. Aktor podaje tytuł.</li> <li>3. Dla wszystkich sal zajętych przez wystawę system potwierdza zwolnienie sali.</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	<ol style="list-style-type: none"> <li>2a. Wystawa o podanym tytule nie została odnaleziona, system informuje o tym i odpytuje aktora o chęć kontynuowania.             <ol style="list-style-type: none"> <li>2aa. Aktor potwierdza chęć ponownego podania tytułu wystawy.</li> <li>2ab. Aktor rezygnuje z kontynuowania przypadku użycia.</li> </ol> </li> <li>2b. Wystawa o podanym tytule nie została zamknięta, system informuje o tym i kończy przypadek użycia.</li> <li>2c. Wystawa o podanym tytule nie jest wystawą czasową, system informuje o tym i kończy przypadek użycia.</li> </ol>
<b>Zakończenie</b>	Określone w treści scenariusza.
<b>Warunek końcowy</b>	Jeśli wystawa została zamknięta, dla każdej z sal, zajmowanych przez wystawę, status sali przyjmuje wartość „wolna”.

#### 6.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

- Rozwiązywanie zaprezentowane na diagramie Rys. 6-2 jest przykładowe, tym niemniej nie powinni się tu znaleźć tacy aktorzy, jak *Podsystem zliczający liczbę widzów*, ponieważ jest on częścią systemu, a nie bytem z jego otoczenia.
- Błędne nazwy przypadków, jak np.: *Liczba widzów wystawy*. Nazwa przypadku powinna sugerować rodzaj zadania zleconego systemowi przez aktora (a tak nie

jest dla podanego przypadku, którego nazwa jest nazwą danych). Błędny jest również przypadek użycia *Remontuj salę*. Remont sali odbywa się poza systemem i bez wsparcia z jego strony.

## 6.3 Analiza strukturalna

### 6.3.1 Schemat pojęciowy

*Polecenie:* Dla podanego tekstu wymagań skonstruuj schemat pojęciowy (nie redukując liczności).

Schemat pojęciowy [35, 40, 41, 45] dla systemu *Muzeum* został przedstawiony na Rys. 6-3.

### 6.3.2 Analiza elementów pochodnych

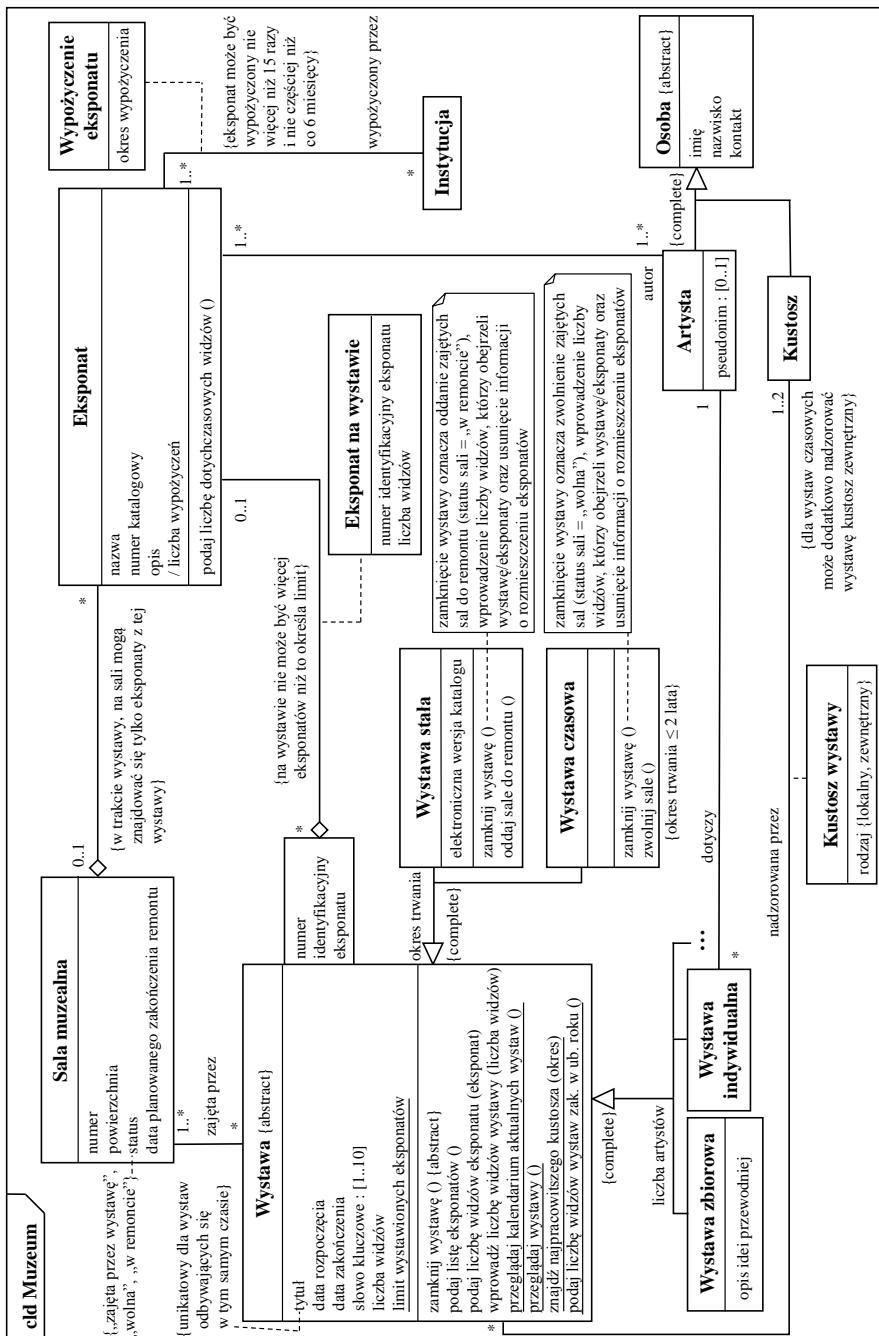
Na schemacie pojęciowym z Rys. 6-3 w klasie *Eksponat* znajduje się atrybut pochodny *liczba wypożyczeń*. Wartość atrybutu jest wyliczana na podstawie liczby wypożyczeń eksponatu (tj. liczby powiązań *wypożyczony przez* dla eksponatu). Złożoność obliczeniowa ewentualnej metody wyliczającej liczbę wypożyczeń byłaby niewielka. Eksponaty będą wypożyczane raczej rzadko. Na dalszym etapie konieczne będzie wprowadzenie metody w klasie *Eksponat*, np. *wypożycz eksponat (instytucja, okres)*, rejestrującej dane wypożyczenia oraz weryfikującej ograniczenia nań nałożone i dlatego – aby przypomnieć o konieczności dostarczenia jego obsługi w trakcie wypożyczeni i zwrotów eksponatów – atrybut *liczba wypożyczeń* powinien pozostać pochodny. Schemat pojęciowy pozostanie więc bez zmian, jak na Rys. 6-3.

### 6.3.3 Analiza wartości początkowych

Analizując schemat pojęciowy z Rys. 6-3, warto rozważyć, czy występują nań atrybuty z wartościami początkowymi (domyślnymi) lub metody z parametrami domyślnymi. Na schemacie pojęciowym z Rys. 6-3 nie ma metod z określonymi parametrami domyślnymi. Gdyby dla funkcjonalności opisanej w punkcie 11.8 tekstu wymagań okresem domyślnym zostało przyjęte pięć ostatnich lat, to specyfikacja metody w klasie *Wystawa* znajdującej najbardziej pracowitego kustosza z zadанego okresu byłaby następująca:

*znajdź najpracowitszego kustosza (okres = pięć ostatnich lat)*

Lista atrybutów z określonymi wartościami początkowymi została przedstawiona w Tab. 6-2. Schemat pojęciowy uwzględniający zaproponowane w Tab. 6-2 wartości początkowe został zaprezentowany na Rys. 6-4.

Rys. 6-3 Schemat pojęciowy dla systemu *Muzeum*

Tab. 6-2 Lista atrybutów z określonymi wartościami początkowymi

Klasa	Atrybut	Wartość początkowa (domyślna)
<i>Wystawa</i>	<i>liczba widzów</i>	<i>0</i>
<i>Eksponat</i>	<i>/ liczba wypożyczeń</i>	<i>0</i>
<i>Sala muzealna</i>	<i>status</i>	<i>„wolna”</i>
	<i>data planowanego zakończenia remontu</i>	<i>null</i>
<i>Eksponat na wystawie</i>	<i>liczba widzów</i>	<i>0</i>

### 6.3.4 Analiza wartości granicznych

Podczas dalszej analizy schematu pojęciowego (z Rys. 6-4) powinny zostać rozważone występujące na nim ograniczenia oraz wartości graniczne. W szczególności należy zastanowić się nad takimi modyfikacjami schematu pojęciowego, aby zmiany wartości granicznych nie wymagały konieczności zmiany w kodzie.

Syntetyczne zestawienie wybranych wartości granicznych wraz z proponowanymi modyfikacjami schematu pojęciowego przedstawiono w Tab. 6-3.

Tab. 6-3 Zestawienie wartości granicznych, ograniczeń i proponowanych zmian w schemacie pojęciowym

Wartość graniczna	Ograniczenie	Zmiany w schemacie pojęciowym
1-10 słów kluczowych	liczność <i>1..10</i> atrybutu powtarzanego <i>słowo kluczowe</i> w klasie <i>Wystawa</i>	dodatkowe atrybuty o zasięgu klasowym z zainicjalizowanymi wartościami w klasie <i>Wystawa</i> : <u>min. liczba słów kluczowych = 1</u> <u>maks. liczba słów kluczowych = 10</u>  zmiana liczności na * dla atrybutu <i>słowo kluczowe</i> w klasie <i>Wystawa</i>  dodatkowe ograniczenie

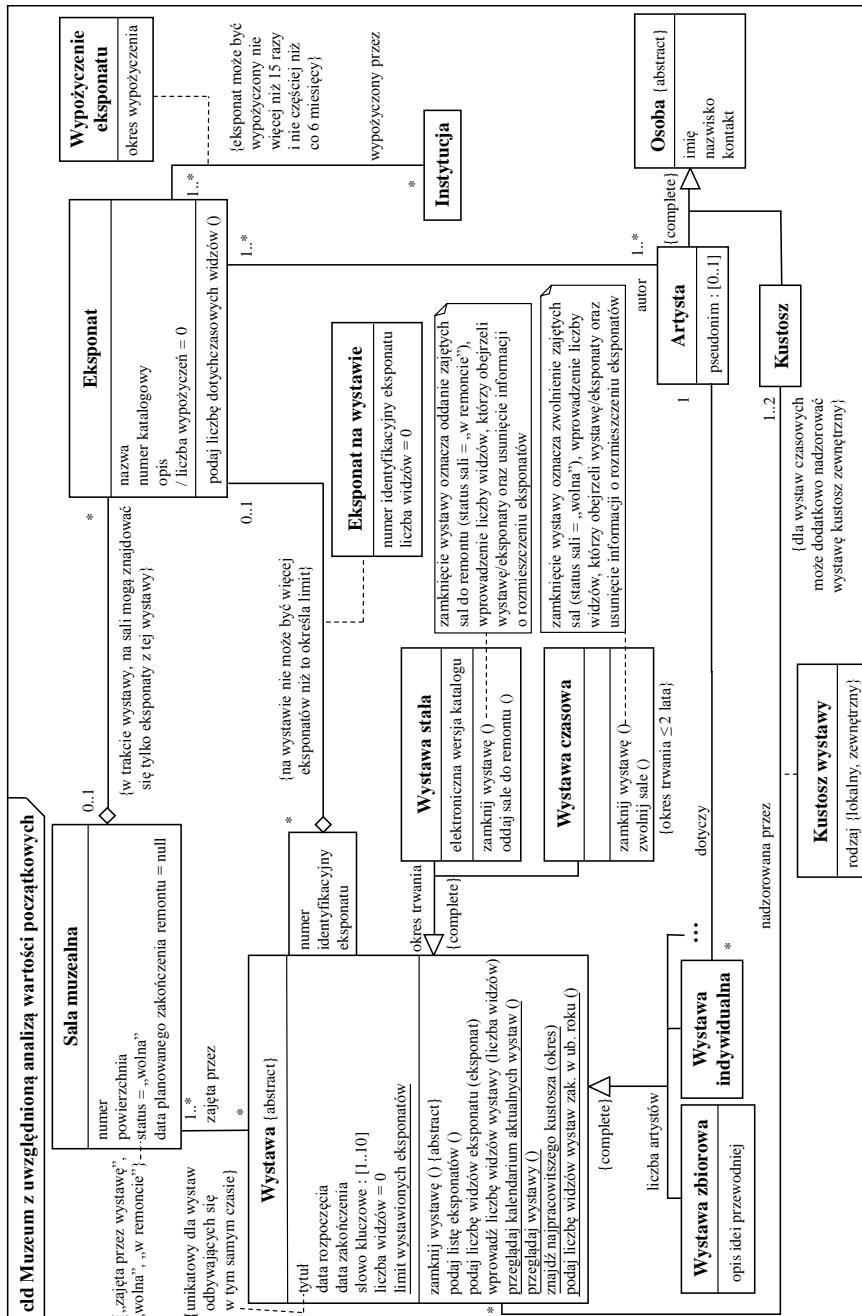
		<p><i>{liczba słów kluczowych mieści się w zakresie min. - maks.}</i>  nałożone na atrybut <i>słowo kluczowe</i> w klasie <i>Wystawa</i></p>
2 lata, <i>okres trwania</i>	<p><i>{okres trwania ≤ 2 lata}</i>  nałożone na obiekt <i>Wystawa czasowa</i></p>	<p>dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością w klasie <i>Wystawa czasowa</i>:  <u><i>maks. okres trwania = 2 lata</i></u></p> <p>zmiana ograniczenia do postaci  <i>{okres trwania ≤ maks. okres trwania}</i></p>
15, <i>limit wypożyczeń</i>	<p><i>{eksponat może być wypożyczony nie więcej niż 15 razy}</i>  nałożone na asocjację między klasą <i>Eksponat</i> a klasą <i>Instytucja</i></p>	<p>dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością w klasie <i>Eksponat</i>:  <u><i>limit wypożyczeń = 15</i></u></p> <p>zmiana ograniczenia do postaci  <i>{nie więcej niż określa limit wypożyczeń}</i></p>
6 miesięcy, <i>okres między kolejnymi wypożyczeniami eksponentu</i>	<p><i>{eksponat może być wypożyczony nie częściej niż co 6 miesięcy}</i>  nałożone na asocjację między klasą <i>Eksponat</i> a klasą <i>Instytucja</i></p>	<p>dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością w klasie <i>Eksponat</i>:  <u><i>min. przerwa pomiędzy wypożyczeniami = 6 miesięcy</i></u></p> <p>zmiana ograniczenia do postaci  <i>{nie częściej niż określa min. przerwa między wypożyczeniami}</i></p>

Schemat pojęciowy uwzględniający proponowane zmiany po analizie wartości granicznych został zaprezentowany na Rys. 6-5.

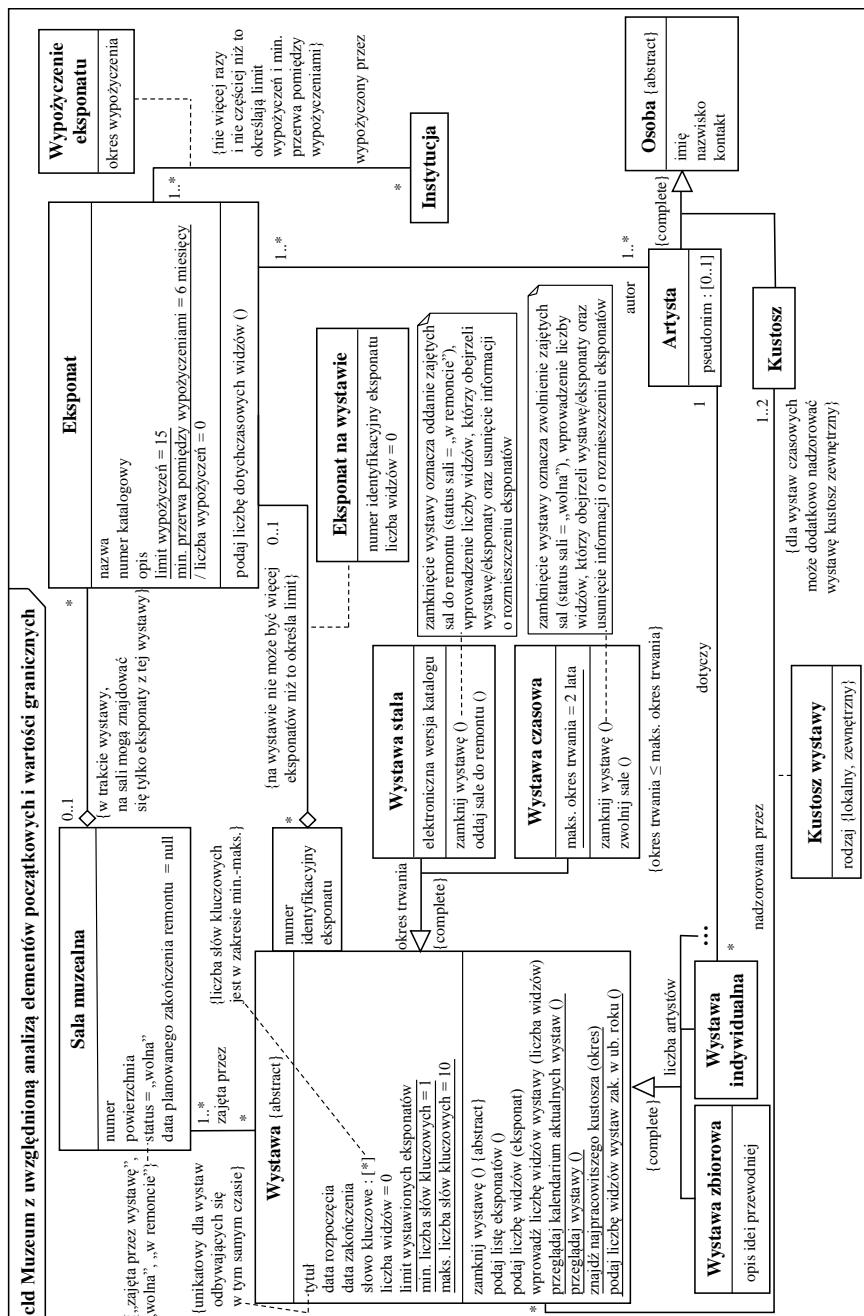
### 6.3.5 Zadania podstawowe

Zadania omówione w tym rozdziale są realizowane w oparciu o schemat pojęciowy z Rys. 6-5.

**Zad. 1:** Podkreśl te rodzaje dziedziczenia (rozłączne, nierożłączne, kompletne, niekompletne, jednokrotne, wielokrotne, jednoaspektowe, wieloaspektowe, dynamiczne, elipsa), które uważasz za potrzebne do wykorzystania na schemacie pojęciowym dla systemu *Muzeum*. Dla każdego wykorzystanego podziału podaj zastosowane rodzaje dziedziczenia.



Rys. 6-4 Schemat pojęciowy dla systemu *Muzeum* z uwzględnioną analizą elementów pochodnych i wartości początkowych



Rys. 6-5 Schemat pojęciowy dla systemu *Muzeum* z uwzględnioną analizą elementów pochodnych, wartości początkowych i wartości granicznych

- Hierarchia dla osób: dziedziczenie rozłączne, kompletne, jednokrotne, jednoaspektowe;
- hierarchia dla wystaw: dziedziczenie wieloaspektowe, dwa aspekty: *liczba artystów, okres trwania*;
- aspekt *liczba artystów* dla wystaw: rozłączne, kompletne, jednokrotne, elipsa;
- aspekt *okres trwania* dla wystaw: rozłączne kompletne, jednokrotne.

**Zad. 2:** Jakie rodzaje metod (abstrakcyjne, zaimplementowane) może zawierać klasa abstrakcyjna, a jakie klasa konkretna? Podaj przykłady.

Klasa abstrakcyjna może zawierać oba rodzaje metod, a klasa konkretna tylko metody zaimplementowane.

- Metoda abstrakcyjna w klasie abstrakcyjnej: *zamknij wystawę ()* w klasie *Wystawa*;
- metoda zaimplementowana w klasie abstrakcyjnej: *podaj listę eksponatów ()* w klasie *Wystawa*;
- metoda zaimplementowana w klasie konkretnej: *zamknij wystawę ()* w klasie *Wystawa czasowa*.

**Zad. 3:** Podaj przykłady metody klasowej, metody obiektu, atrybutu klasowego i atrybutu obiektu.

- Metoda obiektu: *podaj listę eksponatów ()* w klasie *Wystawa*;
- metoda klasowa: *znajdź najpracowitszego kustosza (okres)* w klasie *Wystawa*;
- atrybut obiektu: *tytuł* w klasie *Wystawa*;
- atrybut klasowy: *maks. okres trwania* z zainicjalizowaną wartością początkową *2 lata* w klasie *Wystawa czasowa*.

**Zad. 4:** Wskaż przykład zastosowania polimorfizmu metod.

Metody polimorficzne to np. metody *zamknij wystawę ()* w klasach hierarchii zbudowanej dla wystaw dla podziału związanego z aspektom *okres trwania*.

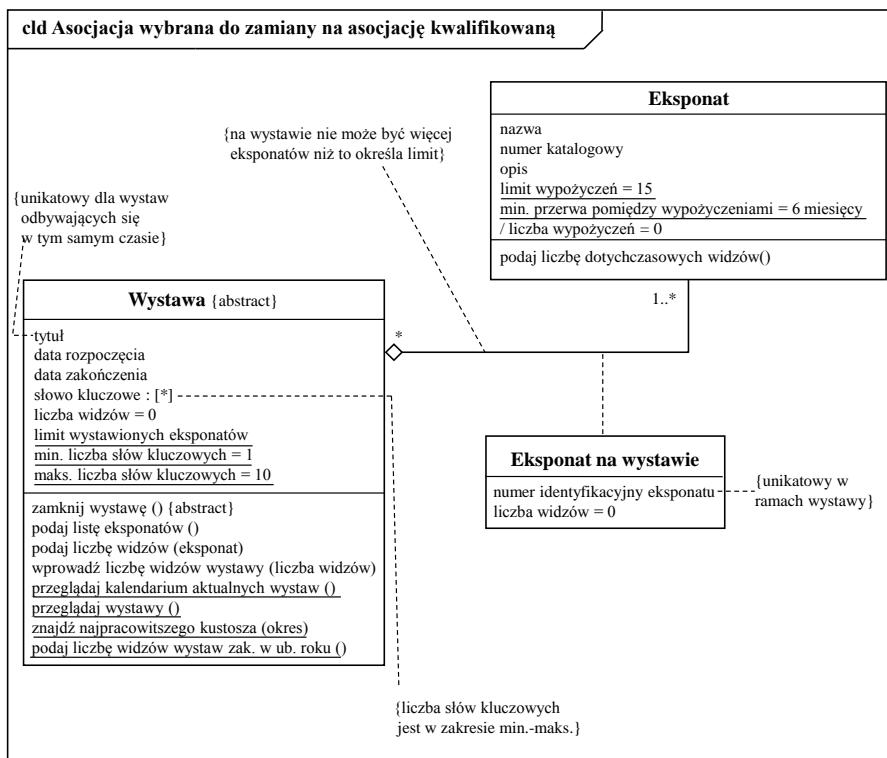
**Zad. 5:** Asocjację z diagramu klas, posiadającą niesymetryczne liczności oraz atrybut (lub klasę asocjacji), zamień na asocjację kwalifikowaną. Uzasadnij, dlaczego poddałeś zamianie właśnie tę asocjację.

Na schemacie pojęciowym umieszczono asocjację kwalifikowaną pomiędzy klasami *Wystawa* i *Eksponat*. Wskazuje ona, że kluczem do przeszukiwania zbioru eksponatów przypisanych do danej wystawy może być *numer identyfikacyjny eksponatu*.

Rys. 6-6 przedstawia fragment diagramu przed zamianą asocjacji „zwykłej” na asocjację kwalifikowaną.

*Uwaga:* Asocjację kwalifikowaną wykorzystuje się do wskazania klucza użytkowego do przeszukiwania dużych zbiorów danych (a nie zbiorów jednoelementowych). W tym konkretnie przypadku, kwalifikator *numer identyfikacyjny*

*eksponatu* może być wykorzystany jako klucz do przeszukiwania zbioru eksponatów przypisanych do pojedynczej wystawy. Wprowadzenie asocjacji kwalifikowanej zmienia licznosć tego końca asocjacji, który znajduje się naprzeciwko kwalifikatora. Nowa licznosć bezpośrednio wskazuje „siłę” klucza, czyli np. licznosć = 0..1 mówi o tym, że wartość kwalifikatora pozwala na jednoznaczną identyfikację eksponatu w ramach danej wystawy. Licznosć 0 dla roli *Eksponat* wskazuje, że dopuszczalna jest sytuacja, kiedy żadnemu z eksponatów danej wystawy nie przypisano numeru identyfikacyjnego o wartości równej danej wartości kwalifikatora.



Rys. 6-6 Fragment diagramu bez wykorzystania asocjacji kwalifikowanej

**Zad. 6:** Wskaź na schemacie pojęciowym asocjację, która jest agregacją lub kompozycją (lub może kwalifikować się do zamiany na takową). Objaśnij, dlaczego zdecydowałeś się na wybór agregacji/kompozycji, a nie „zwykłej” asocjacji? Ponadto, dlaczego wybrałeś agregację, a nie kompozycję (czy też odwrotnie)?

Na schemacie pomiędzy klasami: *Wystawa* i *Eksponat* występuje agregacja, co wskazuje – lepiej niż zwykła asocjacja – na związek częścь-całość występujący pomiędzy nimi. Wykorzystano agregację, a nie kompozycję, ponieważ w tym

przypadku cykl życiowy części nie zawiera się w cyklu życiowym całości, tzn. usunięcie z systemu obiektu klasy *Wystawa* nie będzie skutkowało usunięciem tych obiektów klasy *Eksponat*, które były związane z daną wystawą. Ponadto, na przestrzeni czasu, każdy z eksponatów mógł być związany z wieloma wystawami – stąd liczność \* przy klasie *Wystawa*.

**Zad. 7:** Wskaż na schemacie pojęciowym przykładowe ograniczenie. Na jaki element modelu zostało ono nałożone? Czy jest to ograniczenie statyczne czy dynamiczne? Zaproponuj, jaka metoda i w jakiej klasie mogłaby być odpowiedzialna za jego zapewnienie.

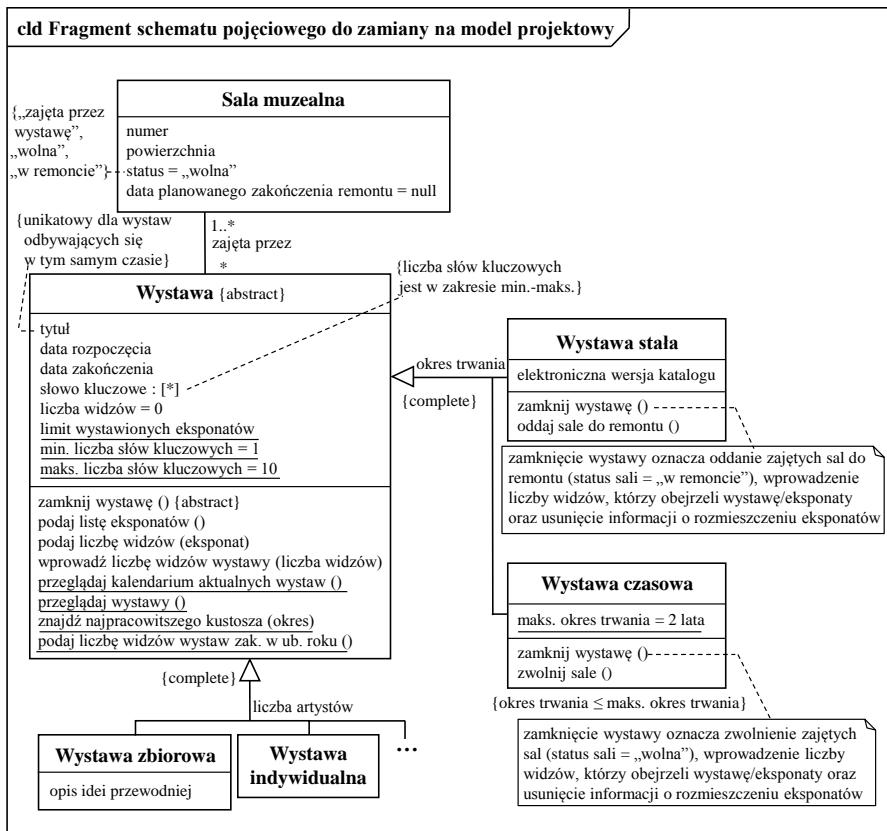
Np. ograniczenie określające liczbę i częstotliwość wypożyczeń eksponatu instytucjom, brzmiące: „*nie więcej razy i nie częściej niż to określają limit wypożyczeń i min. przerwa pomiędzy wypożyczeniami*”. Jest to ograniczenie dynamiczne, nałożone na asocjację pomiędzy klasami *Eksponat* i *Instytucja*. Ograniczenie to mogłoby być zrealizowane w metodzie *wypożycz eksponat (instytucja, okres)* umieszczonej w klasie *Eksponat*.

### 6.3.6 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzących przez osoby nauczane/uczące się

- Wprowadzanie do diagramów informacji, które ani bezpośrednio, ani pośrednio nie zostały umieszczone w tekście wymagań, jak np. *nazwa instytucji, imiona i nazwiska* osób odwiedzających salę itp. Jest to traktowane jako nadmiarowość modelu.
- Nazwa klasy to zazwyczaj rzeczownik w liczbie pojedynczej, tzn. jeżeli pojedynczy obiekt przechowuje информацию o jednej wystawie, to klasa nazywa się *Wystawa*, ale gdyby pojedynczy obiekt przechowywał informacje o zbiorze wystaw, w takim przypadku klasa nazywałaby się *Wystawy*.
- W nazwach podklas trzeba umieszczać nazwę nadklasy, czyli nazwa klasy to nie *Czasowa*, ale *Wystawa czasowa*.
- Błędne hierarchie klas, co wynika prawdopodobnie z nieznajomości reguł, które wykorzystujemy w procesie ich identyfikacji. Najważniejsza reguła mówi o tym, że klasy łączy się w hierarchie na zasadzie podobieństwa ich znaczeń, a nie podobieństwa ich własności (atributów czy operacji). To oznacza, że jeśli zbudujemy hierarchię, gdzie klasa *Wystawa indywidualna* dziedziczy po klasie *Rodzaj wystawy*, to następujące zdanie nie jest zdaniem prawdziwym: *Wystawa indywidualna jest rodzajem, szczególnym przypadkiem rodzaju wystawy*.

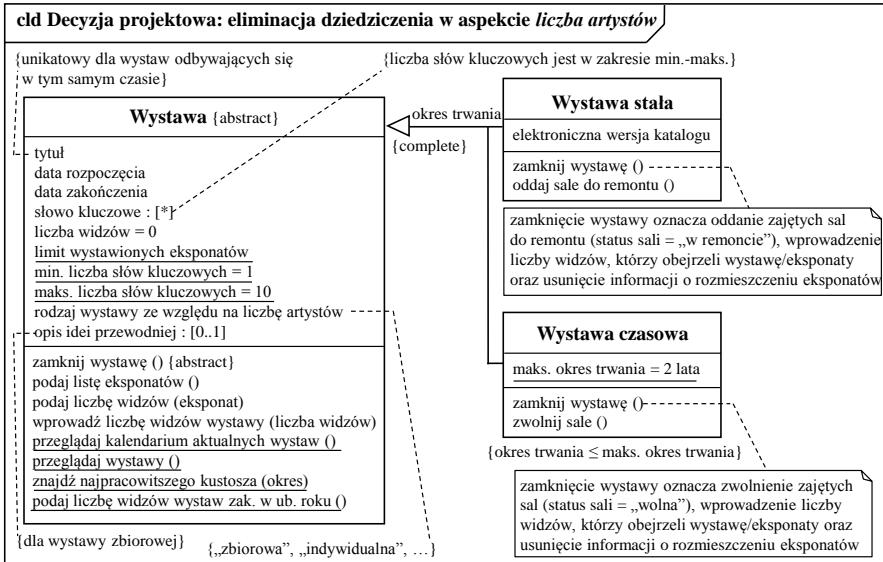
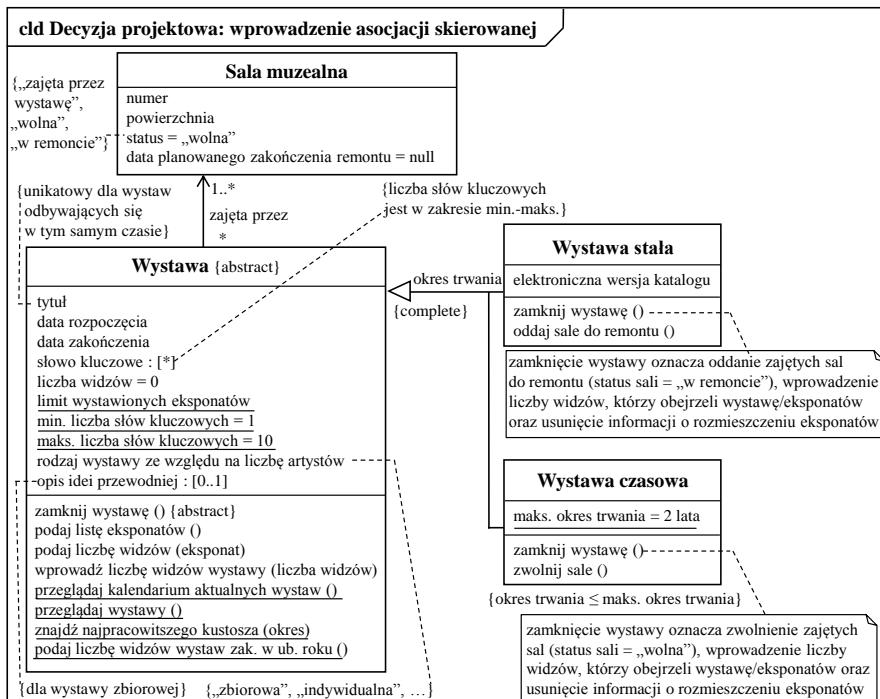
### 6.3.7 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji

Na Rys. 6-7 zilustrowany jest fragment schematu pojęciowego, dla którego zostanie określony odpowiedni diagram projektowy (uwzględniający ograniczenia obiektowego środowiska implementacji) [37, 40], niezbędny do zamodelowania diagramu interakcji dla funkcjonalności *Zwolnij sale zajmowane przez wystawę*.



Rys. 6-7 Fragment schematu pojęciowego do zamiany na schemat projektowy

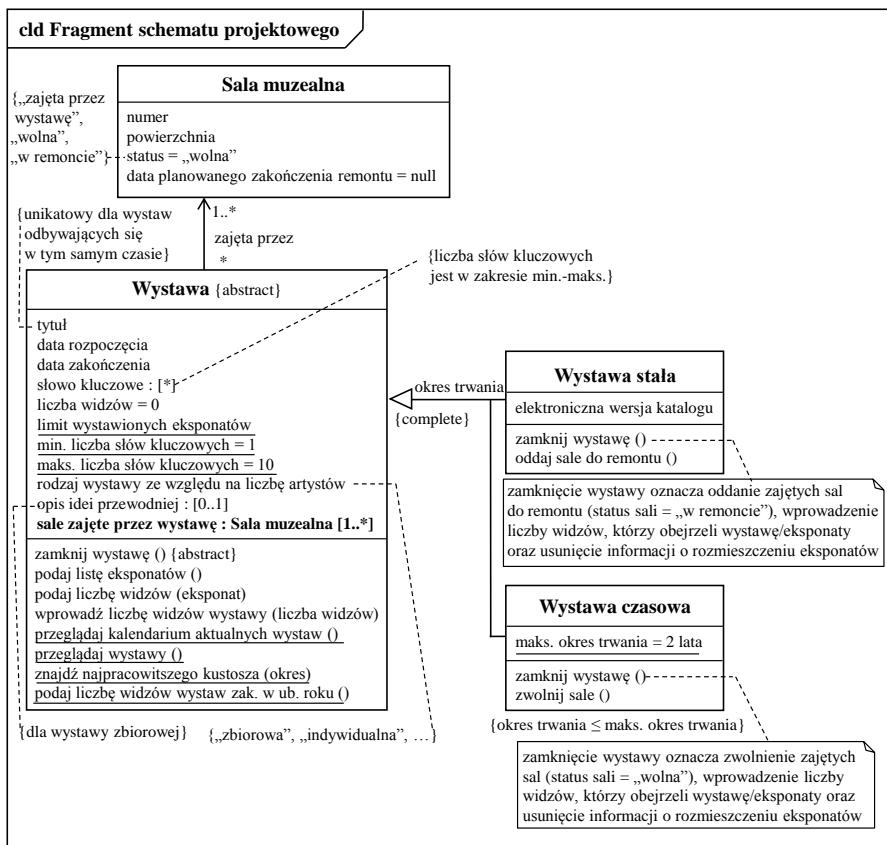
W pierwszym kroku transformacji schematu pojęciowego na schemat projektowy zostanie wyeliminowane dziedziczenie wieloaspektowe. Łatwo zauważyc, że aspekt *liczba artystów* dla hierarchii wystaw można w prosty sposób „zwinąć” do nadklasy *Wystawa* – jak na Rys. 6-8. Po wykonaniu tego kroku, hierarchia dla *Wystaw* posiada tylko jeden aspekt dziedziczenia, który jest kompletny i rozłączny, a zatem implementowalny w natywny sposób w obiektowym języku programowania, jak np. w Javie lub C#.

Rys. 6-8 Eliminacja aspektu *liczba artystów*Rys. 6-9 Wprowadzenie asocjacji skierowanej *zajęta przez*

Ponieważ dla asocjacji *zajęta przez* między klasą *Sala* a klasą *Wystawa* wymagana jest nawigacja tylko w jedną stronę, tj. od wystawy do sal przez nią zajmowanych, w drugim kroku transformacji asocjacja zostaje zamieniona na asocjację skierowaną, jak na Rys. 6-9.

W ostatnim kroku transformacji, do schematu z Rys. 6-9 zostaje wprowadzony atrybut złożony *sale zajmowane przez wystawę* w klasie *Wystawa*, potrzebny w celu implementacji asocjacji skierowanej *zajęta przez*.

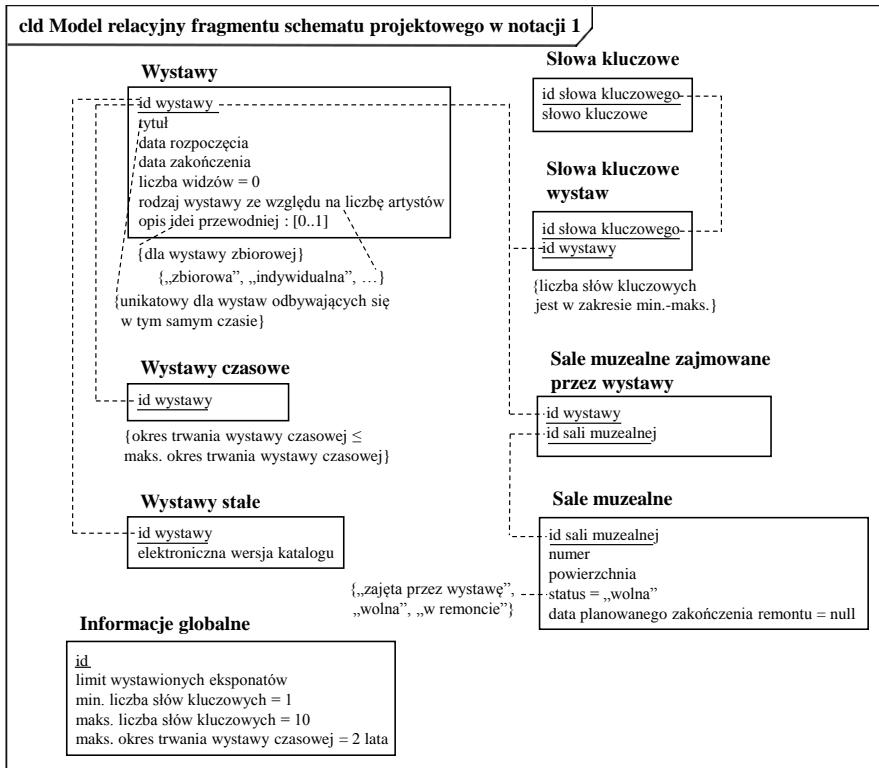
Końcowy fragment schematu projektowego jest umieszczony na Rys. 6-10.



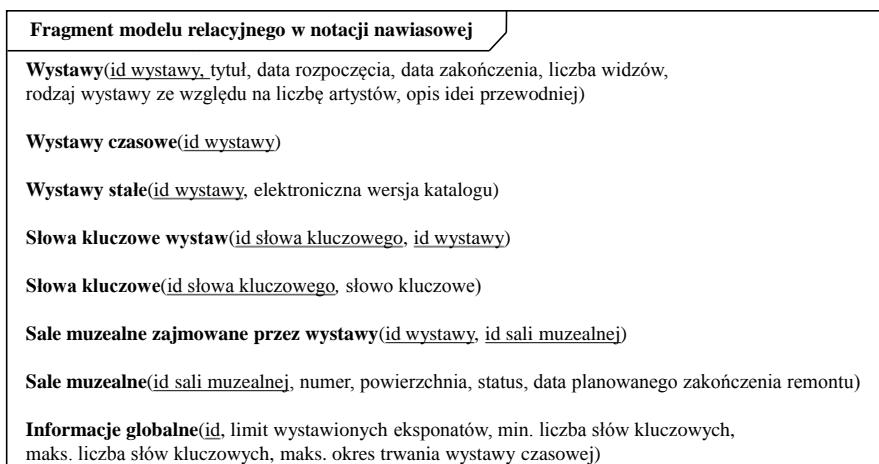
Rys. 6-10 Fragment schematu projektowego

### 6.3.8 Schemat relacyjny dla fragmentu diagramu klas

Na Rys. 6-10 przedstawiono fragment diagramu klas, zawierający cztery klasy, związek dziedziczenia i asocjację. Na Rys. 6-11 pokazano model relacyjny [37, 44] tego fragmentu diagramu klas w notacji 1 [41].

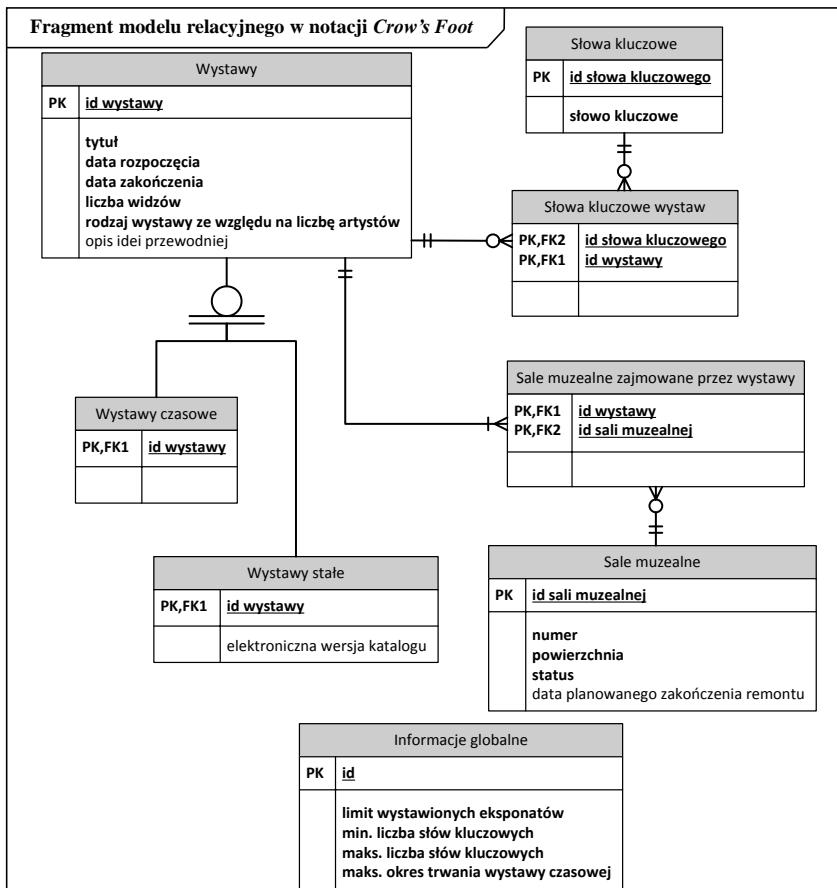


Rys. 6-11 Model relacyjny fragmentu diagramu klas przedstawionego na Rys. 6-10 w notacji 1



Rys. 6-12 Model relacyjny fragmentu diagramu klas w notacji nawiasowej

Na Rys. 6-12 zilustrowany jest adekwatny fragment modelu relacyjnego w notacji nawiązowej, a na Rys. 6-13 – w notacji *Crow's Foot*.



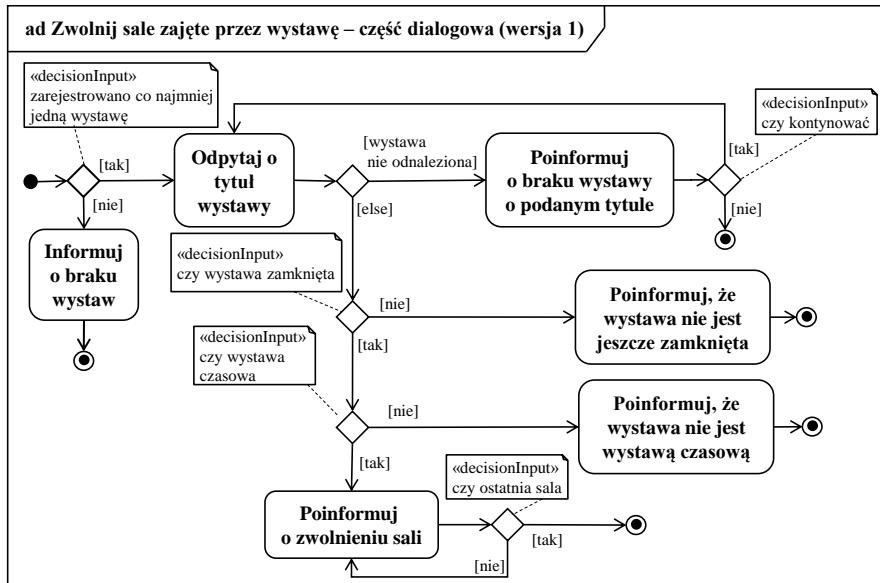
Rys. 6-13 Fragment modelu relacyjnego w notacji *Crow's Foot*

## 6.4 Analiza dynamiczna

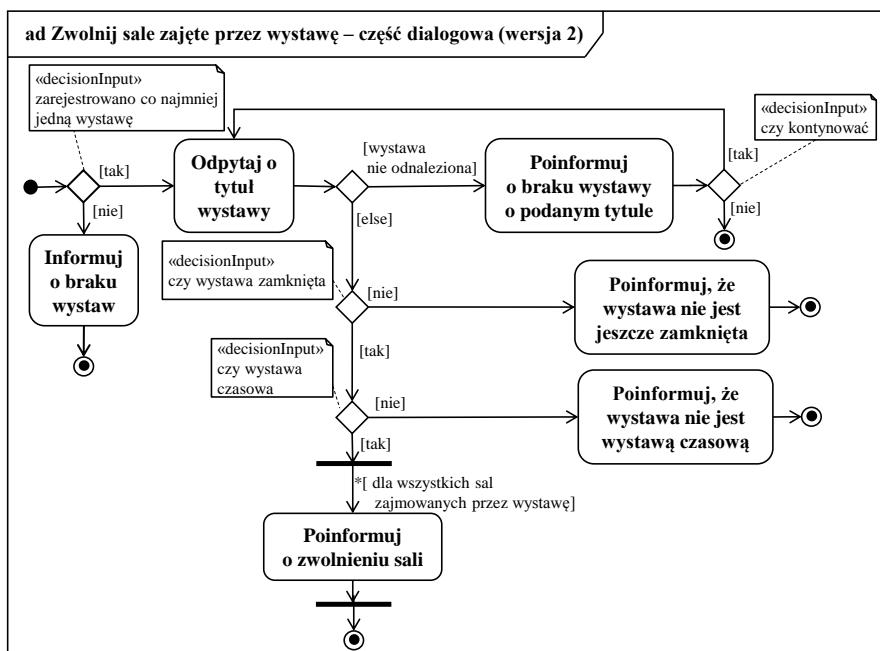
### 6.4.1 Zadania podstawowe

**Zad. 1:** Dla funkcjonalności *Zwolnij sale zajęte przez wystawę* skonstruj diagram aktywności dla części dialogowej zgodny ze scenariuszem z Tab. 6-1.

Przykładowe dwie wersje realizacji scenariusza sporządzonego w Tab. 6-1 ilustrują diagramy aktywności zawarte na Rys. 6-14 (z wykorzystaniem przetwarzania sekwencyjnego) i Rys. 6-15 (z wykorzystaniem przetwarzania współbieżnego).



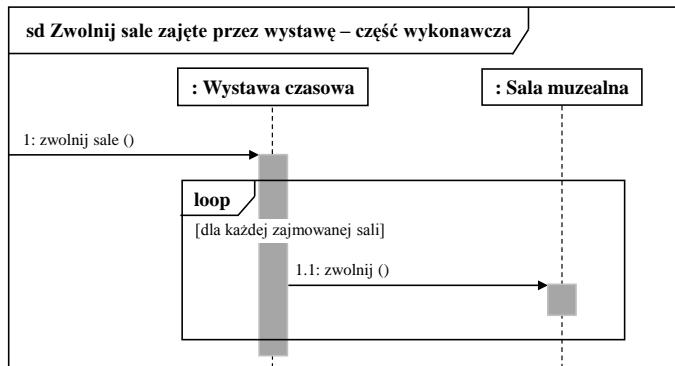
Rys. 6-14 Diagram aktywności dla przypadku użycia *Zwolnij sale zajęte przez wystawę* z wykorzystaniem przetwarzania sekwencyjnego



Rys. 6-15 Diagram aktywności dla przypadku użycia *Zwolnij sale zajęte przez wystawę* z wykorzystaniem przetwarzania współbieżnego

**Zad. 2:** Sporządź diagram interakcji dla części wykonawczej wybranego przypadku użycia.

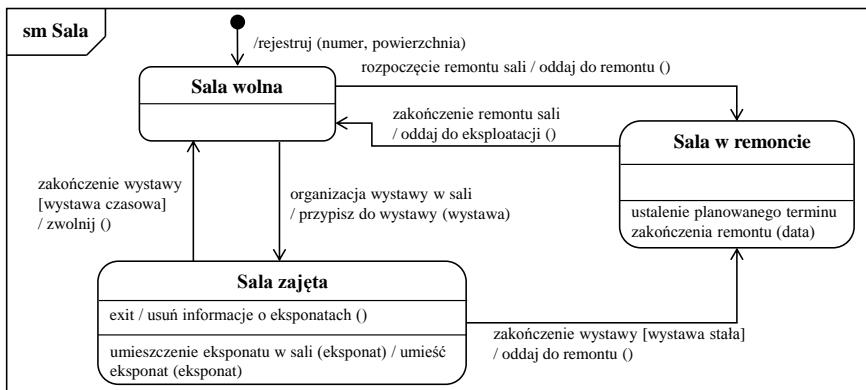
Przykładowy diagram sekwencji dla części wykonawczej przypadku użycia *Zwolnij sale zajęte przez wystawę* ilustruje Rys. 6-16, przy założeniu, że wystawa jest czasowa i została już zamknięta.



Rys. 6-16 Diagram sekwencji dla części wykonawczej przypadku *Zwolnij sale zajęte przez wystawę*

**Zad. 3:** Dla klasy *Sala muzealna* ze schematu pojęciowego wskaż kilka stanów (co najmniej dwa stany, nie licząc stanów początkowego i końcowego), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów, zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

*Uwaga:* W tym zadaniu nie jest wymagana cała maszyna stanowa, wystarczy zaprezentować dwa stany wraz ze specyfikacją przejścia.



Rys. 6-17 Maszyna stanowa dla klasy *Sala*

Maszyna stanowa dla klasy *Sala muzealna* została przedstawiona na Rys. 6-17. Przejście ze stanu *Sala wolna* do stanu *Sala w remoncie* nie wynika bezpośrednio z tekstu wymagań, ale wydaje się, że taka możliwość jest spójna z dziedziną problemową i może polepszyć użyteczność systemu.

#### **6.4.2 Omówienie zmian na schemacie pojęciowym i schemacie projektowym, wynikających z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań**

Po przeprowadzeniu analizy dynamicznej należy uzupełnić schemat pojęciowy i schemat projektowy o nowe elementy, które zostały ujęte na diagramach interakcji oraz diagramach stanów.

Lista zmian została zamieszczona w Tab. 6-4.

Tab. 6-4 Lista zmian na schemacie pojęciowym i schemacie projektowym po analizie dynamicznej

Klasa	Zmiana
<i>Sala muzealna</i>	dodatkowe metody: <i>oddaj do remontu ()</i> <i>oddaj do eksploatacji ()</i> <i>przypisz do wystawy (wystawa)</i> <i>rejestruj (numer, powierzchnia)</i> <i>umieść eksponat (eksponat)</i> <i>ustal planowany termin zakończenia remontu (data)</i> <i>usuń informacje o eksponatach ()</i> <i>zwolnij ()</i>
<i>Wystawa</i>	dodatkowa metoda: <i>znajdź wystawę (tytuł)</i>

#### **6.5 Zadania do samodzielnej pracy**

1. Narysuj podział przypadku użycia *Zwolnij sale zajęte przez wystawę* na pod przypadki, zgodnie ze scenariuszem z Tab. 6-1.
2. Czy podsystem może być aktorem dla danego systemu? Jeśli tak, odpowiedź uzupełnij przykładem spójnym z wymaganiami z zadania.

3. Wyjaśnij pojęcie stereotypu. Podaj odpowiedni przykład dla schematu pojęciowego z Rys. 6-5.
4. Objaśnij różnicę między dziedziczeniem wielokrotnym a wieloaspektowym. Czy któryś z tych dwóch rodzajów dziedziczenia wystąpił na schemacie pojęciowym z Rys. 6-5?
5. Czy klasa abstrakcyjna może zawierać metody zaimplementowane? Odpowiedź uzasadnij. Jeśli odpowiedź jest na „tak”, podaj odpowiedni przykład w oparciu o schemat pojęciowy z Rys. 6-5.
6. Objaśnij pojęcie przesłaniania. Wskaż przykład zastosowania na schemacie pojęciowym z Rys. 6-5. Czy przesłanianie jest rodzajem polimorfizmu metod?
7. Czy zasadę zamienialności można stosować do obiektów należących do różnych hierarchii klas. Odpowiedź uzasadnij. Jeśli odpowiedź jest na „tak”, podaj przykład oparty na schemacie pojęciowym z Rys. 6-5.
8. Dla wybranego przypadku użycia systemu (innego niż omówionego szczegółowo w analizie funkcjonalnej lub dynamicznej), sformułuj scenariusz i narysuj diagram aktywności (określ przynajmniej dwie aktywności, nie licząc początkowej i końcowej).
9. Dla wybranego przypadku użycia systemu (innego niż omówionego szczegółowo w analizie funkcjonalnej lub dynamicznej), sformułuj algorytm i narysuj odpowiedni diagram interakcji, który przedstawia interakcje pomiędzy obiektami co najmniej dwóch różnych klas.
10. Dla dowolnej klasy (innej niż omówionej szczegółowo w analizie dynamicznej) ze schematu pojęciowego wskaż dwa stany (wyłączając stan początkowy i stan końcowy), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

## 6.6 Podsumowanie zadania

W Tab. 6-5 przedstawiono ogólną charakterystykę omawianego zadania. Szczegółową charakterystykę zadania dla grup tematycznych: *Wprowadzenie do przedmiotu*, *Analiza funkcjonalna*, *Analiza strukturalna*, *Analiza dynamiczna* zamieszczono odpowiednio w tabelach Tab. 6-6, Tab. 6-7, Tab. 6-8 i Tab. 6-9.

Tab. 6-5 Ogólna charakterystyka zadania

<b>Ogólna charakterystyka zadania</b>	
<i>Nazwa</i>	Muzeum
<i>Dziedzina</i>	system mógłby znaleźć zastosowanie w każdej dziedzinie, gdzie

<i>problemowa</i>	prowadzona jest działalność wystawiennicza
<i>Cel</i>	wdrożenie systemu ułatwi prowadzenie działalności wystawienniczej oraz sprawozdawczej
<i>Zakres odpowiedzialności</i>	ewidencja dzieł sztuki, wystaw, rejestracja odwiedzających, zarządzanie salami, wypożyczanie eksponatów do innych placówek, tworzenie raportów
<i>Stopień trudności</i>	średnio trudne
<i>Rozmiar</i>	11 klas

*Użyteczność w grupach tematycznych*

<i>Grupa tematyczna</i>	<i>Użyteczność w grupie tematycznej</i>	<i>Użyteczność na etapie zgodnym z modelem efektywnego nauczania</i>
<i>Wprowadzenie do przedmiotu</i>	niska	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> niska <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> niska
<i>Analiza funkcjonalna</i>	średnia	<i>Przygotowanie do nauki:</i> średnia <i>Nauka:</i> średnia <i>Podsumowanie dotychczasowych działań:</i> średnia <i>Ewaluacja:</i> wysoka
<i>Analiza strukturalna</i>	średnia	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> wysoka <i>Podsumowanie dotychczasowych działań:</i> średnia <i>Ewaluacja:</i> średnia
<i>Analiza dynamiczna</i>	średnia	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> średnia <i>Podsumowanie dotychczasowych działań:</i> średnia <i>Ewaluacja:</i> wysoka

Tab. 6-6 Szczegółowa charakterystyka zadania dla grupy tematycznej *Wprowadzenie do przedmiotu*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Wprowadzenie do przedmiotu</i>

<b>Stopień trudności grupy w zadaniu</b>	łatwe		
<b>Poruszane tematy</b>	<i>analiza a projektowanie, mechanizmy rozszerzalności, reguły nazewnictwa, poprawa percepcji diagramu/diagramów</i>		
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>	
<i>analiza a projektowanie</i>	łatwe	<i>model pojęciowy, model projektowy</i>	
<i>mechanizmy rozszerzalności</i>	łatwe	<i>stereotyp, ograniczenie, komentarz</i>	
<i>reguły nazewnictwa</i>	łatwe	reguły nazewnictwa dla aktorów, przypadków użycia, klas	
<i>poprawa percepcji diagramu/diagramów</i>	łatwe	reguły tworzenia hierarchii klas	

Tab. 6-7 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza funkcjonalna*

<b>Szczegółowa charakterystyka zadania</b>			
<b>Grupa tematyczna</b>	<i>Analiza funkcjonalna</i>		
<b>Stopień trudności grupy w zadaniu</b>	średnio trudne		
<b>Poruszane tematy</b>	<i>aktorzy, przypadki użycia, scenariusze, mechanizmy rozszerzalności</i>		
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>	
<i>aktorzy</i>	średnio trudne	<i>aktor abstrakcyjny, aktor Podsystem czasu, hierarchia dla aktorów, stosunkowo duża liczba aktorów</i>	

<i>przypadki użycia</i>	średnio trudne	
<i>scenariusze</i>	łatwe	
<i>mechanizmy rozszerzalności</i>	łatwe	<i>komentarz, ograniczenie nałożone na relację między przypadkami użycia</i>

Tab. 6-8 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza strukturalna*

<b>Szczegółowa charakterystyka zadania</b>			
<b>Grupa tematyczna</b>	<i>Analiza strukturalna</i>		
<b>Stopień trudności grupy w zadaniu</b>	średnio trudne		
<b>Poruszane tematy</b>	<i>atrybuty klasy, metody klasy, struktury generalizacji-specjalizacji, klasa abstrakcyjna a klasa konkretna, asocjacje, analiza wartości, przejście na schemat projektowy, przejście na schemat relacyjny, mechanizmy rozszerzalności</i>		
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>	
<i>atrybuty klasy</i>	łatwe	<i>atrybut powtarzalny, pochodny, obiektowy, klasowy</i>	
<i>metody klasy</i>	łatwe	<i>metoda abstrakcyjna, zaimplementowana, obiektowa, klasowa, polimorfizm metod</i>	
<i>struktury generalizacji-specjalizacji</i>	łatwe	dziedziczenie wieloaspektowe i elipsa	
<i>klasa abstrakcyjna a klasa konkretna</i>	łatwe	<i>klasa abstrakcyjna, klasa konkretna</i>	
<i>asocjacje</i>	średnio trudne	<i>rola asocjacji, klasa asocjacji, agregacja, asocjacja kwalifikowana, stosunkowo duża liczba klas asocjacji</i>	
<i>analiza wartości</i>	łatwe	<i>wartość pochodna, wartość początkowa (domyślne), wartość graniczna</i>	
<i>przejście na schemat projektowy</i>	średnio trudne	obejście dziedziczenia wieloaspektowego, <i>asocjacja skierowana</i>	

<i>przejście na schemat relacyjny</i>	średnio trudne	<i>obejście dla dziedziczenia klas, zamiana asocjacji wiele do wielu, zamiana atrybutów klasowych, hierarchia encji, encje/tabele słownikowe, stosunkowo złożona transformacja do modelu relacyjnego</i>
<i>mechanizmy rozszerzalności</i>	łatwe	<i>ograniczenie, komentarz, stosunkowo duża liczba ograniczeń i komentarzy na schemacie pojęciowym i projektowym</i>

Tab. 6-9 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza dynamiczna*

<b>Szczegółowa charakterystyka zadania</b>			
<b>Grupa tematyczna</b>	<i>Analiza dynamiczna</i>		
<b>Stopień trudności grupy w zadaniu</b>	średnio trudne		
<b>Poruszane tematy</b>	<i>diagramy aktywności, diagramy interakcji, diagramy stanów, podsumowanie analizy dynamicznej</i>		
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>	
<i>diagramy aktywności</i>	średnio trudne	romb decyzyjny <i>fork, join, branch</i> i <i>merge</i> , dwie wersje diagramu aktywności: z wykorzystaniem przetwarzania sekwencyjnego i współbieżnego	
<i>diagramy interakcji</i>	łatwe	fragmenty wyodrębnione <i>loop</i> na diagramach sekwencji	
<i>diagramy stanów</i>	łatwe		
<i>podsumowanie analizy dynamicznej</i>	łatwe	modyfikacje schematu pojęciowego i schematu projektowego o dodatkowe metody	

## Spis treści rozdziału

Rozdział 7 Kwatery do wynajęcia .....	280
7.1 Tekst wymagań.....	280
7.2 Analiza funkcjonalna .....	281
7.2.1 Diagram kontekstowy .....	281
7.2.2 Model przypadków użycia.....	282
7.2.3 Scenariusze wybranych przypadków użycia łącznie z podziałem na podprzypadki .....	282
7.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	287
7.3 Analiza strukturalna.....	288
7.3.1 Schemat pojęciowy .....	288
7.3.2 Analiza elementów pochodnych .....	288
7.3.3 Analiza wartości początkowych .....	290
7.3.4 Analiza wartości granicznych.....	290
7.3.5 Zadania podstawowe .....	292
7.3.6 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	296
7.3.7 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji .....	297
7.3.8 Schemat relacyjny dla fragmentu diagramu klas .....	305
7.4 Analiza dynamiczna .....	306
7.4.1 Zadania podstawowe .....	306
7.4.2 Omówienie zmian na schemacie pojęciowym i schemacie projektowym, wynikających z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań .....	311
7.5 Zadania do samodzielnej pracy .....	313
7.6 Podsumowanie zadania.....	314

# Rozdział 7

## Kwatery do wynajęcia

*Ewa Stemposz, Andrzej Jodłowski*

### 7.1 Tekst wymagań

W jednym ze znanych polskich biur turystycznych zarząd postanowił wdrożyć system informatyczny wspierający rezerwację pokojów w kwaterach prywatnych.

1. Każda zarejestrowana w systemie kwatera jest opisana przez nazwę (jeśli ją posiada), adres, standard pokojów („wysoki”, „średni”, „niski”) oraz informacje o właściwemu. Pokoje do wynajęcia są przeznaczone na określona liczbę osób, wiadomo czy posiadają kuchnię i/lub łazienkę oraz są ponumerowane. Dla każdego pokoju znana jest cena wynajmu za dzień.
2. Ponieważ klienci indywidualni (imię, nazwisko, adres, PESEL) są szczególnie zainteresowani niektórymi lokalizacjami kwater, dlatego wyróżniono pewne ich kategorie, jak np. kwatery nadmorskie (dla których ma być przechowywana informacja o odległości do morza) czy kwatery w górach (z nazwami okolicznych pasm górskich oraz położeniem kwater n.p.m.). Podział jest kompletny. Aktualnie, aby kwatera mogła zostać uznana jako nadmorska, musi się znajdować w odległości nie przekraczającej 10 km od morza. Z kolei, kwatery górskie muszą być położone na wysokości co najmniej 500 m n.p.m. Zdarzają się kwatery należące do kilku kategorii jednocześnie.
3. Z badań rynkowych zamówionych przez biuro turystyczne wyniknęła także potrzeba podzielenia kwater ze względu na rodzaj potencjalnego klienta, m.in. na kwatery o charakterze rodzinnym (przystosowane do zamieszkania przez rodziny z dziećmi w pewnym przedziale wiekowym ustalonym przez właściciela kwater) oraz na kwatery studenckie – z określonym rabatem dla studentów. Ten podział jest również kompletny. Kwatery rodzinne mogą też być równocześnie np. studenckimi.
4. Klienci za pośrednictwem biura rezerwują pokoje w kwaterach. Pojedyncza rezerwacja może obejmować kilka pokojów, ale tylko w obrębie jednej kwatera, przy czym każdy z pokojów może zostać zarezerwowany na inny okres. W ramach danej rezerwacji pokojom są przypisywane unikatowe identyfikatory. W systemie pamiętała jest data złożenia rezerwacji, łączna wysokość kwoty do zapłaty, wysokość wpłaconej zaliczki, ostateczny termin zapłaty pozostały części kwoty oraz status rezerwacji („złożona”, „wykupiona”, „anulowana”). Rezerwacja jest anulowana w przypadku niedotrzymania przez klienta ostatecznego terminu zapłaty.

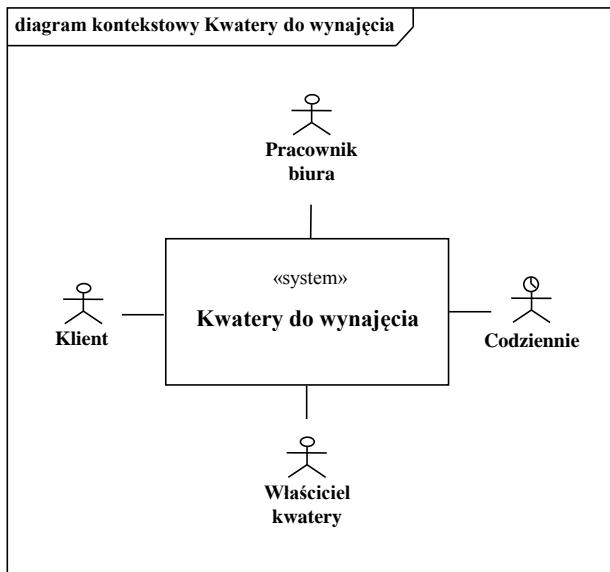
5. Przechowywane mają być informacje wyłącznie o klientach, którzy złożyli co najmniej jedną rezerwację w biurze.
6. Właściciele kwater (opisywani przez imię, nazwisko, adres, PESEL, właściwy urząd skarbowy) ustalają cenę wynajmu każdego z pokojów – cena nie może być zmieniana częściej niż raz na pół roku dla danego pokoju.
7. W systemie ma istnieć możliwość wydruku oferty wybranej kwater, zawierającej podstawowe informacje o niej, takie jak: nazwa, standard, adres i informacje dodatkowe, zależne od rodzaju kwater, jak np. rabat studencki, odległość od morza, przedział wiekowy dzieci. Do oferty dołączane są też informacje o pokojach wchodzących w skład kwater.
8. System ma wspomóc w realizowaniu usług, jak na przykład:
  - 8.1 Wyszukanie kwater (przez klienta, właściciela lub pracownika biura);
  - 8.2 Wyszukanie kwater o określonym standardzie i ewentualnie o odpowiedniej lokalizacji (przez klienta, właściciela lub pracownika biura);
  - 8.3 Wyszukiwanie kwater o odpowiedniej lokalizacji (przez klienta, właściciela lub pracownika biura);
  - 8.4 Rejestracja nowej kwatera z jednoczesnym wprowadzaniem informacji o dostępnych pokojach (przez pracownika biura lub właściciela kwater); oraz z możliwością wprowadzenia danych właściciela – jeśli informacja o nim nie jest już przechowywana w systemie (przez pracownika biura);
  - 8.5 Określenie najbardziej popularnych kwater (z największą liczbą rezerwacji) w danym okresie (przez pracownika biura);
  - 8.6 Anulowanie rezerwacji, które nie zostały opłacone w terminie (codziennie);
  - 8.7 Automatyczne tworzenie listy klientów, którzy nie opłaciли w terminie kosztów wynajmu (codziennie);
  - 8.8 Zmiana ceny wynajmu pokoju (przez właściciela kwater).

## 7.2 Analiza funkcjonalna

### 7.2.1 Diagram kontekstowy

*Polecenie:* Zbuduj diagram kontekstowy w oparciu o tekst wymagań zamieszczony w punkcie 7.1

Diagram kontekstowy dla systemu *Kwatery do wynajęcia* został przedstawiony na Rys. 7-1.



Rys. 7-1 Diagram kontekstowy

### 7.2.2 Model przypadków użycia

*Polecenie:* Zbuduj model przypadków użycia, uwzględniając wyłącznie funkcjonalność sugerowaną w ostatnim punkcie tekstu wymagań.

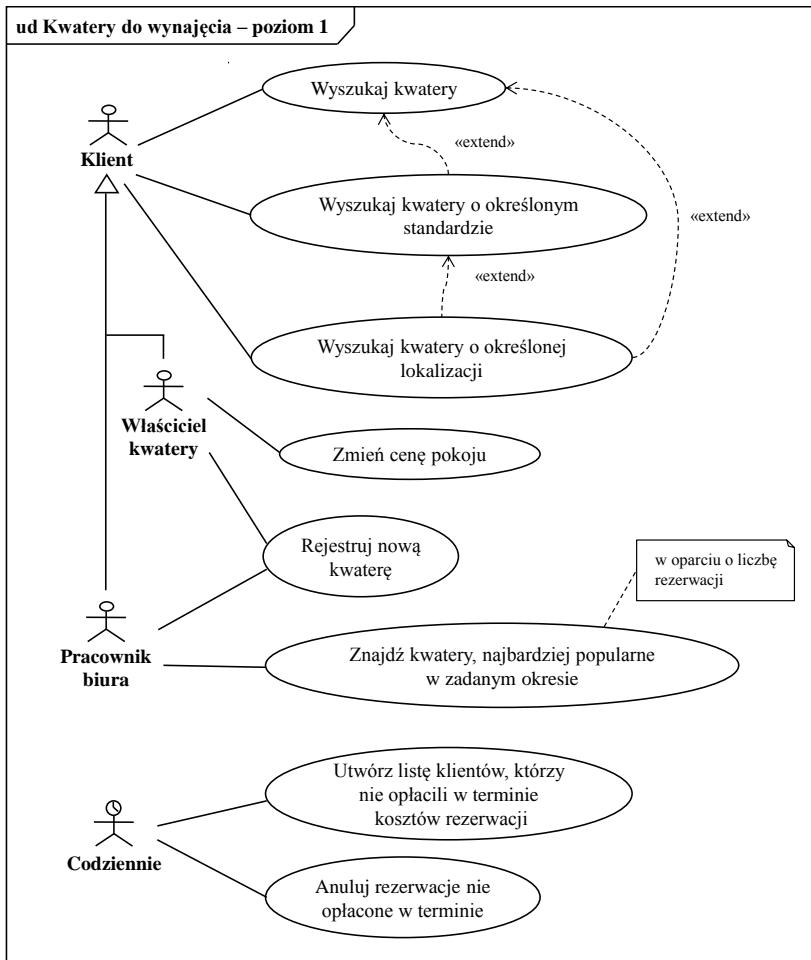
*Uwaga:* Model należy skonstruować z perspektywy aktorów z otoczenia systemu (najwyższy poziom abstrakcji modelu). Należy uwzględnić hierarchię dla aktorów i relacje pomiędzy przypadkami (o ile mają/mogłyby mieć miejsce).

Model przypadków użycia [35, 40, 41, 45] uwzględniający wyłącznie funkcjonalność oraz aktorów określonych w punkcie 8 tekstu wymagań przedstawiono na Rys. 7-2.

### 7.2.3 Scenariusze wybranych przypadków użycia łącznie z podziałem na podprzypadki

*Polecenie:* Dla przypadku użycia *Wyszukaj kwatery o określonej lokalizacji* napisz scenariusz.

Przykładowy scenariusz dla wybranego przypadku użycia został przedstawiony w Tab. 7-1. W scenariuszu uwzględniono wyłącznie aktorów z otoczenia systemu, a pomimo aktorów będących przypadkami użycia, czyli *Wyszukaj kwatery* oraz *Wyszukaj kwatery o określonym standardzie*.



Rys. 7-2 Model przypadków użycia

Tab. 7-1 Scenariusz dla przypadku użycia *Wyszukaj kwatery o określonej lokalizacji*

<b>Nazwa przypadku użycia</b>	Wyszukaj kwatery o określonej lokalizacji
<b>Warunek początkowy</b>	W systemie jest zarejestrowana co najmniej jedna kwatera.
<b>Główny przepływ zdarzeń</b>	1. Aktor <i>Klient</i> , <i>Właściciel kwiatery</i> lub <i>Pracownik biura</i> uruchamia przypadek użycia.

	<p>2. System wyświetla formularz z kryteriami dla wyszukiwania kwater o określonej lokalizacji. Aktor wprowadza kryteria.</p> <p>3. System wyświetla listę kwater spełniających zadane kryteria.</p>
<b>Alternatywne przepływy zdarzeń</b>	<p>2a. Format wprowadzonych danych jest błędny, system informuje o tym aktora i powraca do punktu 2.</p> <p>2b. Nie znaleziono kwater spełniających zadane kryteria, system informuje o tym aktora i odpytuje, czy kontynuować poszukiwania w oparciu o zmienione kryteria.</p> <p>2.ba. Aktor decyduje się na zakończenie poszukiwań, system kończy przypadek użycia.</p> <p>2.bb. Aktor decyduje się na zmianę kryteriów wyszukiwania, system powraca do punktu 2.</p>
<b>Zakończenie</b>	W dowolnym momencie.
<b>Warunek końcowy</b>	Brak

*Polecenie:* Dla przypadku użycia *Zmień cenę pokoju* napisz scenariusz.

Przykładowy scenariusz dla wybranego przypadku użycia został przedstawiony w Tab. 7-2.

Tab. 7-2 Scenariusz dla przypadku użycia *Zmień cenę pokoju*

<b>Nazwa przypadku użycia</b>	Zmień cenę pokoju
<b>Warunek początkowy</b>	W systemie jest zarejestrowana co najmniej jedna kwatera.
<b>Główny przepływ zdarzeń</b>	<p>1. Aktor <i>Właściciel kwatery</i> uruchamia przypadek użycia.</p> <p>2. System wyświetla listę kwater danego właściciela. Aktor wybiera kwaterę.</p> <p>3. System wyświetla listę pokojów wraz</p>

	<p>z informacjami o nich. Aktor wybiera pokój.</p> <p>4. System wyświetla formularz zmiany ceny pokoju za jeden dzień. Aktor wprowadza nową cenę wynajmu.</p> <p>5. System informuje o poprawnym zakończeniu przypadku użycia, wyświetla informacje o kwaterze i informacje o pokoju wraz z nową ceną.</p>
<b>Alternatywne przepływy zdarzeń</b>	<p>2a. Aktor rezygnuje z wyboru kwatery, system kończy przypadek użycia.</p> <p>3a. Aktor rezygnuje z wyboru pokoju, system kończy przypadek użycia.</p> <p>4a. Format wprowadzonych danych jest błędny, system informuje o tym aktora i powraca do punktu 4.</p> <p>4b. Nie upłynęło jeszcze 6 miesięcy od ostatniej zmiany ceny pokoju. System wyświetla informację o tym i kończy przypadek użycia</p>
<b>Zakończenie</b>	W dowolnym momencie.
<b>Warunek końcowy</b>	Wprowadzono nową cenę wynajmu pokoju.

*Polecenie:* Dla przypadku użycia *Rejestruj nową kwaterę*:

- napisz scenariusz,
- zaproponuj podział tego przypadku na podprzypadki.

Przykładowy scenariusz dla wybranego przypadku użycia został przedstawiony w Tab. 7-3.

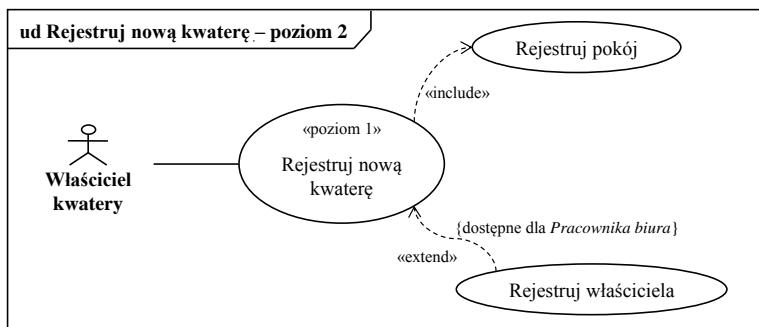
Tab. 7-3 Scenariusz przypadku użycia *Rejestruj nową kwaterę*

<b>Nazwa przypadku użycia</b>	Rejestruj nową kwaterę
<b>Warunek początkowy</b>	Brak
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Przypadek użycia jest wywoływany przez aktora <i>Właściciel kwatery</i> lub aktora <i>Pracownik biura</i>.</li> <li>2. System wyświetla formularz dla rejestracji nowej</li> </ol>

	<p>kwatery, w oparciu o informacje, takie jak: nazwa (niewymagana), adres, standard pokojów, kategoria ze względu na lokalizację oraz rodzaj potencjalnego klienta. Aktor wprowadza odpowiednie dane.</p> <ol style="list-style-type: none"> <li>3. System wyświetla formularz dla wprowadzenia informacji dodatkowych o kwaterze (odpowiednich do wybranej w punkcie 2 kategorii i rodzaju klienta), takie jak np. odległość od morza, nazwy okolicznych pasm górskich, położenie n.p.m., przedział wiekowy dla dzieci, rabat studencki itd. Aktor wprowadza odpowiednie dane.</li> <li>4. System odpytuje o liczbę pokoi dostępnych w kwaterze. Aktor wprowadza liczbę pokoi.</li> <li>5. Kolejno, dla każdego pokoju, system wyświetla formularz rejestracyjny, odpytując: o liczbę osób, czy pokój posiada kuchnię i/lub łazienkę oraz o cenę wynajmu za dzień.</li> <li>6. System wyświetla dane właściciela kwatery: imię, nazwisko, adres, PESEL, właściwy urząd skarbowy.</li> <li>7. System prosi o potwierdzenie danych właściciela kwatery. Aktor potwierdza.</li> <li>8. System informuje aktora o poprawnym zakończeniu procesu rejestracji kwatery.</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	<ol style="list-style-type: none"> <li>2a. Format wprowadzonych danych nie jest poprawny, system informuje o tym aktora i wraca do punktu 2.</li> <li>2b. Kwatera, o podanym adresie jest już zarejestrowana w systemie, system informuje o tym aktora i pyta, czy zakończyć.             <ol style="list-style-type: none"> <li>2ba. Aktor zdecydował o kontynuowaniu przypadku. System powraca do punktu 2.</li> <li>2bb. Aktor wybrał zakończenie przypadku użycia.</li> </ol> </li> <li>3a. Format wprowadzonych danych nie jest poprawny, system informuje aktora o błędzie i powraca do punktu 3.</li> <li>4a. Format danych nie jest poprawny, system informuje aktora o błędzie i ponownie odpytuje o liczbę pokoi w punkcie 4.</li> <li>5a. Format danych wprowadzanych w trakcie</li> </ol>

	<p>rejestrowania pokoju nie jest poprawny, system informuje aktora o błędzie i ponawia odpytywanie o dane tego pokoju.</p> <p>6a. Aktorem jest <i>Pracownik biura</i>, system wyświetla formularz rejestracji danych właściciela kwatery (imię, nazwisko, adres, PESEL, właściwy urząd skarbowy). Aktor wprowadza odpowiednie dane.</p> <p>6aa. Format wprowadzonych przez <i>Pracownika biura</i> danych o właściwemu kwaterze nie jest poprawny, system informuje aktora o błędzie i powraca do punktu 6a.</p> <p>6ab. Osoba o podanych danych została już zarejestrowana jako właściciel kwatery, system informuje o tym <i>Pracownika biura</i>.</p>
<b>Zakończenie</b>	W dowolnym momencie
<b>Warunek końcowy</b>	Zarejestrowanie nowej kwatery dla danego właściciela

Diagram z przykładowym podziałem przypadku „Rejestruj nową kwaterę” przedstawiono na Rys. 7-3.



Rys. 7-3 Diagram z przykładowym podziałem przypadku *Rejestruj nową kwaterę*

#### 7.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

Najczęściej występujące błędy w rozwiązaniach osób nauczanych/uczących się to:

- błędny wybór aktora;
- błędna nazwa aktora;

- brak hierarchii dla aktorów lub błędne hierarchie dla nich.

*Dla przypomnienia:* Na najwyższym poziomie abstrakcji, aktor to byt z otoczenia systemu (dla systemu traktowanego jako całość), a bytem z otoczenia systemu nie jest jego część, np. *Podsystem rejestracji opłat*. Ponadto, trudno zrozumieć, jaki zbiór usług miałyby być dostarczony tak nazwanemu aktorowi.

## 7.3 Analiza strukturalna

### 7.3.1 Schemat pojęciowy

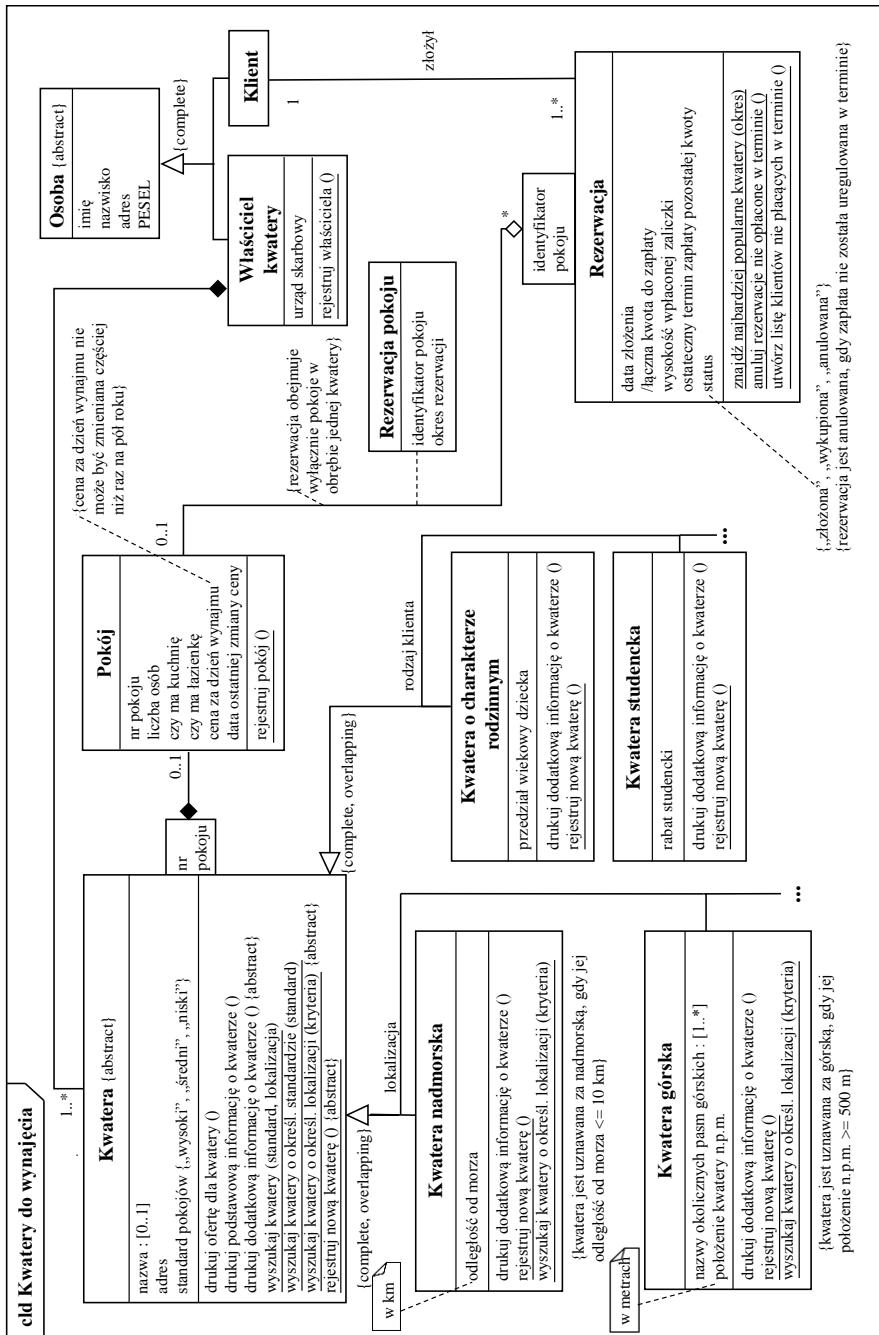
*Polecenie:* Dla podanego tekstu wymagań skonstruuj schemat pojęciowy (nie redukując liczności).

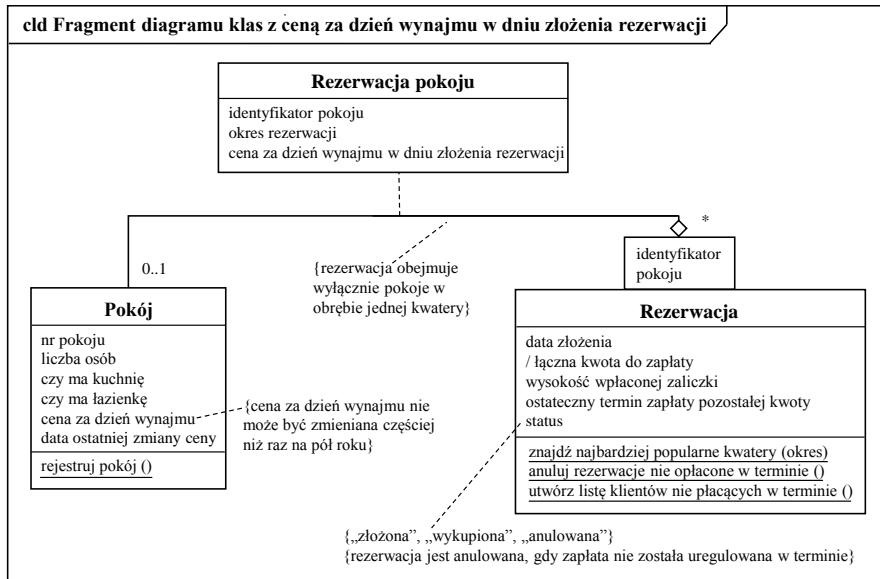
Schemat pojęciowy [36, 41, 42] dla systemu *Kwatery do wynajęcia* przedstawiono na Rys. 7-4.

*Uwaga:* Podczas konstruowania schematu pojęciowego należy również uwzględnić informacje wynikające pośrednio z tekstu wymagań. Na schemacie pojęciowym dla systemu *Kwatery do wynajęcia* umieszczono atrybut *data ostatniej zmiany ceny* w klasie *Pokój*. W tekście wymagań nie został on wymieniony explicite, ale jest niezbędny ze względu na ograniczenie *{cena za dzień wynajmu nie może być zmieniana częściej niż raz na pół roku}*. W przypadku braku daty ostatniej zmiany wartości atrybutu nie można będzie określić, czy ograniczenie jest spełnione.

### 7.3.2 Analiza elementów pochodnych

Wartość atrybutu *łączna kwota do zapłaty* w klasie *Rezerwacja* jest obliczana podczas rejestracji rezerwacji na podstawie informacji o pokojach objętych rezerwacją (z wykorzystaniem wartości atrybutu *okres rezerwacji* w klasie asocjacji *Rezerwacja pokoju* oraz atrybutu *cena za dzień wynajmu* w klasie *Pokój*). Gdyby cena za dzień wynajmu pokoju była ustalona na stałe, wówczas atrybut *łączna kwota do zapłaty* byłby atrybutem pochodnym. Jednakże *cena za dzień wynajmu* może być zmieniana, nawet dwa razy do roku, dlatego taki atrybut pochodny nie przechowowałby faktycznej kwoty rezerwacji (ustalonej na podstawie cen wynajmu właściwych pokojów w dniu rejestracji rezerwacji), ale kwotę obliczoną na podstawie aktualnych cen wynajmu właściwych pokojów. Stąd wynika potrzeba wprowadzenia dodatkowego atrybutu *cena za dzień wynajmu w dniu złożenia rezerwacji* w klasie asocjacji *Rezerwacja pokoju*. Przyjmując takie rozwiążanie, atrybut *łączna kwota do zapłaty* będzie nadal pochodny, gdyż w każdej chwili dostępne byłyby w systemie informacje bazowe umożliwiające wyliczenie jego wartości. Fragment diagramu klas, odpowiadający omówionej propozycji, został przedstawiony na Rys. 7-5.

Rys. 7-4 Schemat pojęciowy dla systemu *Kwatery do wynajęcia*



Rys. 7-5 Fragment diagramu klas z ceną za dzień wynajmu pokoju w dniu złożenia rezerwacji

### 7.3.3 Analiza wartości początkowych

Analizując fragment schematu pojęciowego z Rys. 7-5, w aspekcie określania ewentualnych wartości początkowych dla atrybutów, warto zauważyc, że wartością początkową atrybutu `status` w klasie **Rezerwacja** może być literal tekstowy: „złożona”. Rys. 7-6 ilustruje odpowiedni fragment schematu pojęciowego z uwzględnioną analizą wartości początkowych.

### 7.3.4 Analiza wartości granicznych

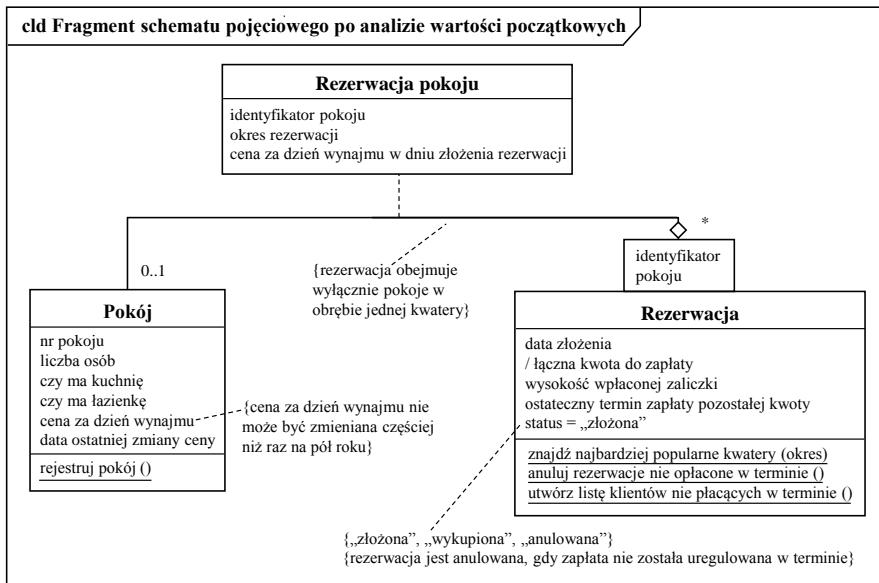
Schemat pojęciowy przedstawiony na Rys. 7-4 zawiera kilka ograniczeń określonych przy pomocy wartości stałych:

- *kwatera jest uznawana za nadmorską, gdy jej odległość od morza <= 10 km;*
- *kwatera jest uznawana za górską, gdy jej położenie n.p.m. >= 500 m n.p.m.;*
- *cena za dzień wynajmu nie może być zmieniana częściej niż raz na pół roku.*

Pierwsze dwa z wymienionych ograniczeń określają warunki, jakie muszą być spełnione, aby kwatera została zakwalifikowana do kategorii kwater górskich lub nadmorskich. Związane z nimi wartości graniczne to: maks. *odległość od morza* i min. *położenie n.p.m.*, i wynoszące aktualnie odpowiednio: *10 km* i *500 m*. Podczas

przyszłej eksploatacji systemu *Kwater do wynajęcia*, nawet w stosunkowo krótkim okresie, może zaistnieć potrzeba określenia nowych wysokości dla tych wartości granicznych (w punkcie 2 w tekście wymagań sformułowano: „*Aktualnie, aby kwatera mogła zostać uznana jako nadmorska, musi się znajdować w odległości nie przekraczającej 10 km od morza.*”), np. może zmniejszyć się maksymalna odległość od morza do 2 km, aby dana kwatera mogła widnieć w systemie jako kwatera nadmorska. Dzięki temu, że *maks. odległość od morza*, jak również *min. położenie n.p.m.* są przechowywane przy pomocy atrybutów klasowych (z zainicjowaną wartością), to wymienione wartości graniczne mogą być modyfikowane podczas działania systemu (bez potrzeby zmiany jego kodu), o ile zostałyby wprowadzone metody (o zasięgu klasowym) umożliwiające takie operacje, np.:

- *ustal maks. odległość od morza (odległość)* w klasie *Kwatera nadmorska*;
- *ustal min. położenie n.p.m. (położenie)* w klasie *Kwatera górska*.



Rys. 7-6 Fragment schematu pojęciowego z Rys. 7-5 z uwzględnioną analizą wartości początkowych

W przypadku trzeciego ograniczenia, wartością graniczną jest minimalny okres, jaki musi upływać, aby można było dokonać zmiany wartości atrybutu *cena za dzień wynajmu* w klasie *Pokój*. Jeśli ograniczenie na zmianę tego atrybutu zostanie „zaszyte” w implementacji przy pomocy wyrażenia stałego (np. z użyciem literałów, operatorów i niektórych funkcji standardowych czy też makr preprocesora), to wówczas ustalenie nowej wartości granicznej, np. wydłużenie minimalnego okresu obowiązywania ceny do co najmniej jednego roku, będzie wymagać modyfikacji

w kodzie (implementacji). Wprowadzenie do klasy *Pokój* atrybutu (o zasięgu klasowym z zainicjowaną wartością) *min. okres obowiązywania ceny = 6 mies.* umożliwiłoby przechowywanie w systemie aktualnej wartości granicznej i późniejszą jej zmianę, bez potrzeby zmiany kodu. Należałoby także przeformułować ograniczenie tak, aby powiązać je z nowym atrybutem, np. do postaci: *cena za dzień wynajmu nie może być zmieniona wcześniej niż po upływie min. okresu obowiązywania ceny.*

Analogicznie, jak w przypadku dwóch pierwszych wartości granicznych, możliwość zmiany wartości granicznej zapewniłoby wprowadzenie w klasie *Pokój* dodatkowej metody (o zasięgu klasowym) *ustal min. okres obowiązywania ceny (okres).*

W wyniku przeprowadzonej analizy wartości granicznych do schematu pojęciowego wprowadzony zostanie w klasie *Pokój* atrybut (o zasięgu klasowym i zainicjowanej wartości) *min. okres obowiązywania ceny = 6 miesięcy*, a także zmodyfikowane jedno ograniczenie. Zaproponowane metody: *ustal maks. odległość od morza (odległość)*, *ustal min. położenie n.p.m. (położenie)* i *ustal min. okres obowiązywania ceny (okres)* to metody typu *setter*, które są jednym ze środków realizacji hermetyzacji w popularnych językach programowania obiektowego. Metody te jednak nie zostaną dołączone do schematu pojęciowego, ze względu na przyjętą (przez autorów) zasadę, że na schemacie pojęciowym nie powinno się umieszczać elementów związanych raczej z fazą projektowania (lub implementacji).

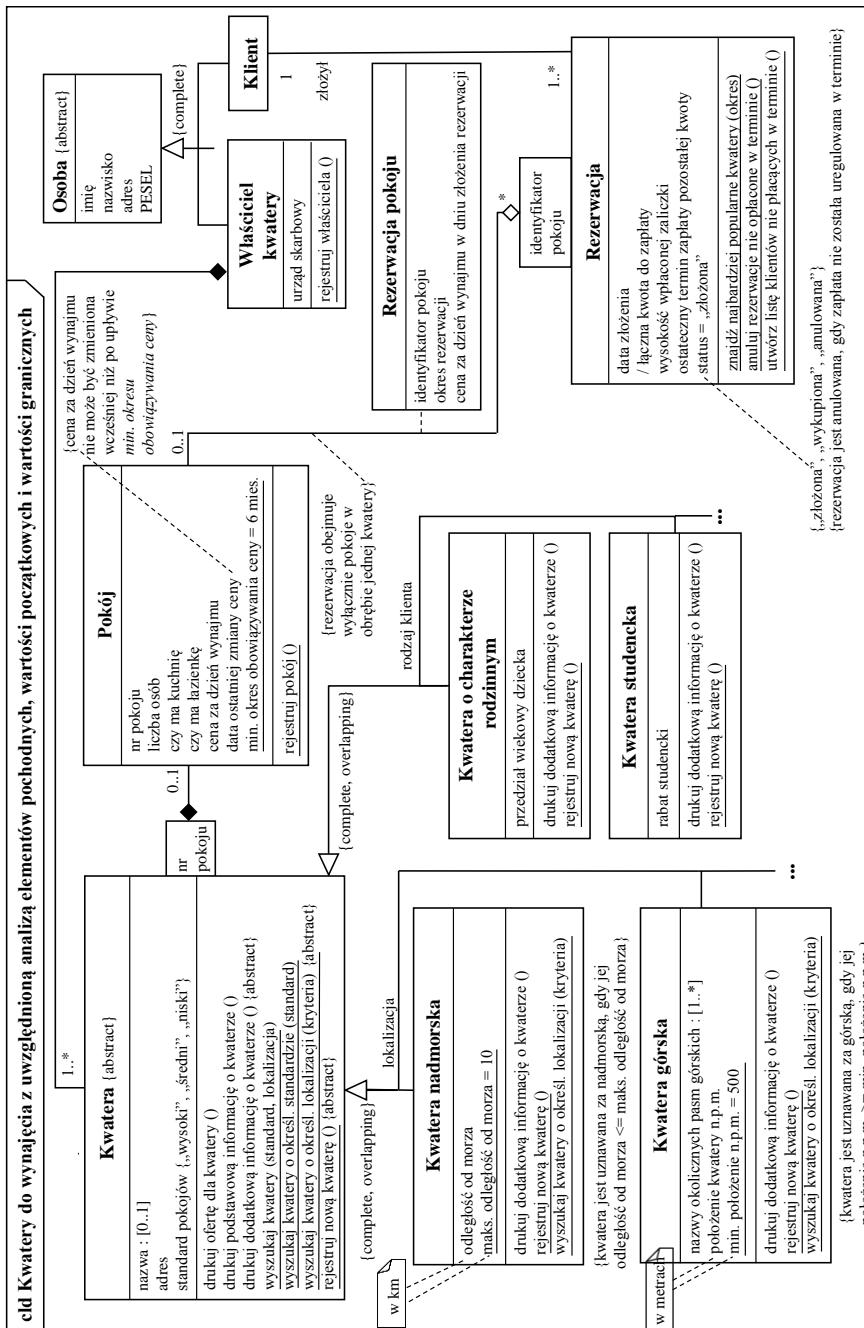
Schemat pojęciowy po zmianach uwzględniających analizę elementów pochodnych, wartości początkowych i wartości granicznych został przedstawiony na Rys. 7-7.

### 7.3.5 Zadania podstawowe

Zadania omówione w tym rozdziale są realizowane w oparciu o schemat pojęciowy z Rys. 7-7.

**Zad. 1:** Podkreśl te rodzaje dziedziczenia (*rozłączne*, *nierozłączne*, *kompletne*, *niekompletne*, *jednokrotne*, *wielokrotne*, *jednoaspektowe*, *wieloaspektowe*, *dynamiczne*, *elipsa*), które uważasz za potrzebne do wykorzystania na schemacie pojęciowym. Dla każdego podkreślonego rodzaju dziedziczenia podaj hierarchię/aspekt, w której ten rodzaj dziedziczenia wystąpił.

- Rozłączne (ang. *disjoint*): hierarchia dla osób;
- nierozłączne (ang. *overlapping*): hierarchia dla kwater w obu aspektach;
- kompletne: wszystkie hierarchie;
- jednokrotne: wszystkie hierarchie;
- jednoaspektowe: hierarchia dla osób;
- wieloaspektowe: hierarchia dla kwater (aspekty *lokalizacja* i *rodzaj klienta*);
- elipsa: hierarchia dla kwater w obu aspektach.



Rys. 7-7 Schemat pojęciowy systemu po analizie elementów pochodnych, wartości początkowych i wartości granicznych

**Zad. 2:** Jakie rodzaje metod (abstrakcyjne, zaimplementowane) może zawierać klasa abstrakcyjna, a jakie klasa konkretna? Podaj przykłady.

- Klasa abstrakcyjna może zawierać oba rodzaje metod, a klasa konkretna tylko metody zaimplementowane;
- metoda abstrakcyjna w klasie abstrakcyjnej: klasa *Kwaterna*, metoda *drukuj dodatkową informację o kwaterze ()*;
- metoda zaimplementowana w klasie abstrakcyjnej: klasa *Kwaterna*, metoda *drukuj podstawową informację o kwaterze ()*;
- metoda zaimplementowana w klasie konkretnej: klasa *Kwaterna nadmorska*, metoda *drukuj dodatkową informację o kwaterze ()*.

**Zad. 3:** Podaj przykłady metody klasowej, metody obiektowej, atrybutu klasowego i atrybutu obiektowego.

- Metoda obiektowa: metoda *drukuj podstawową informację o kwaterze ()* w klasie *Kwaterna*;
- metoda klasowa: rejestruj nową kwaterę () w klasie *Kwaterna*;
- atrybut obiektowy: *adres* w klasie *Kwaterna*;
- atrybut klasowy: *maks.odległość od morza* w klasie *Kwaterna nadmorska* (atrybut z przypisaną wartością początkową).

**Zad. 4:** Zdefiniuj i podaj przykłady dla następujących pojęć: operacja, metoda, komunikat.

- Operacja – czynność, którą można wykonać na obiekcie/obiektach klasy (instancji lub ekstensji), np. *zmień cenę (cena)*;
- metoda – implementacja operacji, np. *zmień cenę (cena)* – metoda zaimplementowana w klasie *Pokój*;
- komunikat – wyrażenie w języku programowania, którego zgłoszenie ma na celu wykonanie pewnej operacji przez adresata komunikatu, tj. instancji klasy, jej ekstensji lub samej klasy – jeśli operacja wykorzystuje inwarianty o zasięgu klasowym, jak np. atrybuty klasowe (w implementacji statyczne), o składni:

*adresat komunikatu . operacja (lista argumentów)*,

np. *p . zmień cenę (100 zł)*, gdzie *p* (adresat komunikatu) oznacza referencję do pewnej instancji klasy *Pokój*.

**Zad. 5:** Podaj przykład wywołania metody (zdefiniowanej na schemacie pojęciowym), dla którego został pominięty (nie został podany) jeden z jej argumentów.

Załóżmy, że metodą wołaną jest metoda *wyszukaj kwatery (standard, lokalizacja)* w klasie *Kwaterna*. Jeśli szukającego kwater interesują wszystkie kwatery o danej lokalizacji (niezależnie od standardu), to przykładowe wywołanie metody może być następujące:

*Kwaterna . wyszukaj kwatery (-, lokalizacja)*,

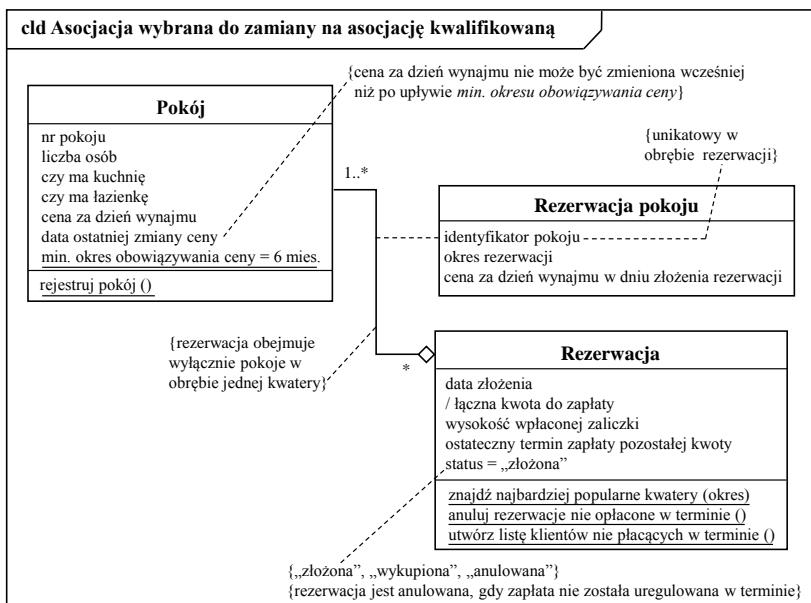
gdzie brak pierwszego argumentu został oznaczony za pomocą znaku minus „-”, zaś adresem jest ekstensja klasy *Kwaterna*.

**Zad. 6:** Wskaż przykład zastosowania polimorfizmu metod na schemacie pojęciowym dla systemu *Kwatery do wynajęcia*.

Metody polimorficzne to metody: *drukuj dodatkową informację o kwaterze ()*, umieszczone w klasach hierarchii zbudowanej dla kwater (w obu aspektach), czy też *wyszukaj kwatery o określ. lokalizacji (kryteria)* (w aspekcie *lokalizacja*; kryteria są określone na podstawie różnych atrybutów: *odległość od morza, położenie kwatera n.p.m.* itd.).

**Zad. 7:** Asocjację ze schematu pojęciowego posiadającą niesymetryczne liczności oraz atrybut (lub klasę asocjacji) zamień na asocjację kwalifikowaną. Uzasadnij, dlaczego poddałeś zamianie właśnie tę asocjację.

Na diagramie umieszczone dwie asocacje kwalifikowane: pierwsza pomiędzy klasami *Kwaterna* i *Pokój*, a druga pomiędzy klasami *Pokój* i *Rezerwacja*. Pierwsza z nich wskazuje, że kluczem do przeszukiwania zbioru pokojów przypisanych do danej kwatery może być numer pokoju. Z kolei druga, że kluczem do przeszukiwania zbioru pokojów przypisanych do danej rezerwacji może być identyfikator pokoju, unikatowy w obrębie danej rezerwacji. Rys. 7-8 przedstawia fragment schematu pojęciowego, dla drugiej z asocjacji kwalifikowanych, sprzed zamiany asocjacji „zwykłej” na asocjację kwalifikowaną.



Rys. 7-8 Fragment schematu pojęciowego bez wykorzystania asocjacji kwalifikowanej

**Uwaga:** Asocjację kwalifikowaną wykorzystuje się do wskazania klucza użytecznego do przeszukiwania dużych zbiorów danych (a nie zbiorów jednoelementowych). W tym konkretnie przypadku kwalifikator *identyfikator pokoju* może być wykorzystany jako klucz do przeszukiwania zbioru pokojów przypisanych do pojedynczej rezerwacji. Wprowadzenie asocjacji kwalifikowanej zmienia liczność tego końca asocjacji, który znajduje się naprzeciwko kwalifikatora. Nowa liczność bezpośrednio wskazuje „siłę” klucza, czyli np. liczność 0..1 mówi o tym, że wartość kwalifikatora pozwala na jednoznaczną identyfikację pokoju w ramach danej rezerwacji.

**Zad. 8:** Wskaż na schemacie pojęciowym asocjację, która jest agregacją lub kompozycją (lub może kwalifikować się do zamiany na takową). Objasnij, dlaczego zdecydowałeś się na wybór agregacji/kompozycji, a nie „zwykłej” asocjacji? Ponadto, dlaczego wybrałeś agregację, a nie kompozycję (czy też odwrotnie)?

Na diagramie występują dwie kompozycje: pierwsza pomiędzy klasami: *Kwaterna* i *Pokój* oraz druga pomiędzy klasami: *Właściciel kwatery* i *Kwaterna*. Obie kompozycje, lepiej niż zwykła asocjacja, wskazują na związek części-całość występujący pomiędzy kwaterną i jej pokojami i odpowiednio, pomiędzy właścicielem kwatery a jego kwaterami. Wykorzystano kompozycje, a nie agregacje, ponieważ w obu przypadkach cykl życiowy części zawiera się w cyklu żywiołowym całości, tzn. usunięcie z systemu obiektu klasy *Kwaterna* będzie skutkowało usunięciem tych obiektów klasy *Pokój*, które były związane z usuwaną kwaterną. Podobnie jest w drugim przypadku. Agregacja natomiast została wykorzystana pomiędzy klasami *Rezerwacja* i *Pokój* – ponieważ pomiędzy rezerwacją a pokojami, które ta rezerwacja obejmuje, występuje związek części-całość, ale cykl życiowy obiektu klasy *Pokój* nie zawiera się w cyklu żywiołowym obiektu *Kwaterna*. Na przestrzeni czasu każdy z pokojów mógł być rezerwowany wiele razy, stąd agregacja, a nie kompozycja.

**Zad. 9:** Wskaż na schemacie pojęciowym przykładowe ograniczenie. Na jaki element modelu zostało ono nałożone? Czy jest to ograniczenie statyczne czy dynamiczne? Zaproponuj, jaka metoda i w jakiej klasie mogłaby być odpowiedzialna za jego zapewnienie.

Np. ograniczenie, że {*kwatera nie może być oddalona o więcej niż 10 km od morza, aby mogła być uznana za kwatery nadmorską*}; jest to ograniczenie statyczne nałożone nie na atrybut *odległość od morza* w klasie *Kwaterna nadmorska*, ale na istnienie obiektu tej klasy. Ograniczenie to powinno być zaimplementowane w metodzie rejestrującej nową kwatery w klasie *Kwaterna nadmorska*.

### 7.3.6 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

- Wybór nieodpowiednich nazw dla klas. Po pierwsze, nazwa klasy to zazwyczaj rzeczownik w liczbie pojedynczej (w nielicznych przypadkach jest inaczej), a po

drugie, powinna ona być zgodna z „zawartością” obiektu, tzn. jeżeli pojedynczy obiekt przechowuje informację o jednej kwaterze, to klasę trzeba nazwać *Kwatera*, ale jeżeli jeden obiekt przechowuje informacje o zbiorze kwater, to w takim przypadku wybieramy nazwę *Kwatery*. W nazwach podklas trzeba umieszczać nazwę nadklasty, czyli nazwa klasy to nie *Rodzinna*, ale *Kwatera rodzinna* (analogicznie poprawne są nazwy klas: *Kwatera nadmorska* i *Kwatera górska*, a niepoprawne: *Nadmorska* czy *Górska*).

- Błędne rozmieszczanie metod na schemacie pojęciowym. Metody umieszczamy tam, gdzie znajdują się dane, na których one operują, a nie tam, gdzie potencjalnie mogą być wywołane. Np. metodę *rejestruj rezerwację()* umieszczamy w klasie *Rezerwacja*, a nie w klasie *Właściciel kwatery*.
- Umieszczanie metod w klasach opisujących aktorów systemu (np. metoda *znajdź najbardziej popularne kwaterę()* umieszczona w klasie *Pracownik*). *Uwaga:* na tym etapie aktorów w ogóle nie należy wprowadzać do diagramu klas.
- Metody tworzące nowe obiekty w tej klasie, w której zostały umieszczone, to metody klasowe. A więc metoda *rejestruj rezerwację()* jest metodą klasową w klasie *Rezerwacja*, natomiast umieszczona w klasie *Właściciel kwatery* byłaby metodą obiektu.
- Brak argumentów dla metod – np. metoda *znajdź najbardziej popularne kwaterę()* w klasie *Rezerwacja*, zgodnie z tekstem wymagań (punkt 8.5) wymaga podania argumentu *okres*.
- Wprowadzanie konstrukcji projektowych do schematu pojęciowego, np. wprowadzenie klas *Rodzinna kwatera nadmorska* i *Studencka kwatera nadmorska* oraz odpowiednio *Rodzinna kwatera górska* i *Studencka kwatera górska* zamiast dziedziczenia wieloaspektowego (lub łącznie z dziedziczeniem wieloaspektowym). Dziedziczenie wieloaspektowe w zupełności wystarcza do opisania problemu. Konstrukcje projektowe pozwolą na jego zaimplementowanie w etapach późniejszych.
- Nieuwzględnianie w trakcie modelowania upływu czasu (którego efektem jest tzw. archiwizowanie danych). Np., na przestrzeni czasu pokój mógł być wynajmowany wielokrotnie, mógł wchodzić w skład wielu rezerwacji, co ma znaczenie dla poprawnego oznaczania liczności asocjacji między klasami *Pokój* i *Rezerwacja*.

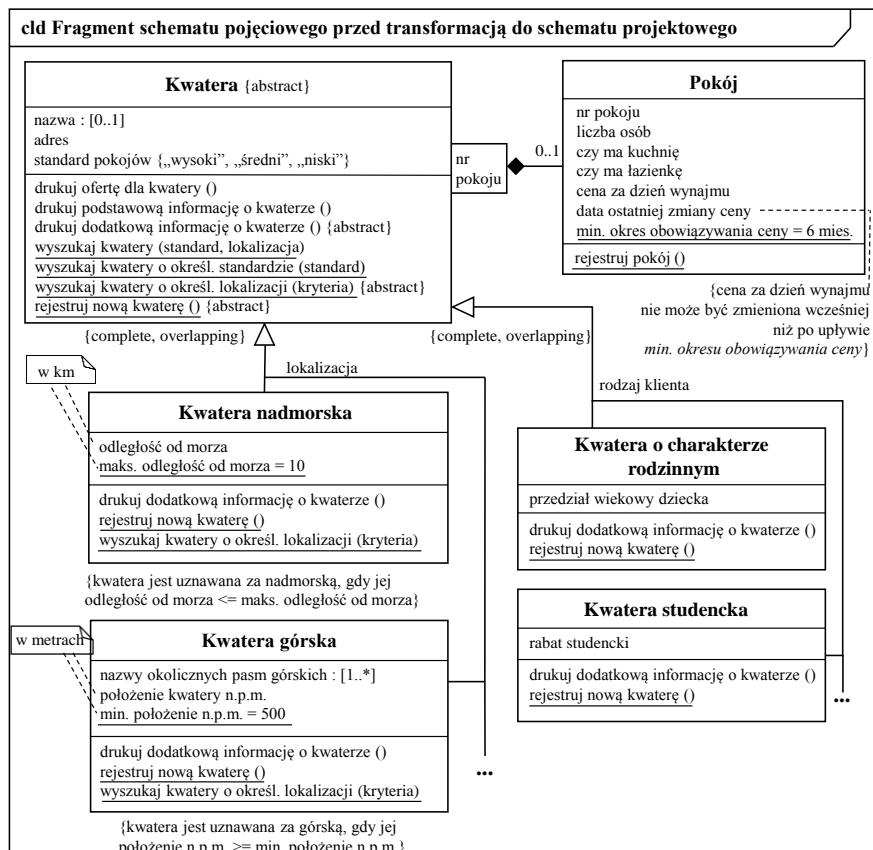
### **7.3.7 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji**

W popularnych obiektowych językach programowania (jak np. Java, C#, czy C++) pewne elementy modelu pojęciowego nie mogą być zaimplementowane w sposób natywny ze względu na ograniczenia języka lub środowiska implementacji [37, 40]. Do takich konstrukcji należą m.in. dziedziczenie nietrywialne (np. nierożłączne,

wieloaspektowe, dynamiczne), asocjacje, niektóre rodzaje atrybutów (np. pochodne).

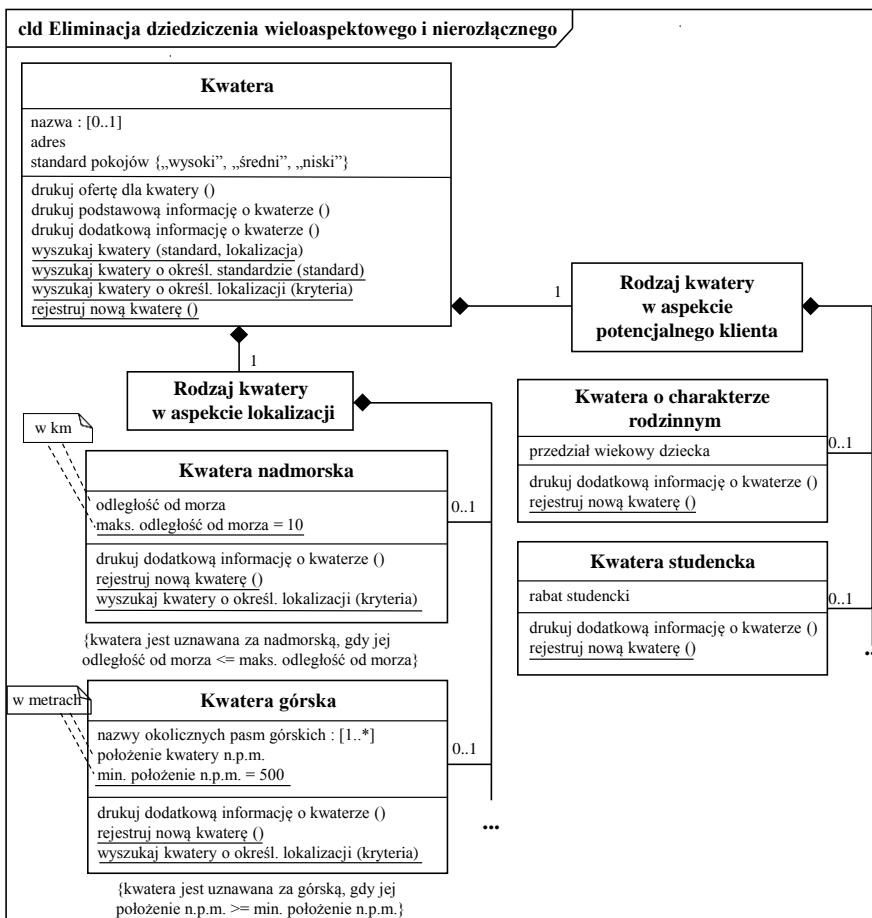
Na Rys. 7-9 przedstawiony jest fragment schematu pojęciowego, niezbędny dla przeprowadzenia analizy dynamicznej dla przypadku użycia *Zmień cenę pokoju* (Zad. 6 w rozdziale 7.4.1). Zawiera hierarchię kwater z dwoma aspektami dziedziczenia, klasę *Pokój* oraz kompozycje kwalifikowaną między nimi.

Ponieważ hierarchie dziedziczenia w obu aspektach dla kwater nie są rozłączne, a także ze względu na zawarte w nich metody polimorficzne wydaje się, że najbardziej odpowiednim sposobem implementacji tej hierarchii będzie zastąpienie dziedziczenia za pomocą kompozycji – jak na Rys. 7-10. Klasa *Kwaterna* przestała być klasą abstrakcyjną, zaś klasy *Rodzaj kwatery w aspekcie lokalizacji* i *Rodzaj kwatery w aspekcie potencjalnego klienta* reprezentują pierwotne aspekty. Mimo że nie mają określonych (aktualnie) żadnych własnych inwarariantów, to umożliwiają zachowanie separacji własności kwater w ramach aspektów.

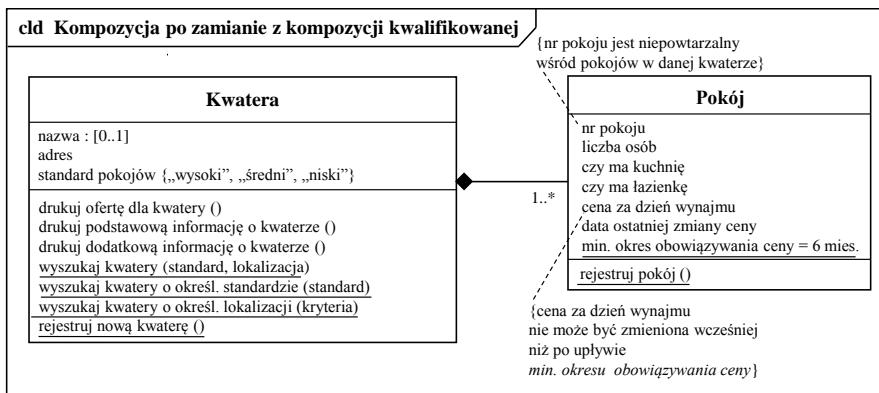


Rys. 7-9 Fragment schematu pojęciowego przed transformacją do schematu projektowego

Realizacja asocjacji w językach obiektowych polega zwykle na wprowadzeniu atrybutów (pół) w klasach połączonych asocjacją, mających za zadanie przechowywanie identyfikatorów (referencji lub identyfikatorów liczbowych) powiązanych obiektów klasy po przeciwej stronie asocjacji. Należy także zaznaczyć, że nie w każdym przypadku niezbędna będzie nawigacja w obu kierunkach pomiędzy powiązanymi obiektem – a na schemacie projektowym należy pozostawić tylko te konstrukcje, które są konieczne dla poprawnej i efektywnej implementacji metod niezbędnych do zrealizowania przypadków użycia wyróżnionych w trakcie analizy funkcjonalnej.



Rys. 7-10 Eliminacja dziedziczenia wieloaspektowego i nierożłącznego za pomocą kompozycji



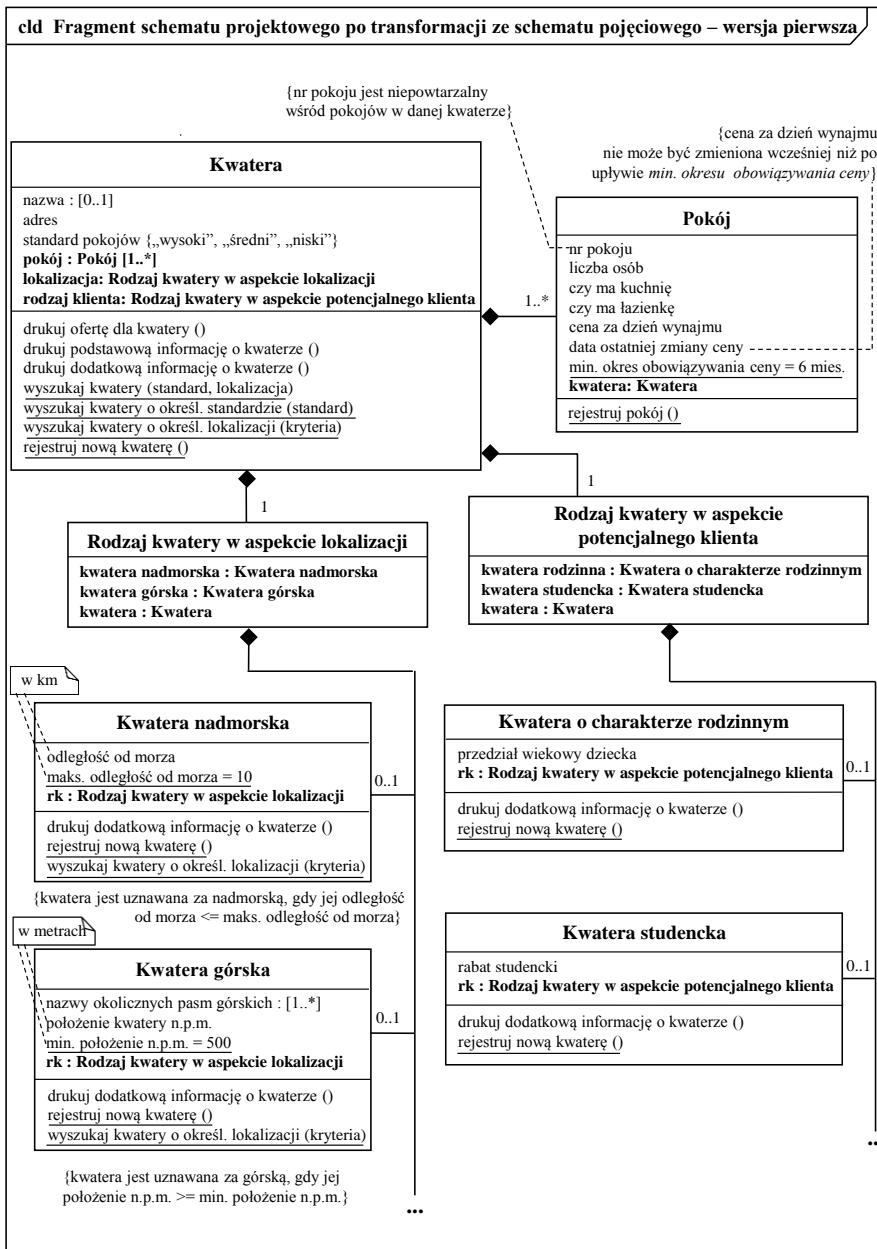
Rys. 7-11 Fragment diagramu klas z asocjacją po zamianie z kompozycji kwalifikowanej

Kompozycja kwalifikowana podczas transformacji do schematu projektowego może zostać zamieniona na „zwykłą” kompozycję. Ponieważ w systemie *Kwatery do wynajęcia* niezbędna będzie nawigacja w obu kierunkach dla powiązanych obiektów klas *Kwatera* i *Pokój* (np. metoda *drukuj informację o kwaterze ()* wymaga nawigacji do wszystkich pokojów z danej kwatery, a metoda *znajdź najbardziej popularne kwatery (okres)* – wymaga nawigacji w drugą stronę, tj. prowadzącej z pokoju do kwater). Odpowiedni fragment diagramu klas zawiera Rys. 7-11.

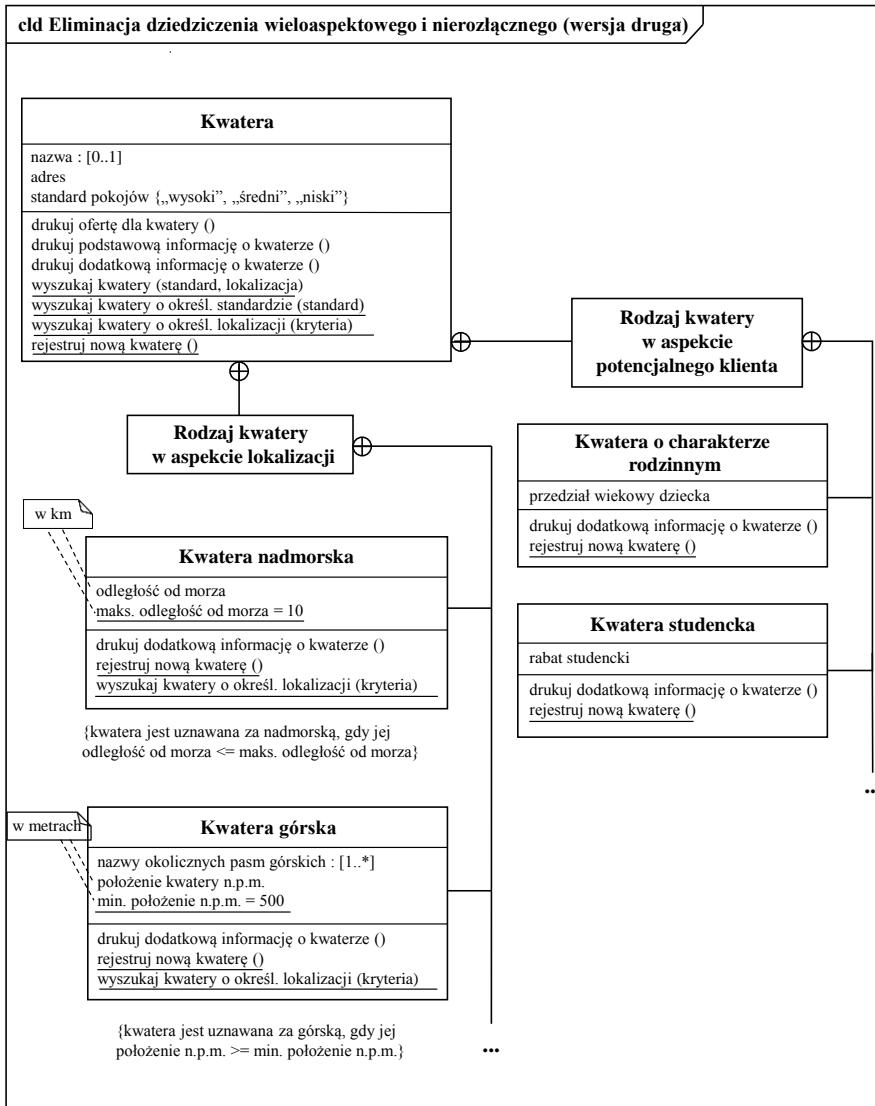
Schemat projektowy odpowiadający fragmentowi schematu pojęciowego z Rys. 7-9 został przedstawiony na Rys. 7-12. Dla lepszej czytelności atrybuty implementujące asocje zostały wytłuszczone.

W alternatywnej propozycji (wersji drugiej) transformacji schematu z Rys. 7-9 do schematu projektowego, założono wykorzystanie klas zagnieżdżonych zamiast implementowania kompozycji za pomocą zmiennych referencyjnych (jak w wersji pierwszej). Rys. 7-13 przedstawia klasę *Kwatera* z określonymi klasami zagnieżdżonymi dla aspektów, tj. klasami *Rodzaj kwatery w aspekcie lokalizacji* oraz *Rodzaj kwatery w aspekcie potencjalnego klienta*. Klasa *Rodzaj kwatery w aspekcie lokalizacji* posiada zagnieżdżone klasy *Kwatera nadmorska* oraz *Kwatera górska*, natomiast klasa *Rodzaj kwatery w aspekcie potencjalnego klienta* definiuje klasy zagnieżdżone *Kwaterna o charakterze rodzinny* oraz *Kwaterna studencka*. Rys. 7-14 przedstawia fragment schematu projektowego dla wersji drugiej, na którym atrybuty *lokalizacja* i *rodzaj kwatery* w klasie *Kwatera* przechowują referencje do jej „podobiektów” – tj. wystąpień klas zagnieżdżonych (obiekty klas zagnieżdżonych mają dostęp do własności obiektów klas nadzujących, dlatego nie jest potrzebne przechowywanie zmiennych referencyjnych dla nawigacji do obiektów klas nadzujących).

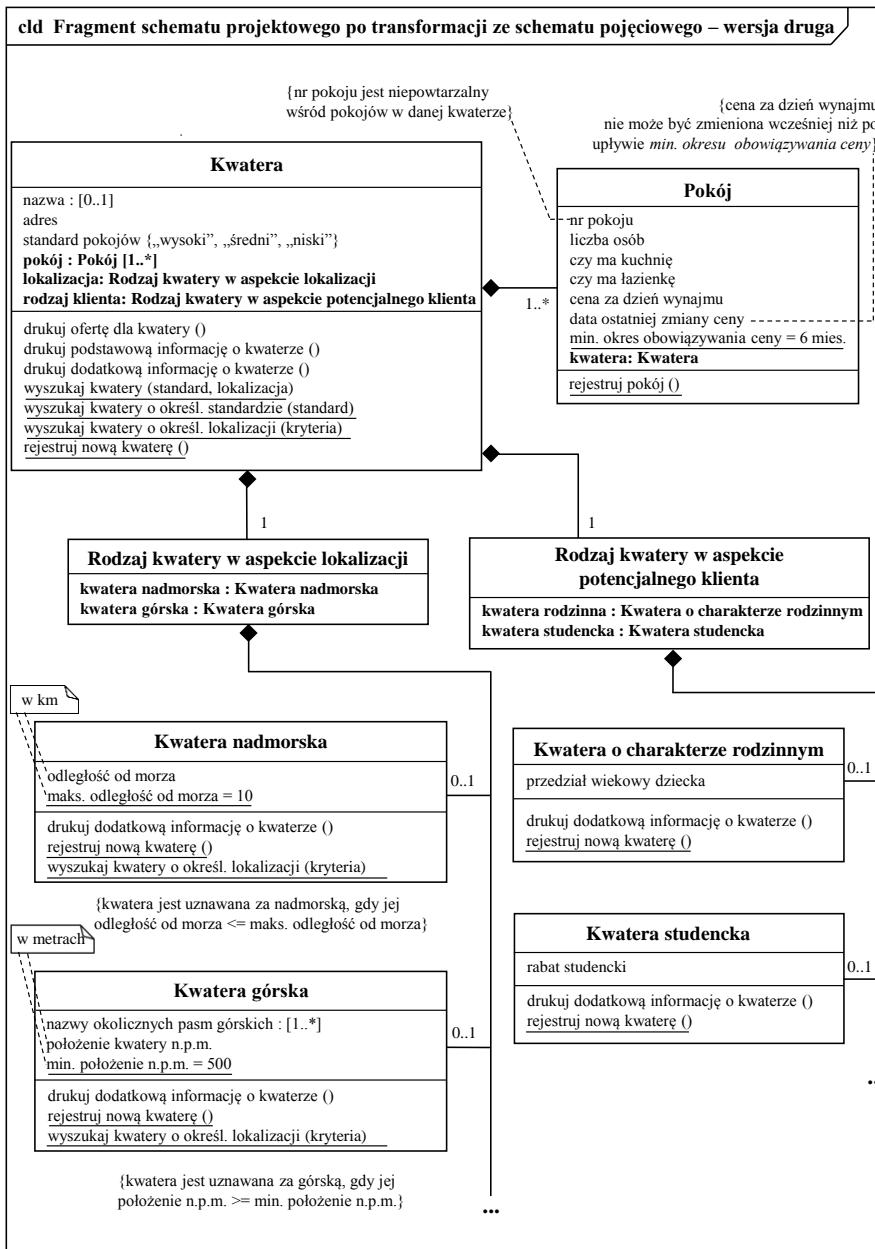
Diagram struktur złożonych dla klasy *Kwatera* z Rys. 7-14 został zilustrowany na Rys. 7-15. Dla części (ang. *parts*) obiektów złożonych pominięto typy atrybutów prostych, jak np. string, integer itp.



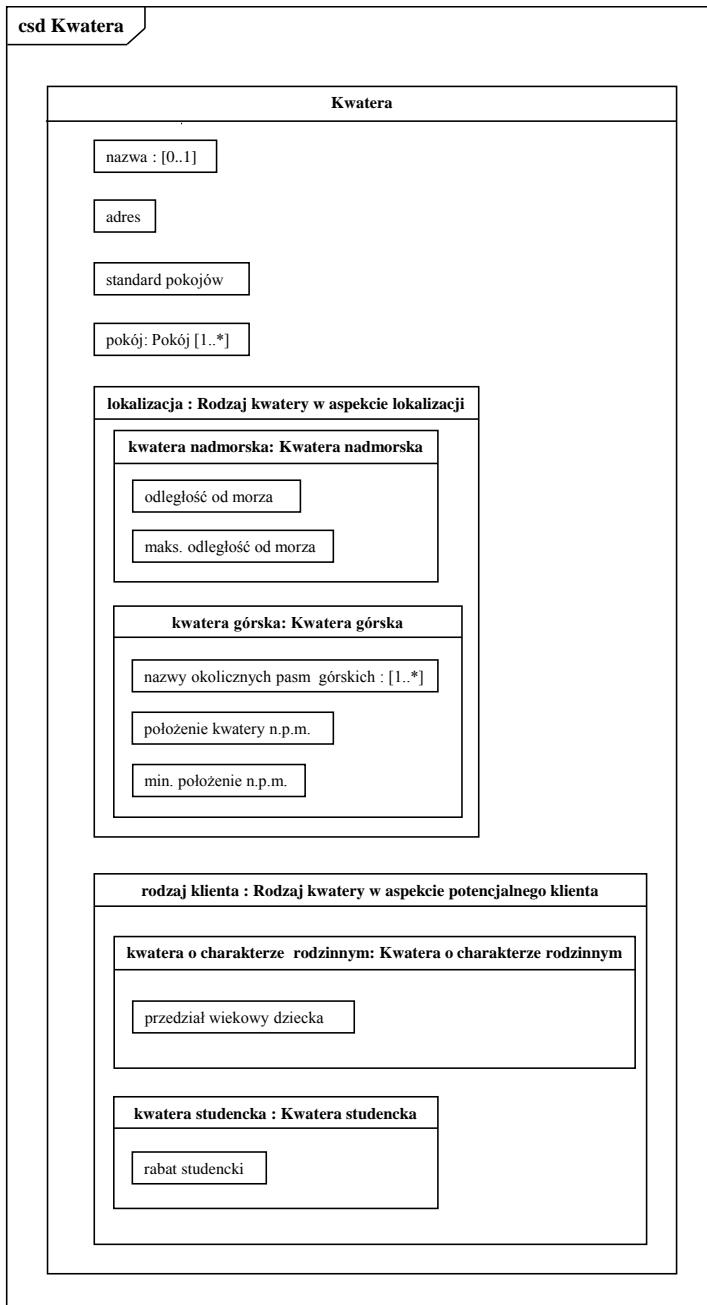
Rys. 7-12 Fragment schematu projektowego po transformacji ze schematu pojęciowego (wersja pierwsza)



Rys. 7-13 Eliminacja dziedziczenia wieloaspektowego i nierożłącznego (wersja druga)



Rys. 7-14 Fragment schematu projektowego po transformacji ze schematu pojęciowego (wersja druga)



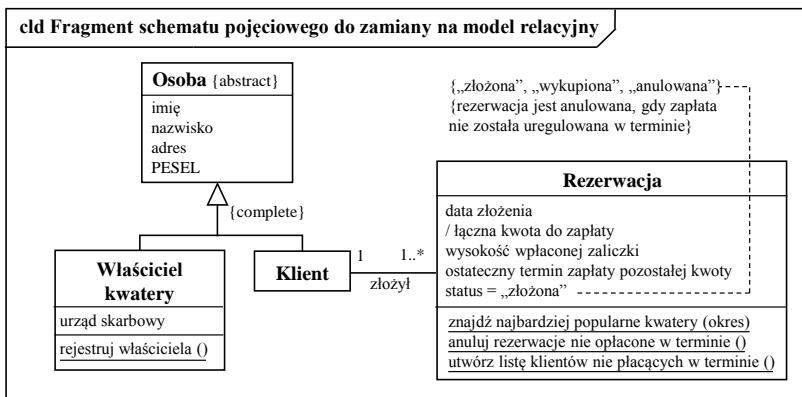
Rys. 7-15 Diagram struktur złożonych dla klasy *Kwatra* dla schematu projektowego z Rys. 7-14

### 7.3.8 Schemat relacyjny dla fragmentu diagramu klas

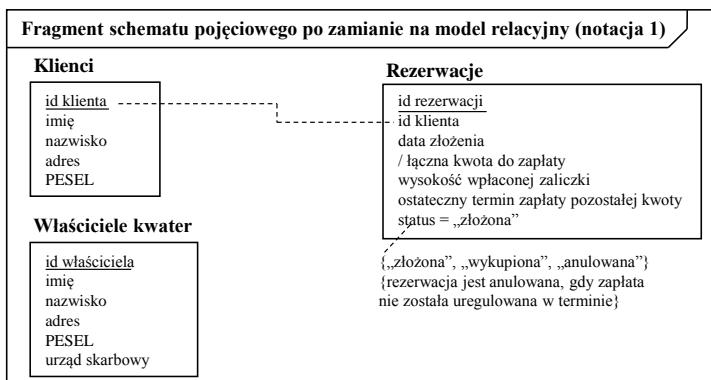
**Polecenie:** Fragment schematu pojęciowego, zawierający co najmniej trzy klasy, związek dziedziczenia oraz asocjacje, przedstaw w modelu relacyjnym.

Na Rys. 7-16 przedstawiono fragment schematu pojęciowego, zawierający cztery klasy, związek dziedziczenia i asocjację.

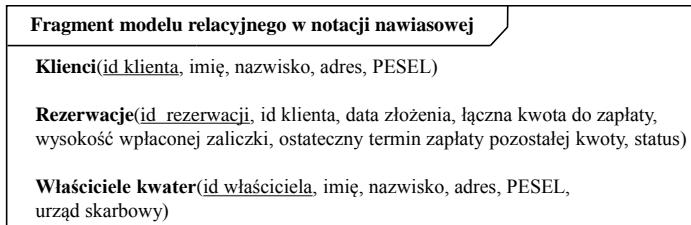
Na Rys. 7-17 pokazano model relacyjny [47, 48] tego fragmentu w notacji 1 [41]. Rys. 7-18 przedstawia odpowiedni model relacyjny w notacji nawiasowej, zaś Rys. 7-19 – w notacji *Crow's Foot*.



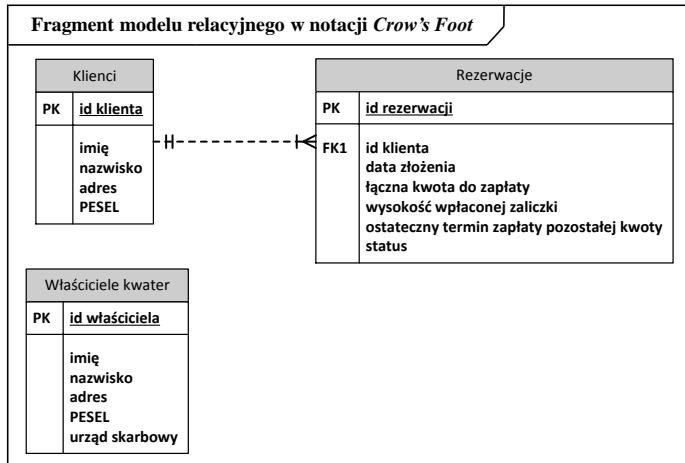
Rys. 7-16 Fragment schematu pojęciowego podlegający zamianie na model relacyjny



Rys. 7-17 Model relacyjny fragmentu schematu pojęciowego przedstawionego na Rys. 7-16 (w notacji 1)



Rys. 7-18 Model relacyjny fragmentu schematu pojęciowego przedstawionego na Rys. 7-16 w notacji nawiasowej



Rys. 7-19 Model relacyjny fragmentu diagramu klas przedstawionego na Rys. 7-16 (w notacji *Crow's Foot*)

## 7.4 Analiza dynamiczna

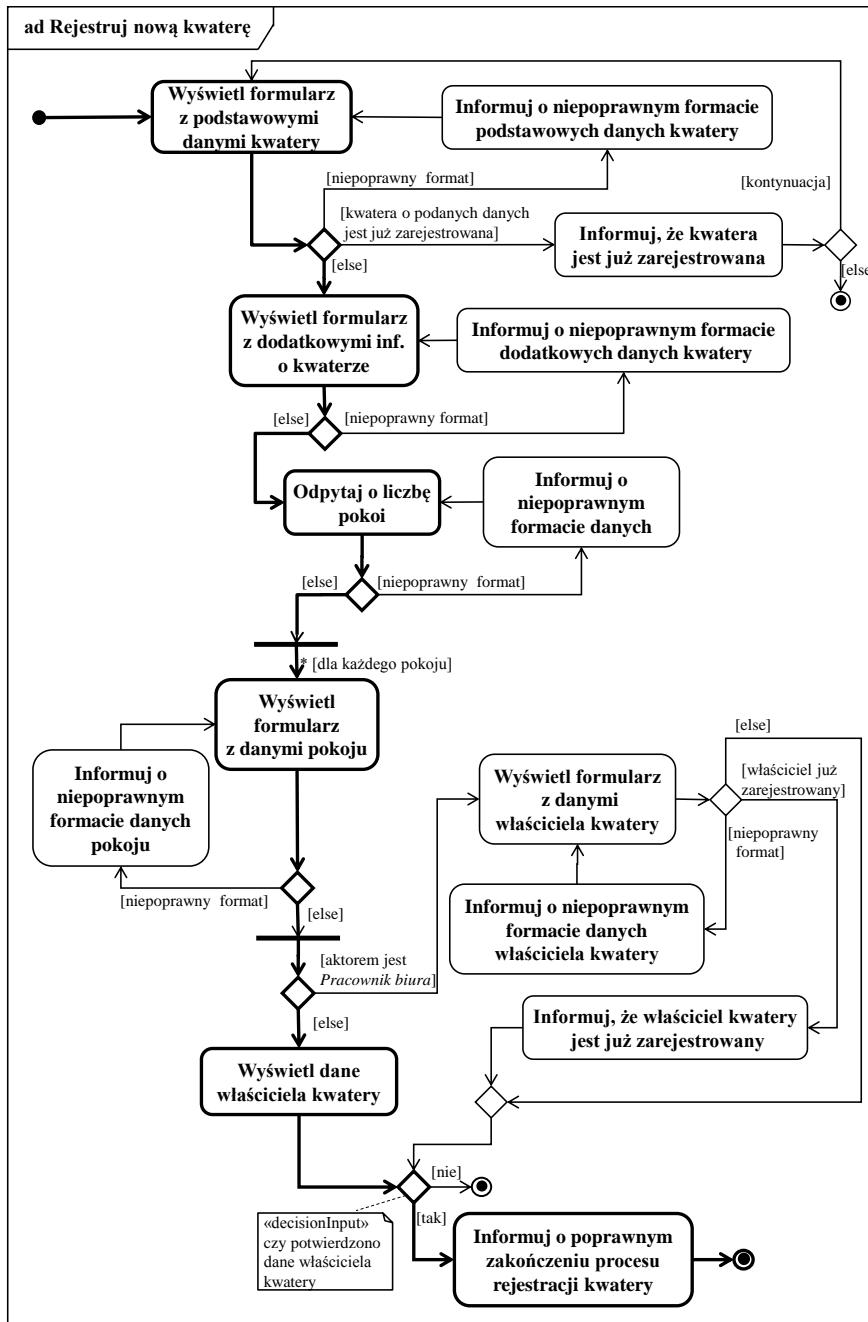
### 7.4.1 Zadania podstawowe

**Zad. 1:** Dla funkcjonalności *Rejestruj nową kwaterę* napisz scenariusz.

Przykładowy scenariusz dla wybranego przypadku użycia został przedstawiony w Tab. 7-3.

**Zad. 2:** Dla funkcjonalności *Rejestruj nową kwaterę* skonstruuj diagram aktywności zgodny ze scenariuszem z zadania poprzedniego.

Diagram aktywności zawarty na Rys. 7-20 ilustruje przykładową realizację dla scenariusza, o którym mowa w zadaniu poprzednim. Dla zwiększenia czytelności diagramu główny przepływ zdarzeń (aktywności i przejścia) zostały narysowane nieco grubszą kreską niż elementy należące do przepływu alternatywnych.

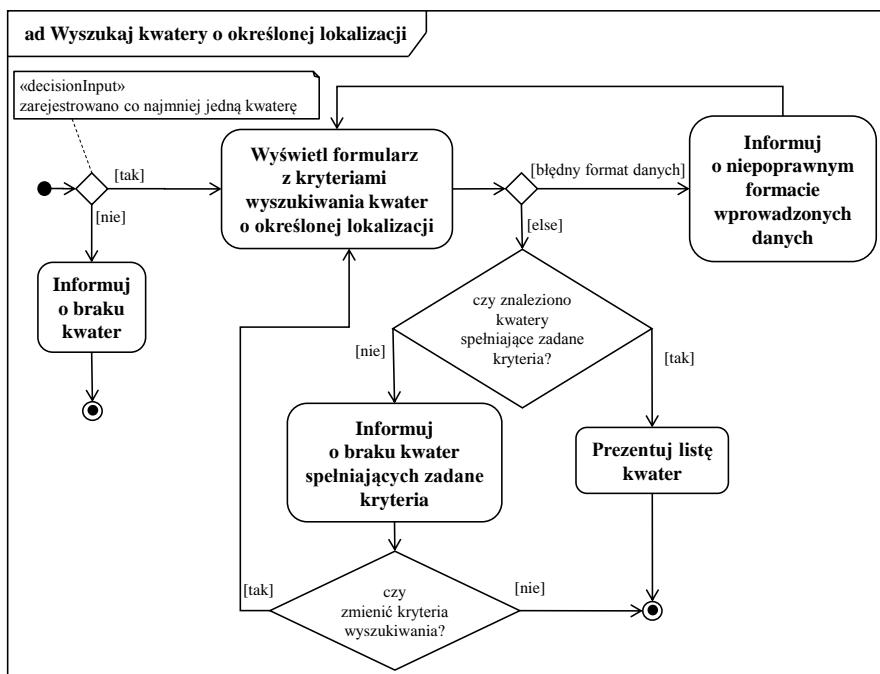
Rys. 7-20 Diagram aktywności dla przypadku użycia *Rejestruj nową kwaterę*

**Zad. 3:** Dla funkcjonalności *Wyszukaj kwaterę o określonej lokalizacji* napisz scenariusz.

Przykładowy scenariusz dla wybranego przypadku użycia został przedstawiony w Tab. 7-1.

**Zad. 4:** Dla funkcjonalności *Wyszukaj kwaterę o określonej lokalizacji* skonstruuj diagram aktywności.

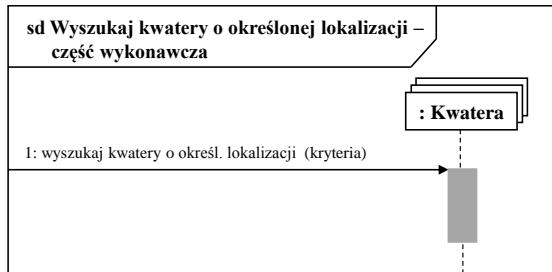
Diagram aktywności zawarty na Rys. 7-21 ilustruje przykładową realizację dla scenariusza sporzązonego w zadaniu poprzednim.



Rys. 7-21 Diagram aktywności dla przypadku użycia *Wyszukaj kwaterę o określonej lokalizacji*

**Zad. 5:** Sporządź diagram interakcji dla funkcjonalności *Wyszukaj kwaterę o określonej lokalizacji*.

Przykładowy diagram sekwencji dla przypadku użycia *Wyszukaj kwaterę o określonej lokalizacji* ilustruje Rys. 7-22. Metoda wywołuje odpowiednią metodę w GUI, służącą do wprowadzenia kryteriów dla lokalizacji. Na diagramie została przedstawiona wyłącznie część wykonawcza, natomiast część dialogowa interakcji (GUI) została pominięta.

Rys. 7-22 Diagram sekwencji dla przypadku *Wyszukaj kwaterę o określonej lokalizacji*

**Zad. 6:** Dla funkcjonalności *Zmień cenę pokoju* napisz scenariusz, narysuj diagram aktywności oraz diagram interakcji.

Przykładowy scenariusz dla wybranego przypadku użycia został przedstawiony w Tab. 7-2.

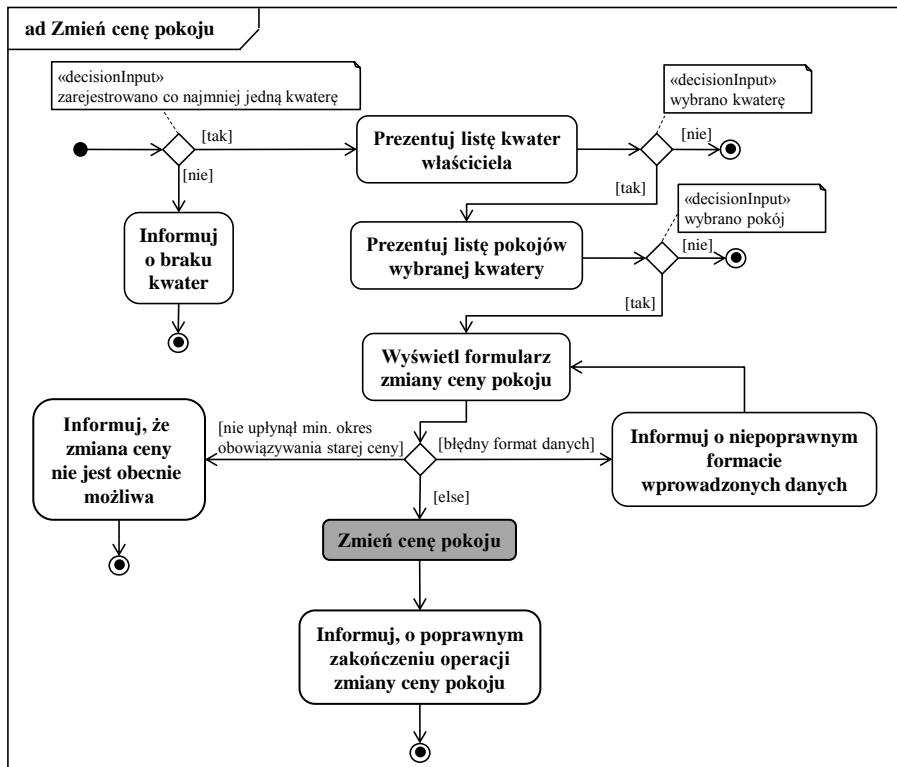
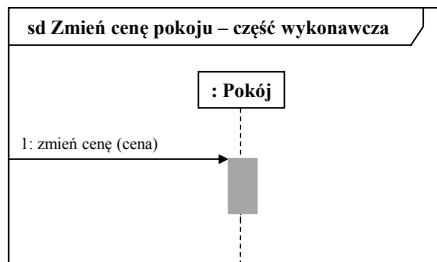
Rys. 7-23 Diagram aktywności dla przypadku użycia *Zmień cenę pokoju*

Diagram aktywności zawarty na Rys. 7-23 ilustruje przykładową realizację dla podanego scenariusza. Część wykonawcza została oznaczona ciemniejszym tłem.

Diagram sekwencji dla części wykonawczej omawianego przypadku użycia został przedstawiony na Rys. 7-24. Przyjęte zostało dlań założenie, że pokój, którego cena jest zmieniana, został już odszukany (zgodnie ze scenariuszem i diagramem aktywności) przez część dialogową przypadku użycia (GUI) oraz że cena została podana w odpowiednim formacie, a także minął minimalny okres obowiązywania ceny (obecnie 6 miesięcy).



Rys. 7-24 Diagram sekwencji dla przypadku użycia *Zmień cenę pokoju* (część wykonawcza)

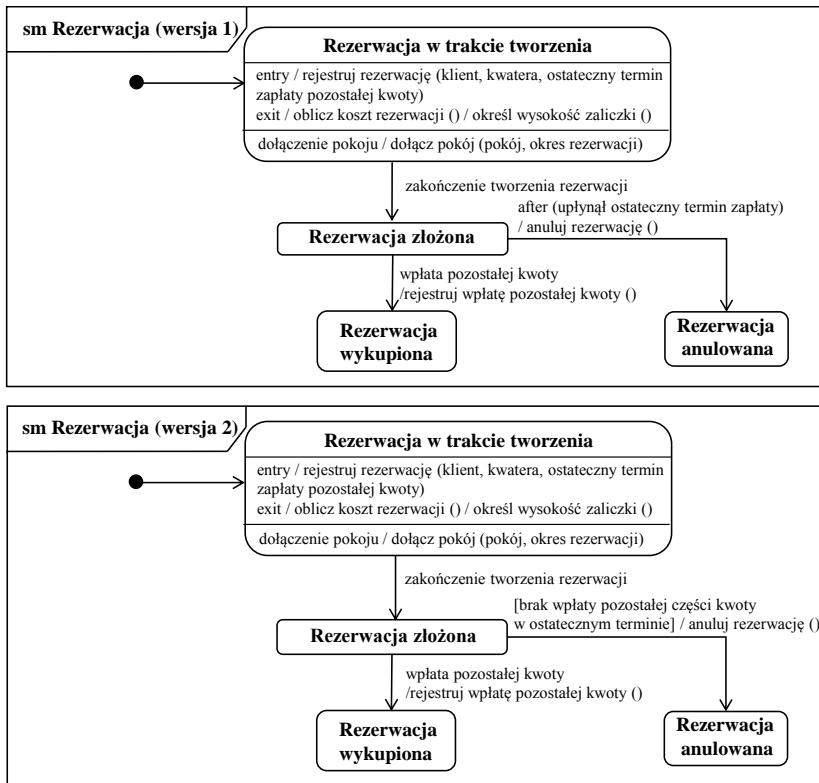
**Zad. 7:** Dla klasy *Rezerwacja* z diagramu klas wskaż kilka stanów (co najmniej dwa stany, nie licząc stanów początkowego i końcowego), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów, zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

*Uwaga:* W tym zadaniu nie jest wymagana cała maszyna stanowa, wystarczy zaprezentować dwa stany wraz ze specyfikacją przejścia.

Dwie wersje maszyny stanowej zachowania dla klasy *Rezerwacja* zostały przedstawione na Rys. 7-25.

W wersji pierwszej nieopłacona rezerwacja jest anulowana przez akcję (operację) *anuluj rezerwację()* w wyniku obsługi zdarzenia typu *after*, które może zostać automatycznie wyzwolone po upływie ostatecznego terminu zapłaty dla danej rezerwacji, jeśli nie została ona jeszcze opłacona w całości. Natomiast wersja druga zakłada tylko, że nieopłacona w pełni rezerwacja może być anulowana w przypadku, gdy spełnione są ograniczenia związane z upływem czasu za pomocą (także) akcji (operacji) *anuluj rezerwację()*, ale nie ustala, co i w jaki sposób powoduje uruchomienie tej akcji.

*Uwaga:* Należy zwrócić uwagę na brak informacji w tekście wymagań o warunkach usuwania rezerwacji. W związku z powyższym diagram nie zawiera stanu końcowego.

Rys. 7-25 Maszyna stanowa dla klasy *Rezerwacja*, dwie wersje

#### 7.4.2 Omówienie zmian na schemacie pojęciowym i schemacie projektowym, wynikających z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań

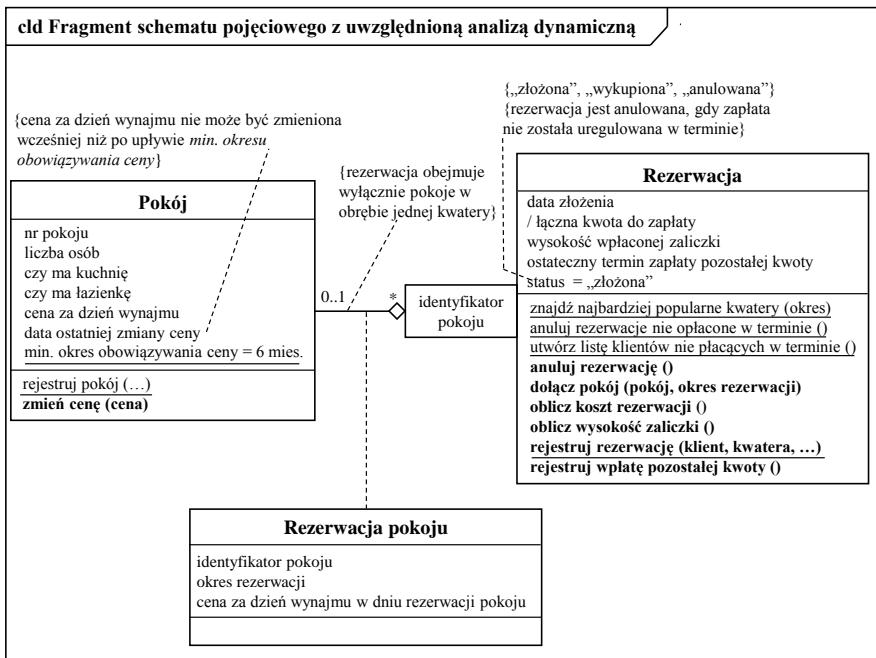
Po przeprowadzeniu analizy dynamicznej należy uzupełnić schemat pojęciowy o nowe elementy, które zostały ujęte na diagramach interakcji oraz diagramach stanów. Lista zmian została zamieszczona w Tab. 7-4.

Tab. 7-4 Lista zmian na schemacie pojęciowym i schemacie projektowym po analizie dynamicznej

Klasa	Zmiana
<i>Pokój</i>	dodatkowa metoda: <i>zmień cenę (cena)</i>

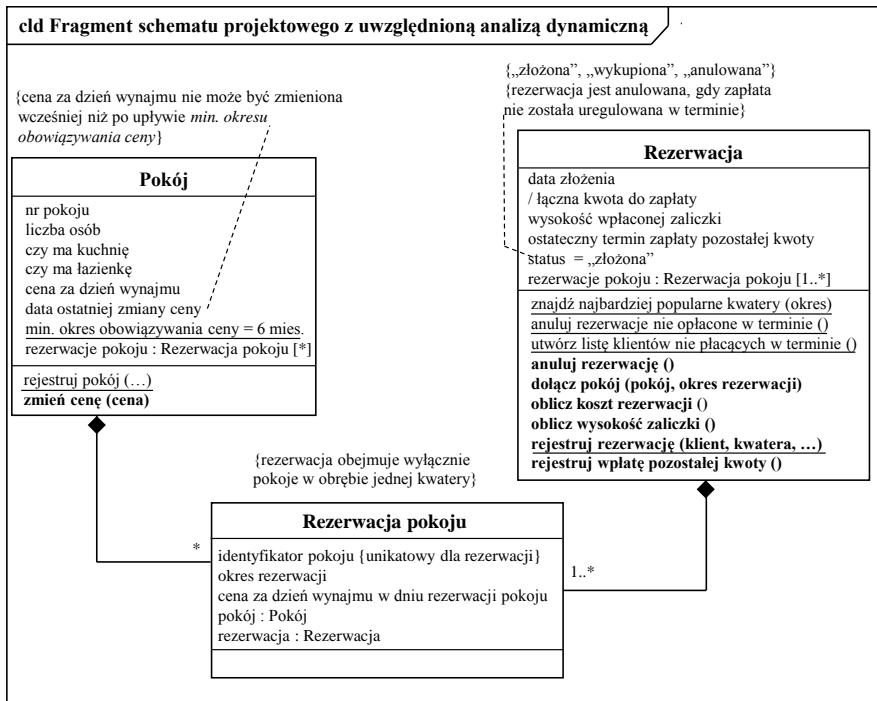
<i>Rezerwacja</i>	dodatkowe metody:  <u>anuluj rezerwację ()</u> <u>dolacz pokój (pokój, okres rezerwacji)</u> <u>oblicz koszt rezerwacji ()</u> <u>oblicz wysokość zaliczki ()</u> <u>rejestruj rezerwację (klient, kwatera, ostateczny termin zapłaty pozostały kwoty)</u> <u>rejestruj wpłatę pozostały kwoty ()</u>
-------------------	--

Rys. 7-26 ilustruje fragment schematu pojęciowego, na którym zostały wprowadzone metody opisane w Tab. 7-4.



Rys. 7-26 Fragment schematu pojęciowego z uwzględnioną analizą dynamiczną z Tab. 7-4

Rys. 7-27 przedstawia fragment schematu projektowego, na którym zostały wprowadzone metody opisane w Tab. 7-4.



Rys. 7-27 Fragment schematu projektowego po uwzględnieniu analizy dynamicznej z Tab. 7-4

## 7.5 Zadania do samodzielnej pracy

1. Podaj scenariusz dla wybranego (innego niż omówionego w analizie funkcjonalnej lub dynamicznej) przypadku użycia i zaproponuj jego podział na podprzypadki.
2. Objaśnij różnicę między dziedziczeniem wielokrotnym (ang. *multiple inheritance*) a wieloaspektowym (ang. *multiple aspect inheritance*). Czy któryś z tych obu rodzajów dziedziczenia wystąpił na schemacie pojęciowym?
3. Objaśnij różnicę między instancją klasy a ekstensją klasy. Podaj przykładowe ekstensje klas wchodzących w skład hierarchii dla kwater na schemacie pojęciowym.
4. Objaśnij pojęcie polimorfizm metod. Wskaż przykład zastosowania na schemacie pojęciowym. Czy polimorfizm metod wiąże się zawsze z przesłanianiem metod?
5. Objaśnij różnicę pomiędzy komentarzem a ograniczeniem. Podaj odpowiednie przykłady dla schematu pojęciowego.

6. Wskaż na schemacie pojęciowym asocjację wiele-do-wielu z niesymetrycznymi licznosciami dla obu końców asocjacji oraz ze zdefiniowaną klasą lub atrybutem asocjacji. Na jakim etapie cyklu życia systemu klasę asocjacji powinno się zamienić na konstrukcję zawierającą klasę pośredniczącą? Narysuje odpowiedni przykład (z atrybutami i metodami) na podstawie wskazanego uprzednio fragmentu diagramu.
7. Dla schematu pojęciowego podaj przykład ograniczenia na istnienie obiektu. Czy jest to ograniczenie statyczne czy dynamiczne? Do jakiej metody i w jakiej klasie włożyłbyś implementację tego ograniczenia?
8. Dla wybranego przypadku użycia systemu (innego niż omówionego szczegółowo w analizie funkcjonalnej lub dynamicznej), sformułuj scenariusz i narysuje diagram aktywności (określ przynajmniej dwie aktywności, nie licząc początkowej i końcowej).
9. Dla wybranego przypadku użycia systemu (innego niż omówionego szczegółowo w analizie funkcjonalnej lub dynamicznej), sformułuj algorytm i narysuje odpowiedni diagram interakcji, który przedstawia interakcje pomiędzy obiektami co najmniej dwóch różnych klas.
10. Dla dowolnej klasy (innej niż omówionej szczegółowo w analizie dynamicznej) ze schematu pojęciowego wskaż dwa stany (wyłączając stan początkowy i stan końcowy), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów, zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

## 7.6 Podsumowanie zadania

W Tab. 7-5 przedstawiono ogólną charakterystykę omawianego zadania. Szczegółową charakterystykę zadania dla grup tematycznych: *Wprowadzenie do przedmiotu*, *Analiza funkcjonalna*, *Analiza strukturalna*, *Analiza dynamiczna* zamieszczono odpowiednio w tabelach Tab. 7-6, Tab. 7-7, Tab. 7-8 i Tab. 7-9.

Tab. 7-5 Ogólna charakterystyka zadania

<b>Ogólna charakterystyka zadania</b>	
<i>Nazwa</i>	Kwatery do wynajęcia
<i>Dziedzina problemowa</i>	system mógłby znaleźć zastosowanie w branży turystycznej i hotelowej
<i>Cel</i>	wprowadzenie systemu ułatwi obsługę wynajmu pokojów w kwaterach prywatnych

<i>Zakres odpowiedzialności</i>	klasyfikacja i ewidencja kwater, ewidencja właścicieli kwater, ewidencja klientów, wyszukiwanie kwater według określonych kryteriów, zarządzanie rezerwacjami kwater, tworzenie raportów dotyczących rezerwacji kwater		
<i>Stopień trudności</i>	średnio trudne		
<i>Rozmiar</i>	10 klas		
<i>Użyteczność w grupach tematycznych</i>			
<i>Grupa tematyczna</i>	<i>Użyteczność w grupie tematycznej</i>	<i>Użyteczność na etapie zgodnym z modelem efektywnego nauczania</i>	
<i>Wprowadzenie do przedmiotu</i>	niska	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> niska <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> niska	
<i>Analiza funkcjonalna</i>	średnia	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> wysoka <i>Podsumowanie dotychczasowych działań:</i> średnia <i>Ewaluacja:</i> wysoka	
<i>Analiza strukturalna</i>	wysoka	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> wysoka <i>Podsumowanie dotychczasowych działań:</i> wysoka <i>Ewaluacja:</i> wysoka	
<i>Analiza dynamiczna</i>	wysoka	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> wysoka <i>Podsumowanie dotychczasowych działań:</i> średnia <i>Ewaluacja:</i> średnia	

Tab. 7-6 Szczegółowa charakterystyka zadania dla grupy tematycznej *Wprowadzenie do przedmiotu*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Wprowadzenie do przedmiotu</i>
<b>Stopień trudności grupy w zadaniu</b>	łatwe

<b>Poruszane tematy</b>	<i>analiza a projektowanie, dekompozycja diagramów, mechanizmy rozszerzalności, reguły nazewnictwa, poprawa percepcji diagramu/diagramów</i>		
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>	
<i>analiza a projektowanie</i>	łatwe	<i>model pojęciowy, model projektowy</i>	
<i>mechanizmy rozszerzalności</i>	łatwe	<i>stereotyp, ograniczenie, komentarz</i>	
<i>dekompozycja diagramów</i>	łatwe	<i>podział modelu przypadków użycia na diagramy</i>	
<i>reguły nazewnictwa</i>	łatwe	<i>reguły nazewnictwa aktorów, klas oraz metod</i>	
<i>poprawa percepcji diagramu/diagramów</i>	łatwe	<i>modelowanie upływu czasu i danych historycznych</i>	

Tab. 7-7 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza funkcjonalna*

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>	<i>Analiza funkcjonalna</i>	
<b>Stopień trudności grupy w zadaniu</b>	średnio trudne	
<b>Poruszane tematy</b>	<i>aktorzy, przypadki użycia, scenariusze, dekompozycja modelu/diagramu przypadków użycia, mechanizmy rozszerzalności</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>aktorzy</i>	średnio trudne	<i>aktor Podsystem czasu, hierarchia dla aktorów</i>

<i>przypadki użycia</i>	średnio trudne	
<i>scenariusze</i>	trudne	stosunkowo duża złożoność scenariuszy
<i>dekompozycja modelu/diagramu przypadków użycia</i>	łatwe	<i>dekompozycja pionowa</i>
<i>mechanizmy rozszerzalności</i>	łatwe	ograniczenie nałożone na relację między przypadkami użycia

Tab. 7-8 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza strukturalna*

<b>Szczegółowa charakterystyka zadania</b>			
<b>Grupa tematyczna</b>	<i>Analiza strukturalna</i>		
<b>Stopień trudności grupy w zadaniu</b>	średnio trudne		
<b>Poruszane tematy</b>	<i>trybuty klasy, metody klasy, struktury generalizacji-specjalizacji, klasa abstrakcyjna a klasa konkretna, asocjacje, analiza wartości. diagramy struktur złożonych, przejście na schemat projektowy, przejście na schemat relacyjny, mechanizmy rozszerzalności</i>		
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>	
<i>trybuty klasy</i>	łatwe	<i>trybut pochodny, obiektowy, klasowy</i>	
<i>metody klasy</i>	średnio trudne	<i>metoda abstrakcyjna, zaimplementowana, obiektowa, klasowa, polimorfizm metod, operacja a metoda i komunikat, stosunkowo duża liczba metod klasowych, wywołanie metody z podaną niepełną listą argumentów</i>	
<i>struktury generalizacji-specjalizacji</i>	średnio trudne	dziedziczenie nierożłączne, wieloaspektowe i elipsa	
<i>klasa abstrakcyjna a klasa konkretna</i>	łatwe	<i>klasa abstrakcyjna, klasa konkretna</i>	
<i>asocjacje</i>	średnio	<i>klasa asocjacji, agregacja, kompozycja,</i>	

	trudne	<i>asocjacja kwalifikowana</i>
<i>analiza wartości</i>	łatwe	<i>wartość pochodna, wartość początkowa (domyślna), wartość graniczna</i>
<i>diagramy struktur złożonych</i>	średnio trudne	<i>diagram struktur złożonych</i>
<i>przejście na schemat projektowy</i>	trudne	stosunkowo duża złożoność transformacji schematu pojęciowego do schematu projektowego, dwa sposoby obejścia dziedziczenia wieloaspektowego i nierożłącznego: za pomocą kompozycji i klas zagnieżdzonych, zamiana asocjacji kwalifikowanej
<i>przejście na schemat relacyjny</i>	średnio trudne	obejście dziedziczenia nierożłącznego
<i>mechanizmy rozszerzalności</i>	średnio trudne	<i>ograniczenie, komentarz, stosunkowo duża liczba ograniczeń na schemacie pojęciowym i projektowym</i>

Tab. 7-9 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza dynamiczna*

<b>Szczegółowa charakterystyka zadania</b>			
<b>Grupa tematyczna</b>	<i>Analiza dynamiczna</i>		
<b>Stopień trudności grupy w zadaniu</b>	trudne		
<b>Poruszane tematy</b>	<i>diagramy aktywności, diagramy interakcji, diagramy stanów, podsumowanie analizy dynamicznej</i>		
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>	
<i>diagramy aktywności</i>	trudne	stosunkowo duża złożoność diagramów aktywności, romb decyzyjny <i>branch, merge, fork i join</i> , wyróżniony przebieg główny	
<i>diagramy interakcji</i>	łatwe	obiekt wielokrotny	

<i>diagramy stanów</i>	średnio trudne	akcja <i>entry</i> , akcja <i>exit</i> , zdarzenie <i>after</i>
<i>podsumowanie analizy dynamicznej</i>	łatwe	modyfikacje schematu pojęciowego i schematu projektowego o dodatkowe metody

## Spis treści rozdziału

Rozdział 8 Osiedlowy klub kartingowy.....	322
8.1 Tekst wymagań.....	322
8.2 Analiza funkcjonalna .....	324
8.2.1 Diagram kontekstowy .....	324
8.2.2 Model przypadków użycia.....	325
8.2.3 Scenariusze dla wybranych przypadków użycia łącznie z podziałem na podprzypadki .....	327
8.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	332
8.3 Analiza strukturalna.....	332
8.3.1 Schemat pojęciowy .....	332
8.3.2 Analiza elementów pochodnych .....	334
8.3.3 Analiza wartości początkowych .....	335
8.3.4 Analiza wartości granicznych.....	338
8.3.5 Zadania podstawowe .....	343
8.3.6 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	345
8.3.7 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji .....	349
8.3.8 Schemat relacyjny dla fragmentu schematu pojęciowego .....	353
8.4 Analiza dynamiczna .....	355
8.4.1 Zadania podstawowe .....	355
8.4.2 Omówienie zmian na schemacie pojęciowym i schemacie projektowym wynikających z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań .....	360
8.5 Zadania do samodzielnej pracy .....	363
8.6 Podsumowanie zadania.....	363

## Rozdział 8

# Osiedlowy klub kartingowy

Ewa Stemposz, Andrzej Jodłowski

### 8.1 Tekst wymagań

Z powodu gwałtownego rozwoju osiedlowego klubu kartingowego kierownik doszedł do wniosku, że aby móc sprawować efektywny nadzór i na bieżąco orientować się w sprawach klubowych, będzie potrzebował odpowiedniego systemu.

1. System ma przechowywać informacje o zatrudnionych trenerach oraz o zawodnikach itd. Trener, oprócz podstawowych danych (takich jak: imię, nazwisko, adres, rok urodzenia, data zatrudnienia, wiek), musi mieć zapamiętywany także staż pracy w klubie oraz staż pracy w zawodzie przed przyjęciem do klubu – o ile nie jest to jego pierwsza praca.
2. Zawodnicy dzielą się na poczatkujących i zaawansowanych. Dla zawodników obu grup ma być pamiętane: imię, nazwisko, adres, rok urodzenia, data wstąpienia do klubu, data wystąpienia z klubu oraz wiek. Zawodnikiem zaawansowanym jest zawodnik należący do klubu co najmniej od roku. Dodatkowo, dla zawodnika zaawansowanego ma być wiadomo, w zadnym okresie, ile razy brał udział w zawodach oraz ile razy był członkiem zwycięskiego zespołu w zawodach. Zawodnik zaawansowany może być jednocześnie trenerem. System powinien co pół roku kontrolować, czy można przekwalifikować zawodnika poczatkującego na zawodnika zaawansowanego.
3. Dla wszystkich trenerów i zawodników należy przechowywać informację o liczbie godzin spędzonych na treningach.
4. W danym momencie każdy zawodnik ma przydzielonego jednego trenera, który się nim opiekuje, a trener może mieć co najwyżej ośmiu podopiecznych. Pamiętany jest tylko aktualny stan przydziału zawodników do trenerów.
5. Zawodnicy jeżdżą na gokartach, dla których znana jest maksymalna prędkość i pojemność silnika (nie większa niż 0,25 litra), a także maksymalny przebieg, po którym gokart powinien zostać wycofany z użytku – taki sam dla wszystkich gokartów (aktualnie wynosi on 100 000 km). Jeden gokart może być przydzielony do maksymalnie dwóch zawodników, a każdy z zawodników jeździ w danym momencie na jednym gokarcie (ma być pamiętany tylko aktualny przydział). Ma być wiadomo, czy gokart w danym momencie jest wolny, używany lub serwisowany.

6. Klub posiada kilka torów, na których odbywają się treningi. Dla każdego toru mają być przechowywane informacje, takie jak: długość, stopień trudności („łatwy”, „średni”, „trudny”), lista punktów kontrolnych (co najmniej 3) oraz najlepszy czas, w jakim tor został kiedykolwiek pokonany. W treningu bierze udział jeden trener oraz maksymalnie czterech zawodników (tylko ci, którzy aktualnie są przydzieleni do tego trenera). Dla pojedynczego treningu mają być pamiętane następujące informacje: tor, data, godzina rozpoczęcia, czas trwania (aktualnie nie dłużej niż 4 godziny, ale wartość ta może w przyszłości ulec zmianie), a także najlepszy czas uzyskany przez zawodników uczestniczących w danym treningu. Również dla każdego zawodnika uczestniczącego w treningu należy zapamiętać najlepszy czas, jaki osiągnął w jego trakcie.
7. Na torach trudnych mogą trenować tylko zawodnicy zaawansowani.
8. Zawodnicy zaawansowani, grupowani w zespoły, mogą brać udział w zawodach kartingowych; jeden zespół może brać udział w wielu zawodach. Każdy zespół – o unikatowej nazwie – składa się z dwóch takich zawodników i opiekującego się nimi trenera. W danym momencie zespołem opiekuje się jeden trener, a zawodnik może być członkiem tylko jednego zespołu. Należy pamiętać datę utworzenia oraz datę rozwiązania zespołu. Zawodnicy, jak również i trenerzy mogą migrować między zespołami (z możliwością powrotów do tych samych zespołów), należałoby pamiętać okres ich przynależności do każdego z nich.
9. W systemie mają być rejestrowane zawody, ale tylko i wyłącznie takie, w których startowały zespoły klubu. Klub, jeśli w ogóle bierze udział w zawodach, to wystawia co najmniej 5 swoich zespołów. Dla zawodów mają być pamiętane informacje takie jak: ich unikatowa nazwa, termin przeprowadzenia, informacja o tym, czy wliczane są do klasyfikacji generalnej, a także: które zespoły z klubu w nich uczestniczyły, jaki numer startowy miał każdy z nich oraz które miejsce zajął. Dla każdego zespołu system powinien dostarczyć możliwość sprawdzenia, w ilu zawodach startował w danym okresie oraz w ilu z nich zwyciężył.
10. Po zwolnieniu trenera lub wystąpieniu zawodnika z klubu informacje o nich mają być nadal przechowywane.
11. System ma wspomagać użytkowników między innymi w:
  - 11.1 Zwolnieniu trenera z klubu, jeżeli zespół, którym się opiekował, nie wygrał co najmniej jednych z dziesięciu kolejno odbywających się zawodów lub jeśli trener ukończył 67 lat;
  - 11.2 Grupowaniu zawodników w zespoły na zawody (czyli tworzeniu zespołów startujących na zawodach) z ewentualnym równoczesnym przydzieleniem trenera do opieki nad zespołem i z ewentualną zmianą przydziału gokartów do zawodników;
  - 11.3 Ustalaniu statystyk zawodnika zaawansowanego za dany okres, tzn. system powinien dostarczać informacji o tym, w ilu zawodach dany

zawodnik brał udział i ile z nich wygrał (tj. był członkiem zwycięskiego zespołu);

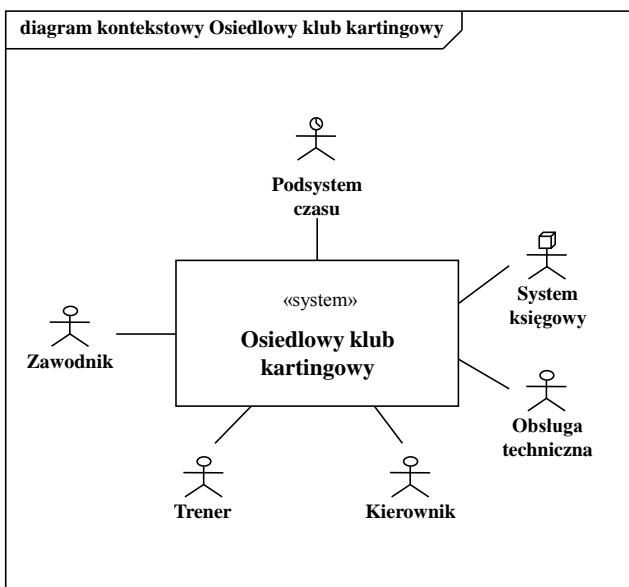
- 11.4 Wyliczaniu wynagrodzenia miesięcznego dla zawodnika i trenera. Wynagrodzenie trenera wyliczane jest w oparciu o liczbę godzin, jaką spędził na treningach, a zawodnika – w oparciu o liczbę godzin, jaką spędził na treningach oraz o to, ile razy zespół, którego był członkiem, zwyciężył w czasie rozważanego miesiąca w zawodach;
- 11.5 Znajdowaniu najlepszego zawodnika w zadany okresie (tzn. tego, który był najwięcej razy członkiem zwycięskiego zespołu w tym okresie).

## 8.2 Analiza funkcjonalna

### 8.2.1 Diagram kontekstowy

*Polecenie:* Zbuduj diagram kontekstowy w oparciu o tekst wymagań z punktu 8.1.

Diagram kontekstowy przedstawiający aktorów z otoczenia systemu *Osiedlowy klub kartingowy* wybranych na podstawie analizy tekstu wymagań, został zaprezentowany na Rys. 8-1.



Rys. 8-1 Diagram kontekstowy

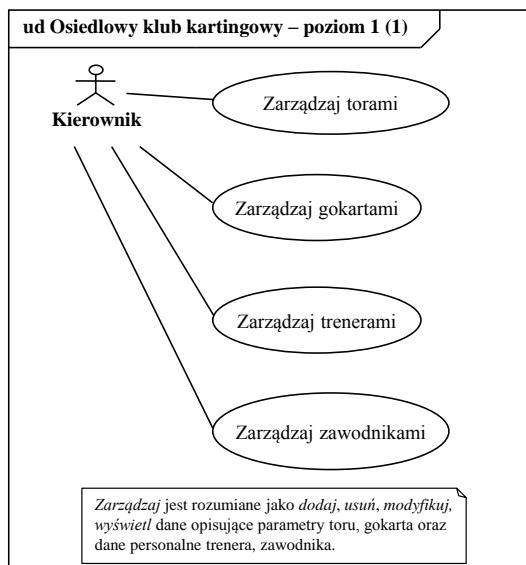
## 8.2.2 Model przypadków użycia

**Polecenie:** Zbuduj model przypadków użycia, uwzględniając przede wszystkim funkcjonalność sugerowaną w ostatnim punkcie tekstu wymagań (tzn. w punkcie 11).

Uwaga: Model należy skonstruować z perspektywy aktorów z otoczenia systemu oraz z uwzględnieniem hierarchii dla nich i relacji pomiędzy przypadkami (o ile mają/mogły mieć miejsce).

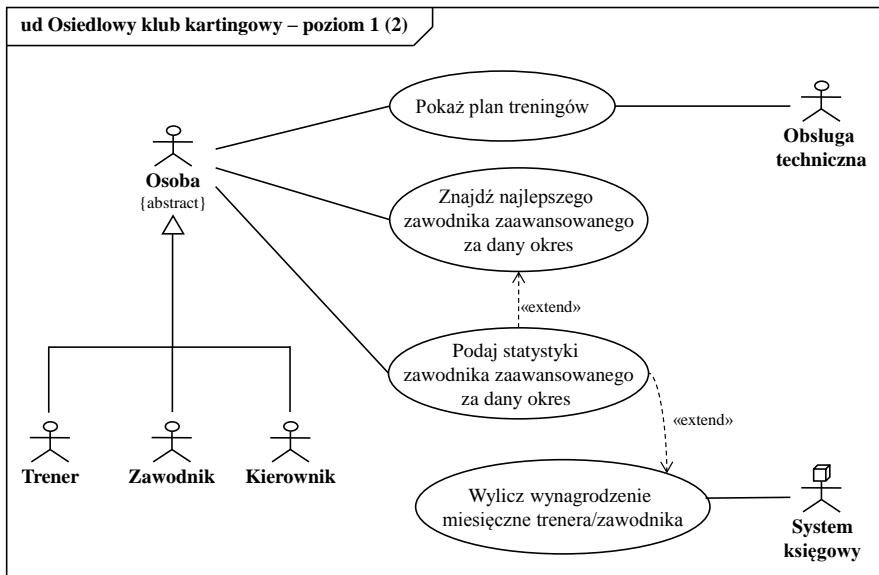
Dla przejrzystości, model przypadków użycia [36, 41, 42, 46] został opisany za pomocą czterech diagramów umieszczonych na Rys. 8-2, Rys. 8-3, Rys. 8-4 oraz Rys. 8-5. Diagramy zostały zbudowane z perspektywy aktorów z otoczenia systemu. W nagłówkach oznaczono numer poziomu abstrakcji modelu, a w nawiasach podano numery kolejnych diagramów. Rys. 8-2 zawiera przypadki użycia typu CRUD (Create, Read, Update, Delete), które na tym etapie analizy mogłyby zostać pominięte.

Diagramy przedstawione na Rys. 8-3, Rys. 8-4 i Rys. 8-5 ilustrują funkcjonalność systemu proponowaną na podstawie tekstu wymagań.

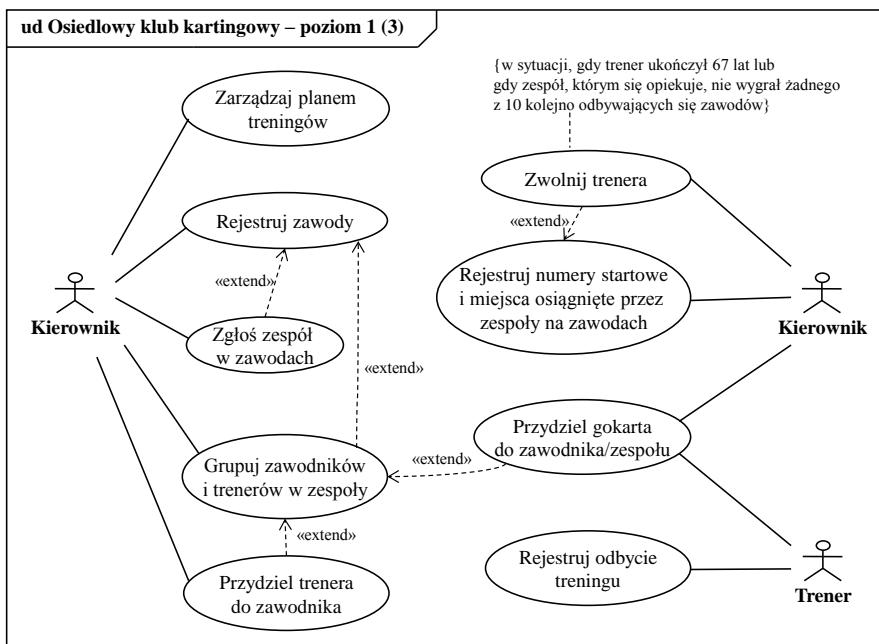


Rys. 8-2 Model przypadków użycia (diagram pierwszy)

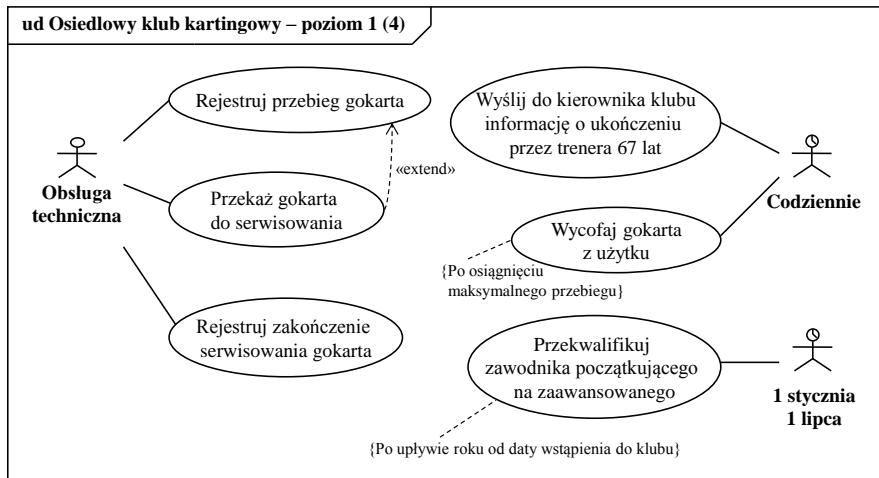
Należy pamiętać o tym, że w kolejnych krokach ten wstępny zbiór usług oferowanych przez system powinien podlegać dalszej weryfikacji, modyfikacji, rozszerzeniu itd.



Rys. 8-3 Model przypadków użycia (diagram drugi)



Rys. 8-4 Model przypadków użycia (diagram trzeci)



Rys. 8-5 Model przypadków użycia (diagram czwarty)

### 8.2.3 Scenariusze dla wybranych przypadków użycia łącznie z podziałem na podprzypadki

*Polecenie:* Dla funkcjonalności *Podaj statystyki zawodnika zaawansowanego za dany okres* (punkt 11.3 tekstu wymagań) napisz scenariusz.

Przykładowy scenariusz dla przypadku *Podaj statystyki zawodnika zaawansowanego za dany okres* został umieszczony w Tab. 8-1.

Tab. 8-1 Scenariusz dla przypadku *Podaj statystyki zawodnika zaawansowanego za dany okres*

<b>Nazwa przypadku użycia</b>	Podaj statystyki zawodnika zaawansowanego za dany okres
<b>Warunek początkowy</b>	W systemie jest zarejestrowany co najmniej jeden zawodnik zaawansowany i co najmniej jedne zawody.
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>Przypadek użycia <i>Podaj statystyki zawodnika zaawansowanego za dany okres</i> uruchamia aktor <i>Osoba</i> lub jeden z przypadków użycia: <i>Znajdź najlepszego zawodnika zaawansowanego za dany okres</i>, <i>Wylicz wynagrodzenie miesięczne trenera/zawodnika</i>.</li> <li>System prezentuje alfabetyczną listę zawodników</li> </ol>

	zaawansowanych. Aktor wybiera zawodnika. 3. System odpytuje o okres, dla którego mają być sporządzone statystyki. Aktor wprowadza okres. 4. System prezentuje statystyki zawodnika.
<b>Alternatywne przepływy zdarzeń</b>	2a. Aktor rezygnuje z kontynuowania przypadku. 2b. Przypadek został wywołany z przypadku użycia <i>Znajdź najlepszego zawodnika zaawansowanego za dany okres albo Wylicz wynagrodzenie miesięczne trenera/zawodnika</i> , system przechodzi do punktu 3 w przepływie głównym. 3a. Niepoprawny format którejś z dat określających okres: system informuje o błędzie i odpytuje ponownie o okres. 3b. Niepoprawne daty (data początkowa > data końcowa): system informuje o błędzie i odpytuje ponownie o okres. 3c. Niezgodność podanego okresu z okresem członkostwa zawodnika w klubie: system informuje o błędzie i odpytuje ponownie o okres. 3ca. Aktor rezygnuje z kontynuowania przypadku.
<b>Zakończenie</b>	W dowolnym momencie.
<b>Warunek końcowy</b>	Brak

**Polecenie:** Dla przypadku użycia *Grupuj zawodników i trenerów w zespoły* (punkt 11.2 tekstu wymagań):

- napisz scenariusz;
- zaproponuj podział tego przypadku na podprzypadki.

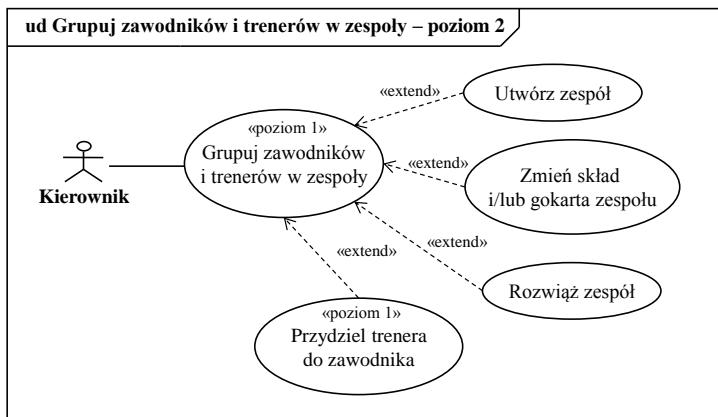
Przykładowy scenariusz dla przypadku użycia *Grupuj zawodników i trenerów w zespoły* został umieszczony w Tab. 8-2.

Z kolei, przykładowy scenariusz dla przypadku użycia *Utwórz zespół* został umieszczony w Tab. 8-3. Przyjęte zostało założenie, że przydzielenie gokarta do zespołu polega na przydzieleniu jednego i tego samego gokarta do dwóch zawodników zaawansowanych, będących członkami tego zespołu oraz że przydzielenie trenera do zespołu oznacza, iż zawodnicy – członkowie zespołu podlegają tym samym opiece tego samego trenera również na treningach w klubie.

Tab. 8-2 Scenariusz dla przypadku użycia *Grupuj zawodników w zespoły*

<b>Nazwa przypadku użycia</b>	Grupuj zawodników i trenerów w zespoły
<b>Warunek początkowy</b>	W systemie jest zarejestrowanych co najmniej dwóch zawodników zaawansowanych, co najmniej jeden trener i co najmniej jeden gokart.
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Przypadek użycia uruchamia aktor <i>Kierownik</i>.</li> <li>2. System prezentuje formatkę z trzema opcjami do wyboru: <i>Utwórz zespół</i>, <i>Zmień skład i/lub gokarta zespołu</i> oraz <i>Rozwiąż zespół</i>. Aktor wybiera jedną z opcji.</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	
<b>Zakończenie</b>	W dowolnym momencie.
<b>Warunek końcowy</b>	Stan systemu, po zrealizowaniu przypadku, jest uzależniony od wyboru opcji.

Na Rys. 8-6 został przedstawiony przykładowy podział przypadku użycia *Grupuj zawodników i trenerów w zespoły* zgodny ze scenariuszem z Tab. 8-2.

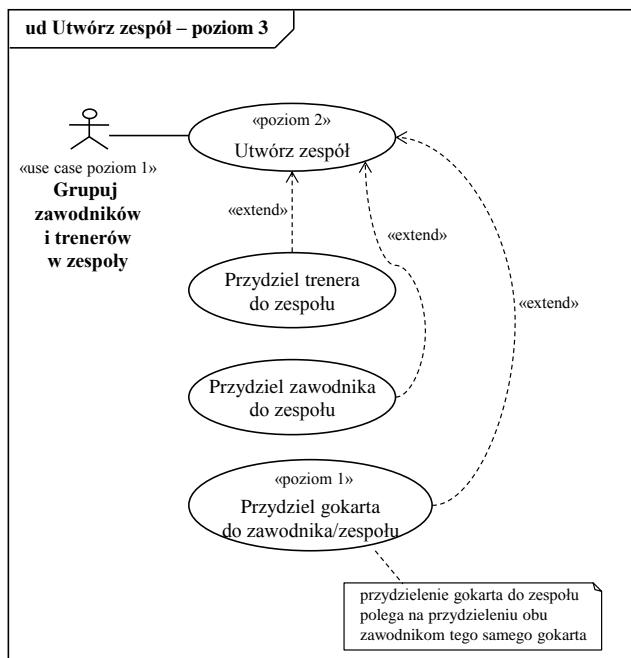
Rys. 8-6 Diagram z przykładowym podziałem przypadku użycia *Grupuj zawodników i trenerów w zespoły*

Tab. 8-3 Scenariusz dla przypadku użycia *Utwórz zespół*

<b>Nazwa przypadku użycia</b>	Utwórz zespół
<b>Warunek początkowy</b>	brak
<b>Główny przepływ zdarzeń</b>	<p>1. Przypadek użycia <i>Utwórz zespół</i> jest wywoływany z przypadku użycia <i>Grupuj zawodników i trenerów w zespoły</i>.</p> <p>2. System prezentuje formularz z: polem dla wprowadzenia nazwy zespołu, alfabetyczną listą zawodników zaawansowanych (łącznie z informacją o przypisaniu zawodników do aktualnie istniejących zespołów, o ile są już gdzieś przypisani), trenerów (z alfabetyczną listą zawodników aktualnie pozostających pod ich opieką) oraz gokartów (z listą zawodników doń przypisanych). <i>Kierownik</i> wprowadza nazwę zespołu, dokonuje wyboru trenera, dwóch zawodników i gokarta.</p> <p>3. System potwierdza utworzenie nowego zespołu i prezentuje opisujące go informacje takie jak: nazwa, skład, przydzielony gokart.</p>
<b>Alternatywne przepływy zdarzeń</b>	<p>2a. Jest już zarejestrowany zespół o nazwie podanej przez aktora lub wybrany gokart jest już przydzielony do innego zawodnika (innego niż jeden z dwóch wybranych w formularzu), lub co najmniej jeden z wybranych zawodników ma już przydzielony inny gokart, lub co najmniej jeden z wybranych w formularzu zawodników ma już przydzielonego innego trenera, system informuje o tym i pyta, czy ma kontynuować.</p> <p>2aa. Aktor potwierdził wolę kontynuowania, system wraca do punktu 2.</p> <p>2b. Aktor nie wybrał żadnego zawodnika lub wybrał tylko jednego zawodnika, lub nie wybrał trenera, lub nie wybrał gokarta, system prosi o uzupełnienie danych w formularzu wyboru zawodników, trenera oraz gokarta i wraca do punktu 2.</p>

<b>Zakończenie</b>	Przypadek może zostać zakończony w dowolnym momencie.
<b>Warunek końcowy</b>	Jeżeli proces tworzenia zespołu zostanie poprawnie zakończony, to: zostanie utworzony nowy zespół o podanej nazwie, z ustalonym składem (trener i dwóch zawodników zaawansowanych), przypisany do zawodników gokartem oraz ewentualnie zostanie podana informacja o zmianie przypisania gokarta.

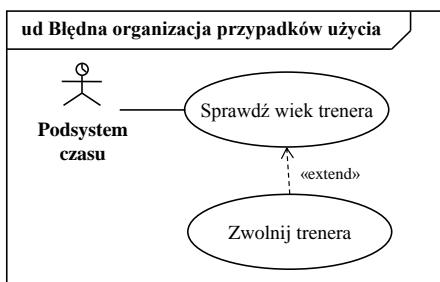
Jak widać, zrezygnowano tutaj ze specyfikowania warunku początkowego, ponieważ przypadek *Utwórz zespół* jest wywoływany z przypadku *Grupuj zawodników i trenerów w zespoły*, dla którego zostały określone odpowiednie warunki początkowe – zgodne z ograniczeniami dla przypadku *Utwórz zespół* – i będą one podlegały weryfikacji przed wywołaniem *Utwórz zespół*. Na Rys. 8-7 został przedstawiony przykładowy podział przypadku użycia *Grupuj zawodników i trenerów w zespoły* zgodny ze scenariuszem z Tab. 8-2.



Rys. 8-7 Diagram z przykładowym podziałem przypadku *Utwórz zespół* na podprzypadki

### 8.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

- Nazwa przypadku nie jest nazwą zadania zlecanego systemowi, ale określa dane, np. nazwa taka jak *Gokarty* – trudno się tu zorientować, jaki ma być *obserwowalny rezultat* tego przypadku?
- Niewłaściwa wzajemna organizacja przypadków, jak np. na Rys. 8-8, gdzie zadaniem docelowym jest zwolnienie trenera, natomiast sprawdzanie wieku trenera jest czynnością pomocniczą. Zamiana kolejności wywołania przypadków (łącznie ze zmianą kierunku relacji z «extend» na «include») również nie byłaby rozwiązaniem do końca poprawnym, ponieważ do diagramu zostałby wprowadzony *przypadek wewnętrzny*, czyli taki, który nie jest bezpośrednio wywoływany przez aktora z danego poziomu abstrakcji (tu przez aktora z otoczenia systemu). Innymi słowy, w takiej sytuacji diagram powinien zawierać wyłącznie aktora *Podsystem czasu* oraz jeden przypadek użycia *Wyślij do kierownika klubu informację o ukończeniu przez trenera 67 lat*. Automatyczne zwalnianie trenera i usuwanie jego danych nie wydaje się praktyczne i pożądane przez przyszłego użytkownika systemu. Oczywiście, aktora można by było określić dokładniej, np. *Codziennie*.



Rys. 8-8 Przykład niewłaściwej wzajemnej organizacji przypadków

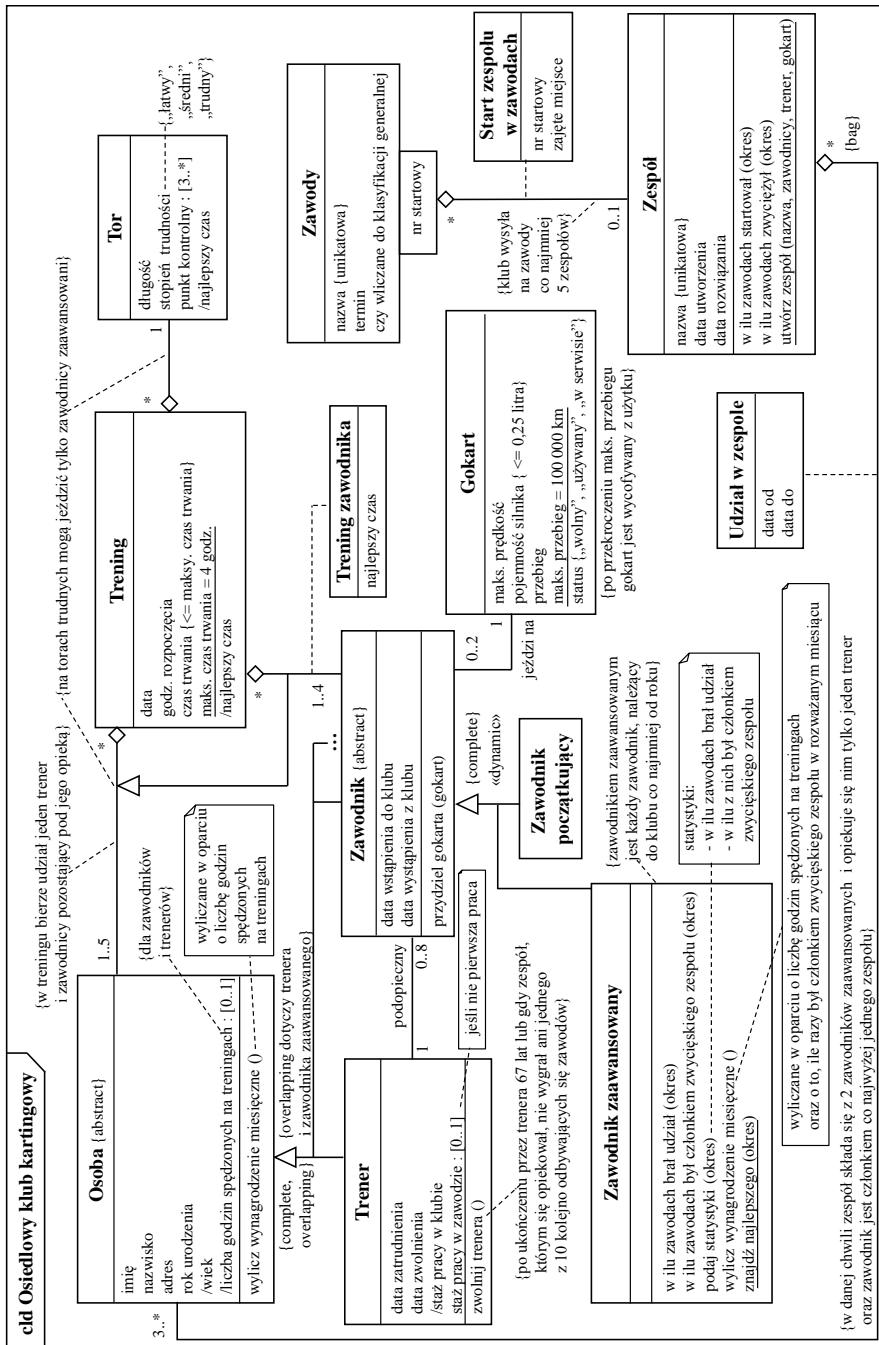
## 8.3 Analiza strukturalna

### 8.3.1 Schemat pojęciowy

*Polecenie:* Dla podanych założeń skonstruuj schemat pojęciowy (nie redukując liczności).

Schemat pojęciowy [35, 41, 43, 45] dla projektowanego systemu został przedstawiony na Rys. 8-9.

Warto zwrócić uwagę na określoną na diagramie hierarchię agregacji. Agregacja z klasą asocjacji *Trening zawodnika* między klasą *Trening* a klasą *Zawodnik* jest specjalizacją agregacji między klasą *Trening* a klasą *Osoba*.

Rys. 8-9 Schemat pojęciowy dla systemu *Osiedlowy klub kartingowy*

### 8.3.2 Analiza elementów pochodnych

Na schemacie pojęciowym (Rys. 8-9) można zidentyfikować następujące atrybuty pochodne:

- *wiek* w klasie *Osoba*;
- *liczba godzin spędzonych na treningach* w klasie *Osoba*;
- *staż pracy w klubie* w klasie *Trener*;
- *najlepszy czas* w klasie *Trening*;
- *najlepszy czas* w klasie *Tor*.

Wartość atrybutu *wiek* w klasie *Osoba* jest obliczana na podstawie roku urodzenia osoby i aktualnej daty systemowej, co nie jest złożone obliczeniowo, ponadto nie wydaje się, aby wartość ta była często wykorzystywana w innych obliczeniach. Dlatego wystarczy wyłącznie wprowadzenie metody *podaj wiek*, a atrybut *wiek* może zostać opuszczony. W analogiczny sposób może zostać opuszczony atrybut *staż pracy w klubie* w klasie *Trener*, którego wartość jest wyliczana na podstawie daty zatrudnienia trenera oraz aktualnej daty systemowej; wystarczy więc jedynie wprowadzenie metody *podaj staż pracy w klubie ()* w klasie *Trener*.

Atrybut opcjonalny *liczba godzin spędzonych na treningach* przechowuje liczbę godzin, jaką spędził w ostatnim miesiącu dany trener lub zawodnik (nie dotyczy każdej osoby i dlatego został oznaczony jako opcjonalny) na wszystkich treningach, które się odbyły z ich udziałem. Jest obliczany jako suma długości (tj. wartości atrybutu *czas trwania* w klasie *Trening*) odpowiednich treningów. Nietrudno byłoby wprowadzić – bez konieczności przechowywania wartości atrybutu – metodę, która wyliczałaby potrzebną informację, ale ponieważ ta informacja będzie zmieniała się w ścisłe określonej sytuacji (podczas rejestracji treningu), to wydaje się, że w przypadku pozostawienia atrybutu *liczba godzin spędzonych na treningach* zostałoby osiągnięte wystarczające rozwiążanie, a ponadto łatwiejsze do zaimplementowania. Zakładając, że zostałoby wprowadzona metoda *rejestruj trening (data, godz. rozpoczęcia, czas trwania, osoby)* w klasie *Trening*, która dodawałaby czas trwania treningu do aktualnej wartości atrybutu *liczba godzin spędzonych na treningach*, atrybut ten stanie się wtedy atrybutem bazowym.

Wartość atrybutu *najlepszy czas* w klasie *Trening* jest wyliczana jako minimalny czas z najlepszych czasów zawodników (w atrybucie *najlepszy czas* w klasie *Trening zawodnika*), biorących udział w danym treningu. Ponieważ wartość tego atrybutu jest ustalana jednorazowo (po zakończeniu treningu) i już nie jest zmieniana, dlatego atrybut pochodny *najlepszy czas* w klasie *Trening* nie zostanie usunięty z diagramu.

Wartość atrybutu *najlepszy czas* w klasie *Tor* jest obliczana jako minimalna wartość z najlepszych czasów osiągniętych na tym torze – na treningach. Złożoność obliczeniowa potencjalnej metody wyliczającej nie byłaby duża, ale ponieważ ewentualna zmiana atrybutu byłaby przeprowadzana po wprowadzeniu informacji o danym treningu na określonym torze (i tylko w przypadku, jeśli uzyskany *najlepszy czas* dla treningu byłby krótszy niż dotychczasowy rekord na tym torze),

to znacznie prostszym rozwiązaniem jest pozostawienie atrybutu *najlepszy czas* w klasie *Tor* jako atrybutu bazowego wyliczanego po zakończeniu treningu na danym torze w oparciu o najlepszy czas na treningu i warunkowanego poprzednią wartością tego atrybutu.

Diagram pojęciowy z uwzględnioną analizą elementów pochodnych został przedstawiony na Rys. 8-10.

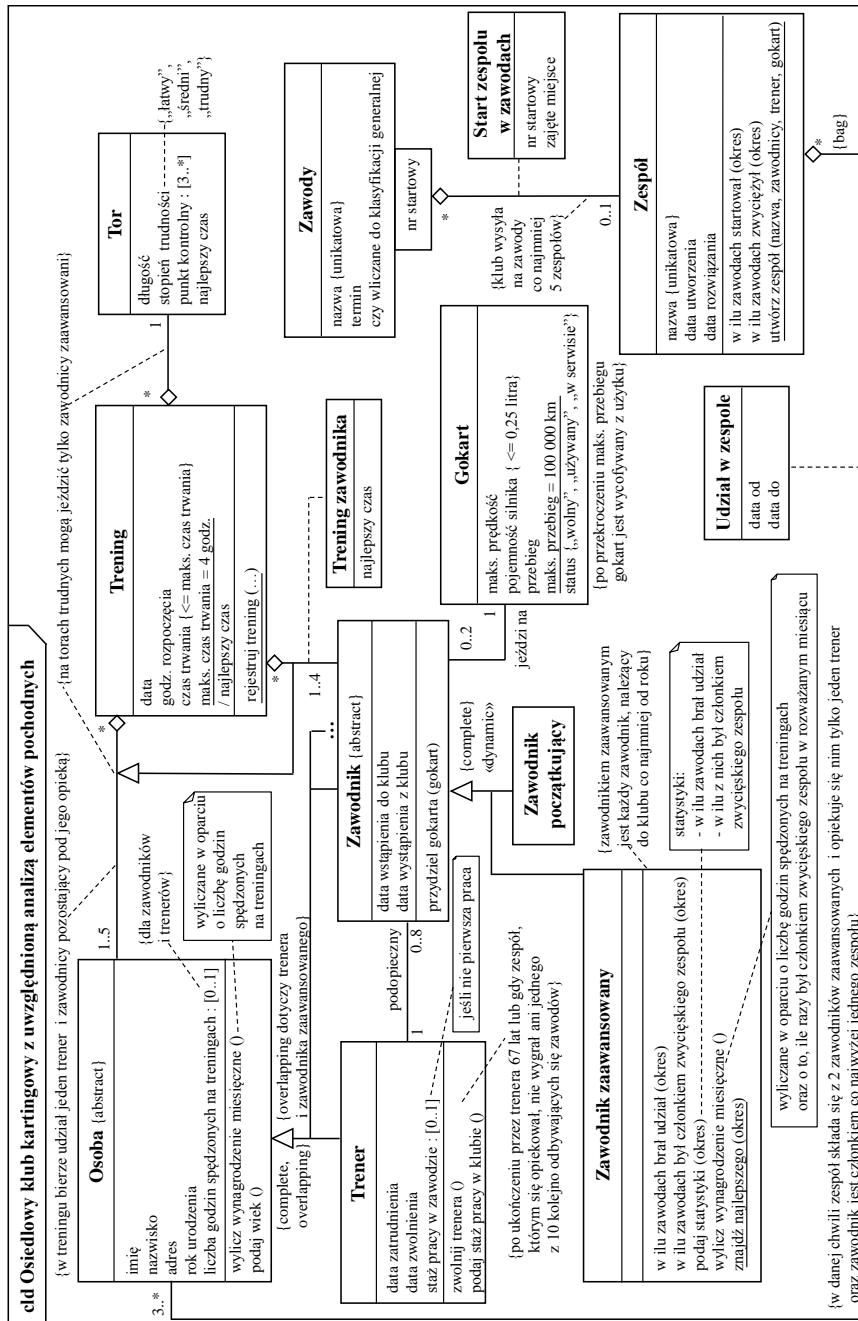
### 8.3.3 Analiza wartości początkowych

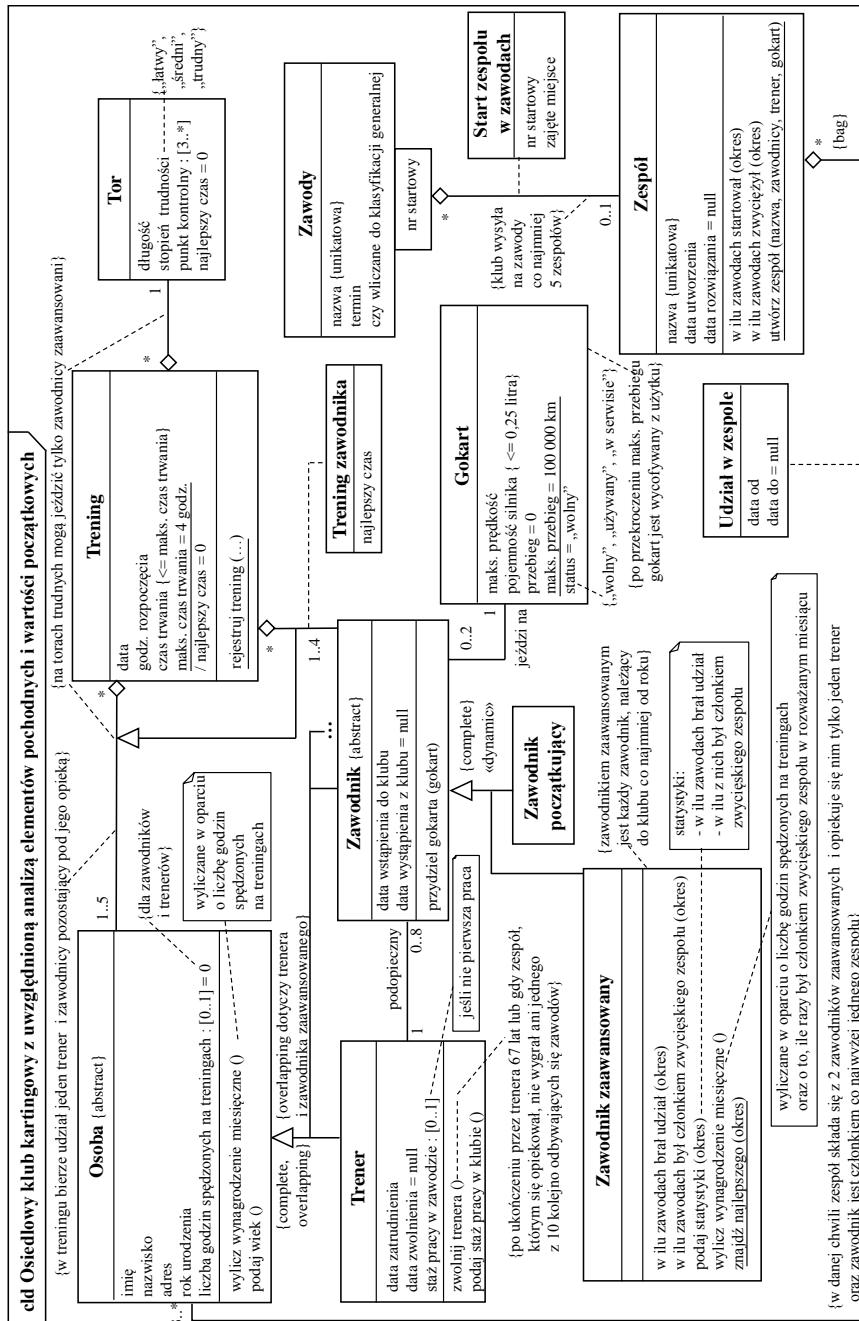
W dalszej analizie schematu pojęciowego z Rys. 8-10 zostaną zidentyfikowane domyślne wartości początkowe dla atrybutów oraz domyślne parametry dla metod. Na Rys. 8-10 nie ma metod z określonymi parametrami domyślnymi. Lista atrybutów z określonymi domyślnymi wartościami początkowymi została przedstawiona w Tab. 8-4.

Tab. 8-4 Lista atrybutów z określonymi wartościami początkowymi

Klasa	Atrybut	Wartość początkowa (domyślna)
<i>Gokart</i>	<i>przebieg</i>	<i>0</i>
	<i>status</i>	<i>„wolny”</i>
<i>Osoba</i>	<i>liczba godzin spędzonych na treningach</i>	<i>0</i>
<i>Tor</i>	<i>najlepszy czas</i>	<i>0</i>
<i>Trener</i>	<i>data zwolnienia</i>	<i>null</i>
<i>Trening</i>	<i>/ najlepszy czas</i>	<i>0</i>
<i>Zawodnik</i>	<i>data wystąpienia z klubu</i>	<i>null</i>
<i>Udział w zespole</i>	<i>data do</i>	<i>null</i>
<i>Zespół</i>	<i>data rozwiązania</i>	<i>null</i>

Schemat pojęciowy uwzględniający zaproponowane w Tab. 8-4 wartości początkowe został zaprezentowany na Rys. 8-11.

Rys. 8-10 Schemat pojęciowy dla systemu *Osiedlowy klub kartingowy z uwzględnioną analizą elementów pochodnych*



Rys. 8-11 Schemat pojęciowy dla systemu *Osiedlowy klub kartingowy* z uwzględnioną analizą elementów pochodnych i wartości początkowych

### 8.3.4 Analiza wartości granicznych

Podczas dalszej analizy schematu pojęciowego z Rys. 8-11 należy zidentyfikować wartości graniczne i zastanowić się, czy podczas ewentualnej eksploatacji systemu (wraz z upływem czasu) możliwe byłyby zmiany wartości granicznych bez wprowadzania zmian w kodzie.

Syntetyczne zestawienie przykładowych wartości granicznych, powiązanych z nimi ograniczeń oraz proponowanych zmian w schemacie pojęciowym przedstawiono w Tab. 8-5.

Tab. 8-5 Zestawienie wartości granicznych, ograniczeń i proponowanych zmian w schemacie pojęciowym

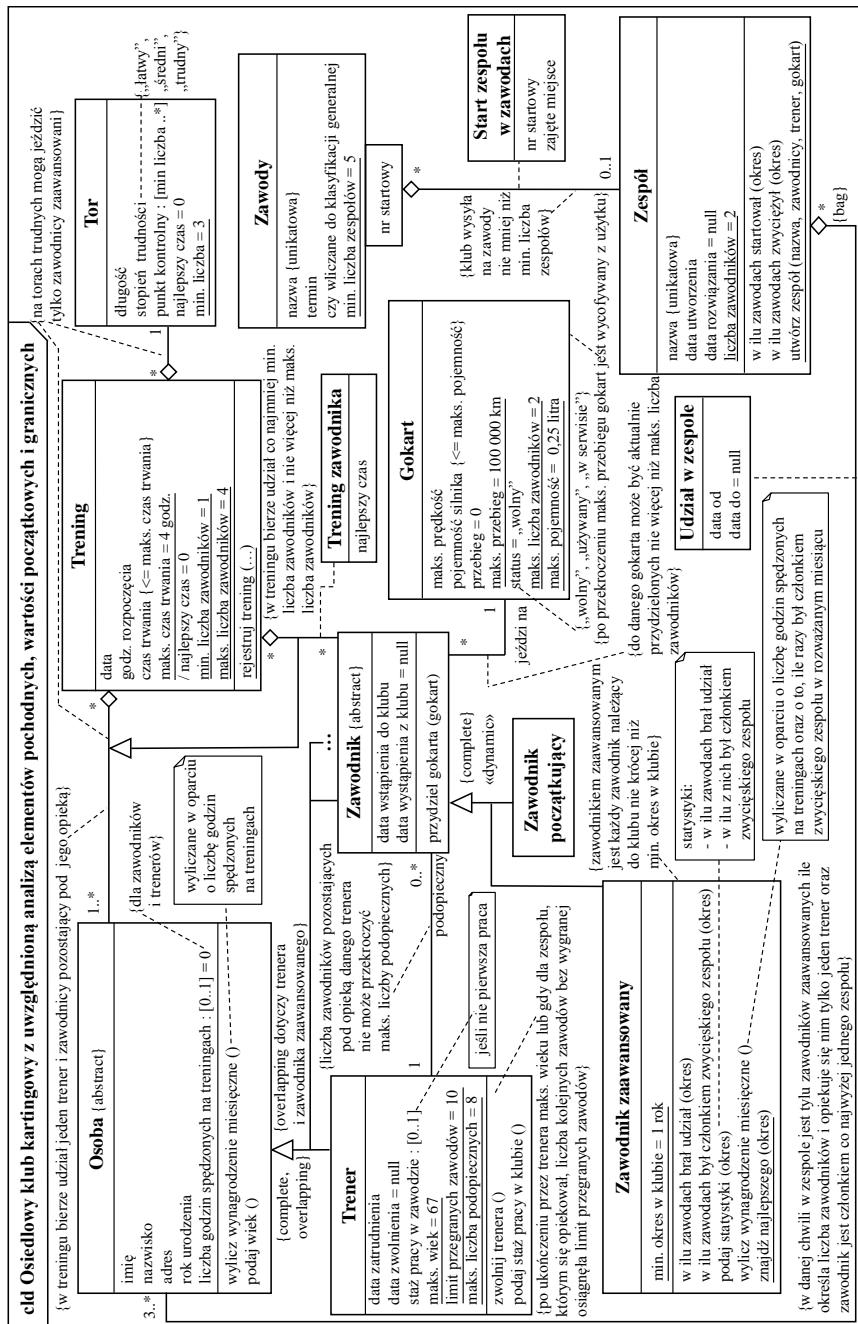
Wartość graniczna	Ograniczenie	Zmiany schematu pojęciowego
67 lat	<p><i>{po ukończeniu przez trenera 67 lat lub ...}</i></p> <p>nałożone na atrybut <i>data zwolnienia</i> klasy <i>Trener</i></p> <p>ograniczenie zaimplementowane jako warunek wstępny w metodzie <i>zwolnij trenera ()</i> w klasie <i>Trener</i></p> <p><i>Uwaga:</i> trener nie jest zwalniany automatycznie po osiągnięciu 67 lat, lecz na polecenie <i>Kierownika</i>.</p>	<p>dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością umieszczony w klasie <i>Trener</i>:</p> <p><u><i>maks. wiek = 67</i></u></p> <p>zmiana ograniczenia do postaci  <i>{po ukończeniu przez trenera maks. wieku lub ...}</i></p>
10 kolejnych zawodów	<p><i>{... gdy zespół, którym się opiekował, nie wygrał ani jednego z 10 kolejno odbywających się zawodów}</i></p> <p>nałożone na atrybut <i>data zwolnienia</i> klasy <i>Trener</i></p>	<p>dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością umieszczony w klasie <i>Trener</i>:</p> <p><u><i>limit przegranych zawodów = 10</i></u></p> <p>zmiana ograniczenia do postaci  <i>{... gdy dla zespołu, którym się opiekował, liczba kolejnych zawodów</i></p>

	ograniczenie zaimplementowane jako warunek wstępny dla metody <i>zwolnij trenera()</i> w klasie <i>Trener</i>	<i>bez wygranej osiągnęła limit przegranych zawodów}</i>
4 zawodników	nałożone na liczbosć 1..4 dla roli <i>Zawodnik</i> agregacji między klasą <i>Trening</i> a klasą <i>Zawodnik</i>	dodatkowe atrybuty o zasięgu klasowym z zainicjalizowaną wartością umieszczone w klasie <i>Trening</i> : $\underline{\text{min. liczba zawodników} = 1}$ $\underline{\text{maks. liczba zawodników} = 4}$  dodatkowe ograniczenie nałożone na agregację między klasą <i>Trening</i> a klasą <i>Zawodnik</i> : <i>{w treningu bierze udział co najmniej min. liczba i nie więcej niż maks. liczba zawodników}</i>  zmiana liczności do * dla roli <i>Zawodnik</i> agregacji między klasą <i>Trening</i> a klasą <i>Zawodnik</i>
4 godziny	<i>{czas trwania &lt;= 4 godz.}</i>  nałożone na atrybut <i>czas trwania</i>	bez zmian, ponieważ w klasie <i>Trening</i> już istnieje atrybut klasowy z zainicjalizowaną wartością: $\underline{\text{maks. czas trwania} = 4 godz.}$ oraz ograniczenie <i>{&lt;= maks. czas trwania}</i> nałożone na atrybut <i>czas trwania</i> w klasie <i>Trening</i>
0-8 zawodników	nałożone na liczbosć 0..8 dla roli <i>podopieczny</i> , asocjacji między klasą <i>Trener</i> a klasą <i>Zawodnik</i>	dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością umieszczony w klasie <i>Trener</i> : $\underline{\text{maks. liczba podopiecznych} = 8}$  zmiana liczności do 0..* dla roli <i>podopieczny</i> asocjacji między klasą <i>Trener</i> a klasą <i>Zawodnik</i>  dodatkowe ograniczenie:

		<p><i>{liczba zawodników pozostających pod opieką danego trenera nie może przekroczyć maks. liczby podopiecznych}</i>  nałożone na rolę <i>podopieczny</i>, asocjacji między klasą <i>Trener</i> a klasą <i>Zawodnik</i></p>
2 zawodników	<p><i>{w danej chwili zespół składa się z 2 zawodników zaawansowanych (...)</i>  oraz zawodnik jest członkiem co najwyżej jednego zespołu}</p> <p>liczność 3..* nałożone na rolę <i>Osoba</i> agregacji między klasą <i>Zespół</i> a klasą <i>Osoba</i></p>	<p>dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością umieszczony w klasie <i>Zespół</i>:  <u><i>liczba zawodników = 2</i></u>  zmiana ograniczenia do postaci  <i>{w danej chwili w zespole jest tylu zawodników zaawansowanych ile określa liczba zawodników (...) i zawodnik jest członkiem co najwyżej jednego zespołu}</i></p> <p>zmiana liczności do 0..* dla roli <i>Osoba</i> agregacji między klasą <i>Zespół</i> a klasą <i>Osoba</i></p>
0-2 zawodników	liczność 0..2 dla roli <i>Zawodnik</i> asocjacji między klasą <i>Zawodnik</i> a klasą <i>Gokart</i>	<p>dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością umieszczony w klasie <i>Gokart</i>:  <u><i>maks. liczba zawodników = 2</i></u>  dodatkowe ograniczenie:  <i>{do danego gokarta może być aktualnie przydzielonych nie więcej niż maks. liczba zawodników}</i></p> <p>zmiana liczności do * dla roli <i>Zawodnik</i> asocjacji między klasą <i>Zawodnik</i> a klasą <i>Gokart</i></p>
0,25 litra	<p><i>{ &lt;= 0,25 litra}</i>  nałożone na atrybut <i>pojemność silnika</i> w klasie <i>Gokart</i></p>	<p>dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością umieszczony w klasie <i>Gokart</i>:  <u><i>maks. pojemność = 0,25 litra</i></u></p>

		zmiana ograniczenia nałożonego na atrybut <i>pojemność silnika</i> w klasie <i>Gokart</i> do postaci $\{ \leq \text{maks. pojemność} \}$
co najmniej 3 punkty kontrolne	liczność 3..* dla atrybutu powtarzalnego <i>punkt kontrolny</i> w klasie <i>Tor</i>	dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością umieszczony w klasie <i>Tor</i> : <u><i>min. liczba = 3</i></u>  zmiana liczności do <i>min liczba..*</i> dla atrybutu <i>punkt kontrolny</i> w klasie <i>Tor</i>
co najmniej 5 zespołów	<i>{klub wysyła na zawody co najmniej 5 zespołów}</i>  nałożone na rolę <i>Zespół</i> asocjacji między klasą <i>Zawody</i> a klasą <i>Zespół</i>	dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością umieszczony w klasie <i>Zawody</i> : <u><i>min. liczba zespołów= 5</i></u>  zmiana ograniczenia do postaci <i>{klub wysyła na zawody nie mniej niż min. liczba zespołów}</i>
100 000 km	<i>{po przekroczeniu 100 000 km gokart jest wycofywany z użytku}</i>  nałożone na obiekt klasy <i>Gokart</i>	bez zmian, ponieważ w klasie <i>Gokart</i> już istnieje atrybut klasowy z zainicjalizowaną wartością: <u><i>maks. przebieg = 100 000 km</i></u> oraz istnieje ograniczenie <i>{po przekroczeniu maks. przebiegu gokart jest wycofywany z użytku}</i>
1 rok	<i>{zawodnikiem zaawansowanym jest każdy zawodnik należący do klubu co najmniej od roku}</i>  nałożone na obiekt klasy <i>Zawodnik zaawansowany</i>	dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością w klasie <i>Zawodnik zaawansowany</i> : <u><i>min. okres w klubie = 1 rok</i></u>  zmiana ograniczenia do postaci <i>{zawodnikiem zaawansowanym jest każdy zawodnik należący do klubu nie krócej niż min. okres w klubie}</i>

Schemat pojęciowy uwzględniający proponowane zmiany w wyniku analizy wartości granicznych został zaprezentowany na Rys. 8-12.

Rys. 8-12 Schemat pojęciowy dla systemu *Osiedlowy klub kartingowy z uwzględnioną analizą elementów pochodnych, wartości początkowych i wartości granicznych*

### 8.3.5 Zadania podstawowe

Zadania w tym rozdziale odnoszą się do schematu pojęciowego z Rys. 8-12.

**Zad. 1:** Podkreśl te rodzaje dziedziczenia (rozłączne, nierozłączne, kompletne, niekompletne, jednokrotne, wielokrotne, jednoaspektowe, wieloaspektowe, dynamiczne, elipsa), które uważaś za potrzebne do wykorzystania na schemacie pojęciowym. Dla każdego podkreślonego rodzaju dziedziczenia podaj hierarchię/aspekt, w której ten rodzaj dziedziczenia wystąpił.

- Rozłączne (ang. *disjoint*): hierarchia dla osób i zawodników poczynających oraz dla uczestnictwa osób na treningach;
- nierozłączne (ang. *overlapping*): hierarchia dla osób (klasy *Trener* i *Zawodnik zaawansowany*);
- kompletne: hierarchia dla osób i zawodników;
- niekompletne: hierarchia dla uczestnictwa osób na treningach;
- jednokrotne: hierarchia dla osób i zawodników oraz dla uczestnictwa osób na treningach;
- jednoaspektowe: hierarchia dla osób i zawodników oraz dla uczestnictwa osób na treningach;
- dynamiczne: hierarchia dla zawodników;
- elipsa: hierarchia dla osób.

**Zad. 2:** Jakie rodzaje metod (abstrakcyjne, zaimplementowane) może zawierać klasa abstrakcyjna, a jakie klasa konkretna?

Klasa abstrakcyjna może zawierać oba rodzaje metod, a klasa konkretna tylko metody zaimplementowane.

- Metoda abstrakcyjna w klasie abstrakcyjnej: *metoda wylicz wynagrodzenie miesięczne ()* w klasie *Osoba* mógłaby być metodą abstrakcyjną;
- metoda zaimplementowana w klasie abstrakcyjnej: mógłaby nią być np. metoda *zmień adres (adres)* w klasie *Osoba*;
- metoda zaimplementowana w klasie konkretnej: *wylicz wynagrodzenie miesięczne ()* w klasie *Zawodnik zaawansowany*.

**Zad. 3:** Podaj przykłady metod i atrybutów, klasowych i obiektowych.

- Metoda obiektowa: *wylicz wynagrodzenie miesięczne ()* w klasie *Osoba*;
- metoda klasowa: znajdź najlepszego (okres) – operuje na ekstensji klasy *Zawodnik zaawansowany* czy metoda utwórz zespół (nazwa, zawodnicy, trener, gokart) – tworzy nowy obiekt klasy *Zespół*;
- atrybut obiektowy: *nazwa* w klasie *Zespół*;
- atrybut klasowy: *maks. czas trwania* w klasie *Trening*.

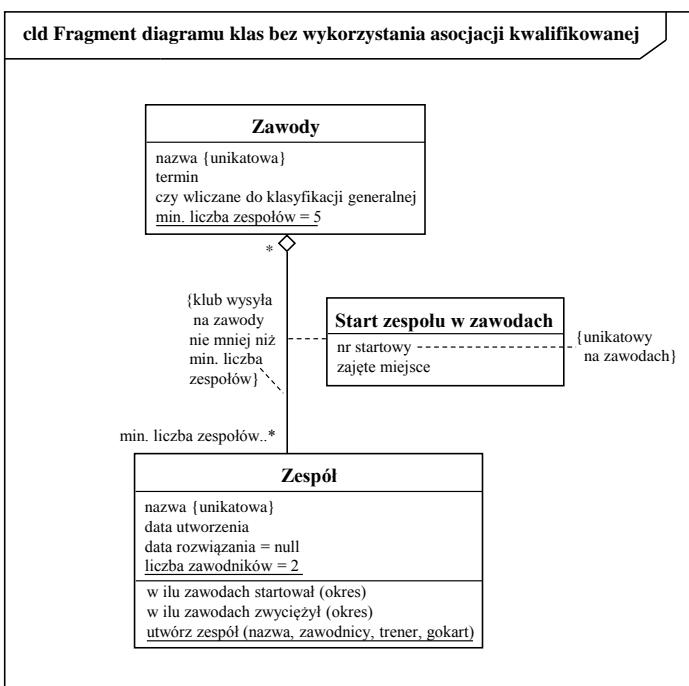
**Zad. 4:** Wskaż przykład zastosowania polimorfizmu metod.

Metody polimorficzne to np. metody: *wylicz wynagrodzenie miesięczne ()* w klasach hierarchii zbudowanej dla osób.

**Zad. 5:** Asocjację z diagramu klas, posiadającą niesymetryczne liczności oraz atrybut (lub klasę asocjacji) zamień na asocjację kwalifikowaną. Uzasadnij, dlaczego poddałeś zamianie właśnie tę asocjację.

Na diagramie umieszczono asocjację kwalifikowaną pomiędzy klasami *Zawody* i *Zespół*. Wskazuje ona, że kluczem do przeszukiwania zbioru zespołów przypisanych do danych zawodów może być numer startowy zespołu w zawodach. Rys. 8-13 przedstawia fragment diagramu przed zamianą asocjacji „zwykłej” na asocjację kwalifikowaną.

Uwaga: Asocjację kwalifikowaną wykorzystuje się do wskazania klucza użytkowego do przeszukiwania dużych zbiorów danych (a nie zbiorów jednoelementowych). W tym konkretnie przypadku, kwalifikator *nr startowy* może być wykorzystany jako klucz do przeszukiwania zbioru zespołów przypisanych do pojedynczych zawodów. Wprowadzenie asocjacji kwalifikowanej zmienia licznosć tego końca asocjacji, który znajduje się naprzeciwko kwalifikatora. Nowa licznosć bezpośrednio wskazuje „siłę” klucza, czyli np. licznosć = 1 mówi o tym, że wartość kwalifikatora pozwala na jednoznaczną identyfikację zespołu w ramach danych zawodów.



Rys. 8-13 Fragment diagramu bez wykorzystania asocjacji kwalifikowanej

**Zad. 6:** Wskaż na diagramie klas asocjację, która jest agregacją lub kompozycją (lub może kwalifikować się do zamiany na takową). Objaśnij, dlaczego zdecydowałeś się na wybór agregacji/kompozycji, a nie „zwykłej” asocjacji? Ponadto, dlaczego wybrałeś agregację, a nie kompozycję (czy też odwrotnie)?

Na diagramie występuje kilka agregacji: np. ta pomiędzy klasami: *Trening* i *Zawodnik*. Agregacja (lepiej niż zwykła asocjacja) wskazuje na związek części-całość występujący pomiędzy treningiem i zawodnikami biorącymi w nim udział. Ponieważ w tym przypadku cykl życiowy części nie zawiera się w cyklu życiowym całości, tzn. usunięcie z systemu obiektu klasy *Trening* nie będzie skutkowało usunięciem tych obiektów klasy *Zawodnik*, które były związane z danym treningiem, dlatego wykorzystano agregację, a nie kompozycję. Ponadto, na przestrzeni czasu każdy z zawodników mógł być związany z wieloma treningami – stąd liczność \* dla roli *Trening*.

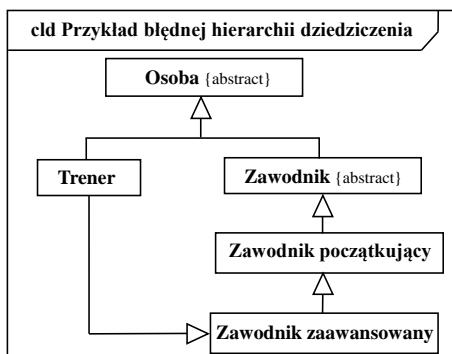
**Zad. 7:** Wskaż na diagramie klas przykładowe ograniczenie. Na jaki element modelu zostało ono nałożone? Czy jest to ograniczenie statyczne czy dynamiczne? Zaproponuj, jaka metoda i w jakiej klasie mogłaby być odpowiedzialna za jego zapewnienie.

Np. ograniczenie, że pojemność silnika gokarta nie może przekroczyć 0,25 litra. Jest to ograniczenie statyczne, nałożone na atrybut *pojemność silnika* w klasie *Gokart*. Ograniczenie to mogłoby być zrealizowane w metodzie tworzącej nowy obiekt tej klasy.

### 8.3.6 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

- Błędem jest wprowadzanie do diagramów informacji, która ani bezpośrednio, ani pośrednio nie wynika z tekstu wymagań, jak np. *id zawodnika*, *id zespołu*, *id trenera*, *poziom umiejętności zawodnika*, *nr toru*.
- Należy pamiętać, że nazwa klasy to zazwyczaj rzeczownik w liczbie pojedynczej, tzn. jeżeli pojedynczy obiekt przechowuje informację o jednym zawodniku, to klasa nazywa się *Zawodnik*, ale gdyby pojedynczy obiekt przechowywał informacje o zbiorze zawodników, to w takim przypadku klasa nazywałaby się *Zawodnicy*.
- W nazwach podklas trzeba umieszczać nazwę nadklasy, czyli nazwa klasy to nie *Początkujący*, ale *Zawodnik początkujący*.
- Stosunkowo często konstruowane są nieprawidłowe hierarchie klas, co prawdopodobnie wynika z nieznajomości reguł, które wykorzystywane są w tym procesie. Klasa łączy się w hierarchie na zasadzie podobieństwa ich znaczeń, a nie podobieństwa ich własności (atrybutów czy operacji). To oznacza, że jeśli zbudujemy hierarchię, gdzie klasa *Zawodnik* dziedziczy po klasie *Osoba*, to następujące zdanie musi być prawdziwe: *zawodnik jest rodzajem, szczególnym*

*przypadkiem osoby* (obiekt klasy *Zawodnik* jest pośrednim wystąpieniem klasy *Osoba*). Spójrzmy na Rys. 8-14. Opisuję on strukturę, gdzie wprowadzono dwie konstrukcje: 1) *zawodnik zaawansowany jest rodzajem, szczególnym przypadkiem zawodnika początkującego* (co nie jest prawdą) i 2) *trener jest rodzajem, szczególnym przypadkiem zawodnika zaawansowanego (i pośrednio też zawodnika początkującego)* – co też nie jest prawdą. Te dwa nieprawdziwe zdania oznaczają, że hierarchia została zbudowana błędnie. Ponadto, przypadkowo udało się tu zbudować hierarchię z dziedziczeniem wielokrotnym: klasa *Trener* dziedziczy po klasach: *Osoba* i *Zawodnik zaawansowany*.

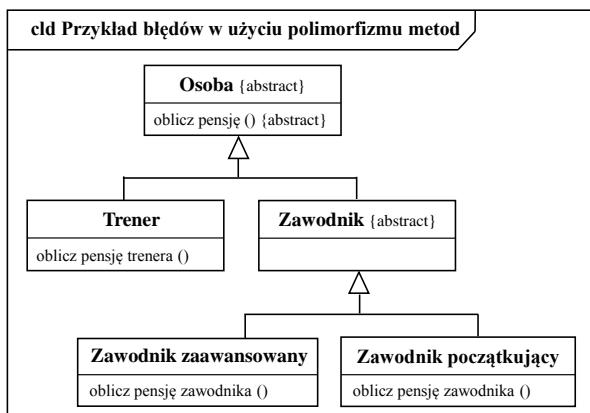


Rys. 8-14 Przykład błędnej hierarchii osób

- Nierzadko pojawiają się problemy z oznaczaniem rodzajów atrybutów, a w szczególności – atrybutów pochodnych występujących w dużej liczbie na schemacie pojęciowym w tym zadaniu. *Dla przypomnienia:* wartość atrybutu pochodnego da się wyliczyć lub wyprowadzić w oparciu o dane bazowe przechowywane w całej strukturze (a nie tylko w tym samym obiekcie). Wprowadzanie atrybutów pochodnych ma znaczenie optymalizacyjne. Analityk wie, że atrybut pochodny jest elementem redundantnym, czyli nie powinien być wprowadzony do diagramu ze względu na pogorszenie percepcji (ponieważ zwiększa się złożoność diagramu), ale sugeruje projektantowi – który może nie znać dziedziny problemowej – potrzebę przechowywania takiego atrybutu (poprzedzając go znakiem ukośnika). Sytuacja taka ma miejsce, gdy pewne dane nigdy nie ulegają zmianom po wyliczeniu ich wartości, proces wyliczenia jest kosztowny i na dodatek dane te są często wykorzystywane. Podsumowując, nie tylko atrybut *wiek* (wyliczany z *roku urodzenia*) jest atrybutem pochodnym, ale też np. atrybut *ile razy brał udział w zawodach* z klasy *Zawodnik zaawansowany*, pochodny w efekcie wprowadzenia asocjacji łączących klasę *Osoba* i *Zespół* oraz *Zespół* i *Zawody*.
- Uwaga: metody tworzące nowe obiekty tej klasy, w której zostały umieszczone, muszą być metodami klasowymi. A więc, metoda utwórz zespół (czy też

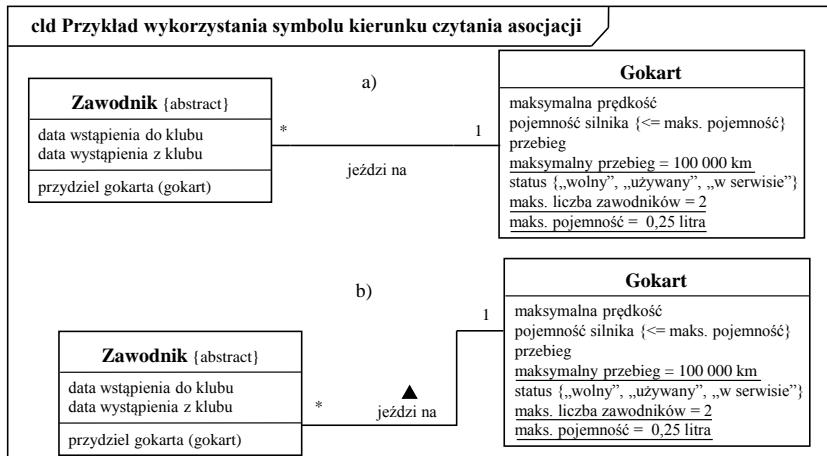
rejestruj zespół) jest metodą klasową w klasie *Zespół*. Gdyby metodę tworzącą nowy zespół umieszczono w klasie *Kierownik* (pod warunkiem że taka klasa została wprowadzona do diagramu, choć nie byłoby to poprawne, ponieważ na tym etapie nie modelujemy w strukturze aktorów systemu), to byłaby metodą obiektową. Z kolei, metody usuwające pojedyncze obiekty klasy są metodami obiektowymi – czyli metoda *zwolnij trenera ()* (jednego trenera) umieszczona w klasie *Trener* jest metodą obiektową, natomiast metoda *zwolnij trenerów ()* byłaby metodą klasową. Wniosek: należy zwracać uwagę na dobór nazw nadawanych w trakcie budowy modelu struktury (oczywiście dotyczy to również wszystkich innych modeli), aby nie wprowadzać w błąd innych uczestników projektu.

- Dla przypomnienia: *Polimorfizm metod* nie oznacza istnienia jednej metody, która wykonywana jest w różny sposób w zależności od tego, na jakim obiekcie została uruchomiona, ale wielu metod (co najmniej dwóch), które mają to samo zadanie do wykonania – co widzimy dzięki temu, że metody te mają identyczne sygnatury. Każda z metod polimorficznych realizuje zadanie określone poprzez sygnaturę (dlatego właściwy dobór nazw jest tak ważny w analizie), w inny sposób (w zależności od obiektu, do którego został wysłany komunikat). Wniosek – metody polimorficzne różnią się ciałami (algorytmami, na których te ciała oparto). Na Rys. 8-15 pokazano diagram, który miał służyć za ilustrację dla polimorfizmu metod. Otóż, tylko metody *oblicz pensję zawodnika ()* w klasach *Zawodnik początkujący* i *Zawodnik zaawansowany* można uznać za metody polimorficzne, ponieważ mają identyczne sygnatury (te same nazwy, puste listy argumentów i zostały umieszczone w jednej hierarchii). Metoda *oblicz pensję trenera ()* nie wchodzi w skład rozważanej grupy metod polimorficznych, gdyż ma inną nazwę. Ponadto, żadnej z trzech wyżej wymienionych metod nie można uznać za implementację metody abstrakcyjnej *oblicz pensję ()*, umieszczonej w klasie abstrakcyjnej *Osoba*, z tego samego powodu, co poprzednio, czyli ze względu na inną nazwę.



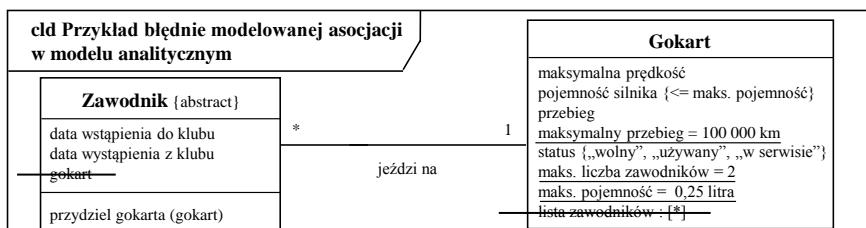
Rys. 8-15 Przykład dla zilustrowania błędnego wykorzystania polimorfizmu metod

- Dla przypomnienia:* nie należy nadużywać symbolu zamalowanego „trójkątka”, który wskazuje kierunek czytania nazwy asocjacji. Za domyślny przyjmuje się kierunek z lewa na prawo i z góry na dół. Na Rys. 8-16, w obu przypadkach: a) i b), zostało sformułowane, że **zawodnik jeździ na gokarcie**, a nie odwrotnie – co może nie być oczywiste w sytuacji, gdy projektant nie zna dziedziny problemowej i opiera się wyłącznie na informacjach przekazywanych na diagramach przez analityka.



Rys. 8-16 Ilustracja wykorzystania symbolu „trójkątka” do zmiany kierunku czytania

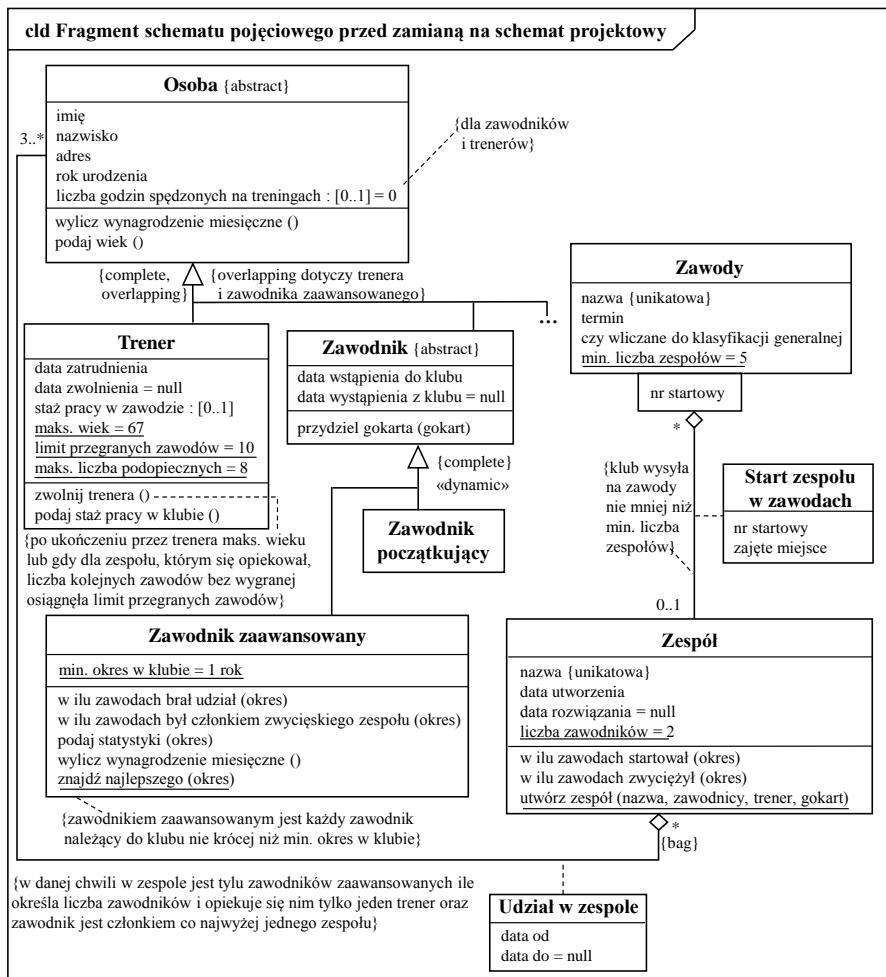
- Dla przypomnienia:* Do modelowania związków typu *asocjacja*, występujących pomiędzy bytami wyróżnialnymi w tym fragmencie dziedziny problemowej, który podlega analizie, wykorzystujemy (na tym etapie) wyłącznie asocjacje („asocjacje zwykłe”, agregacje, kompozycje, asocjacje kwalifikowane, asocjacje n-arne) – bez dodatkowych, skojarzonych z tymi asocjacjami atrybutów. Czyli, jak to zostało zilustrowane na Rys. 8-17, nie należało wprowadzać do diagramu, oprócz asocjacji *jeździ na*, dodatkowych atrybutów modelujących ten związek (tutaj: atrybut *gokart* w klasie *Zawodnik* i atrybut *lista zawodników* w klasie *Gokart*). W modelowaniu obiektowym, w takiej sytuacji i na tym etapie, wykorzystywane są właśnie asocjacje i tylko one.



Rys. 8-17 Błędne modelowanie związków asocjacji – niepotrzebne atrybuty

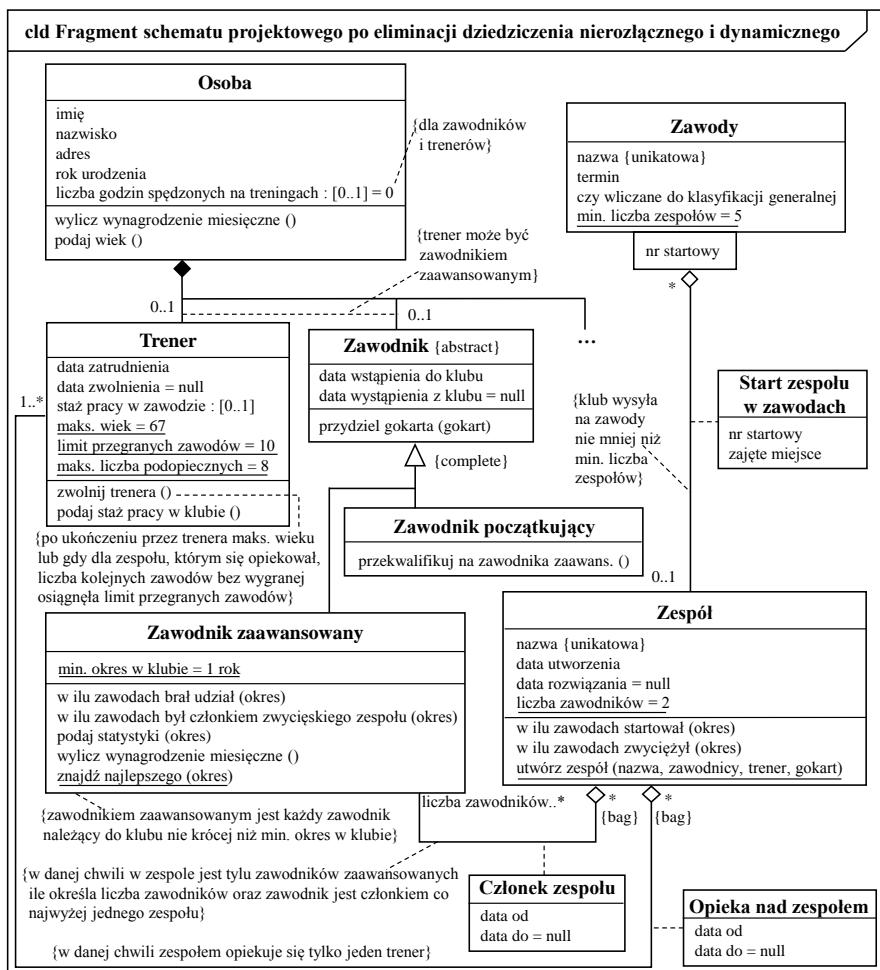
### 8.3.7 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji

Rys. 8-18 ilustruje fragment schematu pojęciowego (z pominięciem komentarzy), niezbędny dla zamodelowania na schemacie projektowym [37, 40] metody *podaj statystyki (okres)* w klasie *Zawodnik zaawansowany* za pomocą diagramów interakcji (Zad. 2 w części analizy dynamicznej). Fragment ten zawiera klasy hierarchii osób (*Osoba*, *Trener*, *Zawodnik*, *Zawodnik początkujący*, *Zawodnik zaawansowany*), *Zawody* i *Zespół*, agregację z klasą asocjacji *Udział w zespole* oraz agregację kwalifikowaną z klasą asocjacji *Start zespołu w zawodach*.



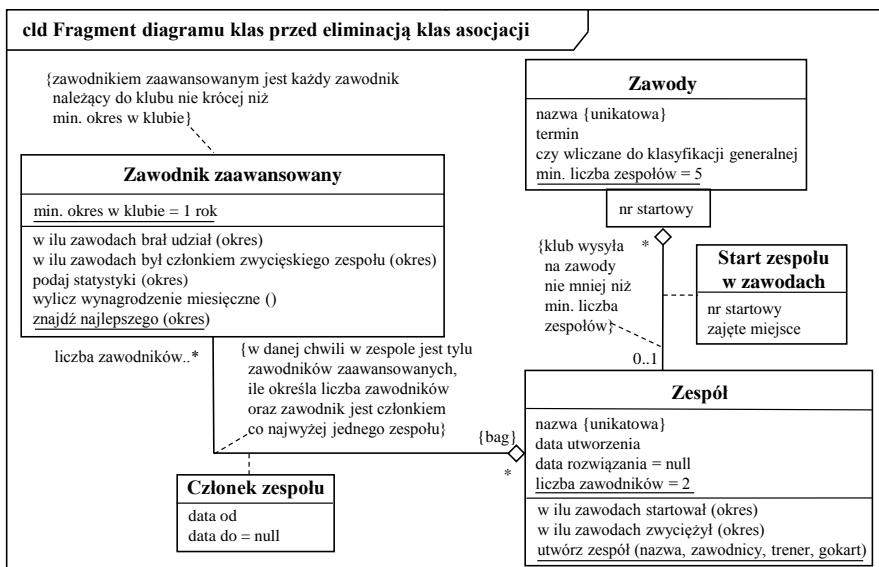
Rys. 8-18 Fragment schematu pojęciowego przed zamianą na schemat projektowy

W typowym języku obiektowym (np. w Javie) nie występuje dziedziczenie nierożłączne, dziedziczenie dynamiczne, klasa asocjacji ani asocjacja kwalifikowana. Dlatego też w pierwszym kroku zamiany schematu pojęciowego na schemat projektowy zostanie „przerwana” hierarchia dziedziczenia między klasą *Osoba* a jej bezpośrednimi specjalizacjami: *Trenerem* i *Zawodnikiem*, a następnie zastąpiona za pomocą kompozycji oraz zamianie dziedziczenia dynamicznego dla zawodników za pomocą dziedziczenia rozłącznego (i kompletnego) wraz z wprowadzeniem w klasie *Zawodnik początkujący* metody *przekwalifikuj na zawodnika zaawans.* (), realizującej migrację (zmianę przynależności do klasy) obiektu z klasy *Zawodnik początkujący* do klasy *Zawodnik zaawansowany*. Pierwszy krok transformacji został zilustrowany na Rys. 8-19.



Rys. 8-19 Fragment schematu projektowego po eliminacji dziedziczenia nierożłącznego i dynamicznego – krok pierwszy

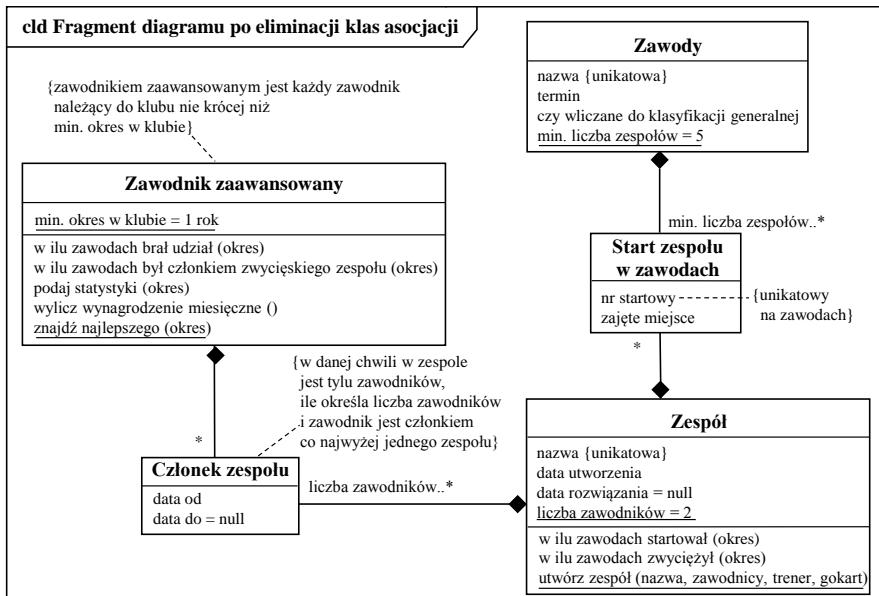
Ponieważ część elementów umieszczonych na diagramie z Rys. 8-19 nie jest niezbędna do zamodelowania metody *podaj statystyki (okres)*, dlatego (w celu lepszej czytelności) drugi krok, polegający na eliminacji klas asocjacji, zostanie przeprowadzony dla fragmentu diagramu klas, który jest przedstawiony na Rys. 8-20. Klasy asocjacji *Członek zespołu* oraz *Start zespołu w zawodach* wymagają zastąpienia za pomocą odpowiednich klas pośredniczących jak na Rys. 8-21. Warto zwrócić uwagę na to, że po wprowadzeniu klas pośredniczących znika ograniczenie *{bag}* dla asocjacji między klasami *Zawodnik zaawansowany* i *Zespół*.



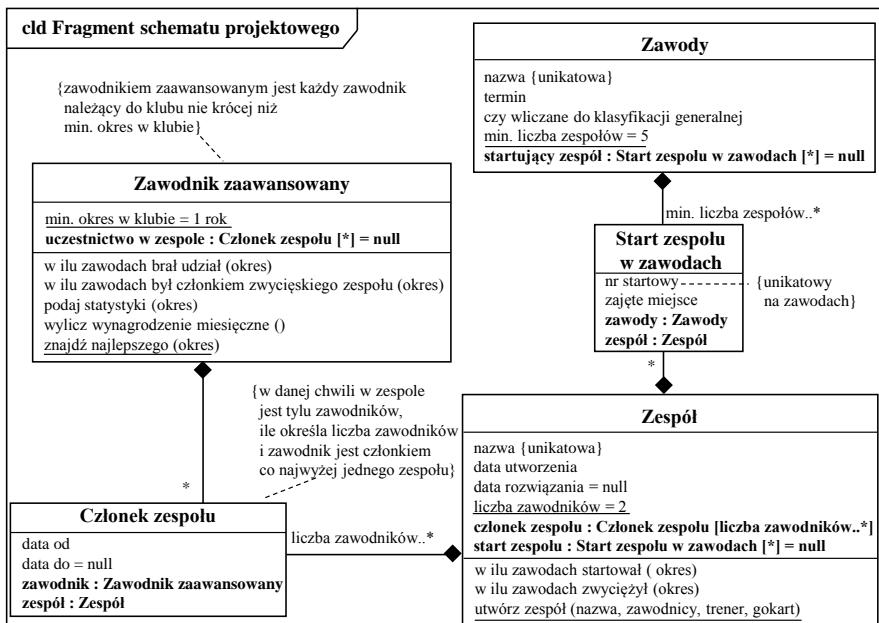
Rys. 8-20 Fragment schematu projektowego przed eliminacją klas asocjacji

Chcemy wiedzieć, czy i jakiego zespołu członkiem jest/był dany zawodnik zaawansowany i chcemy także mieć możliwość podawania składu zespołu. Wynika stąd potrzeba posiadania dwukierunkowej nawigacji między odpowiednimi obiektami klas *Zawodnik zaawansowany*, *Członek zespołu* i *Zespół*. Podobna sytuacja występuje w przypadku obiektów klas *Zawody*, *Start zespołu w zawodach* i *Zespół*. Aby można było ustalić, w jakich zawodach brał udział dany zespół (i jakie zajął na nich miejsce) oraz np. w celu weryfikacji zespołów z klubu, które brały udział w danych zawodach (lub weryfikacji ograniczenia na minimalną liczbę startujących zawodów), potrzebna będzie nawigacja dwustronna między odpowiednimi obiektami wymienionych klas.

W typowym języku obiektowym implementacja roli asocjacji, która umożliwia nawigację, wymaga wprowadzenia dodatkowego atrybutu zawierającego referencję do obiektu klasy z przeciwej strony asocjacji.



Rys. 8-21 Fragment schematu projektowego po eliminacji klas asocjacji – krok drugi



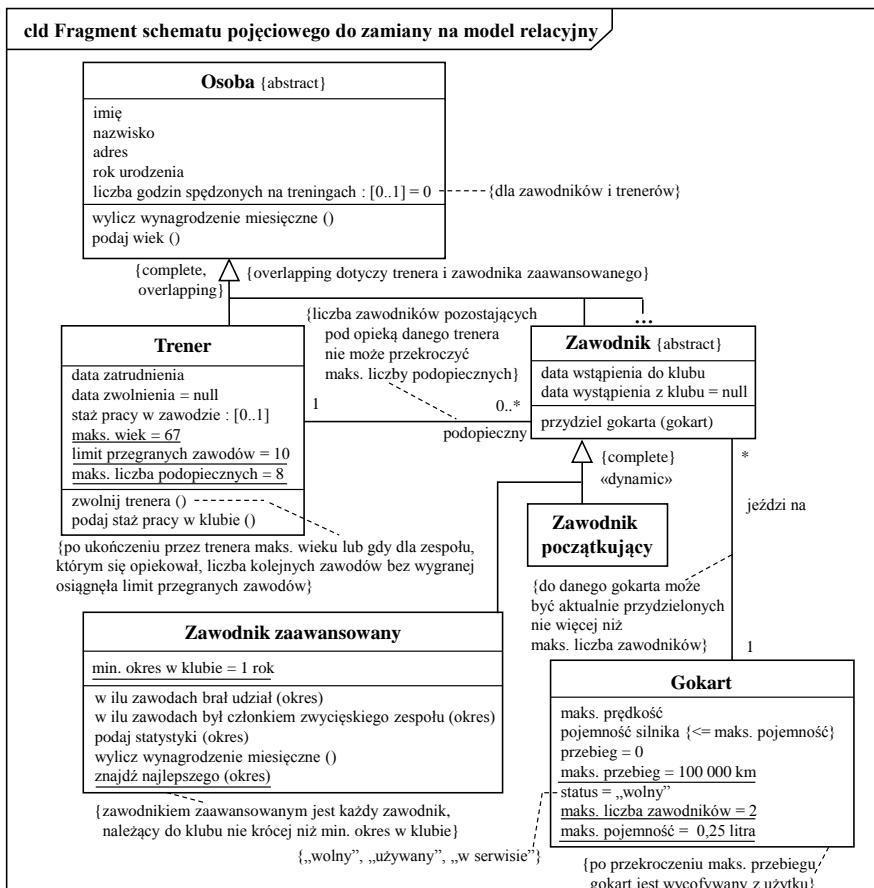
Rys. 8-22 Fragment schematu projektowego po zakończonej transformacji ze schematu pojęciowego

Schemat projektowy dla wskazanego na Rys. 8-20 fragmentu diagramu klas został przedstawiony na Rys. 8-22. Dla lepszej czytelności wytnięte zostały atrybuty służące do implementacji asocjacji. Należy także zaznaczyć, że typy atrybutów, argumentów metod oraz typy zwracanych wartości zostały w większości pominięte w zaprezentowanych rozważaniach.

### 8.3.8 Schemat relacyjny dla fragmentu schematu pojęciowego

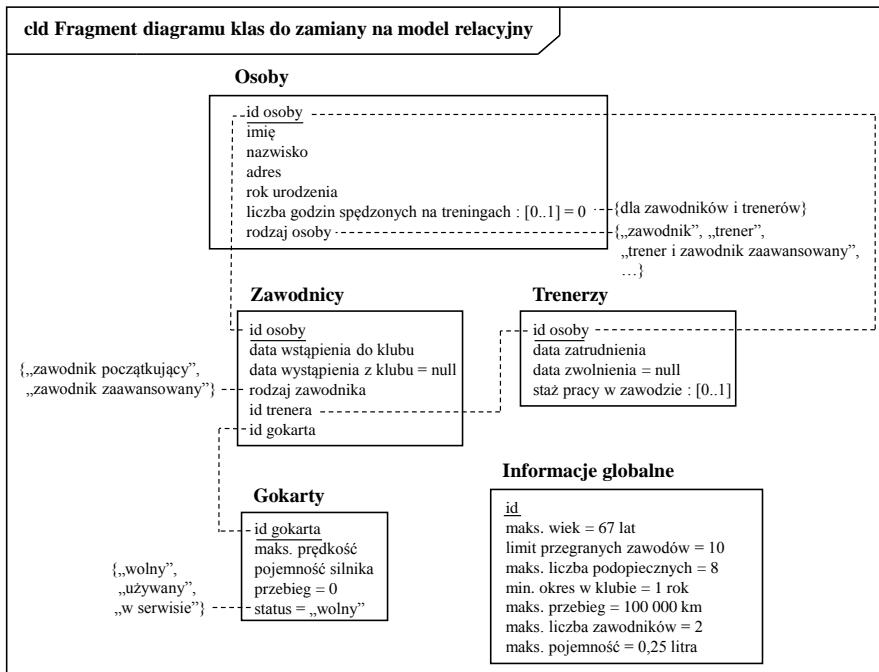
*Polecenie:* Fragment schematu pojęciowego, zawierający przynajmniej trzy klasy, związek dziedziczenia oraz asocjację, przedstaw w modelu relacyjnym.

Na Rys. 8-23 przedstawiony został fragment schematu pojęciowego zawierający hierarchię osób i dwie asocjacje: pierwsza – łącząca trenera z jego podopiecznymi (zawodnikami), zaś druga – łącząca zawodnika z przydzielonym do niego gokartem.

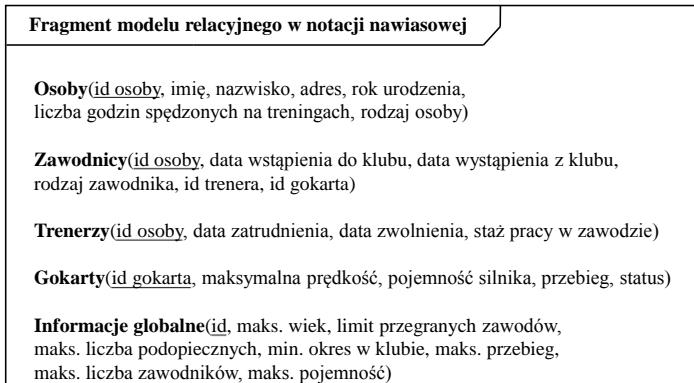


Rys. 8-23 Fragment schematu pojęciowego podlegający zamianie na model relacyjny

Dla zwiększenia czytelności diagramu wszystkie komentarze zostały pominięte, natomiast pozostawiono ograniczenia. Rys. 8-24 zawiera model relacyjny [37, 47] tego fragmentu diagramu klas w notacji 1 [41]. Na Rys. 8-25 zilustrowano odpowiedni model relacyjny w notacji nawiasowej.

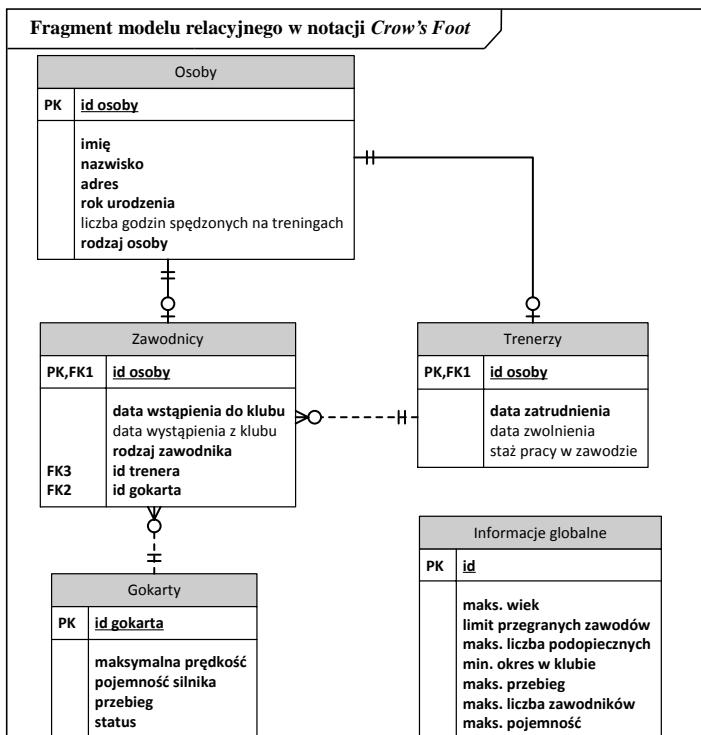


Rys. 8-24 Model relacyjny fragmentu diagramu klas (w notacji 1)



Rys. 8-25 Model relacyjny fragmentu diagramu klas (w notacji 2)

Natomiast Rys. 8-26 przedstawia odpowiedni model relacyjny w notacji *Crow's Foot*.



Rys. 8-26 Model relacyjny fragmentu diagramu klas (w notacji *Crow's Foot*)

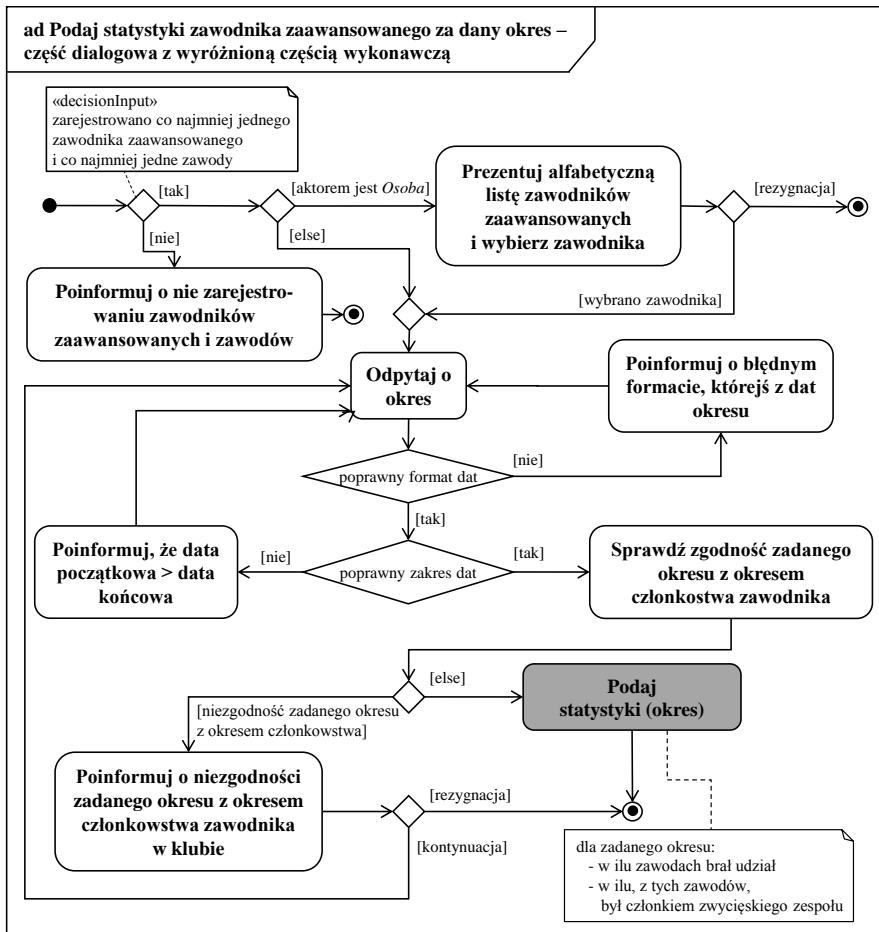
## 8.4 Analiza dynamiczna

### 8.4.1 Zadania podstawowe

**Zad. 1:** Dla funkcjonalności *Podaj statystyki zawodnika zaawansowanego za dany okres* (punkt 11.3 tekstu wymagań) skonstruuj diagram aktywności zgodny ze scenariuszem.

Scenariusz dla przypadku użycia *Podaj statystyki zawodnika zaawansowanego za dany okres* został określony w Tab. 8-1.

Rys. 8-27 przedstawia diagram aktywności zgodny z podanym scenariuszem w Tab. 8-1. Aktywność zaciemniona oznacza część wykonawczą dla przypadku użycia *Podaj statystyki zawodnika zaawansowanego za dany okres*.

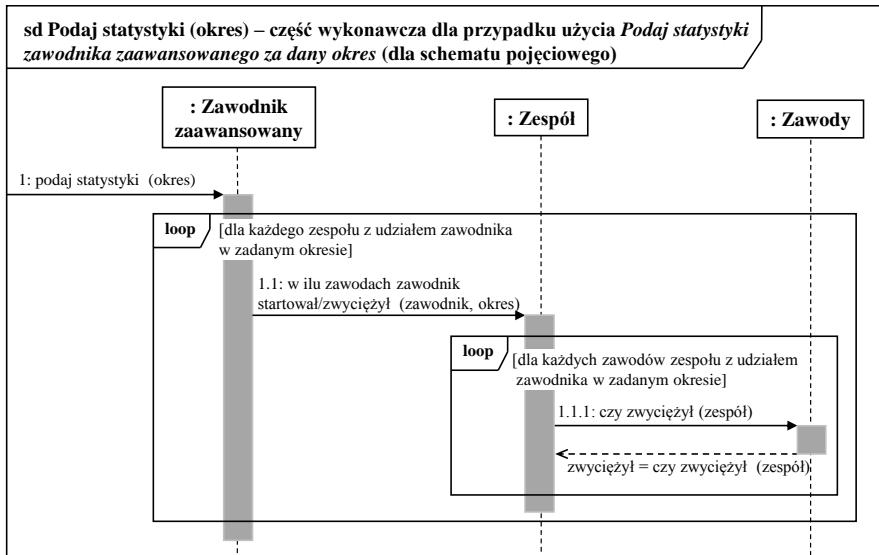


Rys. 8-27 Diagram aktywności dla przypadku użycia *Podaj statystyki zawodnika zaawansowanego za dany okres*

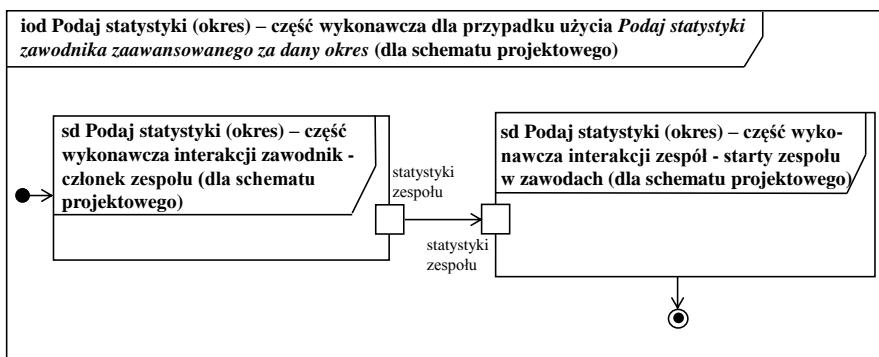
**Zad. 2:** Sporządź diagram interakcji dla części wykonawczej przypadku użycia *Podaj statystyki zawodnika*. Diagram powinien przedstawać interakcje pomiędzy obiektami co najmniej dwóch klas.

Diagram sekwencji dla części wykonawczej przypadku użycia *Podaj statystyki zawodnika* dla schematu pojęciowego z Rys. 8-12 został zilustrowany na Rys. 8-28.

Rys. 8-29 zawiera diagram przeglądu interakcji dla aktywności realizującej część wykonawczą przypadku użycia *Podaj statystyki zawodnika zaawansowanego za dany okres* dla schematu projektowego z Rys. 8-22.

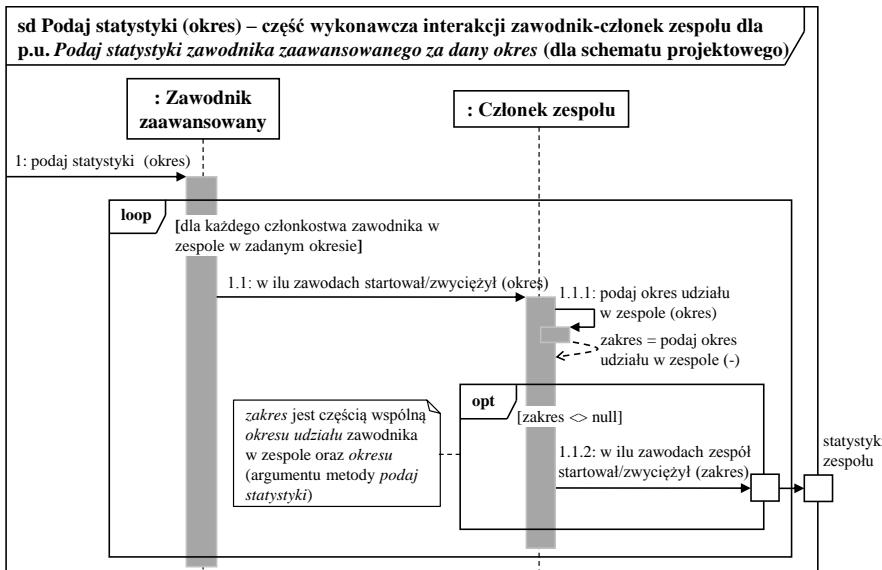


Rys. 8-28 Diagram sekwencji dla przypadku użycia *Podaj statystyki zawodnika za dany okres* (dla schematu pojęciowego)

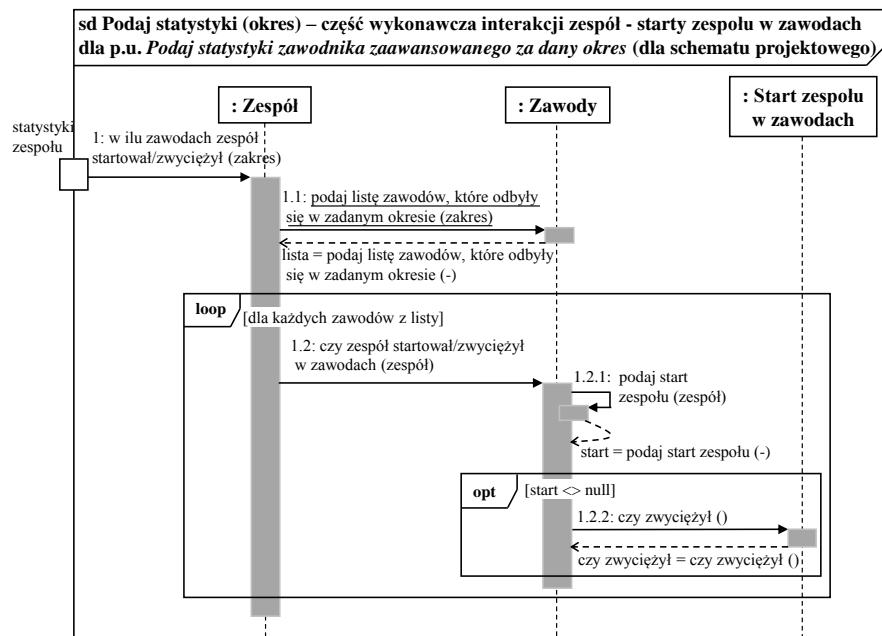


Rys. 8-29 Diagram przeglądu interakcji dla przypadku użycia *Podaj statystyki zawodnika zaawansowanego za dany okres* dla schematu projektowego

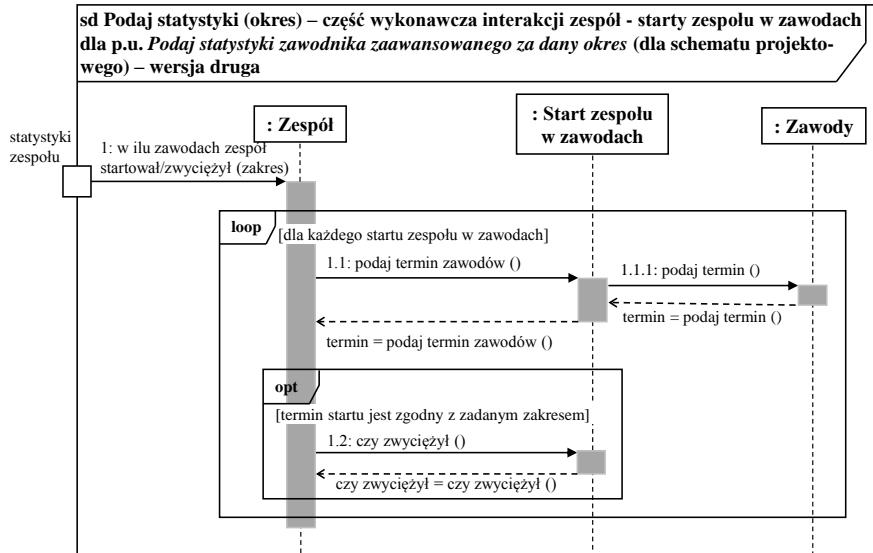
Rys. 8-30 przedstawia diagram sekwencji zawodnik-członek zespołu dla części wykonawczej przypadku użycia *Podaj statystyki zawodnika za dany okres*, odpowiedni dla schematu projektowego. Natomiast Rys. 8-31 i Rys. 8-32 zawierają dwie wersje diagramów sekwencji zespołu – stary zespołu w zawodach (dla schematu projektowego), różniące się algorytmami wyliczania statystyk zespołu, którego członkiem jest/był zawodnik zaawansowany.



Rys. 8-30 Diagram sekwencji zawodnik-członek zespołu dla przypadku użycia *Podaj statystyki zawodnika za dany okres* (dla schematu projektowego)



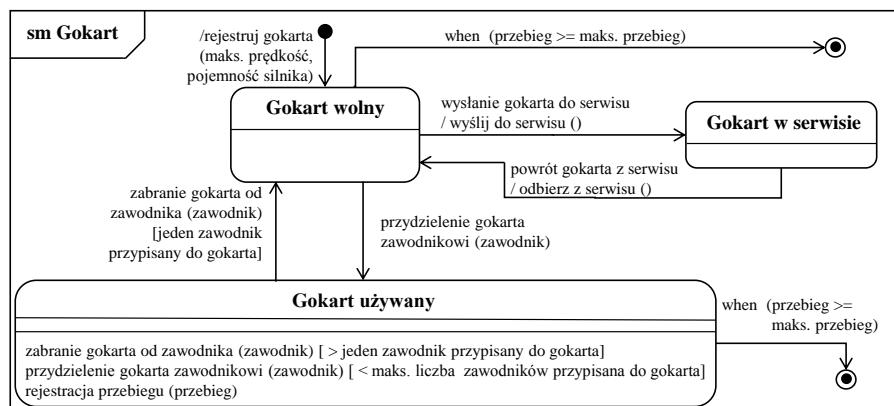
Rys. 8-31 Diagram sekwencji zespół – stary zespołu w zawodach dla przypadku użycia *Podaj statystyki zawodnika za dany okres* (dla schematu projektowego) – wersja pierwsza



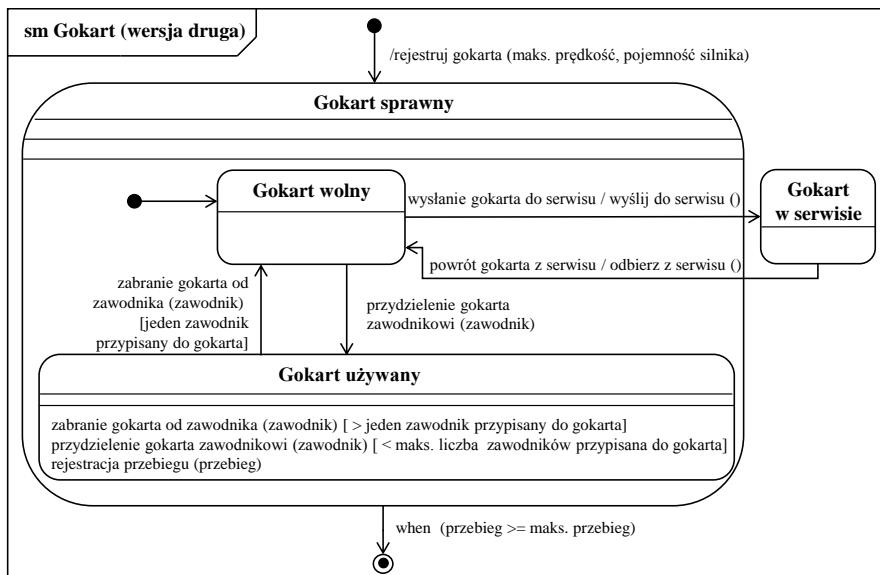
Rys. 8-32 Diagram sekwencji zespół – stary zespołu w zawodach dla przypadku użycia *Podaj statystyki zawodnika za dany okres* (dla schematu projektowego) – wersja druga

**Zad. 3:** Dla klasy *Gokart* na schemacie pojęciowym z Rys. 8-12 wskaż kilka stanów (co najmniej dwa stanów, nie licząc stanów początkowego i końcowego), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów, zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

Diagram stanów dla klasy *Gokart* został przedstawiony na Rys. 8-33. Alternatywne rozwiązanie, z wykorzystaniem stanu złożonego *Gokart sprawny* ilustruje Rys. 8-34.



Rys. 8-33 Maszyna stanowa dla klasy *Gokart*



Rys. 8-34 Maszyna stanowa dla klasy *Gokart* (wersja druga ze stanem złożonym sekwencyjnym)

#### **8.4.2 Omówienie zmian na schemacie pojęciowym i schemacie projektowym wynikających z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań**

Po przeprowadzeniu analizy dynamicznej należy uzupełnić schemat pojęciowy i schemat projektowy o nowe elementy, które zostały ujęte na diagramach interakcji oraz diagramach stanów.

Lista zmian na schemacie pojęciowym została zamieszczona w Tab. 8-6. Natomiast lista zmian na schemacie projektowym, zgodnie z diagramami interakcji z Rys. 8-30 i Rys. 8-31, została zamieszczona w Tab. 8-7 i Tab. 8-8.

Tab. 8-6 Lista zmian na schemacie pojęciowym po analizie dynamicznej

Klasa	Zmiana
Gokart	dodatkowe metody: <i>odbierz z serwisu ()</i> <i>przydziel zawodnikowi (zawodnik)</i> <u><i>rejestruj gokarta (maks. prędkość, pojemność silnika)</i></u> <u><i>rejestruj przebieg (przebieg)</i></u>

	<i>wyslij do serwisu () usuń () zabierz od zawodnika (zawodnik)</i>
Zawody	dodatkowa metoda: <i>czy zwyciężył (zespół)</i>
Zespół	dodatkowa metoda: <i>w ilu zawodach startował/zwyciężył (zawodnik, okres)</i>

Tab. 8-7 Lista zmian na schemacie projektowym po analizie dynamicznej (wersja pierwsza)

Klasa	Zmiana
Członek zespołu	dodatkowe metody: <i>podaj okres udziału w zespole (okres) w ilu zawodach startował/zwyciężył (okres)</i>
Gokart	dodatkowe metody: <i>odbierz z serwisu () przydziel zawodnikowi (zawodnik) <u>rejestruj gokarta (maks. predkość, pojemność silnika)</u> <u>rejestruj przebieg (przebieg)</u> <i>wyslij do serwisu () usuń () zabierz od zawodnika (zawodnik)</i></i>
Start zespołu w zawodach	dodatkowa metoda: <i>czy zwyciężył ()</i>
Zawody	dodatkowe metody: <i>czy zespół startował/zwyciężył w zawodach (zespół) <u>podaj listę zawodów, które odbyły się w zadanym okresie (zakres)</u> <u>podaj start zespołu (zespół)</u></i>

<i>Zespół</i>	dodatkowa metoda: <i>w ilu zawodach zespół startował/zwyciężył (zakres)</i>
---------------	--

Lista zmian na schemacie projektowym dla diagramu interakcji z Rys. 8-30 i wersji drugiej diagramu interakcji zespół-statystyki zespołu w zawodach z Rys. 8-32 została zamieszczona w Tab. 8-8.

Tab. 8-8 Lista zmian na schemacie projektowym po analizie dynamicznej (wersja druga)

Klasa	Zmiana
<i>Gokart</i>	dodatkowe metody: <i>odbierz z serwisu ()</i> <i>przydziel zawodnikowi (zawodnik)</i> <i>rejestruj gokarta (maks. predkość, pojemność silnika)</i> <i>rejestruj przebieg (przebieg)</i> <i>wyslij do serwisu ()</i> <i>usuń ()</i> <i>zabierz od zawodnika (zawodnik)</i>
<i>Członek zespołu</i>	dodatkowe metody: <i>podaj okres udziału w zespole (okres)</i> <i>w ilu zawodach startował/zwyciężył (zawodnik, okres)</i>
<i>Start zespołu w zawodach</i>	dodatkowe metody: <i>czy zwyciężył ()</i> <i>podaj termin zawodów ()</i>
<i>Zawody</i>	dodatkowa metoda: <i>podaj termin ()</i>
<i>Zespół</i>	dodatkowa metoda: <i>w ilu zawodach zespół startował/zwyciężył (zakres)</i>

## 8.5 Zadania do samodzielnej pracy

1. Objaśnij pojęcie: zależność. Podaj przykład zastosowania w modelu funkcjonalnym, strukturalnym i dynamicznym skonstruowanym na podstawie tekstu wymagań.
2. Czy system zewnętrzny lub podsystem wewnętrzny mogą być aktorami? Jeśli tak, odpowiedź uzupełnij przykładami spójnymi z tekstem wymagań.
3. Objaśnij różnicę między dziedziczeniem nierożłącznym (ang. overlapping) a dziedziczeniem dynamicznym (ang. dynamic inheritance). Czy któryś z tych obu rodzajów dziedziczenia wystąpił na schemacie pojęciowym skonstruowanym na podstawie tekstu wymagań?
4. Dla schematu pojęciowego skonstruowanego na podstawie tekstu wymagań podaj przykłady: operacji, metody i komunikatu.
5. Objaśnij pojęcie: przesłanianie metod. Wskaż przykład jego zastosowania na schemacie pojęciowym skonstruowanym na podstawie tekstu wymagań. Czy przesłanianie jest rodzajem polimorfizmu metod?
6. Czy zasadę zamienialności można stosować wyłącznie do obiektów czy także do powiązań? Odpowiedź uzasadnij. Jeśli odpowiedź jest na „tak”, wskaż przykład jej zastosowania na schemacie pojęciowym skonstruowanym na podstawie tekstu wymagań.
7. Dla schematu pojęciowego skonstruowanego na podstawie tekstu wymagań podaj przykład ograniczenia, które odnosi się do dwóch asocjacji. Do jakiej metody i w jakiej klasie włożyłbyś implementację tego ograniczenia?
8. Dla wybranego przypadku użycia systemu (innego niż omówionego szczegółowo w analizie funkcjonalnej lub dynamicznej) sformułuj algorytm i narysuj diagram aktywności, określając przynajmniej dwie aktywności, nie licząc początkowej i końcowej.
9. Dla przypadku użycia wybranego w zadaniu 8 narysuj diagram interakcji, zgodny ze sformułowanym algorytmem, który przedstawia interakcje pomiędzy obiektami co najmniej dwóch różnych klas.
10. Dla dowolnej klasy (innej niż omówionej szczegółowo w analizie dynamicznej) ze schematu pojęciowego wskaż dwa stany (wyłączając stan początkowy i stan końcowy), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów, zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

## 8.6 Podsumowanie zadania

W Tab. 8-9 przedstawiono ogólną charakterystykę omawianego zadania. Szczegółową charakterystykę zadania dla grup tematycznych: *Wprowadzenie do*

*przedmiotu, Analiza funkcjonalna, Analiza strukturalna, Analiza dynamiczna* zamieszczono odpowiednio w tabelach Tab. 8-10, Tab. 8-11, Tab. 8-12 i Tab. 8-13.

Tab. 8-9 Ogólna charakterystyka zadania

<b>Ogólna charakterystyka zadania</b>		
<i>Nazwa</i>	Osiedlowy klub kartingowy	
<i>Dziedzina problemowa</i>	system mógłby znaleźć zastosowanie w niewielkim klubie sportowym, szkole sportowej itp.	
<i>Cel</i>	wprowadzenie systemu ułatwi zarządzanie klubem kartingowym, np. w rejestraniu zawodników, zespołów i zawodów	
<i>Zakres odpowiedzialności</i>	ewidencja trenerów, zawodników, gokartów, treningów, udziału w zawodach, zarządzanie zespołami, wyliczanie wynagrodzenia, tworzenie raportów o zawodnikach, zespołach i ich udziale w zawodach	
<i>Stopień trudności</i>	trudne	
<i>Rozmiar</i>	10 klas	
<i>Użyteczność w grupach tematycznych</i>		
<i>Grupa tematyczna</i>	<i>Użyteczność w grupie tematycznej</i>	<i>Użyteczność na etapie zgodnym z modelem efektywnego nauczania</i>
<i>Wprowadzenie do przedmiotu</i>	niska	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> niska <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> niska
<i>Analiza funkcjonalna</i>	wysoka	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> wysoka <i>Podsumowanie dotychczasowych działań:</i> wysoka <i>Ewaluacja:</i> wysoka
<i>Analiza strukturalna</i>	wysoka	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> wysoka <i>Podsumowanie dotychczasowych działań:</i> wysoka <i>Ewaluacja:</i> wysoka
<i>Analiza dynamiczna</i>	wysoka	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> wysoka

		<i>Podsumowanie dotychczasowych działań: wysoka Ewaluacja: wysoka</i>
--	--	---

Tab. 8-10 Szczegółowa charakterystyka zadania dla grupy tematycznej *Wprowadzenie do przedmiotu*

<b>Szczegółowa charakterystyka zadania</b>			
<b>Grupa tematyczna</b>	<i>Wprowadzenie do przedmiotu</i>		
<b>Stopień trudności grupy w zadaniu</b>	łatwe		
<b>Poruszane tematy</b>	<i>analiza a projektowanie, dekompozycja diagramów, mechanizmy rozszerzalności, reguły nazewnictwa, poprawa percepacji diagramu/diagramów</i>		
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>	
<i>analiza a projektowanie</i>	łatwe	<i>model pojęciowy, model projektowy</i>	
<i>mechanizmy rozszerzalności</i>	łatwe	<i>stereotyp, ograniczenie, komentarz</i>	
<i>dekompozycja diagramów</i>	łatwe	podział modelu przypadków użycia na diagramy	
<i>reguły nazewnictwa</i>	łatwe	reguły nazewnictwa przypadków użycia, klas, atrybutów, metod i asocjacji	
<i>poprawa percepacji diagramu/diagramów</i>	łatwe	reguły organizacji przypadków użycia i budowy hierarchii klas, określanie kierunku czytania nazwy asocjacji	

Tab. 8-11 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza funkcjonalna*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Analiza funkcjonalna</i>

<b>Stopień trudności grupy w zadaniu</b>	średnio trudne	
<b>Poruszane tematy</b>	<i>aktorzy, przypadki użycia, scenariusze, dekompozycja modelu/diagramu przypadków użycia, mechanizmy rozszerzalności</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>aktorzy</i>	średnio trudne	aktor <i>Podsystem czasu</i> , aktor - system zewnętrzny, <i>hierarchia dla aktorów</i> , aktor <i>abstrakcyjny</i> , stosunkowo duża liczba aktorów
<i>przypadki użycia</i>	trudne	stosunkowo duża liczba przypadków użycia i relacji między przypadkami użycia
<i>scenariusze</i>	średnio trudne	
<i>dekompozycja modelu/diagramu przypadków użycia</i>	średnio trudne	<i>dokompozycja pozioma</i> , <i>dokompozycja pionowa</i> , przypadek użycia w roli aktora
<i>mechanizmy rozszerzalności</i>	łatwe	<i>ograniczenie, komentarz</i>

Tab. 8-12 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza strukturalna*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Analiza strukturalna</i>
<b>Stopień trudności grupy w zadaniu</b>	trudne
<b>Poruszane tematy</b>	<i>atrybuty klasy, metody klasy, struktury generalizacji-specjalizacji, klasa abstrakcyjna a konkretna, asocjacje, analiza wartości, przejście na schemat projektowy, przejście na schemat relacyjny, mechanizmy rozszerzalności</i>

<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>atrybuty klasy</i>	średnio trudne	<i>atrybut opcjonalny, powtarzalny, pochodny, obiektowy, klasowy, stosunkowo duża liczba atrybutów pochodnych</i>
<i>metody klasy</i>	średnio trudne	<i>metoda abstrakcyjna, zaimplementowana, obiektowa, klasowa, polimorfizm metod</i>
<i>struktury generalizacji-specjalizacji</i>	trudne	<i>dziedziczenie klas, dziedziczenie nierożłączone, niekompletne, dynamiczne i elipsa</i>
<i>klasa abstrakcyjna a konkretna</i>	łatwe	<i>klasa abstrakcyjna, klasa konkretna</i>
<i>asocjacje</i>	trudne	<i>asocjacja, rola asocjacji, asocjacja kwalifikowana, hierarchia dla asocjacji, stosunkowo duża liczba agregacji, klas asocjacji oraz ograniczeń nałożonych na asocjacje</i>
<i>analiza wartości</i>	trudne	<i>wartość pochodna, wartość początkowa, wartość graniczna, stosunkowo duża liczba wartości pochodnych, wartości początkowych oraz wartości granicznych</i>
<i>przejście na schemat projektowy</i>	trudne	<i>obejście dziedziczenia nierożłącznego i dynamicznego, zamiana asocjacji, asocjacja skierowana, zamiana asocjacji kwalifikowanej oraz asocjacji z klasą asocjacji</i>
<i>przejście na schemat relacyjny</i>	trudne	<i>obejście dziedziczenia nierożłącznego i dynamicznego, zamiana atrybutów klasowych</i>
<i>mechanizmy rozszerzalności</i>	średnio trudne	<i>ograniczenie, komentarz, stereotyp («dynamic»), stosunkowo duża liczba ograniczeń i komentarzy na schemacie pojeciowym i projektowym</i>

Tab. 8-13 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza dynamiczna*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Analiza dynamiczna</i>

<b>Stopień trudności grupy w zadaniu</b>	trudne	
<b>Poruszane tematy</b>	<i>diagramy aktywności, diagramy interakcji, diagramy stanów, podsumowanie analizy dynamicznej</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>diagramy aktywności</i>	średnio trudne	
<i>diagramy interakcji</i>	bardzo trudne	komunikaty zwrotne (powrotu sterowania), zagnieżdżone fragmenty wyodrębnione <i>loop i opt, diagramy przeglądu interakcji</i>
<i>diagramy stanów</i>	trudne	zdarzenie <i>when</i> , stan zagnieżdżony
<i>podsumowanie analizy dynamicznej</i>	średnio trudne	stosunkowo duża liczba dodatkowych metod na schemacie pojęciowym i projektowym

## Spis treści rozdziału

Rozdział 9 Internetowa książka kucharska .....	370
9.1 Tekst wymagań.....	370
9.2 Analiza funkcjonalna .....	371
9.2.1 Diagram kontekstowy .....	371
9.2.2 Model przypadków użycia.....	372
9.2.3 Scenariusze dla wybranych przypadków użycia łącznie z ich podziałem na podprzypadki.....	376
9.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	380
9.3 Analiza strukturalna.....	381
9.3.1 Schemat pojęciowy.....	381
9.3.2 Analiza elementów pochodnych.....	385
9.3.3 Analiza wartości początkowych .....	385
9.3.4 Analiza wartości granicznych.....	385
9.3.5 Zadania podstawowe .....	391
9.3.6 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	393
9.3.7 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji .....	397
9.3.8 Schemat relacyjny dla fragmentu diagramu klas .....	400
9.4 Analiza dynamiczna .....	403
9.4.1 Zadania podstawowe .....	403
9.4.2 Omówienie zmian na schemacie pojęciowym i schemacie projektowym, wynikającym z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań .....	406
9.5 Zadania do samodzielnej pracy .....	408
9.6 Podsumowanie zadania.....	409

# Rozdział 9

## Internetowa książka kucharska

Ewa Stemposz, Andrzej Jodłowski

### 9.1 Tekst wymagań

Poniżej zamieszczono wymagania dla systemu umożliwiającego współpracę wielu osób nad tworzeniem ogólnodostępnej, popularnej internetowej książki kucharskiej.

1. Mają być przechowywane przepisy dla potraw wywodzących się z różnych regionów świata, np. Włoch, Tajlandii itp. (ma być pamiętaana nazwa regionu/kraju). Przez przepis rozumiany jest opis własności potrawy, czyli np. określenie rodzaju i ilości składników, opis sposobu wykonania itd. Dana potrawa może być znana w wielu regionach świata i w każdym z nich może mieć swoją specyficzną (unikatową) nazwę (regionalną) oraz w pewnym stopniu może różnić się od przepisu bazowego, np. co do sposobu wykonania (chcemy pamiętać opis najbardziej istotnych różnic między przepisem bazowym potrawy a jej wersją regionalną, w postaci tekstu o długości nie większej niż 2560 znaków). Za przepis bazowy dla potrawy uważamy ten, który został wprowadzony jako pierwszy.
2. Każda potrawa ma być opisywana za pomocą informacji, takich jak: słowa kluczowe (co najmniej 3, nie więcej niż 10), nazwa (unikatowa), lista składników, łącznie z ich ilościami (co najmniej 3 składniki dla każdej potrawy), opis sposobu wykonania, czas przygotowania, liczba porcji potrawy (które uzyskuje się z podanych ilości składników), zdjęcie. Potrawa może składać się z innych potraw, a także może stanowić element składowy innych potraw. Zakłada się, że informacje o potrawach, dla których czas przygotowania byłby dłuższy niż określony czas maksymalny (aktualnie równy 1,5 godziny), nie będą przechowywane. Dla każdego składnika ma być przechowywana unikatowa nazwa oraz jednostka miary (np. dag, ml, szklanka, łyżeczka).
3. Potrawy dzielą się na grupy ze względu na ich rodzaj, jak np. przystawki na zimno czy sosy (dla sosów chcemy pamiętać liczbę składników). Jest możliwe, że dana potrawa może zostać zaliczona do więcej niż jednej z takich grup.
4. Ponadto wszystkie potrawy dzielą się także ze względu na trudność ich wykonania. Podział ten jest przeprowadzany w oparciu o przecienny czas przygotowania: łatwe (krócej niż 15 minut), średnio-trudne (między 15 a 30

minut) oraz trudne (powyżej 30 minut). Dla potraw trudnych mają być dołączane dodatkowe, praktyczne wskazówki wspomagające przyrządzenie (w postaci tekstu).

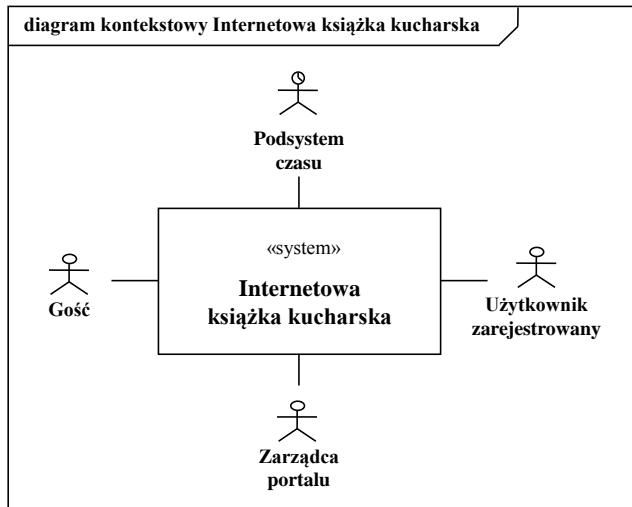
5. Zarejestrowani użytkownicy portalu (login, nick, email) mogą: dodawać nowe potrawy, a te wprowadzone przez siebie modyfikować (nawet więcej niż jeden raz), usuwać, dodawać nazwy regionalne dla potraw łącznie z opisami różnic oraz wprowadzać opinie do przepisów dostarczonych przez innych użytkowników. Powyższe czynności mają być rejestrowane w systemie (ma być wiadomo: kto, co i kiedy zrobił).
6. Każda z opinii dla danej potrawy ma zawierać ocenę użyteczności jej przepisu (w skali od 1 do 5); do opinii można także dołączyć komentarz w postaci tekstu. Zabronione jest opiniowanie potraw wprowadzonych przez siebie, ale można opiniować potrawy innych użytkowników, nawet wielokrotnie te same. Opinie użytkowników mają być uporządkowane w obrębie danej potrawy zgodnie z datami ich wprowadzenia.
7. Oczekuje się, że system będzie wspomagał użytkowników w realizowaniu funkcjonalności, takich jak np.:
  - 7.1 Wyszukanie potrawy po słowie kluczowym (każda osoba, czyli np. gość) z ewentualnym wprowadzeniem opinii o niej;
  - 7.2 Wprowadzenie opinii o potrawie;
  - 7.3 Podanie ogólnego opisu dla potrawy (na podstawie informacji zawartych w punkcie 2);
  - 7.4 Podanie szczegółowego opisu dla potrawy (na podstawie informacji zawartych w punktach 2, 3 i 4);
  - 7.5 Automatyczne usuwanie potraw, dla których średnia ocena użyteczności przepisu nie przekracza wartości 2 (przy minimum 20 wprowadzonych opiniach).

## 9.2 Analiza funkcjonalna

### 9.2.1 Diagram kontekstowy

*Polecenie:* Zbuduj diagram kontekstowy w oparciu o zamieszczony tekst wymagań.

Diagram kontekstowy, przedstawiający aktorów z otoczenia systemu *Internetowa książka kucharska*, wybranych na podstawie analizy tekstu wymagań, został zaprezentowany na Rys. 9-1.



Rys. 9-1 Diagram kontekstowy

### 9.2.2 Model przypadków użycia

*Polecenie:* Zbuduj model przypadków użycia, uwzględniając przede wszystkim funkcjonalność sugerowaną w ostatnim punkcie tekstu wymagań (tzn. w punkcie 7).

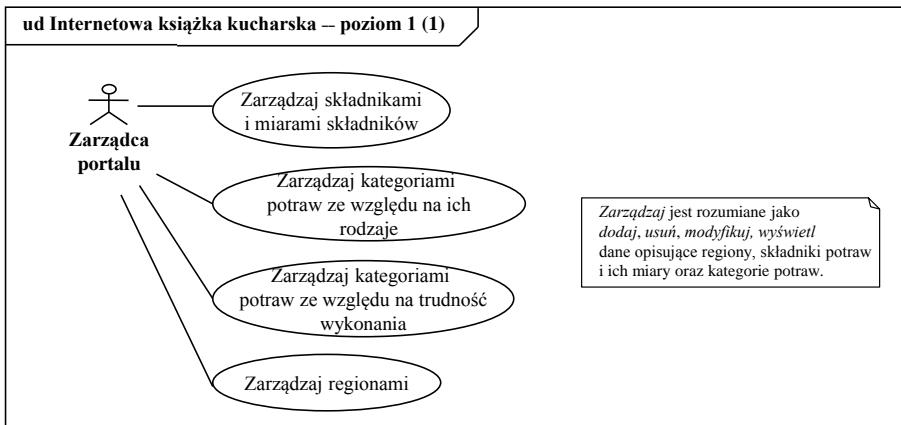
Uwaga: Diagram należy skonstruować z perspektywy aktorów z otoczenia systemu oraz z uwzględnieniem hierarchii aktorów i relacji pomiędzy przypadkami (o ile mają/mogłyby mieć miejsce).

Model przypadków użycia [35, 41, 43, 45] został przedstawiony za pomocą trzech diagramów na Rys. 9-2, Rys. 9-3 oraz Rys. 9-4. W nagłówkach oznaczono numer poziomu abstrakcji modelu, a w nawiasach podano numery jego kolejnych diagramów.

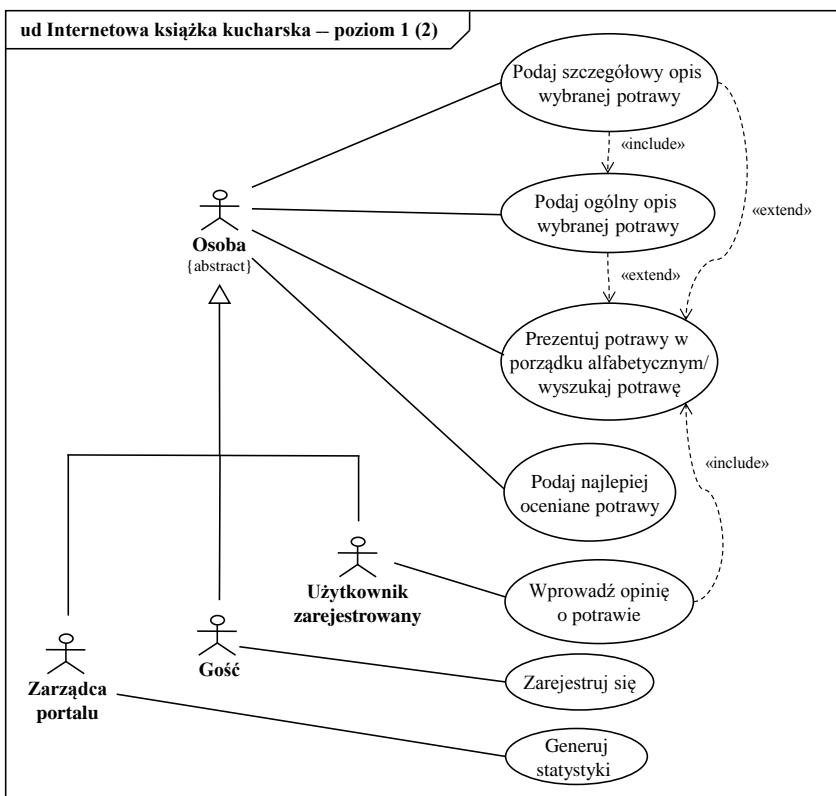
Rys. 9-2 zawiera przypadki użycia typu CRUD (Create, Read, Update, Delete), które na tym etapie analizy mogłyby zostać pominięte.

Diagramy przedstawione na Rys. 9-3 i Rys. 9-4 ilustrują proponowaną funkcjonalność systemu na podstawie punktu 7 tekstu wymagań.

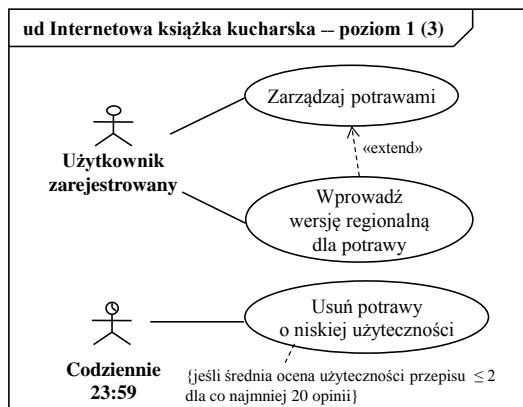
Należy pamiętać o tym, że w kolejnych krokach ten wstępny zbiór usług oferowanych przez system powinien podlegać dalszej weryfikacji, modyfikacji, rozszerzeniu itd.



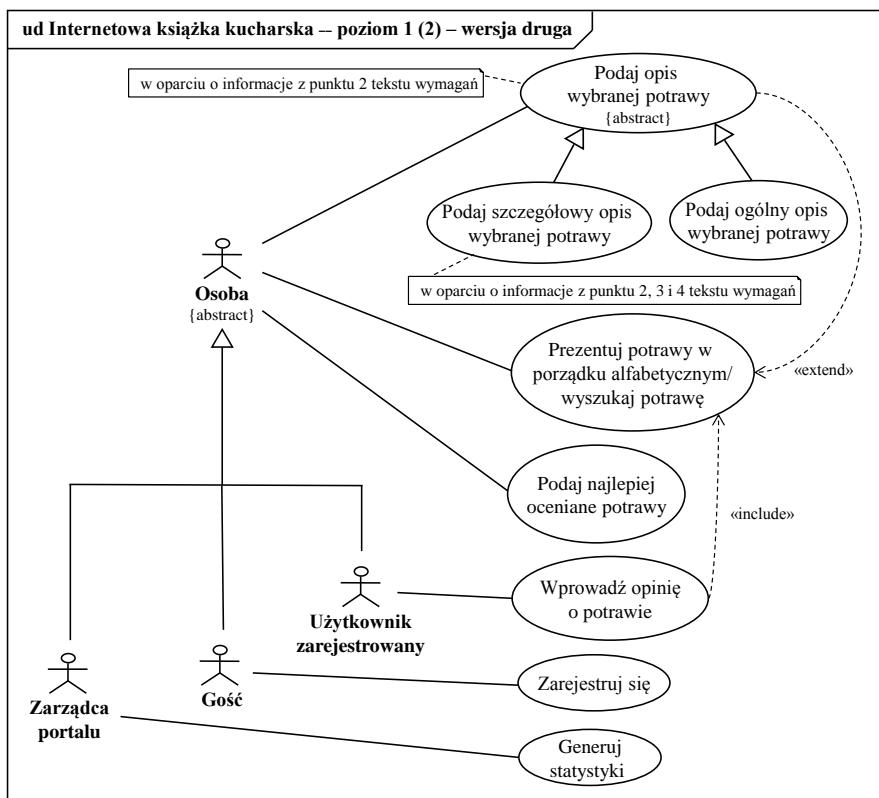
Rys. 9-2 Model przypadków użycia (diagram pierwszy)



Rys. 9-3 Model przypadków użycia (diagram drugi) – wersja pierwsza



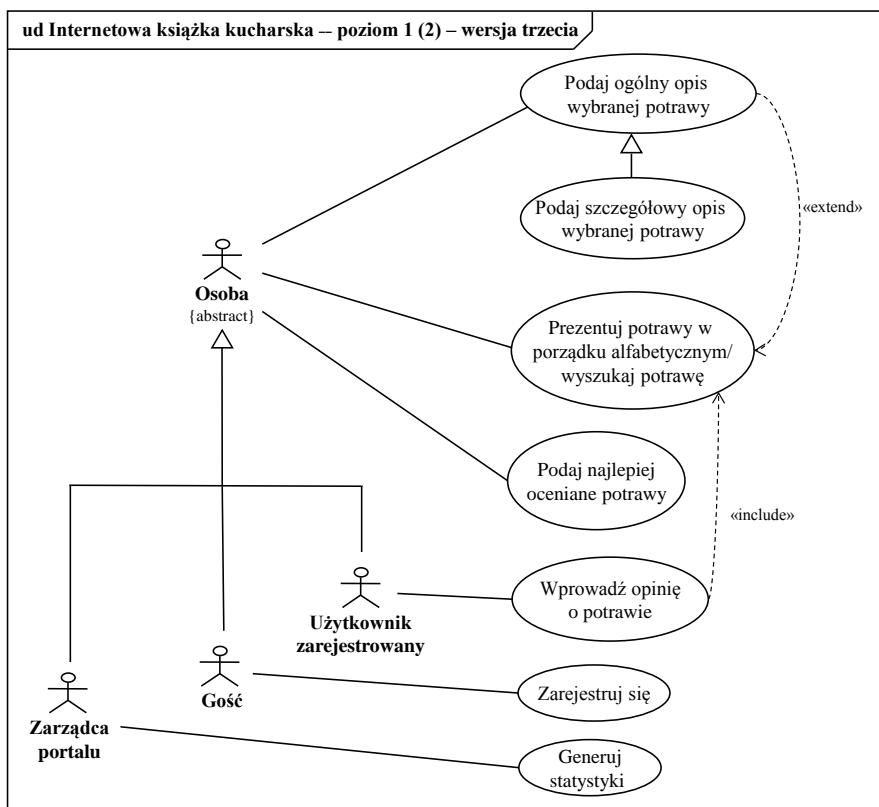
Rys. 9-4 Model przypadków użycia (diagram trzeci)



Rys. 9-5 Model przypadków użycia (diagram drugi) – wersja druga

Rys. 9-5 przedstawia alternatywne rozwiązanie dla drugiego diagramu przypadków użycia (z Rys. 9-3) z wykorzystaniem dziedziczenia między przypadkami. Przypadek użycia *Podaj opis wybranej potrawy* jest abstrakcyjny, co oznacza, że każda jego instancja pośrednia (przebieg, realizacja) jest wystąpieniem przypadku użycia *Podaj szczegółowy opis wybranej potrawy* lub *Podaj ogólny opis wybranej potrawy*.

Rys. 9-6 ilustruje trzecią wersję drugiego diagramu przypadków użycia z Rys. 9-3, na którym każdy przebieg przypadku użycia *Podaj szczegółowy opis wybranej potrawy* zawiera (oddziedziczony) przebieg przypadku użycia *Podaj ogólny opis wybranej potrawy*.



Rys. 9-6 Model przypadków użycia (diagram drugi) – wersja trzecia

### 9.2.3 Scenariusze dla wybranych przypadków użycia łącznie z ich podziałem na podprzypadki

*Polecenie:* Dla funkcjonalności *Podaj ogólny opis wybranej potrawy* (punkt 7.3 tekstu wymagań) napisz scenariusz.

Przykładowy scenariusz dla przypadku użycia *Podaj ogólny opis wybranej potrawy* (punkt 7.3 tekstu wymagań), w oparciu o wersję pierwotną diagramu drugiego modelu przypadków użycia z Rys. 9-3, został umieszczony w Tab. 9-1.

Tab. 9-1 Scenariusz dla przypadku użycia *Podaj ogólny opis potrawy*

<b>Nazwa przypadku użycia</b>	Podaj ogólny opis wybranej potrawy
<b>Warunek początkowy</b>	W systemie jest zarejestrowana co najmniej jedna potrawa.
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Aktor, którym jest tu jeden z przypadków użycia: <i>Podaj szczegółowy opis wybranej potrawy, Prezentuj potrawy w porządku alfabetycznym/wyszukaj potrawę lub Osoba uruchamia przypadek użycia Podaj ogólny opis wybranej potrawy.</i></li> <li>2. System wyświetla ogólne informacje o potrawie (słowa kluczowe, nazwa, składniki łącznie z ich ilościami, opis sposobu wykonania, czas przygotowania, liczbę porcji, zdjęcie, oraz potrawy, w skład których wchodzi i z których się składa).</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	<ol style="list-style-type: none"> <li>2a. Aktorem jest <i>Osoba</i>. System odpytuje o nazwę potrawy. Aktor wprowadza nazwę.</li> <li>2aa. Potrawa o podanej nazwie nie została odnalezione, system informuje o błędzie i odpytuje, czy kontynuować. Aktor potwierdza.</li> <li>2aaa. Aktor rezygnuje; koniec przypadku użycia.</li> </ol>
<b>Zakończenie</b>	Zgodnie ze scenariuszem
<b>Warunek końcowy</b>	Brak

*Polecenie:* Dla przypadku użycia *Wprowadź opinię o potrawie*:

- napisz scenariusz,
- zaproponuj podział tego przypadku na podprzypadki.

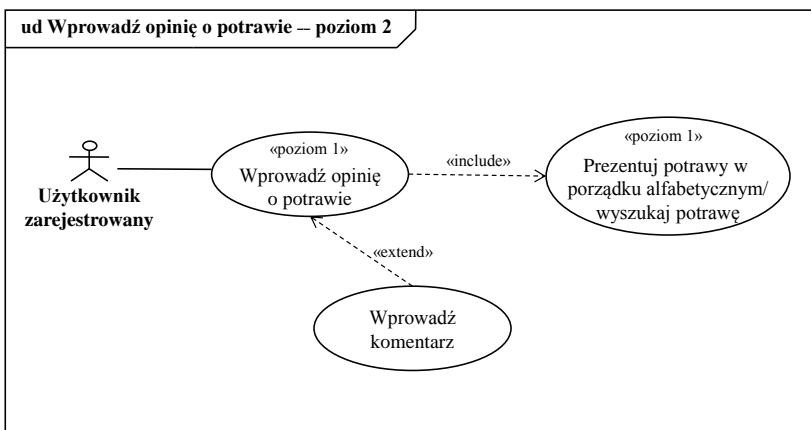
Przykładowy scenariusz dla przypadku użycia *Wprowadź opinię o potrawie* (punkt 7.2 tekstu wymagań) został umieszczony w Tab. 9-2.

Tab. 9-2 Przykładowy scenariusz dla przypadku użycia *Wrowadź opinię o potrawie* – wersja druga

<b>Nazwa przypadku użycia</b>	Wprowadź opinię o potrawie
<b>Warunek początkowy</b>	W systemie jest zarejestrowana co najmniej jedna potrawa.
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Przypadek użycia jest wywoływany przez aktora <i>Użytkownik zarejestrowany</i>.</li> <li>2. System wyświetla formularz z kryteriami wyszukiwania dla potrawy (np. słowa kluczowe i/lub nazwa). Aktor wprowadza kryteria.</li> <li>3. System wyświetla alfabetyczną listę potraw spełniających zadane kryteria. Aktor wybiera potrawę z listy.</li> <li>4. System odpytuje o ocenę użyteczności potrawy. Aktor wprowadza ocenę.</li> <li>5. System odpytuje o komentarz do opinii. Aktor wprowadza komentarz.</li> <li>6. System wyświetla komunikat informujący o zarejestrowaniu opinii dla danej potrawy.</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	<ol style="list-style-type: none"> <li>2a. Nie zadano żadnego kryterium, system wyświetla listę wszystkich potraw uporządkowaną alfabetycznie. System przechodzi do punktu 3 w przepływie głównym.</li> <li>3a. Aktor jest autorem przepisu bazowego, system informuje o tym i kończy przypadek użycia.</li> </ol>

	4a. Ocena użyteczności nie mieści się w przedziale [1, 5], system informuje o błędzie i ponownie odpytuje o ocenę użyteczności. 5a. Aktor rezygnuje z wprowadzenia komentarza.
<b>Zakończenie</b>	W dowolnym momencie.
<b>Warunek końcowy</b>	Dla wybranej potrawy zostanie zarejestrowane: kto wprowadził opinię, data jej wprowadzenia, ocena użyteczności oraz ewentualnie komentarz do opinii.

Przykładowy podział tego przypadku na podprzypadki, zgodny ze scenariuszem Tab. 9-2, został zilustrowany na Rys. 9-7.



Rys. 9-7 Diagram z przykładowym podziałem przypadku użycia *Wprowadź opinię o potrawie* (punkt 7.2 tekstu wymagań)

Druga wersja scenariusza dla przypadku użycia *Wprowadź opinię o potrawie* została przedstawiona w Tab. 9-3. Punkty 2 i 3 oraz 2a scenariusza z Tab. 9-2, określające przebieg przypadku użycia *Prezentuj potrawy w porządku alfabetycznym / wyszukaj potrawę* zostały ujęte za pomocą podprzepływu o takiej samej nazwie.

Podprzepływ *Prezentuj potrawy w porządku alfabetycznym / wyszukaj potrawę* (z Tab. 9-3) może być również bezpośrednio wywoływany przez aktora *Osoba* (jak na Rys. 9-3), dlatego lepszym rozwiązaniem wydaje się umieszczenie scenariusza podprzepływu za pomocą oddzielnej tabeli – jak dla oddzielnego przypadku użycia.

Tab. 9-3 Przykładowy scenariusz dla przypadku użycia *Wprowadź opinię o potrawie – wersja druga*

<b>Nazwa przypadku użycia</b>	Wprowadź opinię o potrawie
<b>Warunek początkowy</b>	W systemie jest zarejestrowana co najmniej jedna potrawa.
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Przypadek użycia jest wywoływany przez aktora <i>Użytkownik zarejestrowany</i>.</li> <li>2. Podprzepływ nazwany: <i>Prezentuj potrawy w porządku alfabetycznym / wyszukaj potrawę</i></li> <li>3. System odpytuje o ocenę użyteczności potrawy. Aktor wprowadza ocenę.</li> <li>4. System odpytuje o komentarz do opinii. Aktor wprowadza komentarz.</li> <li>5. System wyświetla komunikat informujący o zarejestrowaniu opinii dla danej potrawy.</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	<ol style="list-style-type: none"> <li>2a. Aktor jest autorem przepisu bazowego, system informuje o tym i kończy przypadek użycia.</li> <li>3a. Ocena użyteczności nie mieści się w przedziale [1, 5], system informuje o błędzie i ponownie odpytuje o ocenę użyteczności.</li> <li>4a. Aktor rezygnuje z wprowadzenia komentara.</li> </ol>
<b>Zakończenie</b>	W dowolnym momencie.
<b>Warunek końcowy</b>	Dla wybranej potrawy zostanie zarejestrowane: kto wprowadził opinię, data jej wprowadzenia, ocena użyteczności oraz ewentualnie komentarz do opinii.
<b>Podprzepływy nazwane</b>	
<b>Nazwa przypadku użycia</b>	<i>Prezentuj potrawy w porządku alfabetycznym / wyszukaj potrawę</i>
<b>Warunek początkowy</b>	W systemie jest zarejestrowana co najmniej jedna potrawa.

<b>Główny przepływ zdarzeń</b>	1. System wyświetla formularz z kryteriami wyszukiwania dla potrawy (np. słowa kluczowe i/lub nazwa). Aktor wprowadza kryteria. 2. System wyświetla alfabetyczną listę potraw spełniających zadane kryteria. Aktor wybiera potrawę z listy.
<b>Alternatywne przepływy zdarzeń</b>	1a. Nie zadano żadnego kryterium, system wyświetla listę wszystkich potraw uporządkowaną alfabetycznie. Aktor wybiera potrawę z listy. System przechodzi do punktu 3 w przepływie głównym.
<b>Zakończenie</b>	W dowolnym momencie.
<b>Warunek końcowy</b>	Brak.

#### 9.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

- Stosunkowo częstym błędem jest występowanie różnic między zbiorem aktorów określonych na diagramie kontekstowym a zbiorem aktorów umieszczonych na diagramach przypadków użycia. Oczywiście z tego porównania należy wyłączyć aktorów abstrakcyjnych.
- Do modelu są wprowadzani aktorzy będący częścią projektowanego systemu (jak np. *System sprawdzający*), co nie powinno mieć miejsca na rozważanym (najwyższym) poziomie abstrakcji, gdzie interesują nas wyłącznie użytkownicy z otoczenia systemu traktowanego jako całość.
- Warto zadbać o „dobre” nazwy dla przypadków, np. w postaci fraz sugerujących polecenia. Wydaje się, że np. zamiast nazwy takiej jak *system dodaje datę wprowadzenia opinii* lepiej byłoby użyć *Dodaj datę wprowadzenia opinii*.
- *Dla przypomnienia:* przypadki wewnętrzne powinny pojawiać się dopiero na kolejnym poziomie abstrakcji.
- W scenariuszach często są umieszczane sformułowania sugerujące bardziej opis algorytmu realizacji danego przypadku niż opis dialogu aktora z systemem w trakcie wykonania tego przypadku. Np. zamiast frazy *system rejestruje opinię*, lepiej jest wykorzystać frazę *system informuje o zarejestrowaniu opinii*. Pozwala to na wyraźne oddzielenie części dialogowej przypadku (służącej komunikowaniu się z aktorem w celu np. pobierania danych wejściowych, informowaniu o

błędach itp.) od jego części wykonawczej, której zadaniem może być, przykładowo, zapamiętanie danych pobranych od aktora w strukturze systemu.

- Warunek początkowy w postaci  *użytkownik nie jest twórcą przepisu* nie jest możliwy do sprawdzenia w sytuacji, gdy jesteśmy na początku realizowania przypadku użycia i nie jest wiadomo, o jaki przepis w ogóle chodzi – system dopiero w trakcie dialogu z użytkownikiem pozyskuje tę informację. Pozwala to na wyraźne oddzielenie części dialogowej przypadku (służącej komunikowaniu się z aktorem, np. w celu pobierania danych wejściowych, informowaniu o błędach itp.) od jego części wykonawczej, której zadaniem przykładowo może być zapamiętanie w systemie danych pobranych od aktora.
- Warunek początkowy sformułowany następująco:  *użytkownik postanawia dodać swoją opinię do potrawy* nie odnosi się do stanu systemu sprzed wywołania przypadku, ponadto, jak miałoby to być algorytmicznie sprawdzane? Tak określony warunek (jego postać) nie budzi skojarzenia z wyrażeniem logicznym.
- *Dla przypomnienia:* specyfikacja warunku końcowego jest konieczna, jeśli rozważany przypadek zmienia stan systemu.

## 9.3 Analiza strukturalna

### 9.3.1 Schemat pojęciowy

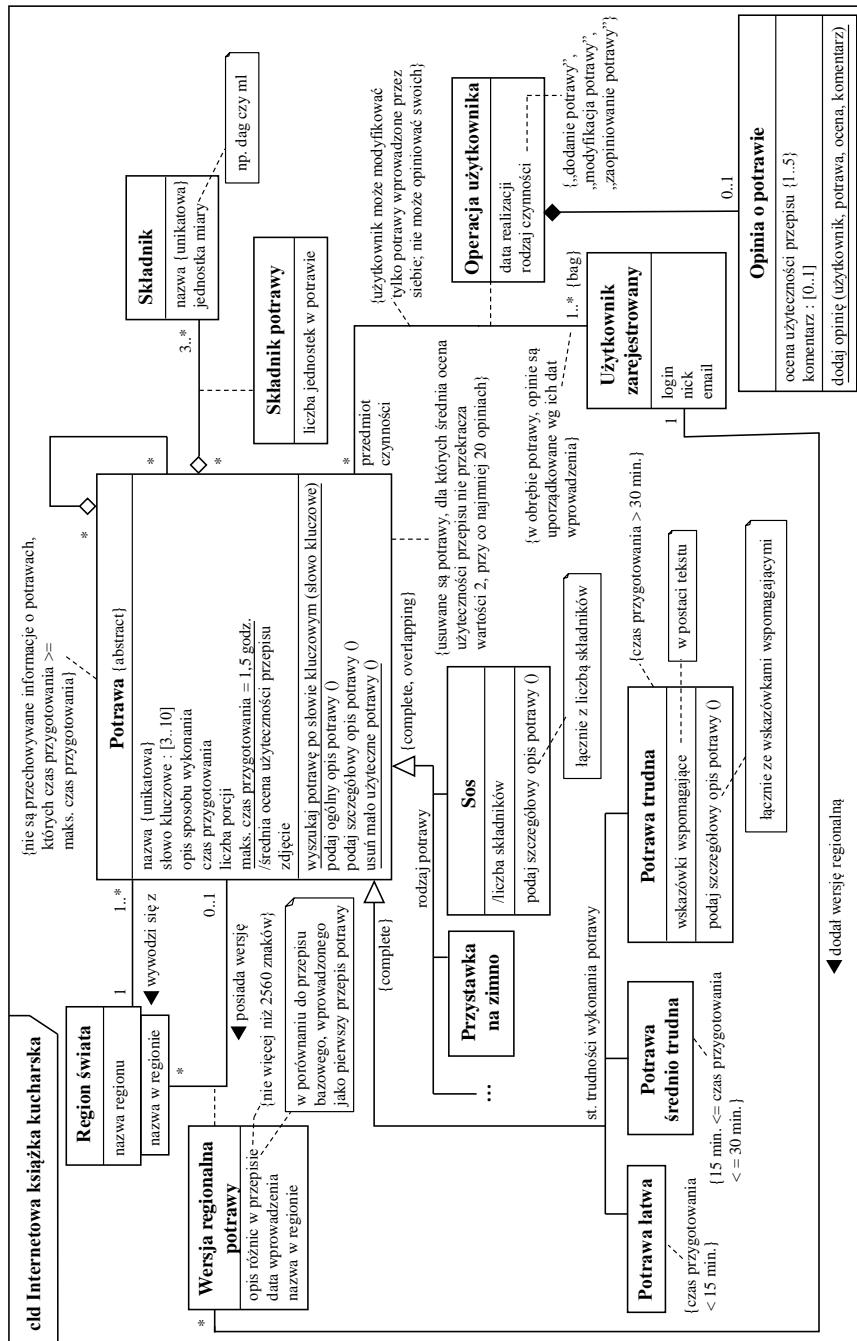
*Polecenie:* Dla podanego tekstu wymagań skonstruuj schemat pojęciowy (nie redukując liczności).

Schemat pojęciowy [36, 38, 41, 45] dla projektowanego systemu przedstawiono na Rys. 9-8.

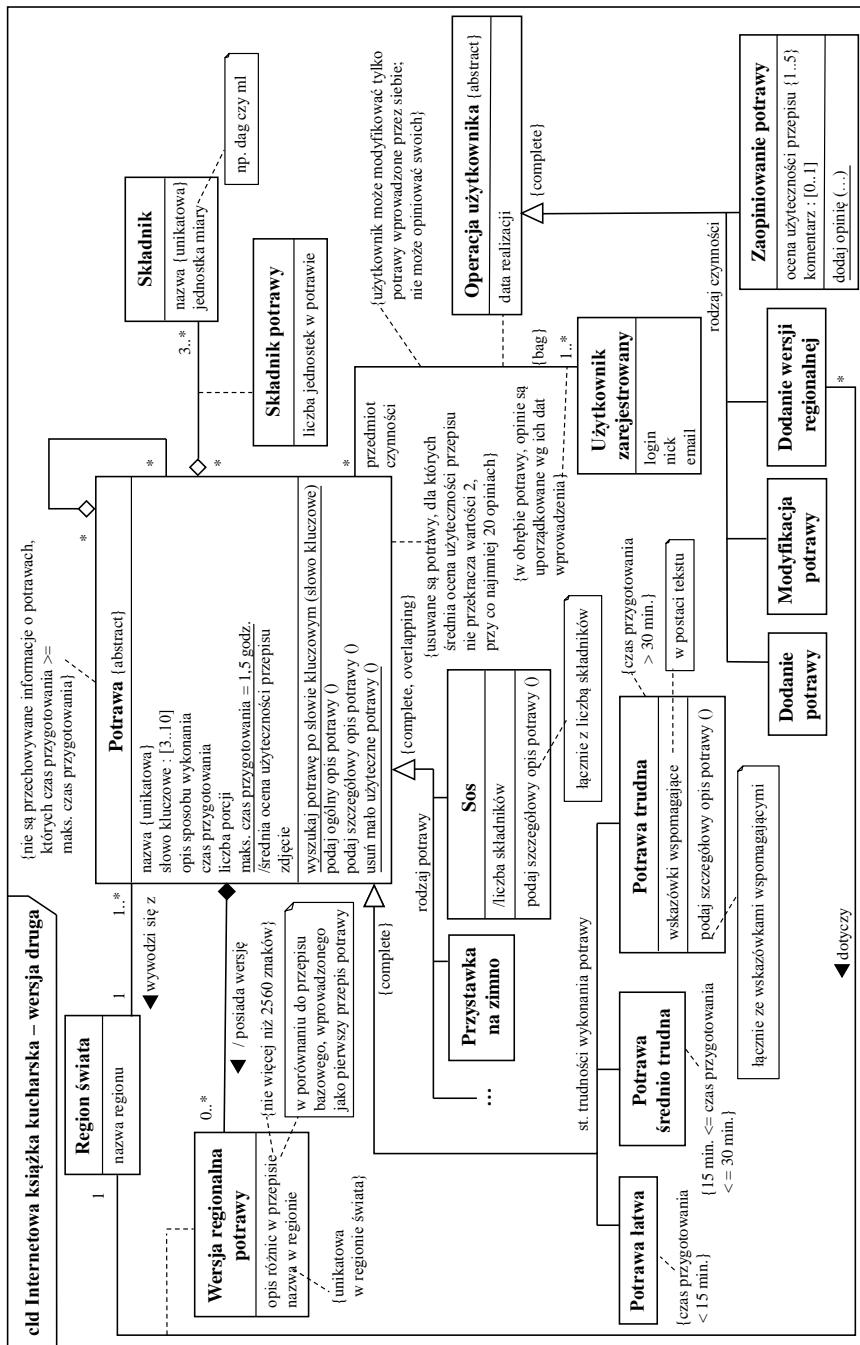
Wprowadzenie, modyfikacja, dodanie wersji regionalnej i dodanie opinii o potrawie są operacjami, które wykonuje użytkownik zarejestrowany na obiekcie klasy *Potrawa*, dlatego możliwe jest wprowadzenie hierarchii dla nich jak na Rys. 9-9. Warto zaznaczyć, że klasa *Wersja regionalna potrawy* w tym przypadku jest klasą asocjacji połączoną z klasą *Potrawa* za pomocą asocjacji pochodnej *posiada wersję* – wersja regionalna potrawy jest wprowadzana w wyniku pewnej operacji użytkownika na określonej potrawie.

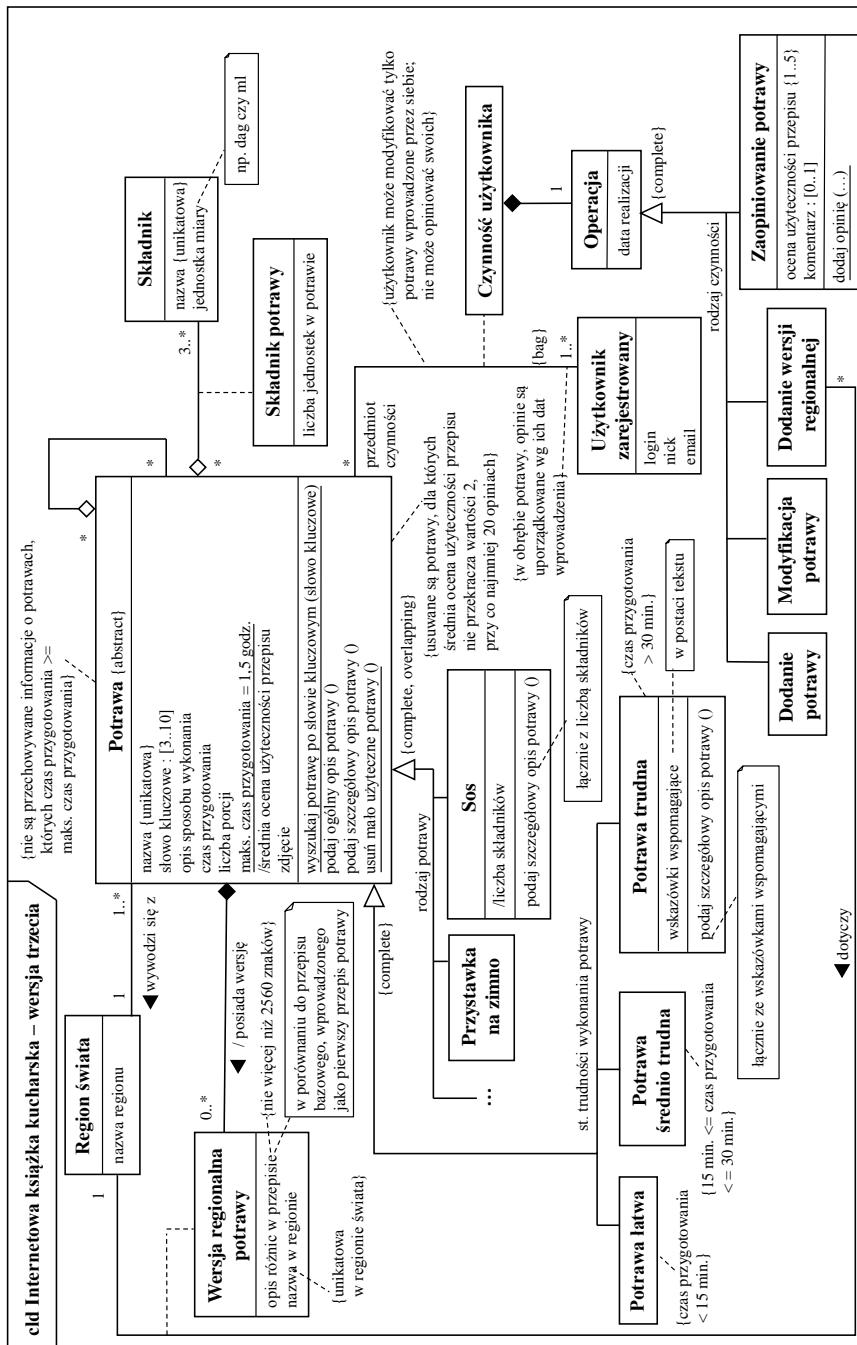
Ponieważ w specyfikacji notacji UML [36] niedozwolone jest wprowadzanie klas asocjacji, które są klasyfikatorami ogólnymi, na Rys. 9-10 zostało przedstawione techniczne rozwiązanie tego problemu.

W dalszej części analizy będzie rozważany schemat pojęciowy, który został zilustrowany na Rys. 9-8.



Rys. 9-8 Schemat pojęciowy dla systemu *Internetowa książka kucharska*

Rys. 9-9 Schemat pojęciowy dla systemu *Internetowa książka kucharska – wersja druga*

Rys. 9-10 Schemat pojęciowy systemu *Internetowa książka kucharska – wersja trzecia*

### 9.3.2 Analiza elementów pochodnych

Na schemacie pojęciowym z Rys. 9-8 można zidentyfikować dwa elementy pochodne:

- atrybut *średnia ocena użyteczności przepisu* w klasie *Potrawa*;
- atrybut *liczba składników* w klasie *Sos*.

Wartość atrybutu *średnia ocena użyteczności przepisu* w klasie *Potrawa* jest wyliczany na podstawie wartości atrybutów *ocena użyteczności przepisu* w odpowiednich obiektach klasy *Opinia o potrawie*. Zmiana wartości tego atrybutu ma miejsce tylko po wprowadzeniu kolejnej opinii dla danej potrawy, dlatego mimo że złożoność obliczeniowa metody wyliczającej jego wartość nie byłaby duża, to wydaje się, że zasadne będzie pozostawienie atrybutu *średnia ocena użyteczności przepisu* w klasie *Potrawa* jako atrybutu pochodnego, ponieważ potencjalnie może on być często wykorzystywany jako element składowy opisu potrawy.

Atrybut *liczba składników* w klasie *Sos* może zostać usunięty z tej klasy, wystarczy umieścić tylko metodę *podaj liczbę składników()* w klasie *Sos*, ponieważ wyliczenie wartości nie jest kosztowne obliczeniowo. Ze względu na nałożone ograniczenie (zgodnie z tekstem wymagań) na liczbę składników potrawy: *co najmniej 3 składniki dla każdej potrawy*, zrealizowane za pomocą liczności roli *Składnik* asocjacji między klasą *Potrawa* a klasą *Składnik*, wydaje się, że zaproponowana metoda powinna zostać przeniesiona do klasy *Potrawa*.

Schemat pojęciowy uwzględniający omówioną analizę elementów pochodnych został przedstawiony na Rys. 9-11.

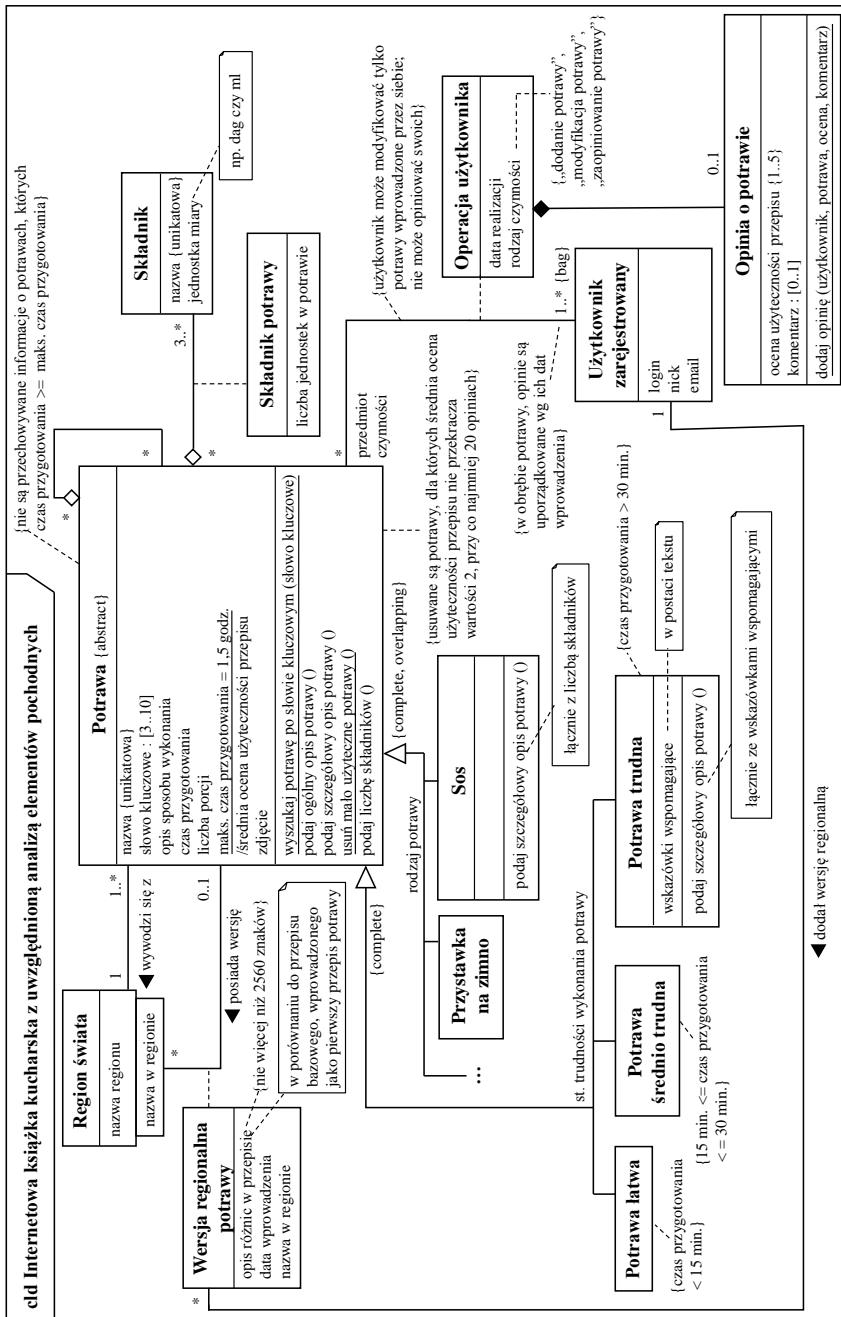
### 9.3.3 Analiza wartości początkowych

Na schemacie pojęciowym z Rys. 9-11 można określić wartość początkową *null* dla atrybutu pochodnego *średnia ocena użyteczności przepisu* w klasie *Potrawa*, jak na Rys. 9-12 (oprócz już zidentyfikowanej wartości początkowej równej *1,5 godz.* dla atrybutu klasowego *maks. czas przygotowania* w tej samej klasie).

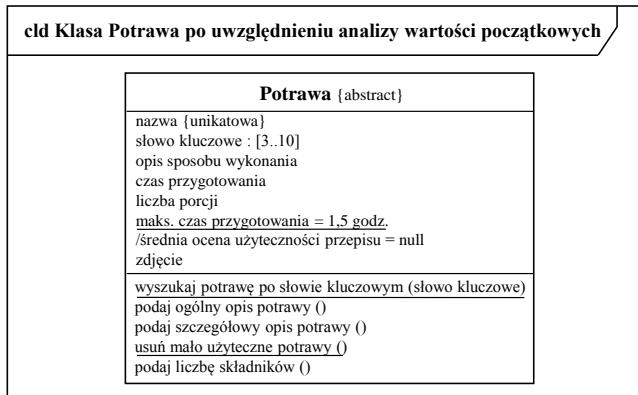
### 9.3.4 Analiza wartości granicznych

Podczas dalszej analizy schematu pojęciowego z Rys. 9-11 należy zidentyfikować wartości graniczne i rozważyć, czy podczas ewentualnej eksploatacji systemu (wraz z upływem czasu) możliwe byłyby ich zmiany bez ponownej komplikacji kodu.

Syntetyczne zestawienie wybranych wartości granicznych, powiązanych z nimi ograniczeń i propozycji zmian w schemacie pojęciowym przedstawiono w Tab. 9-4.



Rys. 9-11 Schemat pojęciowy *Internetowa książka kucharska z uwzględnioną analizą elementów pochodnych*

Rys. 9-12 Klasa *Potrawa* z uwzględnioną analizą wartości początkowych

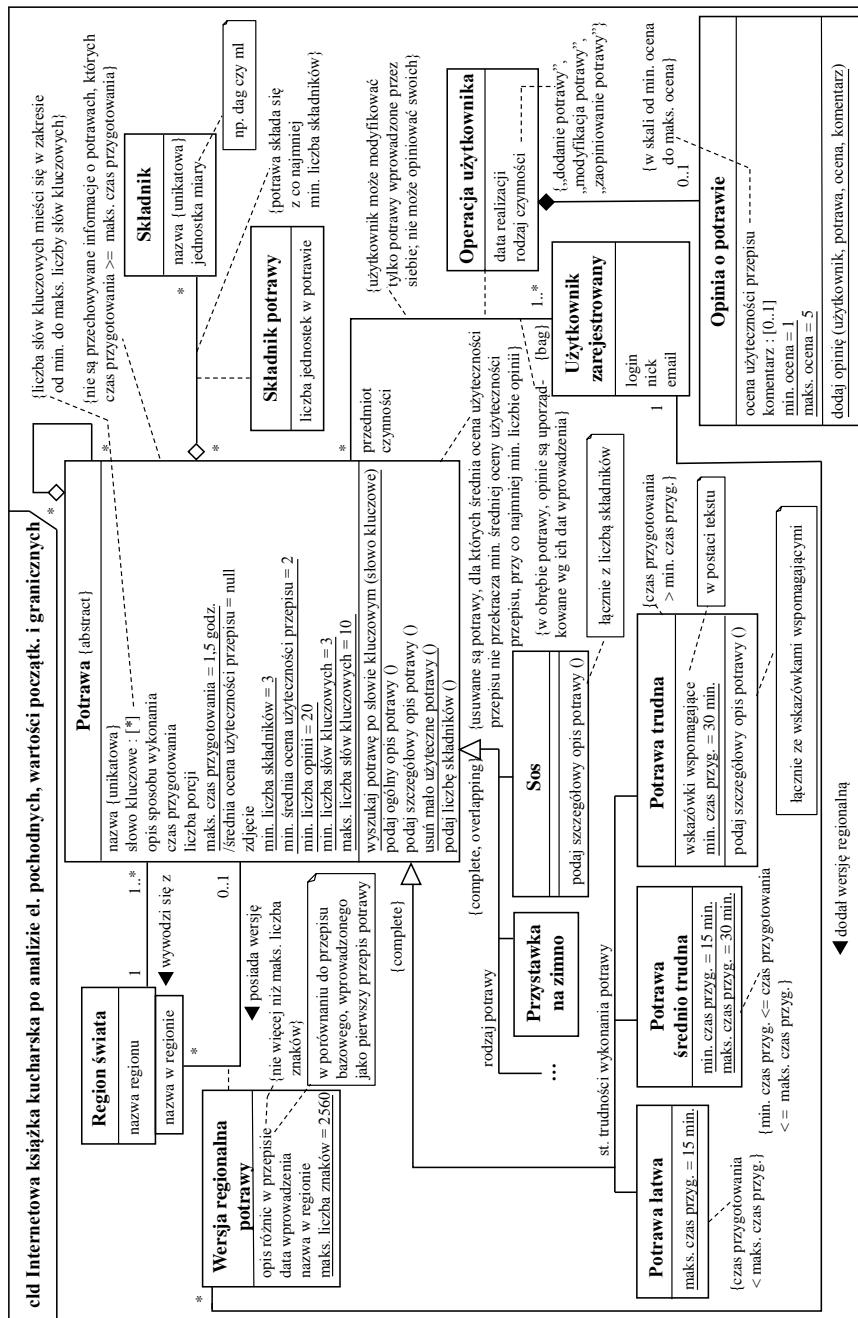
Tab. 9-4 Zestawienie wartości granicznych, ograniczeń i proponowanych zmian w schemacie pojęciowym

Wartość graniczna	Ograniczenie	Zmiany w schemacie pojęciowym
2560 znaków	$\{nie \; więcej \; niż \; 2560 \; znaków\}$ nałożone na atrybut <i>opis różnic w przepisie</i> w klasie asocjacji <i>Wersja regionalna potrawy</i>	dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością w klasie asocjacji <i>Wersja regionalna potrawy</i> : <u>maks. liczba znaków = 2560</u> zmiana ograniczenia do postaci $\{nie \; więcej \; niż \; maks. \; liczba \; znaków\}$
15 minut	$\{czas \; przygotowania \; < \; 15 \; min.\}$ nałożone na obiekt klasy <i>Potrawa łatwa</i>	dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością w klasie <i>Potrawa łatwa</i> : <u>maks. czas przyg. = 15 min.</u> zmiana ograniczenia do postaci $\{czas \; przygotowania \; < \; maks. \; czas \; przyg.\}$
15 minut, 30 minut	$\{15 \; min. \; \leq \; czas \; przygotowania \; \leq \; 30 \; min.\}$ nałożone na obiekt klasy	dodatkowe atrybuty o zasięgu klasowym z zainicjalizowanymi wartościami w klasie <i>Potrawa średnio trudna</i> : <u>min. czas przyg. = 15 min.</u>

	<i>Potrawa średnio trudna</i>	<u>maks. czas przyg. = 30 min.</u>  zmiana ograniczenia do postaci $\{min. \text{ czas przyg.} \leq \text{czas przygotowania} \leq \text{maks. czas przyg.}\}$
30 minut	{czas przygotowania > 30 min.}  nałożone na obiekt klasy <i>Potrawa trudna</i>	dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością w klasie <i>Potrawa trudna</i> : <u>min. czas przyg. = 30 min.</u>  zmiana ograniczenia do postaci $\{czas przygotowania > min. \text{ czas przyg.}\}$
3 składniki	liczność 3..* dla roli <i>Składnik</i> asocjacji między klasą <i>Potrawa</i> a klasą <i>Składnik</i>  nałożone na rolę <i>Składnik</i> asocjacji między klasą <i>Potrawa</i> a klasą <i>Składnik</i>	dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością w klasie <i>Potrawa</i> : <u>min. liczba składników = 3</u>  <u>pierwsze rozwiązanie (wykorzystane)</u> zmiana liczności do * dla roli <i>Składnik</i> asocjacji między klasą <i>Potrawa</i> a klasą <i>Składnik</i>  dodatkowe ograniczenie: $\{potrawa składa się z co najmniej min. liczba składników\}$ nałożone na asocjację między klasą <i>Potrawa</i> a klasą <i>Składnik</i>  <u>drugie rozwiązanie</u> Zmiana liczności na <i>min. liczba składników..*</i> dla roli <i>Składnik</i> asocjacji między klasą <i>Potrawa</i> a klasą <i>Składnik</i>  dodatkowe ograniczenie nie jest potrzebne
1,5 godziny	{nie są przechowywane informacje o potrawach, których czas przygotowania >= maks. czas przy-	bez zmian, ponieważ w klasie <i>Potrawa</i> jest już atrybut klasowy <u>maks. czas przygotowania</u> oraz odpowiednie ograniczenie

	<i>gotowania}</i> nałożone na obiekt klasy <i>Potrawa</i>	
1-5, ocena użyteczności przepisu	{1..5}  nałożone na atrybut <i>ocena użyteczności przepisu</i> w klasie <i>Opinia o potrawie</i>	dodatkowe atrybuty klasowe z zainicjalizowanymi wartościami w klasie <i>Opinia o potrawie</i> : <u>min. ocena = 1</u> <u>maks. ocena = 5</u>  zmiana ograniczenia do postaci {w skali od min. ocena do maks. ocena}
2, średnia ocena użyteczności przepisu  20 opinii o potrawie	{usuwanie są potrawy, dla których średnia ocena użyteczności przepisu nie przekracza wartości 2, przy co najmniej 20 opiniach}  nałożone na obiekt <i>Potrawa</i>	dodatkowe atrybuty klasowe z zainicjalizowanymi wartościami w klasie <i>Potrawa</i> : <u>min. średnia ocena użyteczności przepisu = 2</u> <u>min. liczba opinii = 20</u>  zmiana ograniczenia do postaci {usuwanie są potrawy, dla których średnia ocena użyteczności przepisu nie przekracza min. średniej oceny użyteczności przepisu, przy co najmniej min. liczbie opinii}
3-10 słów kluczowych	liczność [3..10] atrybutu powtarzalnego <i>słowo kluczowe</i> w klasie <i>Potrawa</i>	dodatkowe atrybuty klasowe z zainicjalizowanymi wartościami w klasie <i>Potrawa</i> : <u>min. liczba słów kluczowych = 3</u> <u>maks. liczba słów kluczowych = 10</u>  dodatkowe ograniczenie: {liczba słów kluczowych mieści się w zakresie od min. do maks. liczby słów kluczowych} nałożone na atrybut powtarzalny <i>słowo kluczowe</i> w klasie <i>Potrawa</i>

Schemat pojęciowy uwzględniający proponowane zmiany w wyniku analizy wartości granicznych został zaprezentowany na Rys. 9-13.

Rys. 9-13 Schemat pojęciowy systemu *Internetowa książka kucharska* po analizie elementów pochodnych, wartości początkowych i wartości granicznych

### 9.3.5 Zadania podstawowe

W celu realizacji zadań zamieszczonych poniżej został wykorzystany schemat pojęciowy przedstawiony na Rys. 9-13.

**Zad. 1:** Podkreśl te rodzaje dziedziczenia (rozłączne, nierozłączne, kompletne, niekompletne, jednokrotne, wielokrotne, jednoaspektowe, wieloaspektowe, dynamiczne, elipsa), które według ciebie były potrzebne do wykorzystania na schemacie pojęciowym. Dla każdego podkreślonego rodzaju dziedziczenia podaj hierarchię/aspekt, w której ten rodzaj dziedziczenia wystąpił.

- Rozłączne (ang. *disjoint*): hierarchia dla potraw aspekt *st. trudności wykonania potrawy*;
- nierozłączne (ang. *overlapping*): hierarchia dla potraw aspekt *rodzaj potrawy*;
- kompletne: oba aspekty;
- jednokrotne: oba aspekty;
- dwuaspektowe: hierarchia dla potraw;
- elipsa: hierarchia dla potraw, aspekt *rodzaj potrawy*.

**Zad. 2:** Podaj przykłady metody abstrakcyjnej i metody zaimplementowanej w klasach abstrakcyjnych i konkretnych ze schematu pojęciowego.

- Metoda abstrakcyjna w klasie abstrakcyjnej: do schematu pojęciowego nie wprowadzono metody abstrakcyjnej, ale można by umieścić taką metodę w klasie abstrakcyjnej *Potrawa*, na przykład metodę *podaj szczegółowy opis potrawy ()* – wtedy należałoby umieścić też odpowiednie metody zaimplementowane we wszystkich podklasach klasy *Potrawa* (co nie byłoby najlepszym rozwiązaniem);
- metoda abstrakcyjna w klasie konkretnej: w klasie konkretnej nie można umieszczać metod abstrakcyjnych;
- metoda zaimplementowana w klasie abstrakcyjnej: *podaj ogólny opis potrawy* w klasie *Potrawa*;
- metoda zaimplementowana w klasie konkretnej: *podaj szczegółowy opis potrawy ()* w klasie *Sos*.

**Zad. 3:** Podaj przykłady metod obiektowych i klasowych oraz atrybutów obiektowych i klasowych.

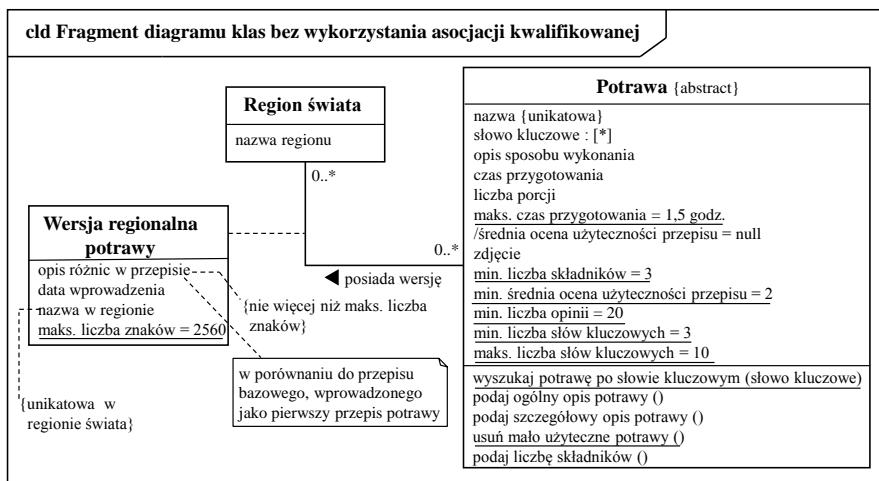
- Metoda obiektowa: *podaj ogólny opis potrawy ()* w klasie *Potrawa*;
- metoda klasowa: usuń mało użyteczne potrawy () (operująca na ekstencji klasy *Potrawa*) lub metoda dodaj opinię (użytkownik, potrawa, ocena, komentarz) (tworząca nowy obiekt klasy *Opinia o potrawie*);
- atrybut obiektowy: *nazwa* w klasie *Potrawa*;
- atrybut klasowy: maks. czas przygotowania w klasie *Potrawa* (zainicjalizowaną wartością 1,5 godz.).

**Zad. 4:** Wskaż przykład wykorzystania polimorfizmu metod.

Metody polimorficzne to metody *podaj szczegółowy opis potrawy ()* w klasach hierarchii zbudowanej dla potraw.

**Zad. 5:** Asocjację ze schematu pojęciowego, posiadającą niesymetryczne liczności oraz atrybut (lub klasę asocjacji), zamień na asocjację kwalifikowaną. Uzasadnij, dlaczego poddałeś zamianie właśnie tę asocjację.

Asocjacja kwalifikowana została umieszczona między klasami *Region świata* i *Potrawa*. Kwalifikatorem jest *nazwa w regionie* (potrawy), unikatowa w obrębie regionu, która może zostać użyta jako klucz do przeszukiwania zbioru potraw, dla których istnieją ich wersje regionalne w danym regionie świata. Rys. 9-14 przedstawia fragment diagramu sprzed zamiany asocjacji „zwykłej” na asocjację kwalifikowaną. Klucz *nazwa w regionie* można też wykorzystać do przeszukiwania zbioru wersji regionalnych w obrębie danej potrawy, ale siła takiego klucza byłaby mniejsza, gdyż dla danej potrawy mogą istnieć jej wersje regionalne o takich samych nazwach, ale pochodzących z różnych regionów świata.



Rys. 9-14 Fragment diagramu bez wykorzystania asocjacji kwalifikowanej

**Zad. 6:** Wskaż na schemacie pojęciowym asocjację, która jest agregacją lub kompozycją (lub może kwalifikować się do zamiany na takową). Objasnij, dlaczego zdecydowałeś się na wybór agregacji/kompozycji, a nie „zwykłej” asocjacji? Ponadto, dlaczego wybrałeś agregację, a nie kompozycję (czy też odwrotnie)?

Na diagramie występuje agregacja pomiędzy klasami: *Potrawa* i *Składnik*, która podkreśla – mocniej niż zwykła asocjacja – związek część-całość, występujący pomiędzy potrawą a składnikami wykorzystywanyimi do jej przygotowania.

Wykorzystano agregację, a nie kompozycję, ponieważ w tym przypadku cykl życiowy części nie zawiera się w cyklu życiowym całości, tzn. usunięcie z systemu obiektu klasy *Potrawa* nie będzie skutkowało usunięciem tych obiektów klasy *Składnik*, które wchodziły w skład potrawy. Dany składnik może być wykorzystany do przygotowania opisu więcej niż jednej potrawy.

**Zad. 7:** Wskaż na schemacie pojęciowym przykładowe ograniczenie. Na jaki element modelu zostało ono nałożone? Czy jest to ograniczenie statyczne czy dynamiczne? Zaproponuj, jaka metoda i w jakiej klasie mogłaby być odpowiedzialna za jego zapewnienie.

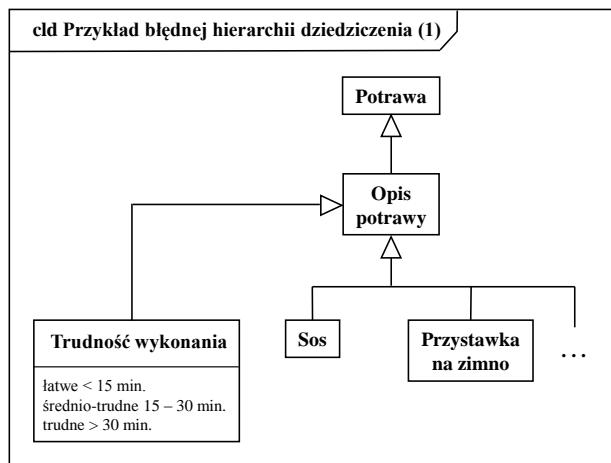
Na przykład ograniczenie, że potrawy łatwe to takie, dla których czas przygotowania jest krótszy niż *maks. czas przyg.* (obecnie 15 minut). Jest to ograniczenie statyczne na istnienie obiektu i implementowane w metodzie tworzącej nowy obiekt klasy *Potrawa łatwa*.

### 9.3.6 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

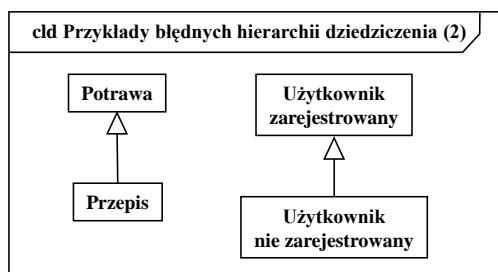
- Nazwa klasy to zazwyczaj rzeczownik w liczbie pojedynczej, co oznacza że jeżeli pojedynczy obiekt przechowuje informację o jednej potrawie, to klasa nazywa się *Potrawa*, ale gdyby pojedynczy obiekt miał przechowywać informacje o zbiorze potraw, to w takim przypadku klasa nazywałaby się *Potrawy*.
- Nazwa nadklasy często jest wykorzystywana do specyfikowania nazwy podklasy i taka sytuacja miała w tym zadaniu miejsce – czyli klasa *Potrawa łatwa* zamiast klasy *Łatwa*, co ma znaczenie wszędzie tam, gdzie wykorzystanie klasy o nazwie *Łatwa* niekoniecznie musiałoby kojarzyć się z potrawą.
- Należy dobrze rozumieć, co opisuje jeden obiekt danej klasy: czy opisuje był pełniący rolę klasyfikatora czy może jedną instancję (egzemplarz) danego klasyfikatora. W tym konkretnie zadaniu jeden obiekt klasy *Potrawa* opisywał jeden egzemplarz – jedną, konkretną potrawę.
- Hierarchia dla osób, w tym klasa *Gość* nie były potrzebne, ponieważ na tym etapie modelowania nie wprowadzamy informacji o aktorach do systemu.
- Często pojawiały się też problemy z konstruowaniem struktur generalizacji/specjalizacji, co jest ważne, ponieważ ta umiejętność należy do najważniejszych w modelowaniu obiektowym. Rys. 9-15, Rys. 9-16, Rys. 9-17 i Rys. 9-18 pokazują przykłady błędnych hierarchii.

Po pierwsze, hierarchie muszą być poprawne semantycznie, co można weryfikować, posiliając się wyrażeniem sformułowanym, jak następuje: „*nazwa podklasy* jest rodzajem *nazwa nadklasy*”. Co oznacza, że nie należy zapominać

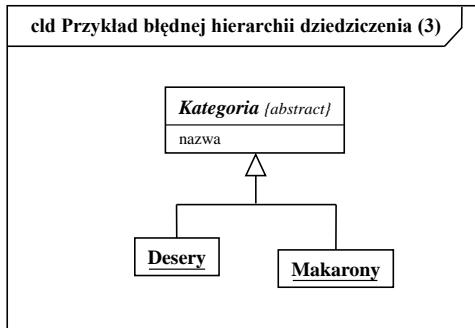
o tym, że obiekt podklasy (czyli wystąpienie bezpośrednie podklasy) jest jednocześnie wystąpieniem pośrednim wszystkich nadklas rozważanej podklasy. Jeśli wyrażenie ma wartość logiczną *true*, to dana konstrukcja jest prawidłowa. Dla przykładu z Rys. 9-15, czy *opis potrawy* jest rodzajem *potrawy* albo *sos* rodzajem *opisu potrawy*? Podobnie, czy *trudność wykonania* jest rodzajem *opisu potrawy*, czy też jest jednym z elementów składowych *opisu potrawy*? Odnosząc się do Rys. 9-16, *przepis* jest opisem (zbiorem) własności *potrawy*, a nie jednym z rodzajów *potrawy*, podobnie *użytkownik niezarejestrowany* nie jest rodzajem *użytkownika zarejestrowanego*. Przykłady, jak widać na zamieszczonych tu rysunkach, można mnożyć. Ale występują też i inne błędy: nieoznaczone rodzaje dziedziczenia (*dla przypomnienia*: umieszcza się je przy symbolu dziedziczenia, czyli przy trójkącie), nieoznaczone aspekty dla dziedziczenia wieloaspektowego, błędnie oznaczone atrybuty (np. jakie wartości miałby przechowywać atrybut o nazwie *latwe*?), ograniczenia nieumieszczone w nawiasach klamrowych, podkreślone nazwy klas (?) czy wprowadzanie informacji redundantnej (nazwa klasy napisana kursywą łącznie z etykietą *{abstract}*), jak na Rys. 9-17).



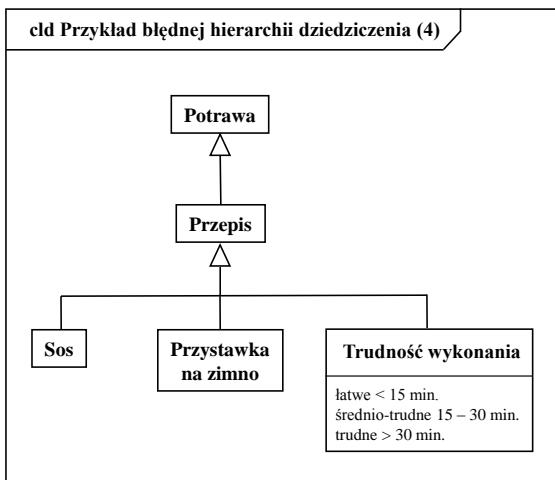
Rys. 9-15 Przykład błędnej hierarchii dziedziczenia



Rys. 9-16 Przykłady błędnych hierarchii dziedziczenia



Rys. 9-17 Przykład błędnej hierarchii dziedziczenia



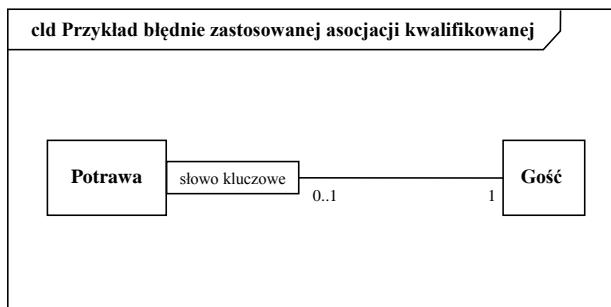
Rys. 9-18 Przykład błędnej hierarchii dziedziczenia

- Niepoprawne jest też wprowadzanie do diagramów informacji, które ani bezpośrednio, ani pośrednio nie zostały umieszczone w tekście wymagań, jak np. *id* dla potraw, czy potrawy takie jak desery lub makarony (ponadto z określona błędna nazwą – w liczbie mnogiej).
- Stosunkowo często są błędnie (lub w ogóle) oznaczane rodzaje atrybutów, np. atrybut *liczba porcji* nie jest atrybutem pochodnym, ponieważ nie określono żadnego sposobu na jego wyliczanie, podaje go arbitralnie autor przepisu. Z kolei atrybut *liczba składników* jest pochodny, ponieważ została wprowadzona asocjacja pomiędzy klasami *Potrawa* i *Składnik*. W czasie run-time'u wystąpienia pośrednie klasy *Potrawa* będą powiązane z odpowiednimi obiektami klasy *Składnik* – czyli będzie możliwe wyliczenie, ile składników

zawiera każdy z sosów. *Dla przypomnienia:* atrybut, którego wartość da się wyliczyć/wyprowadzić w każdym momencie na podstawie innych danych przechowywanych w systemie (tzw. danych bazowych, nieoznaczonych ukośnikiem), jest atrybutem pochodnym. Liczba składników (nie lista składników) jest atrybutem prostym (opisywanym jedną wartością), nie można więc wprowadzać go do diagramu w postaci: *liczba składników : [\*/]* lub *liczba składników : [3..\*]*.

- *Dla przypomnienia:* metody należy umieszczać nie w miejscu potencjalnego wywołania, ale w tej klasie, której obiekty będą przechowywały dane potrzebne do wykonania metody, czyli np. metodę *dodaj opinię ()* wprowadzamy nie do klasy *Użytkownik zarejestrowany*, ale do klasy *Opinia*.
- *Dla przypomnienia:* pojedynczy obiekt usuwamy, wykorzystując metodę obiektową, natomiast grupę obiektów usuwamy metodą klasową (operacja wykonywana jest wtedy na ekstensji klasy).
- Należy zauważać także różnicę pomiędzy wyrażeniami, takimi jak: *ocena użyteczności potrawy : [1..5]* a *ocena użyteczności potrawy {1..5}*. Pierwsze oznacza, że ocena jest atrybutem złożonym i może być opisywana przez od 1 do 5 wartości (które też potencjalnie mogą być złożone). Z kolei drugie wyrażenie oznacza, że ocena jest atrybutem prostym, opisywanym przez jedną wartość z przedziału [1..5].
- Ograniczenie o treści: *nie są przechowywane informacje o potrawach, których czas przygotowania >= maks. czas przygotowania* (obecnie 1,5 godz.) nie jest ograniczeniem na wartość atrybutu *czas przygotowania*, ale jest ograniczeniem na istnienie obiektu – nie będą rejestrowane potrawy (w systemie są one opisywane przez obiekty, będące wystąpieniami klas wchodzących w skład hierarchii dla potraw), dla których czas przygotowania jest większy od akceptowalnego najdłuższego czasu przygotowania.
- Bardzo często występują błędy w konstruowaniu asocjacji kwalifikowanej, jak na Rys. 9-19:
  - (1) klasa *Gość* w ogóle nie powinna zostać wprowadzona na tym etapie do diagramu (*Gość* jest aktorem z otoczenia systemu);
  - (2) klasę *Gość* połączono asocjacją bez nazwy z klasą *Potrawa* – połączenie klas asocjacją oznacza konieczność wprowadzania informacji strukturalnych, tzn., że operacja, zrealizowana na obiekcie którejś z podklas klasy *Potrawa* przez obiekt klasy *Gość*, ma zostać zarejestrowana w systemie (co jest niezgodne z tekstem wymagań);
  - (3) wygląda na to (sądząc po liczności roli asocjacji *przedmiot czynności*, przy kwalifikatorze, równej 0..1), że każdy obiekt klasy *Gość* może zrealizować co najwyżej jedną operację na obiekcie którejś z podklas klasy *Potrawa* (na dodatek, w efekcie braku nazwy asocjacji, nie wiadomo co to za operacja);

(4) przeszukiwanie zbioru gości powiązanych z danym obiektem podklasy klasy *Potrawa* (powiązanych dzięki rejestrowaniu wszystkich operacji wykonanych na tym obiekcie przez obiekty klasy *Gość*), za pomocą kwalifikatora (którym jest tu *słowo kluczowe*), zawsze zwraca jednego gościa (liczność końca asocjacji naprzeciwko kwalifikatora równa jest 1) – oznacza to, że tylko jeden z członków ww. zbioru gości mógł wykorzystać dane słowo kluczowe.



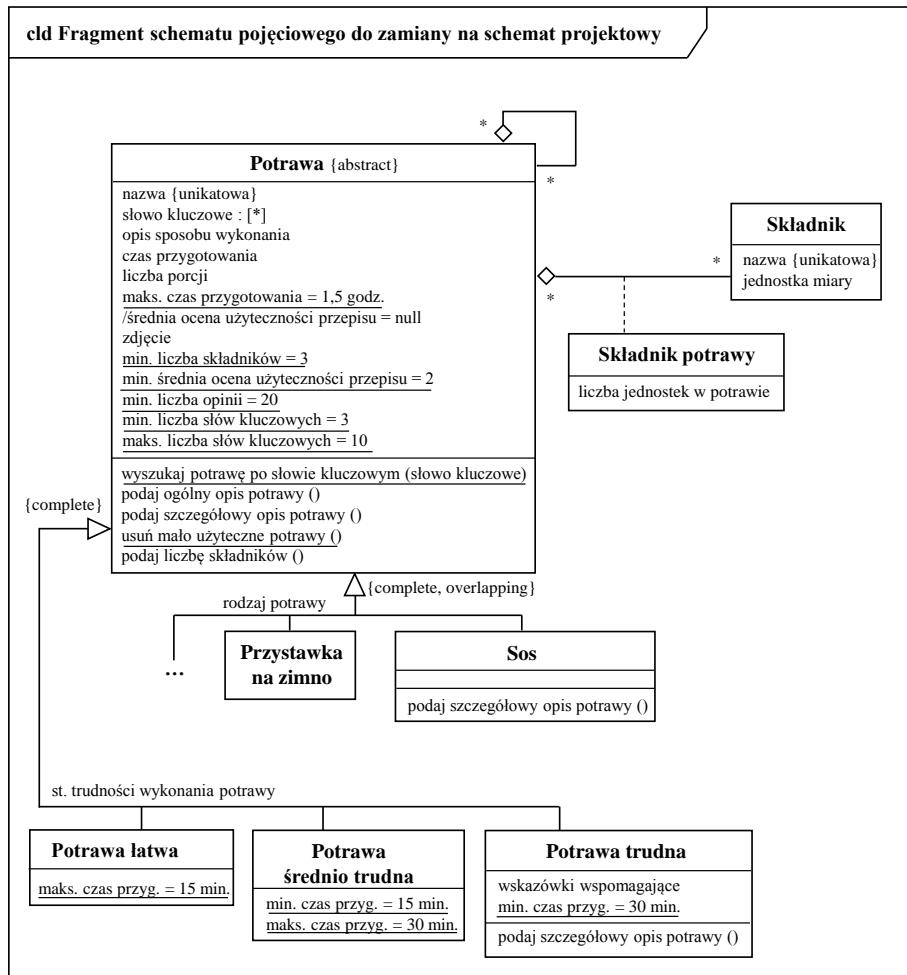
Rys. 9-19 Błędna konstrukcja dla asocjacji kwalifikowanej

### 9.3.7 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji

Rys. 9-20 ilustruje fragment schematu pojęciowego (dla czytelności: z pominięciem ograniczeń i komentarzy), niezbędny dla zamodelowania za pomocą diagramu interakcji metody *Podaj ogólny opis wybranej potrawy* (Zad. 2 w rozdziale 9.4.1 ).

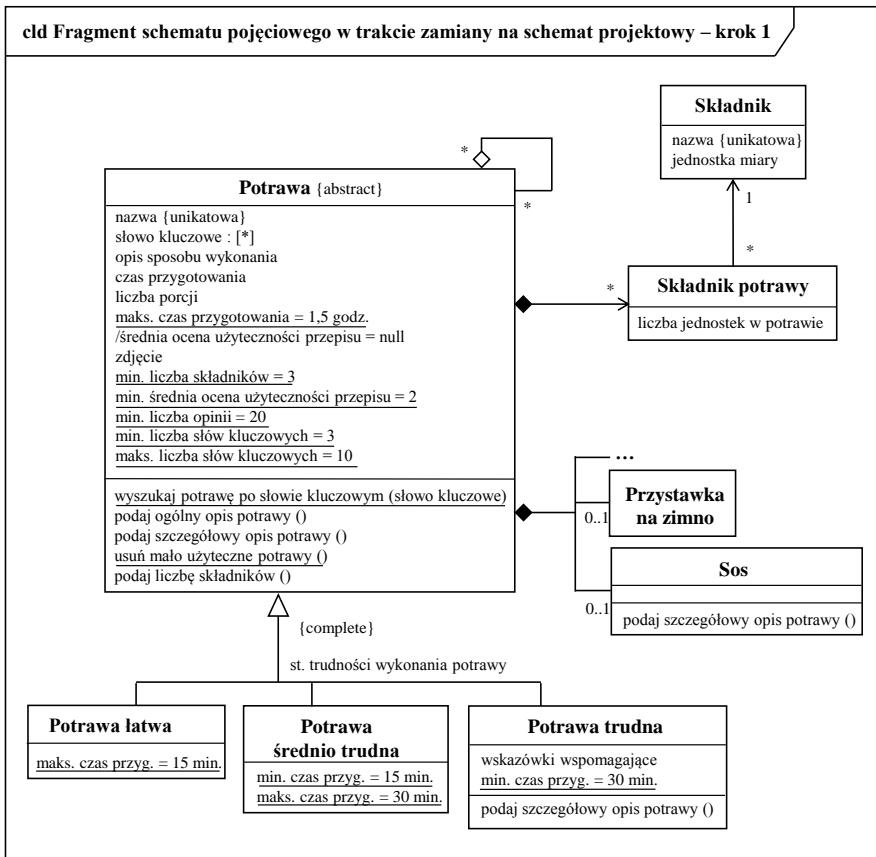
Fragment ten zawiera hierarchię potraw, agregację rekurencyjną między potrawami (składowymi) oraz agregację między klasą *Potrawa* a klasą *Składnik* z klasą asocjacji *Składnik potrawy*.

W typowym języku obiektowym nie są zdefiniowane środki do bezpośredniego implementowania ani dziedziczenia wieloaspektowego, ani dziedziczenia nierożłącznego, ani klasy asocjacji [37, 40] i dlatego w pierwszym kroku fragment hierarchii dziedziczenia dla aspektu *rodzaj potrawy* zostanie zastąpiony z użyciem kompozycji, a klasa asocjacji – zastąpiona przez odpowiednią klasę pośredniczącą. Ewentualne zwinięcie aspektu *rodzaj potrawy* do nadklasy (czyli klasy *Potrawa*) spowodowałoby konieczność „sklejenia” metod polimorficznych o nazwie *podaj szczegółowy opis potrawy()*, podczas gdy transformacja tego fragmentu schematu pojęciowego z wykorzystaniem kompozycji umożliwia zachowanie oddzielnych implementacji metod w klasach *Potrawa*, *Sos*, *Potrawa trudna* itd.



Rys. 9-20 Fragment schematu pojęciowego do zamiany na schemat projektowy

Warto zauważyć, że z tekstu wymagań nie wynika potrzeba nawigacji od składników do potraw, w których są one wykorzystane. Inaczej jest w przypadku agregacji rekurencyjnej dla klasy *Potrawa*, chcemy wiedzieć, z jakich potraw składa dana potrawa i jakich potraw sama jest składnikiem. Dlatego przyjęto na schemacie projektowym, że agregacja wymieniona jako pierwsza powinna zostać zaimplementowana jako asocjacja skierowana, zaś druga z nich – jako asocjacja dwukierunkowa. Kompozycja między klasą *Potrawa* a klasami *Sos*, *Przystawka na zimno* itd., powstała w wyniku transformacji aspektu *rodzaj potrawy*, wymaga nawigacji dwukierunkowej. Fragment schematu pojęciowego po wykonaniu kroku pierwszego zamiany na schemat projektowy został przedstawiony na Rys. 9-21.

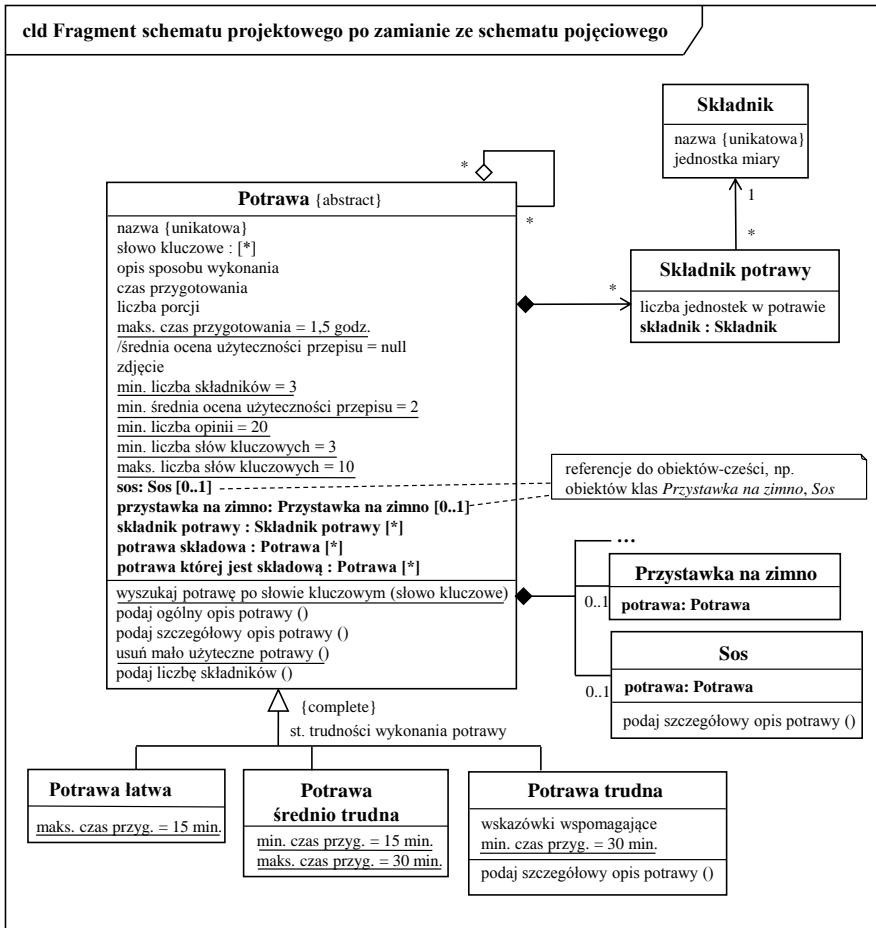


Rys. 9-21 Fragment schematu pojęciowego w trakcie zamiany na schemat projektowy – krok pierwszy

Do rozwiązania zilustrowanego na Rys. 9-21 można by wprowadzić klasę pośrednią połączoną kompozycją (jako klasa-całość) z klasami *Przystawka na zimno*, *Sos* i połączoną kompozycją (jako klasa-część) z klasą *Potrawa*, aby zachować informacje o podziale dwuaspektowym hierarchii dla potraw; ta propozycja nie będzie dalej rozwijana.

W drugim kroku zostaną wprowadzone atrybuty zawierające referencje do obiektów klas z przeciwniej strony asocjacji tylko dla nawigowalnych ról asocjacji, tj. atrybuty *sos*: *Sos* [0..1], *przystawka na zimno* : *Przystawka na zimno* [0..1], *składnik potrawy* : *Składnik potrawy* [\*], *potrawa składowa*: *Potrawa* [\*] i *potrawa której jest składową* :*Potrawa* [\*] w klasie *Potrawa*, atrybut *składnik* : *Składnik* w klasie *Składnik potrawy* oraz atrybuty *potrawa* : *Potrawa* w każdej klasie zawierającej szczegółowe związki z rodzajem potrawy (w klasach *Sos*, *Przystawka na zimno* itd.).

Schemat projektowy dla wskazanego na Rys. 9-20 fragmentu schematu pojęciowego został przedstawiony na Rys. 9-22.

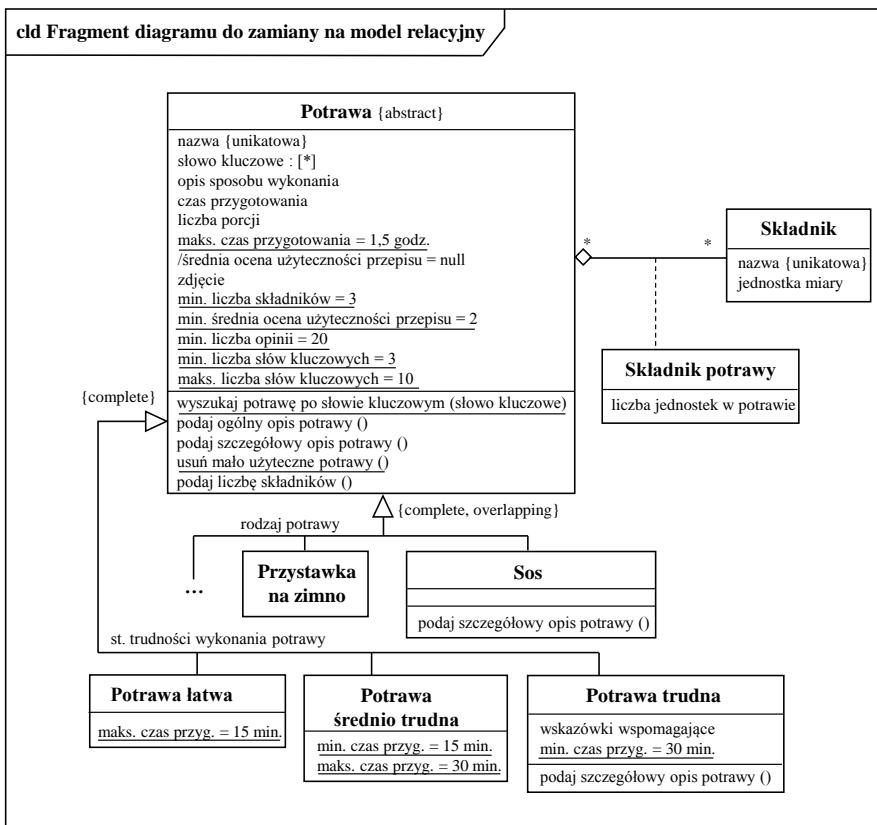


Rys. 9-22 Fragment schematu projektowego po kompletnej zamianie ze schematu pojęciowego

### 9.3.8 Schemat relacyjny dla fragmentu diagramu klas

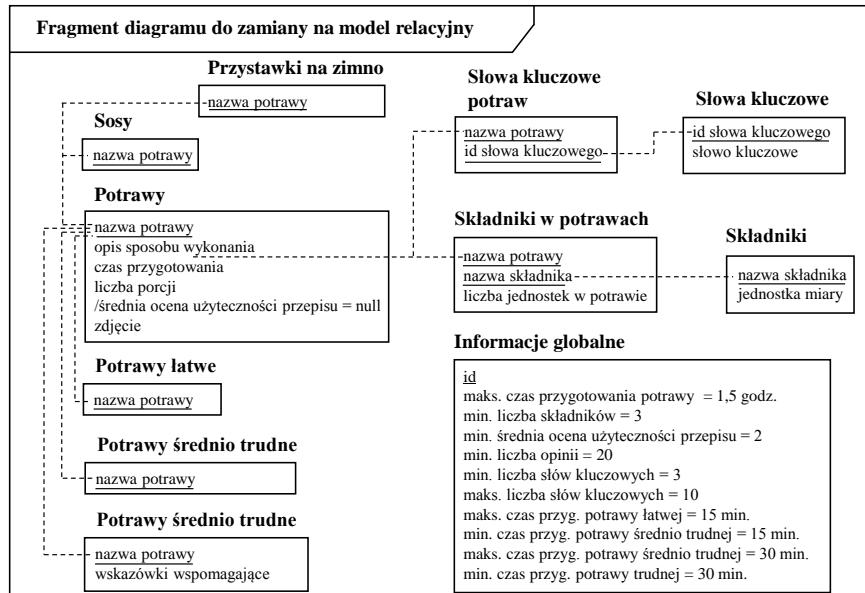
**Polecenie:** Fragment schematu pojęciowego, zawierający przynajmniej trzy klasy, związek dziedziczenia oraz asocjację, przedstaw w modelu relacyjnym.

Na Rys. 9-23 przedstawiony został fragment schematu pojęciowego zawierający hierarchię potraw oraz asocjację (z klasą asocjacji) łączącą potrawy z ich składnikami. Dla zwiększenia czytelności diagramu wszystkie komentarze i ograniczenia zostały pominięte.

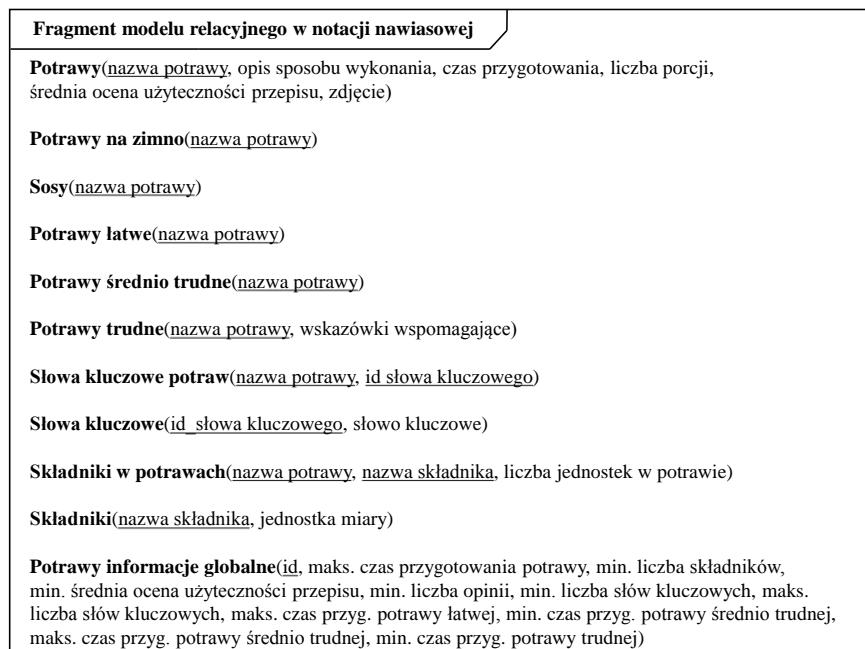


Rys. 9-23 Fragment diagramu klas podlegający zamianie na model relacyjny

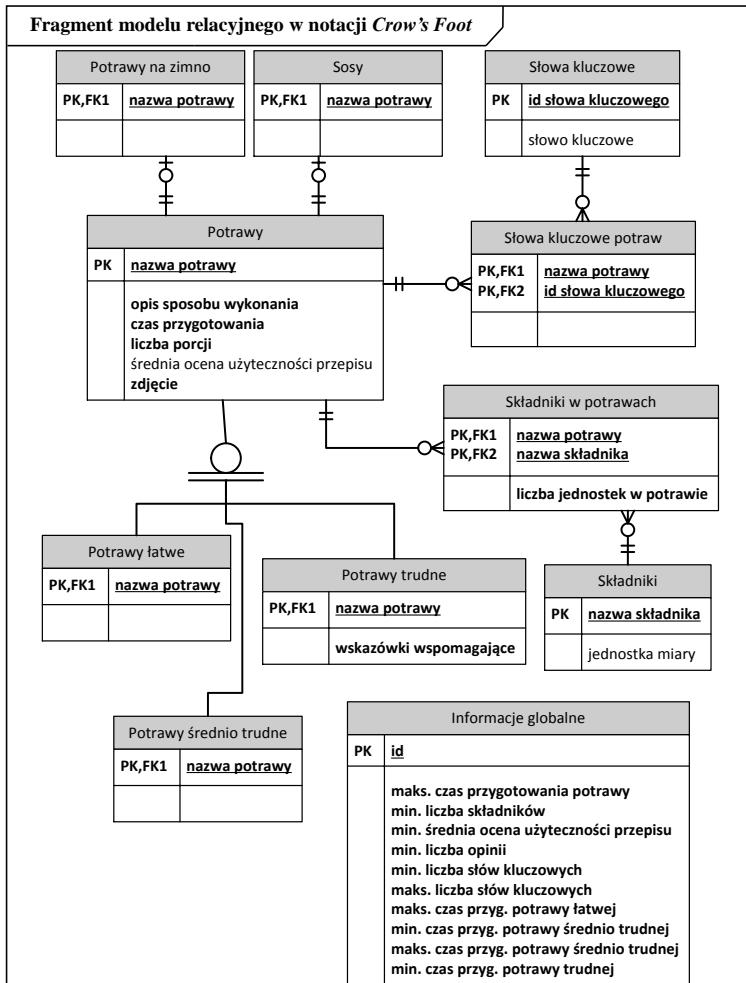
Rys. 9-24 zawiera model relacyjny [37, 48] tego fragmentu diagramu klas w notacji 1 [41]. Na Rys. 9-25 zilustrowano odpowiedni model relacyjny w notacji nawiasowej. Natomiast Rys. 9-26 przedstawia odpowiedni model relacyjny w notacji *Crow's Foot*.



Rys. 9-24 Model relacyjny fragmentu diagramu klas (notacja 1)



Rys. 9-25 Model relacyjny fragmentu diagramu klas (notacja 2)

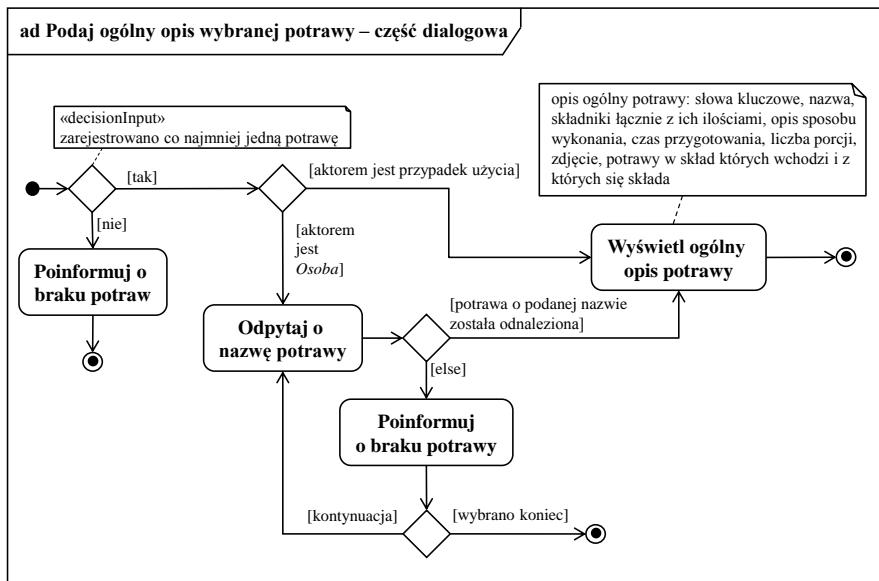
Rys. 9-26 Model relacyjny fragmentu diagramu klas (notacja *Crow's Foot*)

## 9.4 Analiza dynamiczna

### 9.4.1 Zadania podstawowe

**Zad. 1:** Dla funkcjonalności *Podaj ogólny opis potrawy* (punkt 7.3 tekstu wymagań) skonstruuj diagram aktywności zgodny ze scenariuszem z Tab. 9-1.

Diagram aktywności zawarty na Rys. 9-27 ilustruje przykładową realizację dla scenariusza sporządzonego w Tab. 9-1.



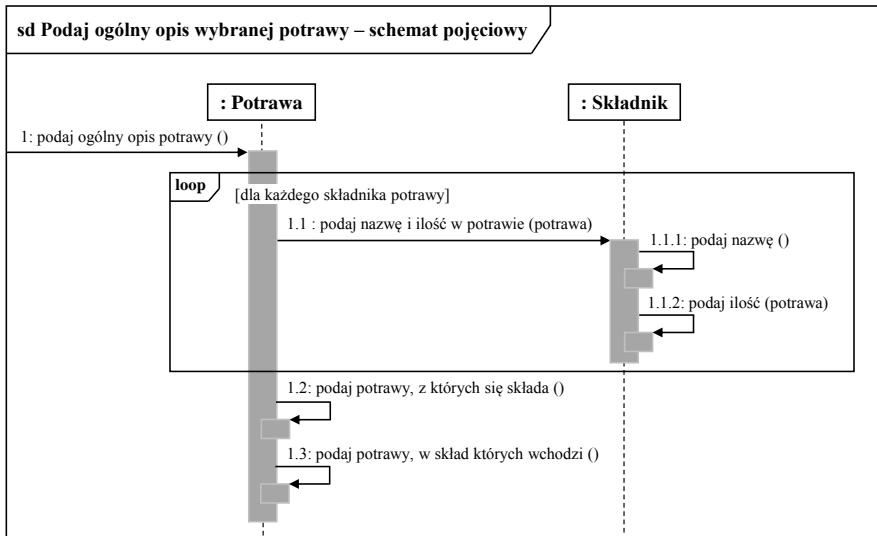
Rys. 9-27 Diagram aktywności dla przypadku użycia *Podaj ogólny opis wybranej potrawy* (część dialogowa)

**Zad. 2:** Sporządź diagram interakcji dla części wykonawczej przypadku użycia *Podaj ogólny opis wybranej potrawy*. Diagram powinien przedstawać interakcję pomiędzy obiektami co najmniej dwóch klas.

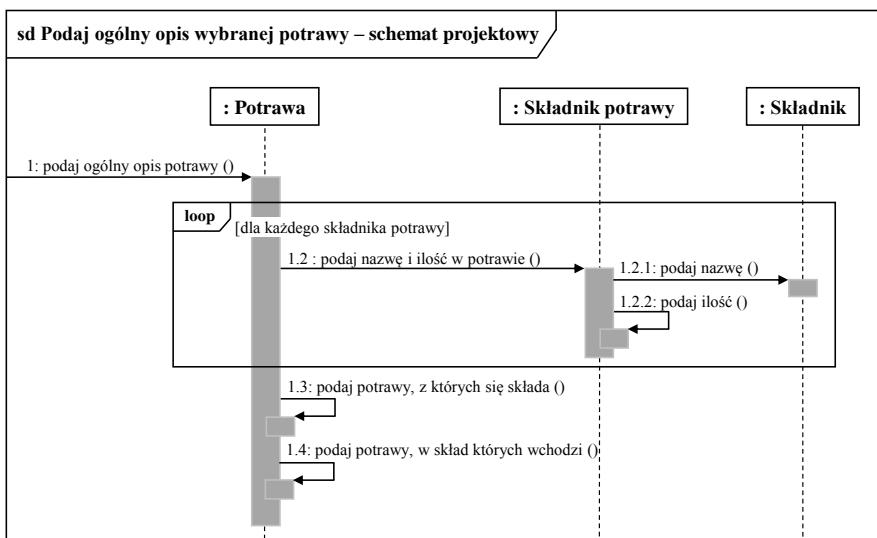
Na Rys. 9-28 i Rys. 9-29 zostały przedstawione diagramy sekwencji dla części wykonawczej przypadku użycia *Podaj ogólny opis wybranej potrawy*, odpowiednio dla schematu pojęciowego i schematu projektowego. Przyjęte zostało założenie, że potrawa została wybrana w trakcie realizowania części dialogowej przypadku użycia, zgodnie z diagramem aktywności z Rys. 9-27.

**Zad. 3:** Dla klasy *Potrawa* ze schematu pojęciowego wskaz kilka stanów (co najmniej dwa stany, nie licząc stanów początkowego i końcowego), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

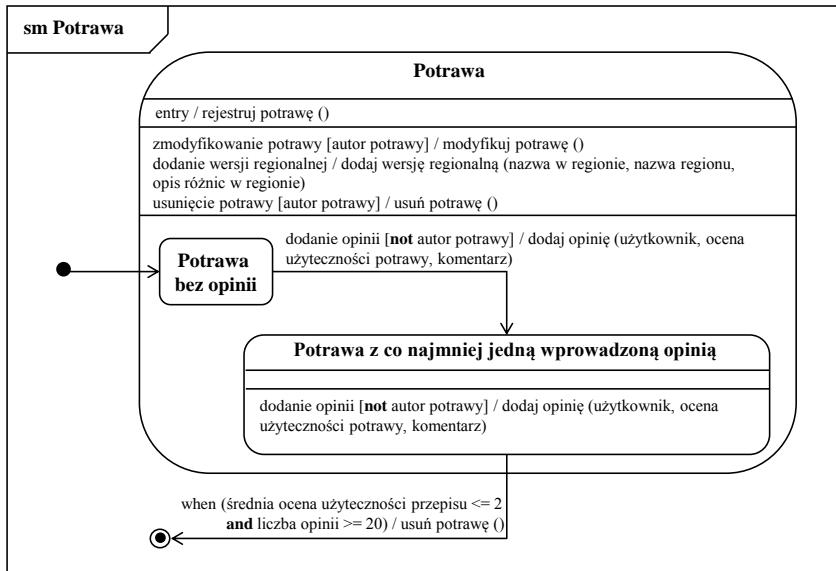
Maszyna stanowa zachowania dla klasy *Potrawa*, ze stanem zagnieżdzonym (sekwencyjnym) została zilustrowana na Rys. 9-30.



Rys. 9-28 Diagram sekwencji zgodny z diagramem aktywności sporządzonym w zadaniu 1 dla schematu pojęciowego



Rys. 9-29 Diagram sekwencji zgodny z diagramem aktywności sporządzonym w zadaniu 1 dla schematu projektowego

Rys. 9-30 Maszyna stanowa dla klasy *Potrawa*

#### 9.4.2 Omówienie zmian na schemacie pojęciowym i schemacie projektowym, wynikającym z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań

W wyniku wykonanej analizy dynamicznej schemat pojęciowy i schemat projektowy należy uzupełnić o nowe elementy, które zostały ujęte na diagramach interakcji oraz diagramach stanów.

Lista zmian na schemacie pojęciowym została zamieszczona w Tab. 9-5, zaś lista zmian na schemacie projektowym – w Tab. 9-6.

Tab. 9-5 Lista zmian na schemacie pojęciowym po analizie dynamicznej

Klasa	Zmiana
<i>Składnik</i>	dodatkowe metody: <i>podaj ilość (potrawa)</i> <i>podaj nazwę ()</i> <i>podaj nazwę i ilość w potrawie (potrawa)</i>

<i>Potrawa</i>	<p>dodatkowe metody:</p> <p><i>dodaj opinię (użytkownik, ocena użyteczności potrawy, komentarz)</i></p> <p><i>dodaj wersję regionalną (nazwa w regionie, nazwa regionu, opis różnic w regionie)</i></p> <p><i>modyfikuj potrawę (słowa kluczowe, lista składników łącznie z ich ilościami, opis sposobu wykonania, czas przygotowania, liczba porcji, zdjęcie, lista potraw w skład których wchodzi, lista potraw z których się składa)</i></p> <p><i>podaj potrawy, z których się składa ()</i></p> <p><i>podaj potrawy, w skład których wchodzi ()</i></p> <p><i>rejestruj potrawę (nazwa, słowa kluczowe, lista składników łącznie z ich ilościami, opis sposobu wykonania, czas przygotowania, liczba porcji, zdjęcie, lista potraw w skład których wchodzi, lista potraw z których się składa)</i></p> <p><i>usuń potrawę ()</i></p>
<i>Potrawa trudna</i>	<p>dodatkowe metody:</p> <p><i>modyfikuj potrawę (słowa kluczowe, lista składników łącznie z ich ilościami, opis sposobu wykonania, czas przygotowania, liczba porcji, zdjęcie, lista potraw w skład których wchodzi, lista potraw z których się składa, wskazówki wspomagające)</i></p> <p><i>rejestruj potrawę (nazwa, słowa kluczowe, lista składników łącznie z ich ilościami, opis sposobu wykonania, czas przygotowania, liczba porcji, zdjęcie, lista potraw w skład których wchodzi, lista potraw z których się składa, wskazówki wspomagające)</i></p>

Tab. 9-6 Lista zmian na schemacie projektowym po analizie dynamicznej

Klasa	Zmiana
<i>Składnik</i>	<p>dodatkowa metoda:</p> <p><i>podaj nazwę ()</i></p>
<i>Składnik potrawy</i>	<p>dodatkowe metody:</p> <p><i>podaj ilość ()</i></p> <p><i>podaj nazwę i ilość w potrawie ()</i></p>

<i>Potrawa</i>	<p>dodatkowe metody:</p> <p><i>dodaj opinię (użytkownik, ocena użyteczności potrawy, komentarz)</i></p> <p><i>dodaj wersję regionalną (nazwa w regionie, nazwa regionu, opis różnic w regionie)</i></p> <p><i>modyfikuj potrawę (słowa kluczowe, lista składników łącznie z ich ilościami, opis sposobu wykonania, czas przygotowania, liczba porcji, zdjęcie, lista potraw w skład których wchodzi, lista potraw z których się składa)</i></p> <p><i>podaj potrawy, z których się składa ()</i></p> <p><i>podaj potrawy, w skład których wchodzi ()</i></p> <p><i>rejestruj potrawę (nazwa, słowa kluczowe, lista składników łącznie z ich ilościami, opis sposobu wykonania, czas przygotowania, liczba porcji, zdjęcie, lista potraw w skład których wchodzi, lista potraw z których się składa)</i></p> <p><i>usuń potrawę ()</i></p>
<i>Potrawa trudna</i>	<p>dodatkowe metody:</p> <p><i>modyfikuj potrawę (słowa kluczowe, lista składników łącznie z ich ilościami, opis sposobu wykonania, czas przygotowania, liczba porcji, zdjęcie, lista potraw w skład których wchodzi, lista potraw z których się składa, wskazówki wspomagające)</i></p> <p><i>rejestruj potrawę (nazwa, słowa kluczowe, lista składników łącznie z ich ilościami, opis sposobu wykonania, czas przygotowania, liczba porcji, zdjęcie, lista potraw w skład których wchodzi, lista potraw z których się składa, wskazówki wspomagające)</i></p>

## 9.5 Zadania do samodzielnej pracy

1. Określ, na czym polega różnica między relacją dziedziczenia a relacją «extend» między przypadkami użycia. Podaj przykłady.
2. Czy system zewnętrzny może być aktorem dla danego systemu? Jeśli tak, odpowiedź uzupełnij przykładem spójnym z wymaganiami z zadania.
3. Objasnij różnicę między dziedziczeniem nieroziłącznym (ang. *overlapping*) a dziedziczeniem wieloaspektowym. Czy któryś z tych dwóch rodzajów dziedziczenia wystąpił na schemacie pojęciowym z zadania?
4. Zdefiniuj pojęcie metoda abstrakcyjna. Wprowadź do schematu pojęciowego zadania przynajmniej jedną metodę abstrakcyjną (wraz z odpowiednimi implementacjami tej metody w podklassach).

5. Czy zasadę zamienialności można stosować do obiektów klas tworzących agregację? Odpowiedź uzasadnij. Jeśli odpowiedź jest na „tak”, podaj przykład w oparciu o diagram z Rys. 9-22.
6. Wskaż na schemacie pojęciowym z Rys. 9-13 asocjację, która w naturalny sposób kwalifikuje się do zamiany na kompozycję lub agregację (inną niż w Zad. 6 z rozdziału 9.3.5). Uzasadnij swój wybór (dlaczego agregacja? dlaczego kompozycja?). Ponadto, dlaczego zdecydowałeś się na wykorzystanie agregacji (lub kompozycji), a nie „zwykłej asocjacji”?
7. Objasnij różnicę pomiędzy komentarzem a ograniczeniem. Podaj odpowiednie przykłady w oparciu o schemat pojęciowy z Rys. 9-13.
8. Dla wybranego przypadku użycia systemu (innego niż omówionego szczegółowo w analizie funkcjonalnej lub dynamicznej), sformułuj scenariusz i narysuj diagram aktywności (określ przynajmniej dwie aktywności, nie licząc początkowej i końcowej).
9. Dla wybranego przypadku użycia systemu (innego niż omówionego szczegółowo w analizie funkcjonalnej lub dynamicznej), sformułuj algorytm i narysuj odpowiedni diagram interakcji, który przedstawia interakcje pomiędzy obiektami co najmniej dwóch różnych klas.
10. Dla dowolnej klasy (innej niż omówionej szczegółowo w analizie dynamicznej) ze schematu pojęciowego wskaż dwa stany (wyłączając stan początkowy i stan końcowy), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

## 9.6 Podsumowanie zadania

W Tab. 9-7 przedstawiono ogólną charakterystykę omawianego zadania. Szczegółową charakterystykę zadania dla grup tematycznych: *Wprowadzenie do przedmiotu*, *Analiza funkcjonalna*, *Analiza strukturalna*, *Analiza dynamiczna* zamieszczono odpowiednio w tabelach Tab. 9-8, Tab. 9-9, Tab. 9-10 i Tab. 9-11.

Tab. 9-7 Ogólna charakterystyka zadania

<b>Ogólna charakterystyka zadania</b>	
<i>Nazwa</i>	Internetowa książka kucharska
<i>Dziedzina problemowa</i>	system mógłby znaleźć zastosowanie w każdej dziedzinie związanej z gotowaniem, jak np. restauracje, szkoły dla kucharzy, portale dla osób poszukujących ciekawych przepisów

<i>Cel</i>	system ma za zadanie ułatwić użytkownikom dostęp do przepisów potraw z różnych regionów świata	
<i>Zakres odpowiedzialności</i>	ewidencja regionów świata, ewidencja i klasyfikacja przepisów potraw, ewidencja użytkowników, wyszukiwanie potraw, wprowadzanie opinii o potrawach	
<i>Stopień trudności</i>	trudne	
<i>Rozmiar</i>	10 klas	
<i>Użyteczność w grupach tematycznych</i>		
<i>Grupa tematyczna</i>	<i>Użyteczność w grupie tematycznej</i>	<i>Użyteczność na etapie zgodnym z modelem efektywnego nauczania</i>
<i>Wprowadzenie do przedmiotu</i>	niska	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> niska <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> niska
<i>Analiza funkcjonalna</i>	wysoka	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> wysoka <i>Podsumowanie dotychczasowych działań:</i> wysoka <i>Ewaluacja:</i> wysoka
<i>Analiza strukturalna</i>	wysoka	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> wysoka <i>Podsumowanie dotychczasowych działań:</i> wysoka <i>Ewaluacja:</i> wysoka
<i>Analiza dynamiczna</i>	wysoka	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> wysoka <i>Podsumowanie dotychczasowych działań:</i> wysoka <i>Ewaluacja:</i> wysoka

Tab. 9-8 Szczegółowa charakterystyka zadania dla grupy tematycznej *Wprowadzenie do przedmiotu*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Wprowadzenie do przedmiotu</i>

<b>Stopień trudności grupy w zadaniu</b>	łatwe		
<b>Poruszane tematy</b>	<i>analiza a projektowanie, klasyfikatory a instancje, dekompozycja diagramów, mechanizmy rozszerzalności, reguły nazewnictwa, poprawa percepacji diagramu/diagramów</i>		
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>	
<i>analiza a projektowanie</i>	łatwe	<i>model pojęciowy, model projektowy</i>	
<i>klasyfikatory a instancje</i>	łatwe	<i>klasyfikator, instancja klasyfikatora</i>	
<i>mechanizmy rozszerzalności</i>	łatwe	<i>stereotyp, ograniczenie, komentarz</i>	
<i>dekompozycja diagramów</i>	łatwe	<i>podział modelu przypadków użycia na diagramy</i>	
<i>reguły nazewnictwa</i>	łatwe	<i>reguły nazewnictwa aktorów, przypadków użycia, klas, atrybutów i metod</i>	
<i>poprawa percepji diagramu/diagramów</i>	łatwe	<i>reguły organizacji przypadków użycia, budowy hierarchii klas, identyfikacji asocjacji kwalifikowanej</i>	

Tab. 9-9 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza funkcjonalna*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	Analiza funkcjonalna
<b>Stopień trudności grupy w zadaniu</b>	trudne
<b>Poruszane tematy</b>	<i>aktorzy, przypadki użycia, scenariusze, dekompozycja modelu/diagramu przypadków użycia, mechanizmy rozszerzalności</i>

<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>aktorzy</i>	średnio trudne	<i>aktor abstrakcyjny, aktor Podsystem czasu, hierarchia dla aktorów</i>
<i>przypadki użycia</i>	trudne	hierarchia dla przypadków użycia (dwie wersje)
<i>scenariusze</i>	trudne	stosunkowo duża złożoność scenariuszy przypadków użycia, podprzepływy nazwane
<i>dekompozycja modelu/diagramu przypadków użycia</i>	średnio trudne	<i>dekompozycja pozioma, dekompozycja pionowa</i>
<i>mechanizmy rozszerzalności</i>	łatwe	<i>ograniczenie, komentarz</i>

Tab. 9-10 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza strukturalna*

<b>Szczegółowa charakterystyka zadania</b>			
<b>Grupa tematyczna</b>	Analiza strukturalna		
<b>Stopień trudności grupy w zadaniu</b>	trudne		
<b>Poruszane tematy</b>	<i>atrybuty klasy, metody klasy, struktury generalizacji specjalizacji, klasa abstrakcyjna a klasa konkretna, asocjacje, analiza wartości, przejście na schemat projektowy, przejście na schemat relacyjny, mechanizmy rozszerzalności</i>		
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>	
<i>atrybuty klasy</i>	średnio trudne	<i>atrybut prosty, złożony, powtarzalny, pochodny, obiektowy, klasowy</i>	
<i>metody klasy</i>	średnio	<i>metoda abstrakcyjna, zaimplementowana,</i>	

	trudne	<i>obiektowa, klasowa, polimorfizm metod</i>
<i>struktury generalizacji-specjalizacji</i>	średnio trudne	<i>dziedziczenie klas, dziedziczenie nierożłączne, wieloaspektowe i elipsa</i>
<i>klasa abstrakcyjna a klasa konkretna</i>	łatwe	<i>klasa abstrakcyjna, klasa konkretna</i>
<i>asocjacje</i>	trudne	<i>rola asocjacji, agregacja, kompozycja, agregacja rekursywna, asocjacja kwalifikowana, stosunkowo duża liczba klas asocjacji, rozważono, czy klasa asocjacji może wystąpić w charakterze klasyfikatora ogólnego</i>
<i>analiza wartości</i>	średnio trudne	<i>wartość pochodna, wartość domyślna (początkowa), wartość graniczna, stosunkowo duża liczba wartości granicznych</i>
<i>przejście na schemat projektowy</i>	trudne	<i>obejście dziedziczenia nierożłącznego, wieloaspektowego i elipsy, zamiana agregacji rekursywnej i asocjacji z klasą asocjacji, asocjacja skierowana</i>
<i>przejście na schemat relacyjny</i>	trudne	<i>obejście dziedziczenia nierożłącznego, wieloaspektowego i elipsy, obejście agregacji rekursywnej i asocjacji z klasą asocjacji, hierarchia encji, encje/tabele słownikowe</i>
<i>mechanizmy rozszerzalności</i>	średnio trudne	<i>ograniczenie, komentarz, stosunkowo duża liczba ograniczeń i komentarzy na schemacie pojęciowym i projektowym</i>

Tab. 9-11 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza dynamiczna*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Analiza dynamiczna</i>
<b>Stopień trudności grupy w zadaniu</b>	średnio trudne
<b>Poruszane tematy</b>	<i>diagramy aktywności, diagramy interakcji, diagramy stanów, podsumowanie analizy dynamicznej</i>

<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>diagramy aktywności</i>	średnio trudne	
<i>diagramy interakcji</i>	trudne	stosunkowo duża złożoność diagramów sekwencji, fragmenty wyodrębnione <i>loop</i> na diagramach sekwencji
<i>diagramy stanów</i>	średnio trudne	zdarzenie <i>when</i> , stan zagnieźdzony
<i>podsumowanie analizy dynamicznej</i>	średnio trudne	stosunkowo duża liczba dodatkowych metod na schemacie pojęciowym i projektowym

## Spis treści rozdziału

Rozdział 10 Klinika jednego dnia.....	416
10.1 Tekst wymagań.....	416
10.2 Analiza funkcjonalna .....	418
10.2.1 Diagram kontekstowy .....	418
10.2.2 Model przypadków użycia.....	419
10.2.3 Scenariusze dla wybranych przypadków użycia łącznie z ich podziałem na podprzypadki.....	422
10.2.4 Omówienie typowych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się.....	428
10.3 Analiza strukturalna.....	430
10.3.1 Schemat pojęciowy .....	430
10.3.2 Analiza elementów pochodnych.....	432
10.3.3 Analiza wartości początkowych .....	433
10.3.4 Analiza wartości granicznych.....	436
10.3.5 Zadania podstawowe .....	440
10.3.6 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	442
10.3.7 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji .....	443
10.3.8 Schemat relacyjny dla fragmentu schematu pojęciowego .....	446
10.4 Analiza dynamiczna .....	448
10.4.1 Zadania podstawowe .....	448
10.4.2 Omówienie zmian na schemacie pojęciowym i projektowym wynikających z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań.....	450
10.5 Zadania do samodzielnej pracy .....	454
10.6 Podsumowanie zadania.....	455

## Rozdział 10

# Klinika jednego dnia

Ewa Stemposz, Andrzej Jodłowski

### 10.1 Tekst wymagań

Właściciel prywatnej kliniki (tzw. *kliniki jednego dnia*) postanowił rozpocząć prace nad budową systemu, który wspomogłby obsługę przeprowadzanych w tej klinice zabiegów.

1. System ma przechowywać dane o osobach związanych z kliniką, zarówno o pracownikach, jak i o pacjentach (dane osobowe, dane teleadresowe). Ponadto, dla pracowników mają być przechowywane także informacje takie jak: data zatrudnienia, data zwolnienia, staż pracy w zawodzie i pensja. Dla pacjentów ma być przechowywana lista uczuleń – zawierająca nie więcej niż 10 pozycji. Pracownik kliniki może być także jej pacjentem.
2. Klinika zatrudnia m. in. lekarzy. Lekarze to chirurdzy, anestezjolodzy i inni.
3. Mają być przechowywane informacje o przeprowadzanych zabiegach: rodzaj zabiegu, nazwa, charakterystyka oraz przedział cen. Nazwa, charakterystyka, przedział cen oraz sugerowany rodzaj znieczulenia są jednakowe dla wszystkich zabiegów danego rodzaju. Dla zabiegów pamiętamy także datę przeprowadzenia zabiegu, godziny jego rozpoczęcia i zakończenia (planowane i rzeczywiste), cenę zabiegu (miesiączącą się w zadanym przedziale cen), stopień trudności (w skali od 1 do 3) oraz status zabiegu („zgłoszony”, „w trakcie planowania”, „zaplanowany”, „trwający”, „przeprowadzony”, „anulowany”).
4. Zabiegi dzielą się na zabiegi wykonywane w znieczulieniu ogólnym oraz zabiegi wykonywane w znieczulieniu miejscowym. Dla pierwszych trzeba pamiętać deklarowany przez anestezjologa czas wybudzenia pacjenta, a dla tych drugich – nazwę leku znieczulającego i jego dawkę.
5. Do wykonania pojedynczego zabiegu jest przydzielana sala i jest powoływany zespół, w skład którego wchodzi od 2 do 4 lekarzy, z których każdy pełni w nim unikatową rolę. Jeden z członków zespołu jest lekarzem odpowiedzialnym za całość realizacji zabiegu. Po przeprowadzeniu zabiegu zespół jest rozwiązywany. Dla zabiegów w znieczulieniu ogólnym do zespołu musi należeć jeden anestezjolog.

6. W momencie zgłoszenia zabiegu przez lekarza zabieg przyjmuje status „zgłoszony”. W chwili rozpoczęcia planowania zabiegu jest powoływany zespół zabiegowy (niekoniecznie kompletny) i od tej pory mogą być ustalane pozostałe informacje, a zabieg otrzymuje status „w trakcie planowania”. Zabieg przyjmuje status „zaplanowany” po ustaleniu planowanej daty, godziny rozpoczęcia i zakończenia zabiegu, jego ceny, stopnia trudności, co najmniej minimalnego składu zespołu (dwóch lekarzy, w tym jeden anestezjolog – o ile jest potrzebny) oraz sali zabiegowej. Zaplanowanie zabiegu wymaga powiadomienia o tym pacjenta i zespół.
7. Lekarz naczelnny może w dowolnym momencie anulować zabieg o statusie „zgłoszony”, „w trakcie planowania” lub „zaplanowany”. Ponadto, zabieg jest automatycznie anulowany, jeśli w określonym terminie (aktualnie wynosi on 3 dni przed planowaną datą zabiegu) nie uda się skompletować zespołu, przydzielić sali lub uzupełnić informacji anestezjologicznych. Anulowanie zabiegu odbywa się z jednociennym powiadomieniem o tym pacjenta i zespołowi. Ma być pamiętaana przyczyna anulowania.
8. Po wykonaniu zabiegów o stopniu trudności 2 lub 3 członkowie zespołu każdorazowo otrzymują premię, w wysokości odpowiednio 3% i 5% ceny zabiegu (są to wartości aktualne, które w przyszłości mogą ulec zmianie).
9. Ponadto anestezjolodzy – z powodu problemów z zatrudnianiem – otrzymują każdorazowo za udział w zabiegu dodatkową premię, której wysokość ustalana przez dyrektora kliniki, jest różna dla różnych anestezjologów.
10. Anestezjolog, który popełnił trzy wykroczenia, spośród takich jak: przekroczenie deklarowanego czasu wybudzenia pacjenta (w związku z czym trzeba pamiętać także rzeczywisty czas wybudzenia), niedostarczenia w terminie informacji dotyczących wybudzenia pacjenta (dla zabiegów w znieczuleniu ogólnym) lub nieokreślenia leku znieczulającego i jego dawki (dla zabiegów w znieczuleniu miejscowym), jest zwalniany z kliniki.
11. Zabiegi przeprowadzane są w salach zabiegowych – jeden zabieg w jednej sali. W danym momencie w sali może być przeprowadzany tylko jeden zabieg.
12. Chcemy przechowywać informację o sprzęcie medycznym, w jaki są wyposażone sale zabiegowe (co najmniej jeden sprzęt w każdej sali).
13. Chcemy także pamiętać, w jakich godzinach jest przeprowadzana dezynfekcja sal – wszystkie sale są dezynfekowane w tym samym czasie (nawet kilkakrotnie w ciągu dnia).
14. System powinien wspomagać między innymi realizację poniższych funkcjonalności:
  - 14.1 Usuwanie informacji o pracownikach zwolnionych wcześniej niż 5 lat temu;
  - 14.2 Prezentowanie listy rodzajów zabiegów przeprowadzanych w klinice z ewentualnym dołączeniem opisu wybranego rodzaju (również gość);

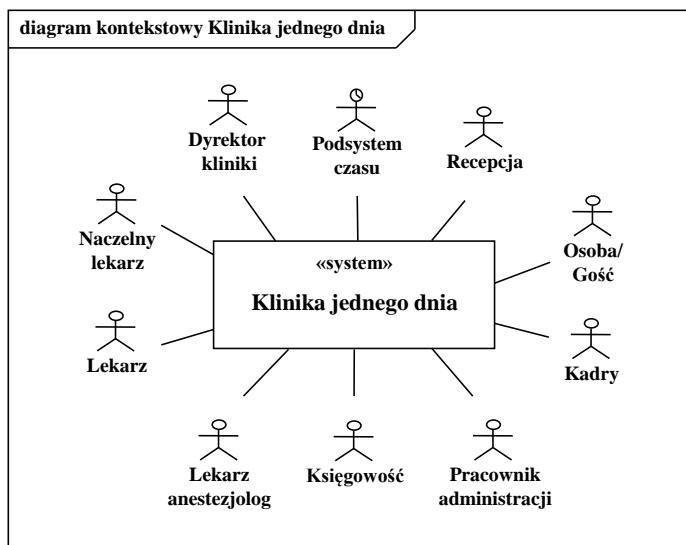
- 14.3 Wyliczanie miesięcznego dochodu lekarza;
- 14.4 Wyliczanie miesięcznej premii lekarza;
- 14.5 Zgłoszenie zabiegu pacjenta łącznie z określeniem rodzaju tego zabiegu przez lekarza i z określeniem rodzaju znieczulenia (opcjonalnie);
- 14.6 Planowanie zabiegu pacjenta łącznie z określeniem terminu, próbą przydzielenia sali, utworzenia zespołu oraz przydzielenia do niego co najmniej 2 lekarzy i wyboru spośród nich lekarza odpowiedzialnego;
- 14.7 Przydzielanie / wyłączanie lekarza z zespołu zabiegowego;
- 14.8 Określanie rodzaju zabiegów, na przeprowadzeniu których w zadanym okresie klinika zarobiła najwięcej.

## 10.2 Analiza funkcjonalna

### 10.2.1 Diagram kontekstowy

*Polecenie:* Zbuduj diagram kontekstowy w oparciu o tekst wymagań zamieszczony punkcie 10.1 .

Diagram kontekstowy dla systemu *Klinika jednego dnia* został przedstawiony na Rys. 10-1.

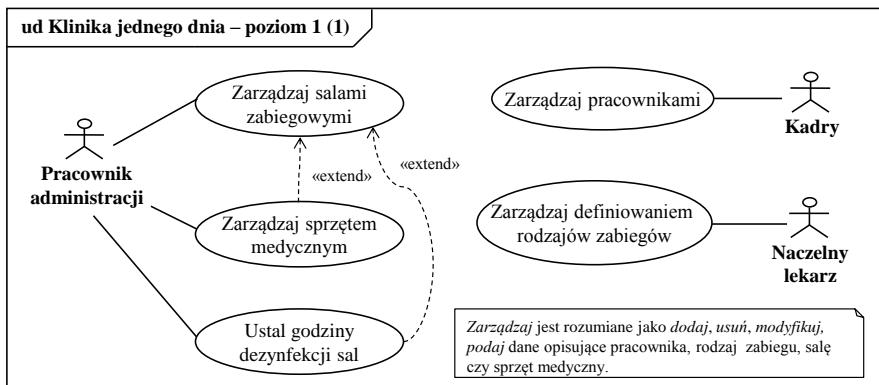


Rys. 10-1 Diagram kontekstowy

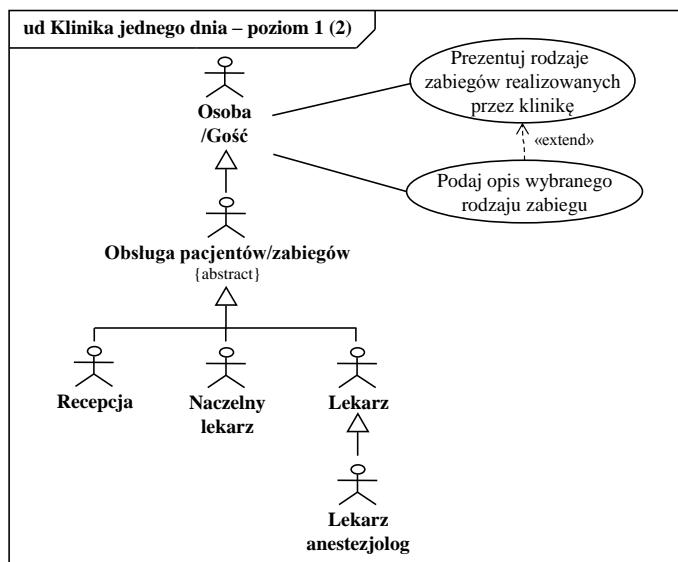
## 10.2.2 Model przypadków użycia

**Polecenie:** Zbuduj model przypadków użycia, uwzględniając między innymi funkcjonalność sugerowaną w ostatnim punkcie tekstu wymagań (tzn. w punkcie 14).

**Uwaga:** Model należy skonstruować z perspektywy aktorów z otoczenia systemu oraz z uwzględnieniem hierarchii aktorów i relacji pomiędzy przypadkami (o ile mają/mogłyby mieć miejsce).



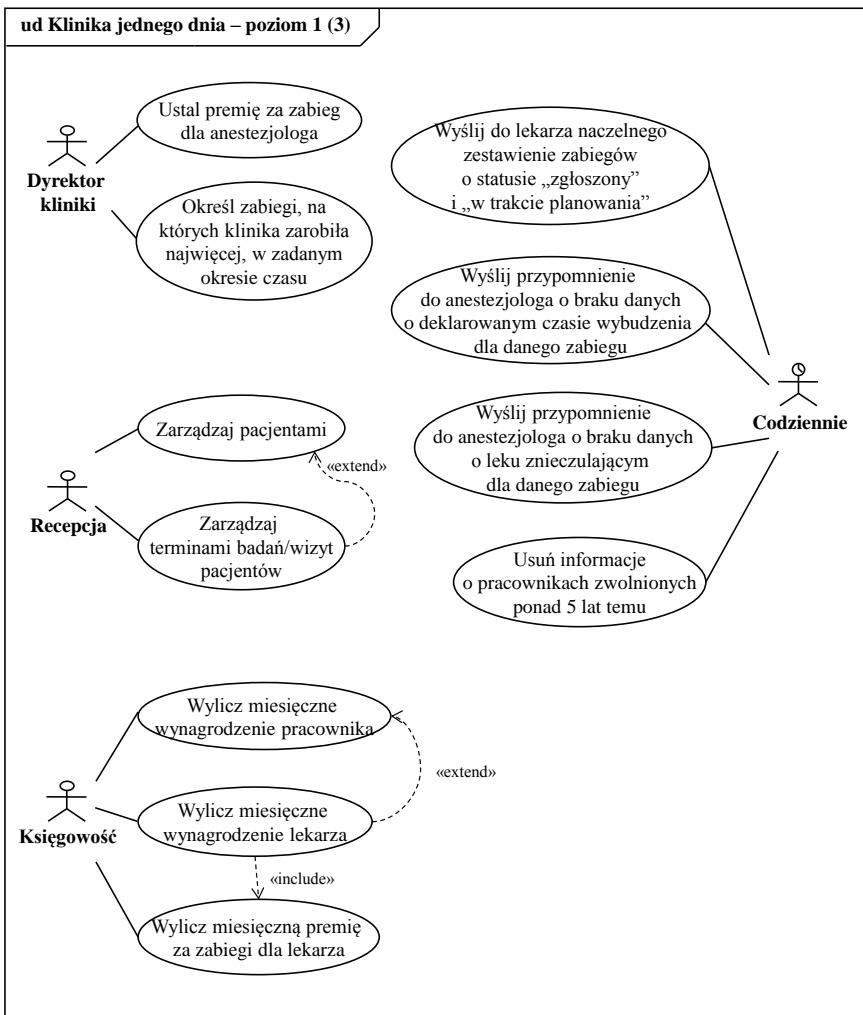
Rys. 10-2 Model przypadków użycia (diagram pierwszy)



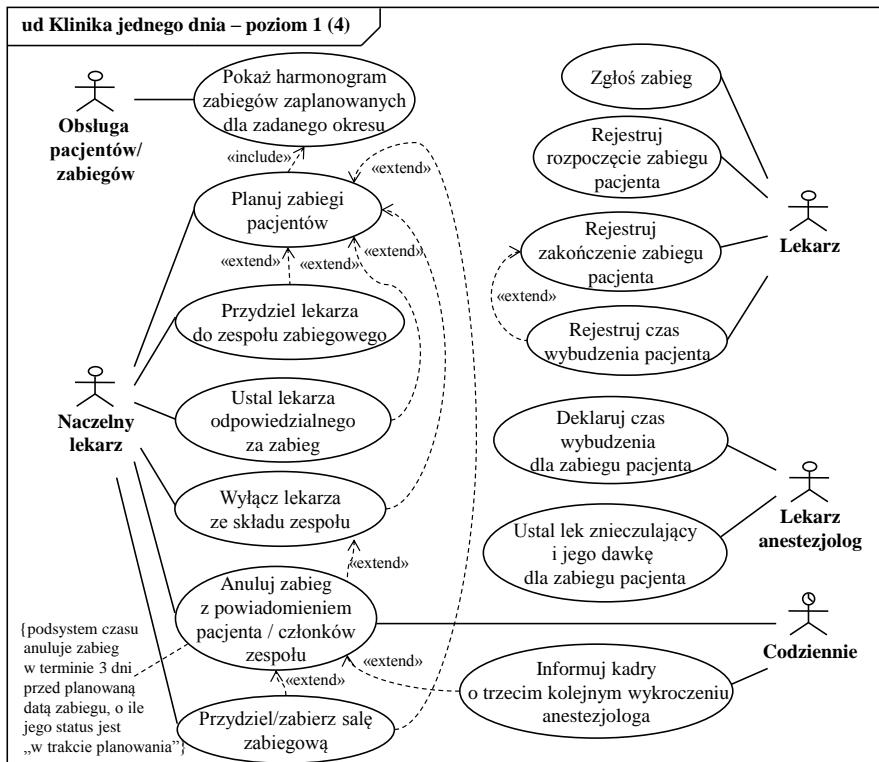
Rys. 10-3 Model przypadków użycia (diagram drugi)

Model przypadków użycia [36, 38, 41, 45] został przedstawiony za pomocą czterech diagramów na Rys. 10-2, Rys. 10-3, Rys. 10-4 oraz Rys. 10-5.

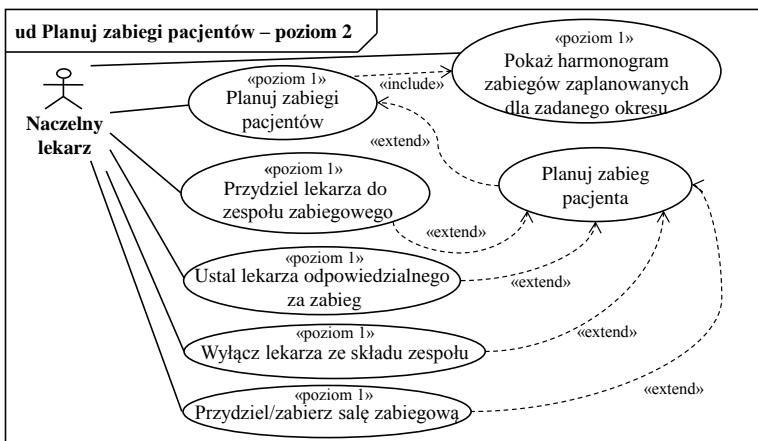
W hierarchii aktorów przedstawionej na Rys. 10-3 celowo pominięto aktorów *Kadry*, *Księgowość* i *Pracownik administracji*, uznając, że funkcjonalność aktora *Gość* nie dotyczy tych aktorów.



Rys. 10-4 Model przypadków użycia (diagram trzeci)



Rys. 10-5 Model przypadków użycia (diagram czwarty)

Rys. 10-6 przedstawia przykładowy podział przypadku *Planuj zabiegi pacjentów*:Rys. 10-6 Diagram z przykładowym podziałem przypadku *Planuj zabiegi pacjentów*

### 10.2.3 Scenariusze dla wybranych przypadków użycia łącznie z ich podziałem na podprzypadki

*Polecenie:* Dla przypadku użycia *Planuj zabieg pacjenta*:

- napisz scenariusz;
- zaproponuj podział tego przypadku na podprzypadki.

Przykładowy scenariusz dla przypadku użycia *Planuj zabieg pacjenta* łącznie z próbą przydzielenia sali, przydzielenia co najmniej 2 lekarzy i wyborem lekarza odpowiedzialnego za zabieg (punkt 14.6 tekstu wymagań) zaprezentowano w Tab. 10-1. Przyjęto założenie, że przebieg główny dotyczy sytuacji, w której status zabiegu ma wartość „zgłoszony”.

Tab. 10-1 Scenariusz dla przypadku użycia *Planuj zabieg pacjenta* łącznie z określeniem terminu zabiegu oraz próbą przydzielenia sali, przydzielenia co najmniej 2 lekarzy i wyborem lekarza odpowiedzialnego za zabieg

<b>Nazwa przypadku użycia</b>	Planuj zabieg pacjenta
<b>Warunek początkowy</b>	Zabieg ma status „zgłoszony” lub „w trakcie planowania”. Dla zabiegu o statusie „w trakcie planowania” został już wcześniej ustalony czas trwania, stopień trudności, cena i utworzony zespół zabiegowy (niekoniecznie kompletny).
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Przypadek użycia jest wywoływany z przypadku <i>Planuj zabiegi pacjentów</i> – łącznie z przekazaniem następujących informacji o zabiegu (określonych przez lekarza zgłoszającego zabieg): którego pacjenta zabieg dotyczy, rodzaj planowanego zabiegu i rodzaj znieczulenia, które ma być zastosowane w trakcie zabiegu.</li> <li>2. System uruchamia podprzepływ nazwany: <i>Ustal czas trwania, stopień trudności, cenę i utwórz zespół bez osób</i>.</li> <li>3. System wyświetla kalendarz w celu określenia okresu, w którym zabieg pacjenta mógłby być przeprowadzony. Aktor wybiera okres.</li> <li>4. System wyświetla aktualny harmonogram zabiegów o statusie „w trakcie planowania” lub „zaplanowa-</li> </ol>

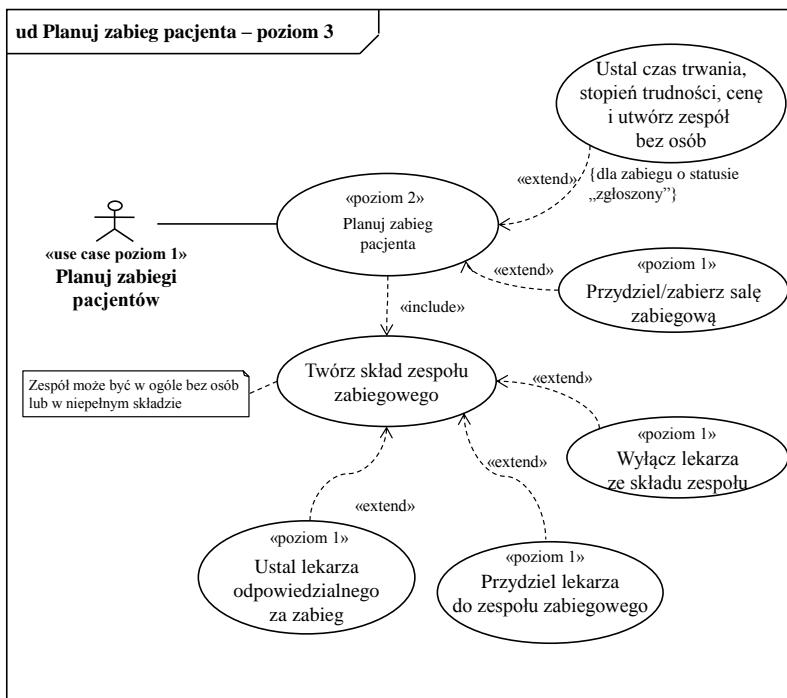
	<p>ny” dla wybranego okresu i prosi o ustalenie terminu zabiegu. Aktor wprowadza planowaną datę i godzinę rozpoczęcia zabiegu.</p> <p>5. System uruchamia przypadek użycia <i>Twórz skład zespołu zabiegowego</i> z przekazaniem informacji o planowanym zabiegu, takimi jak: pacjent, rodzaj zabiegu, data, planowana godzina rozpoczęcia i przybliżony czas trwania oraz czy potrzebny jest anestezjolog.</p> <p>6. System uruchamia przypadek użycia <i>Przydziel salę do zabiegu</i> z przekazaniem informacji o planowanej dacie, planowanej godzinie rozpoczęcia i przybliżonym czasie trwania zabiegu.</p> <p>7. System wyświetla informację o zaplanowanym zabiegu (data, godzina rozpoczęcia i zakończenia zabiegu, skład zespołu zabiegowego, lekarz odpowiedzialny, sala). Aktor potwierdza rejestrację.</p>
<b>Alternatywne przepływy zdarzeń</b>	<p>2a. Zabieg ma status „w trakcie planowania”. System przechodzi do punktu 3.</p> <p>3a. W zadany okresie nie jest możliwe utworzenie zespołu w minimalnym składzie. System pyta, czy kontynuować dla innego okresu.</p> <p>3aa. Aktor potwierdza wolę wprowadzenia innego okresu. System powraca do punktu 3.</p> <p>3ab. Aktor rezygnuje z kontynuowania planowania zabiegu, system kończy przypadek użycia.</p> <p>4a. W zadany terminie (w podanym dniu i o podanej godzinie) nie jest możliwe utworzenie zespołu, w minimalnym składzie (tj. z co najmniej dwoma lekarzami, jednym anestezjologiem – o ile zabieg tego wymaga i wybranym lekarzem odpowiedzialnym). System informuje o tym i pyta, czy kontynuować dla innego okresu lub w innym terminie (dla wybranego wcześniej okresu), czy też podjąć próbę utworzenia zespołu w niepełnym składzie.</p> <p>4aa. Aktor potwierdza kontynuację, system wraca do punktu 3 lub 4, lub 5 – w zależności od wybranej opcji.</p> <p>4ab. Aktor rezygnuje z kontynuowania planowania zabiegu, system kończy przypadek użycia.</p>

	<p>5a. Skład utworzonego zespołu zabiegowego nie spełnia minimalnych wymagań (tj. zostało przydzielonych do zespołu mniej niż dwóch lekarzy, nie przydzielono anestezjologa w przypadku zabiegu ze znieczuleniem ogólnym lub nie wybrano lekarza odpowiedzialnego). System informuje o tym aktora i pyta, czy mimo to spróbować przydzielić salę dla zabiegu.</p> <p>5aa. Aktor potwierdza wolę przydziału sali (pomimo niepełnego składu zespołu), system przechodzi do punktu 6 (w celu przydziału sali).</p> <p>5ab. Aktor rezygnuje z kontynuowania planowania zabiegu, system kończy przypadek użycia.</p> <p>6a. W zadany terminie nie ma wolnej sali, system informuje o tym aktora i odpytuje, czy próbować zaplanować zabieg dla innego okresu lub w innym terminie.</p> <p>6aa. Aktor potwierdza wolę wyboru nowego terminu, system informuje o wyłączeniu lekarzy z zespołu zabiegowego i w zależności od wybranej opcji wraca do punktu 3 (w celu określenia nowego okresu) lub 4 (w celu określeniem nowego terminu).</p> <p>6ab. W przeciwej sytuacji system kończy przypadek użycia.</p> <p>7a. Aktor nie potwierdza rejestracji, system kończy przypadek użycia.</p>
<b>Zakończenie</b>	Zgodnie ze scenariuszem
<b>Warunek końcowy</b>	<p>Jeśli zabieg został zaplanowany (został ustalony termin, sala i skompletowany zespół) zostaje wysłane zawiadomienie o terminie zabiegu do pacjenta i członków zespołu. Jeśli zabieg ma być wykonany w znieczuleniu ogólnym, zostaje także wysłana wiadomość do wybranego anestezjologa z prośbą o zadeklarowanie czasu wybudzenia, a jeśli w znieczuleniu miejscowym – z prośbą o wprowadzenie nazwy i dawki leku znieczulającego.</p> <p>Jeżeli nie uda się skompletować zespołu lub wybrać sali, to status zabiegu po zakończeniu przypadku</p>

	użycia będzie „w trakcie planowania”.
<b>Podprzepływy nazwane</b>	
<b>Nazwa przypadku użycia</b>	<i>Ustal czas trwania, stopień trudności, cenę i utwórz zespół bez osób</i>
<b>Warunek początkowy</b>	Zabieg ma status „zgłoszony”.
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Przypadek użycia jest wywoływany z przypadku <i>Planuj zabieg pacjenta</i>.</li> <li>2. System odpytuje o przybliżony czas trwania zabiegu (w godzinach i minutach). Aktor wprowadza czas.</li> <li>3. System odpytuje o stopień trudności dla planowanego zabiegu. Aktor wprowadza stopień trudności.</li> <li>4. System podaje przedział cen dla danego rodzaju zabiegu. Aktor wprowadza cenę.</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	<ol style="list-style-type: none"> <li>2a. Przybliżony czas zabiegu wprowadzony został nieprawidłowo. System informuje o tym i odpytuje ponownie o czas zabiegu.</li> <li>3a. Wprowadzony stopień trudności nie jest liczbą naturalną z przedziału 1-3, system informuje o błędzie i ponownie odpytuje o stopień trudności.</li> <li>4a. Cena zabiegu nie mieści się w przedziale cen określonym dla danego rodzaju zabiegu, system informuje o błędzie i ponownie odpytuje o cenę.</li> </ol>
<b>Zakończenie</b>	Zgodnie ze scenariuszem.
<b>Warunek końcowy</b>	W systemie dla zabiegu zostaną zapamiętane następujące informacje: przybliżony czas trwania, stopień trudności i cena.

*Uwaga:* przypadek użycia *Ustal czas trwania, stopień trudności, cenę i utwórz zespół bez osób* umożliwia wprowadzenie przybliżonego czasu trwania, stopnia trudności i ceny oraz tworzy nową instancję klasy *Zespół* powiązaną z odpowiednią instancją klasy *Zabieg*. Natomiast przypadek użycia *Twórz skład zespołu zabiegowego* pozwala powiązać odpowiednie instancje klasy *Lekarz* z tą (już utworzoną) instancją klasy *Zespół* (Rys. 10-11).

Diagram ilustrujący podział przypadku użycia *Planuj zabieg pacjenta* na podprzypadki zgodny ze scenariuszem z Tab. 10-1 został zamieszczony na Rys. 10-7.



Rys. 10-7 Diagram z przykładowym podziałem przypadku *Planuj zabieg pacjenta* łącznie z przydzieleniem sali, z przydzieleniem co najmniej 2 lekarzy i lekarza odpowiedzialnego

**Polecenie:** Dla funkcjonalności *Przydziel lekarza do zespołu zabiegowego* (punkt 14.7 tekstu wymagań) napisz scenariusz.

W Tab. 10-2 umieszczony został przykładowy scenariusz dla przypadku użycia *Przydziel lekarza do zespołu zabiegowego*, przy założeniu, że przepływ główny dotyczy sytuacji, gdy przypadek użycia jest wywoływany bezpośrednio przez aktora *Naczelnego lekarza*.

Tab. 10-2 Scenariusz dla przypadku użycia *Przydziel lekarza do zespołu zabiegowego*

<b>Nazwa przypadku użycia</b>	Przydziel lekarza do zespołu zabiegowego
<b>Warunek początkowy</b>	Jeśli aktorem jest <i>Naczelnego lekarza</i> : zarejestrowany jest

	co najmniej jeden lekarz i co najmniej jeden zabieg o statusie w „trakcie planowania” lub „zaplanowany”. Jeśli aktorem jest przypadek <i>Twórz skład zespołu zabiegowego</i> : w składzie zespołu zabiegowego jest nie więcej niż trzech lekarzy.
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Aktor, którym jest <i>Naczelny lekarz</i> lub przypadek użycia <i>Utwórz zespół zabiegowy</i>, uruchamia przypadek użycia <i>Przydziel lekarza do zespołu zabiegowego</i>.</li> <li>2. System wyświetla listę zabiegów o statusie „w trakcie planowania” lub „zaplanowany” i z nie więcej niż trzema lekarzami w składzie zespołu. Aktor wybiera interesujący go zabieg.</li> <li>3. System wyświetla listę lekarzy dostępnych w terminie planowanego zabiegu. Aktor wybiera lekarza.</li> <li>4. System wyświetla listę ról możliwych jeszcze do pełnienia w tworzonym zespole. Aktor wybiera rolę.</li> <li>5. System informuje o przypisaniu lekarza do zespołu zabiegowego (łącznie z przypisaną mu rolą).</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	<ol style="list-style-type: none"> <li>2a. Aktorem jest <i>Naczelny lekarz</i>, a brak zabiegów o statusie „w trakcie planowania” lub „zaplanowany” lub gdy dla wszystkich zabiegów o takim statusie zostało przypisanych czterech lekarzy. System informuje o tym i kończy przypadek użycia.</li> <li>2b. Aktorem jest przypadek użycia <i>Twórz skład zespołu zabiegowego</i>, system przechodzi do punktu 3 w przebiegu głównym.</li> <li>3a. Brak lekarzy dostępnych w terminie planowanego zabiegu lub zabieg ma być przeprowadzony w znieczuleniu ogólnym, do zespołu przypisano już trzech lekarzy, z których żaden nie jest anestezjologiem, a brak anestezjologów dostępnych w terminie planowanego zabiegu. System informuje o niemożności zrealizowania przydziału.</li> <li>3aa. Aktorem jest <i>Naczelny lekarz</i>, system pyta, czy kontynuować dla innego zabiegu.</li> <li>3aaa. Aktor potwierdza kontynuację, system powraca do punktu 2 w przebiegu głównym.</li> <li>3aab. Aktor rezygnuje z kontynuowania</li> </ol>

	<p>przypadku użycia.</p> <p>3ab. Aktorem jest przypadek użycia <i>Twórz skład zespołu zabiegowego</i>, system kończy przypadek.</p> <p>4a. Zabieg ma być przeprowadzony w znieczuleniu ogólnym, do zespołu przypisano już trzech lekarzy i żaden z nich nie jest anestezjologiem, a wybrany lekarz też nie jest anestezjologiem. System informuje o tym i pyta, czy wybrać innego lekarza.</p> <p>4aa. Aktor potwierdza chęć wyboru innego lekarza, system wraca do punktu 3 w przebiegu głównym.</p> <p>4ab. Aktor rezygnuje z kontynuowania przypadku, system kończy przypadek użycia.</p>
<b>Zakończenie</b>	Zgodnie ze scenariuszem
<b>Warunek końcowy</b>	<p>Jeżeli proces przydziału zakończy się pozytywnie, do zespołu zostanie dopisany lekarz, a także zostanie ustalona rola, jaką ma pełnić w zespole.</p> <p>Jeżeli zabieg w chwili rozpoczęcia przypadku użycia miał status „w trakcie planowania”, a w wyniku przydziału lekarza zostały spełnione minimalne warunki dla zaplanowania zabiegu, to status zabiegu zostaje zmieniony na „zaplanowany”.</p>

#### 10.2.4 Omówienie typowych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

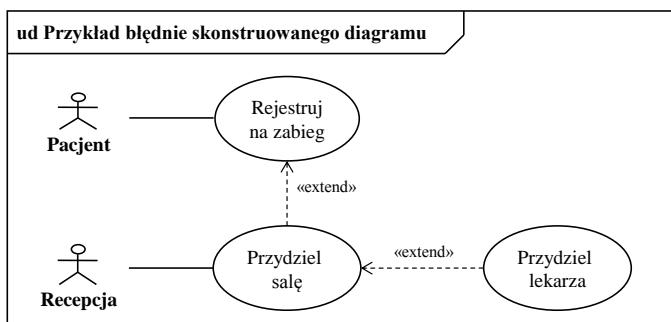
- Zdarza się, że nazwa przypadku nie jest nazwą zadania zlecanego systemowi, ale określa np. dane czy też czynności realizowane „poza systemem” (tzn. bez wsparcia ze strony systemu), jak np. nazwa przypadku taka jak *Zabieg* – trudno się tu zorientować, jaki ma być *obserwowały rezultat* tego przypadku? To samo dotyczy przypadków umieszczenych na Rys. 10-8. *Termin* – nazwa przypadku opisuje dane, a nie czynność; *Hospitalizuj pacjenta*, *Wykonaj zabieg* – czynności realizowane poza systemem.
- Błędna konstrukcja diagramu może skutkować wprowadzaniem informacji niezgodnych z tekstem wymagań i sprzecznych nawet ze stosunkowo ogólną wiedzą posiadaną o danej dziedzinie problemowej. Na Rys. 10-9 pacjent nie dość, że może sam zarejestrować się na zabieg, to może jeszcze przydzielić salę

oraz lekarzy do zabiegu. Podobnie, przydzielenie lekarzy do zabiegu nie powinno być funkcją poboczną dla przydzielania sali, ale ewentualnie dla rejestrowania zabiegu. Również nie wydaje się, aby osoby pracujące w recepcji posiadały wiedzę na poziomie koniecznym do przydzielania lekarzy do wykonania zabiegu.



Rys. 10-8 Przykłady błędnie nazwanych przypadków użycia

- Niezrozumienie sekcji *Warunek wstępny* w dokumentacji przypadku. W tej sekcji umieszczane są warunki nakładane na „świat przed wywołaniem przypadku”. Takie warunki są zazwyczaj sprawdzane na początku realizowania danego przypadku, jeszcze przed rozpoczęciem dialogu aktora z systemem. Czyli warunek wstępny sformułowany następująco: *pacjent musi być zarejestrowany w systemie* nie jest możliwy do sprawdzenia przed uzyskaniem informacji o tym, którego konkretnie pacjenta zabieg ma dotyczyć, taka informacja jest pozyskiwana dopiero w trakcie dialogu aktora z systemem.



Rys. 10-9 Przykład błędnie skonstruowanego diagramu

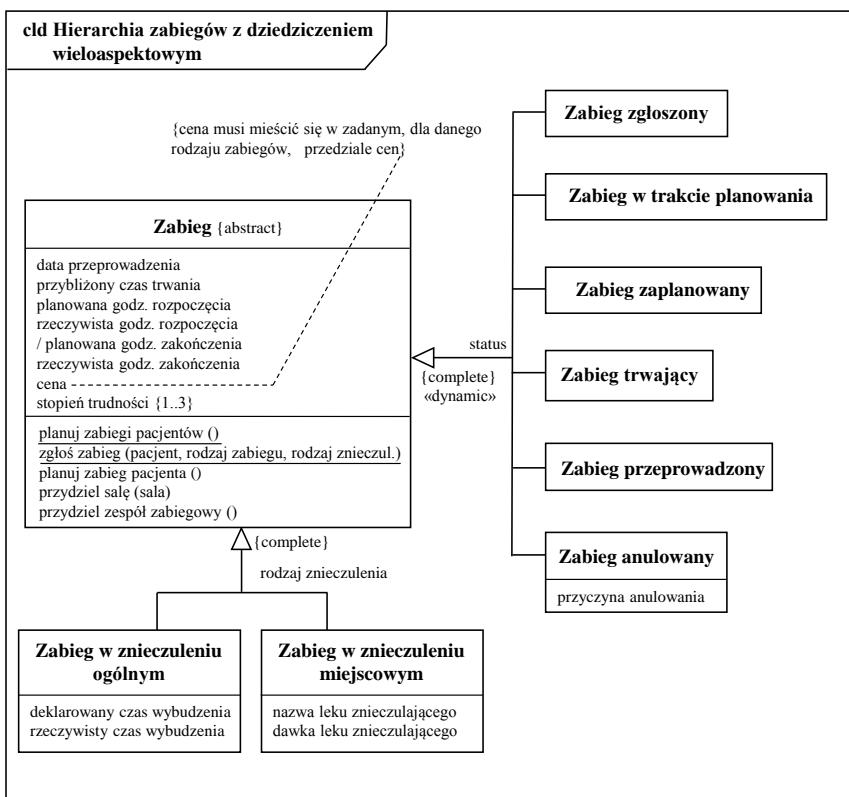
- Stosunkowo częstym błędem jest też umieszczanie kilku przypadków w jednym scenariuszu – mimo że polecenie dotyczy jednego przypadku i na diagramach nie widać przesłanek do takiego postępowania, jak np. w sytuacji, gdy aktor *Rejestracja* rezerwuje salę na zabieg, aktor *Dyrektor* przydziela lekarzy, a aktor *Podsystem czasu* rozwiązuje zespół.

## 10.3 Analiza strukturalna

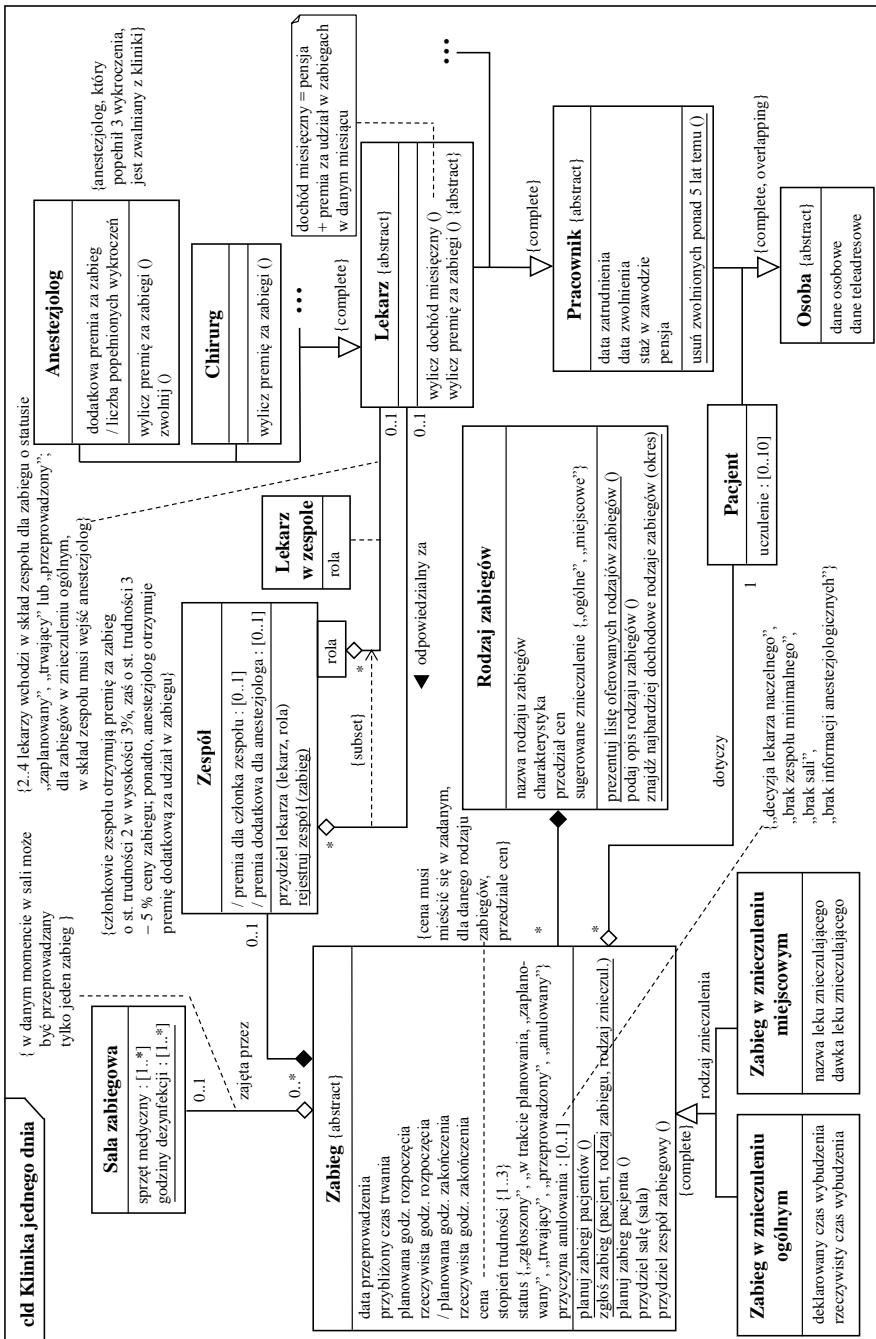
### 10.3.1 Schemat pojęciowy

*Polecenie:* Dla podanych założeń skonstruuj diagram klas (nie redukując liczności).

Schemat pojęciowy [35, 40, 41] dla systemu *Klinika jednego dnia* został przedstawiony na Rys. 10-11. Ze względu na atrybut *status* możliwe byłoby sformułowanie innego podziału zabiegów (z dziedziczeniem dynamicznym), co prowadziłoby do wykorzystania dziedziczenia wieloaspektowego dla modelowania hierarchii zabiegów – jak na Rys. 10-10. To rozwiązanie nie będzie jednak wykorzystywane w dalszej części rozdziału.



Rys. 10-10 Hierarchia zabiegów z rozwiniętym aspektem *status*



Rys. 10-11 Schemat pojęciowy dla systemu *Klinika jednego dnia*

### 10.3.2 Analiza elementów pochodnych

Na schemacie pojęciowym na Rys. 10-11 zostały zidentyfikowane następujące atrybuty pochodne:

- *premia dla członka w zespole* w klasie *Zespół* (opcjonalny);
- *premia dodatkowa dla anestezjologa* w klasie *Zespół* (opcjonalny);
- *liczba wykroczeń* w klasie *Anestezjolog*;
- *planowana godz. zakończenia* w klasie *Zabieg*.

Wartość atrybutu *premia dla członka zespołu* jest obliczana (w oparciu o reguły podane w punkcie 8 tekstu wymagań) wyłącznie dla zabiegów o stopniu trudności 2 lub 3 (dlatego jest to dodatkowo także atrybut opcjonalny), na podstawie wartości atrybutów *stopień trudności* oraz *cena zabiegu* w klasie *Zabieg*. Lekarz anestezjolog – zgodnie z punktem 9 tekstu wymagań – otrzymuje jeszcze premię dodatkową, której wysokość nie jest wliczana do wysokości atrybutu *premia dla członka zespołu*. Od momentu ustalenia wartości premii dla określonego zabiegu ta wartość już nie powinna ulegać zmianom.

Wprowadzenie do schematu pojęciowego tylko metody (bez atrybutu) w klasie *Zespół* wyliczającej wartość premii nie byłoby wystarczające. W przypadku zmiany stawki przyznawanej premii przypadającej na lekarza – uczestnika zespołu, dla zabiegów o odpowiednim stopniu trudności (np. dla zabiegów o stopniu trudności 3 mogłyby nastąpić zmiana z 5% na 6%) taka metoda nie wyliczałaby faktycznej premii dla zabiegu wykonanego pewien czas temu, lecz wyliczałaby jej wysokość według stawki obowiązującej w danym momencie. Wynika z tego, że atrybut *premia dla członka zespołu* w klasie *Zespół* nie może pozostać atrybutem pochodnym, lecz powinien być atrybutem bazowym.

Podobna sytuacja dotyczy atrybutu *premia dodatkowa dla anestezjologa* w klasie *Zespół*, którego wartość jest wyliczana na podstawie atrybutu *dodatkowa premia za zabieg* w klasie *Anestezjolog*. Zgodnie ze schematem pojęciowym na Rys. 10-11, jeśli dyrektor kliniki zmieniłaby wysokość *dodatkowej premii za zabieg* dla ustalonego anestezjologa, to odpowiednio zmieniłaby się także wysokość rozważanego elementu pochodnego dla wszystkich zabiegów z udziałem tego anestezjologa. W związku z tym, atrybut *premia dodatkowa dla anestezjologa* w klasie *Zespół* również nie może pozostać atrybutem pochodnym.

Trzeci rozważany element pochodny – atrybut *liczba wykroczeń* jest wyliczany dla danego anestezjologa na podstawie łącznej liczby stwierdzonych rozbieżności między deklarowanym a rzeczywistym czasem wybudzenia pacjenta dla zabiegów przeprowadzonych w znieczuleniu ogólnym przy udziale w zespole zabiegowym tego anestezjologa oraz przypadków niedostarczenia (w dopuszczalnym terminie) informacji anestezjologicznych. Wartość rozważanego atrybutu pochodnego jest ustalana w ścisłe określonej sytuacji (po każdej zakończonej niepowodzeniem deklaracji czasu wybudzenia lub automatycznie – po przekroczeniu dopuszczalnego terminu uzupełnienia informacji anestezjologicznych). Wraz ze wzrostem zabiegów

z udziałem danego anestezjologa takie rozwiązanie może być korzystniejsze obliczeniowo. Prostszym rozwiązaniem – z punktu widzenia implementacji – będzie zatem, jeśli atrybut ten będzie atrybutem bazowym zainicjalizowanym początkową wartością równą 0.

Atrybut *planowana godz. zakończenia* jest wyliczany na podstawie atrybutów *planowana godz. rozpoczęcia* i *przybliżony czas trwania* w klasie *Zabieg*. Ponieważ wartości obu atrybutów bazowych będą raczej rzadko zmieniane podczas planowania zabiegu, to atrybut *planowana godz. zakończenia* pozostanie pochodnym.

Schemat pojęciowy uwzględniający wnioski z analizy elementów pochodnych został przedstawiony na Rys. 10-12.

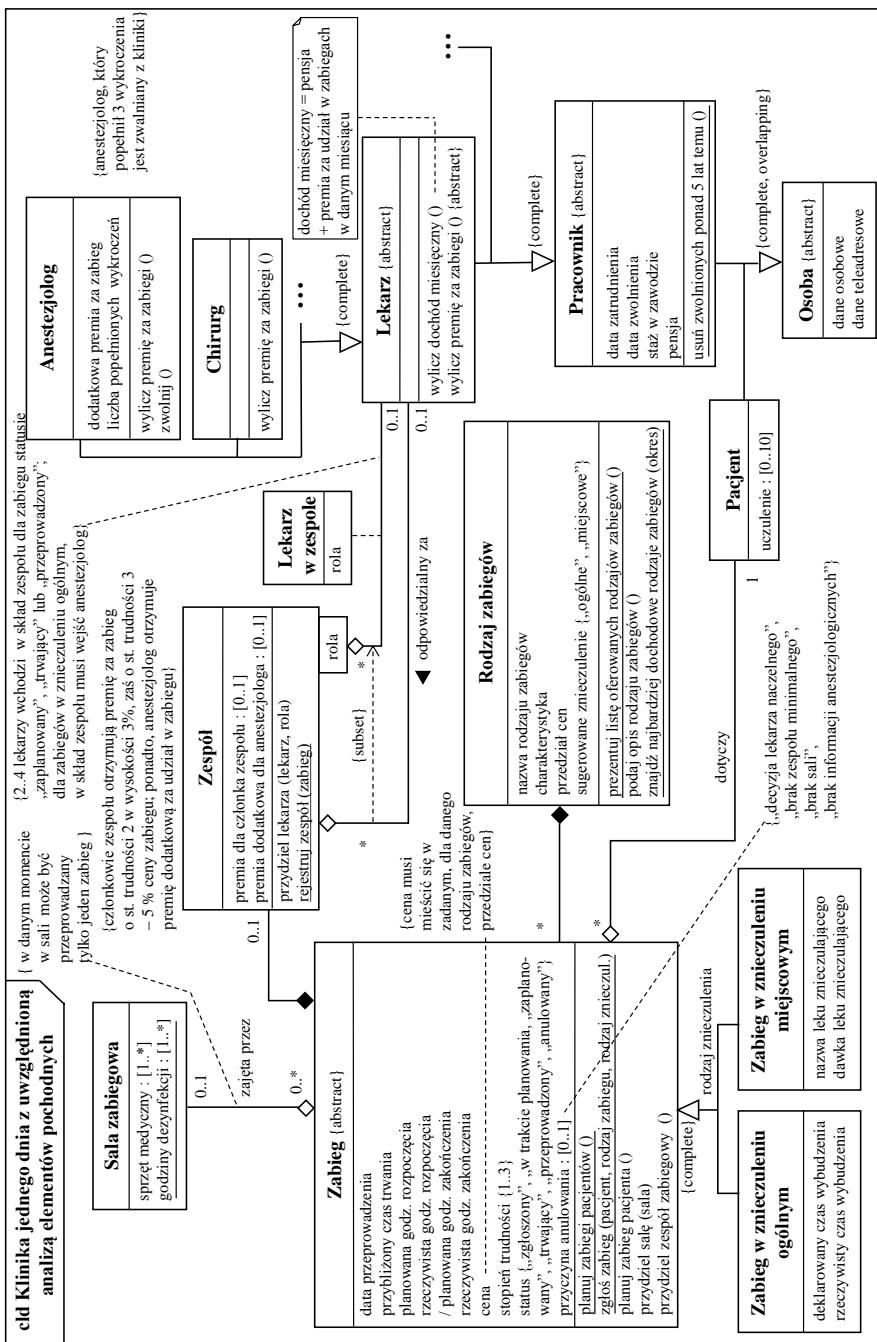
### 10.3.3 Analiza wartości początkowych

Na schemacie pojęciowym z Rys. 10-12 można określić wartości początkowe:

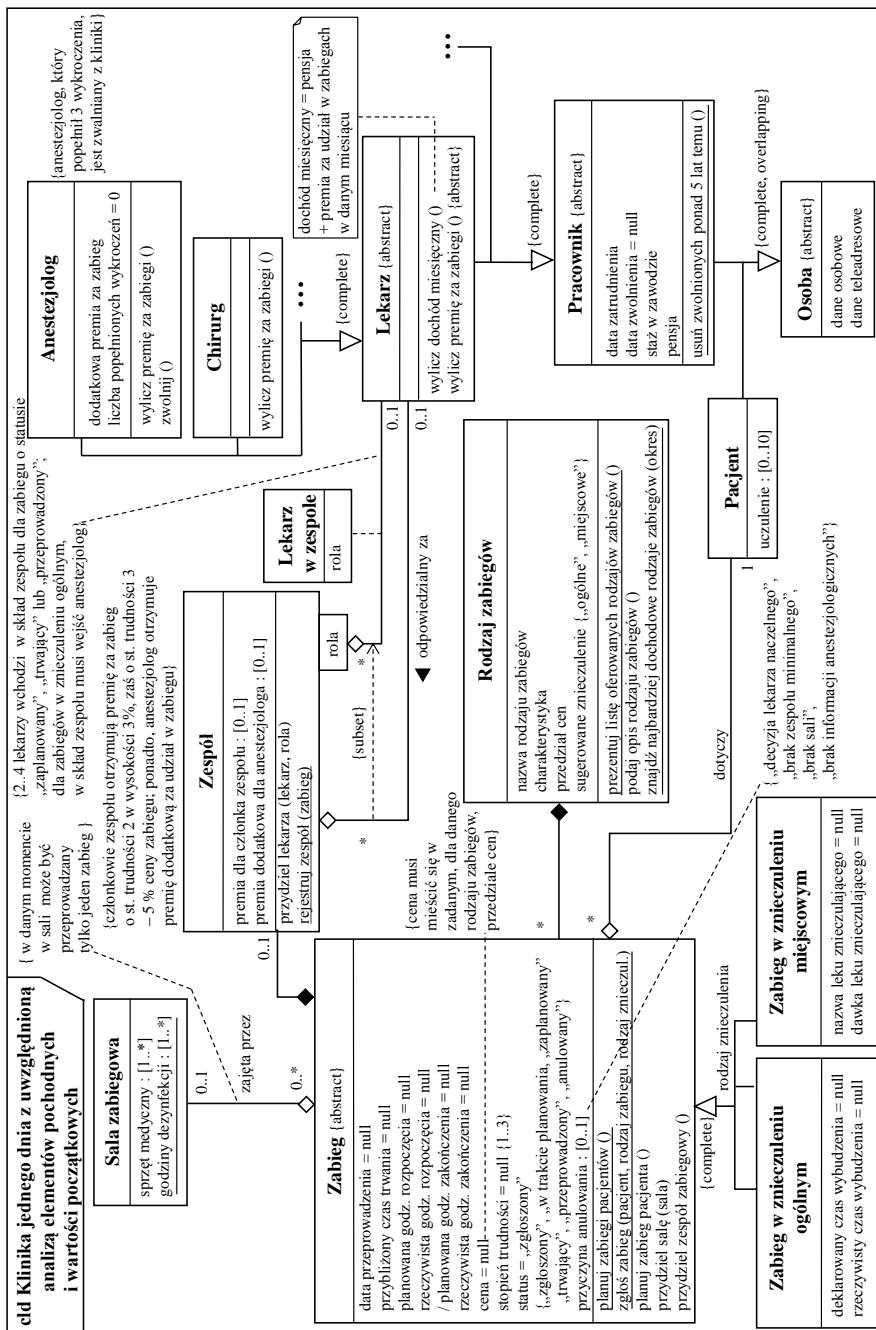
- null dla atrybutu *data przeprowadzenia* w klasie *Zabieg*;
- null dla atrybutu *przybliżony czas trwania* w klasie *Zabieg*;
- null dla atrybutu *planowana godz. rozpoczęcia* w klasie *Zabieg*;
- null dla atrybutu *rzeczywista godz. rozpoczęcia* w klasie *Zabieg*;
- null dla atrybutu *planowana godz. zakończenia* w klasie *Zabieg*;
- null dla atrybutu *rzeczywista godz. zakończenia* w klasie *Zabieg*;
- null dla atrybutu *cena* w klasie *Zabieg*;
- null dla atrybutu *stopień trudności* w klasie *Zabieg*;
- „*zgłoszony*” dla atrybutu *status* w klasie *Zabieg*;
- null dla atrybutu *deklarowany czas wybudzenia* w klasie *Zabieg w znieczuleniu ogólnym*;
- null dla atrybutu *rzeczywisty czas wybudzenia* w klasie *Zabieg w znieczuleniu ogólnym*;
- null dla atrybutu *nazwa leku znieczulającego* w klasie *Zabieg w znieczuleniu miejscowym*;
- null dla atrybutu *dawka leku znieczulającego* w klasie *Zabieg w znieczuleniu miejscowym*;
- null dla atrybutu *data zwolnienia* w klasie *Pracownik*;
- oraz 0 dla atrybutu *liczba wykroczeń* w klasie *Anestezjolog*.

Natomiast nie zostały zdefiniowane metody z parametrami o określonych wartościach domyślnych (początkowych).

Schemat pojęciowy uwzględniający wnioski z analizy wartości początkowych został zilustrowany na Rys. 10-13.



Rys. 10-12 Schemat pojęciowy po uwzględnieniu analizy elementów pochodnych



Rys. 10-13 Schemat pojęciowy po uwzględnieniu analizy elementów pochodnych i wartości początkowych

### 10.3.4 Analiza wartości granicznych

Dalsza analiza schematu pojęciowego (z Rys. 10-13) powinna odpowiedzieć na pytanie czy na schemacie występują wartości graniczne, dla których podczas ewentualnej eksploatacji systemu, wraz z upływem czasu, mogłyby wyniknąć potrzeba zmiany ich wartości i jakie to miałyby znaczenie dla innych elementów modelu. W szczególności należy zastanowić się, czy w wyniku zmiany wartości granicznych wymagane byłyby jakiekolwiek zmiany w kodzie.

Syntetyczne zestawienie wybranych wartości granicznych przedstawiono w Tab. 10-3.

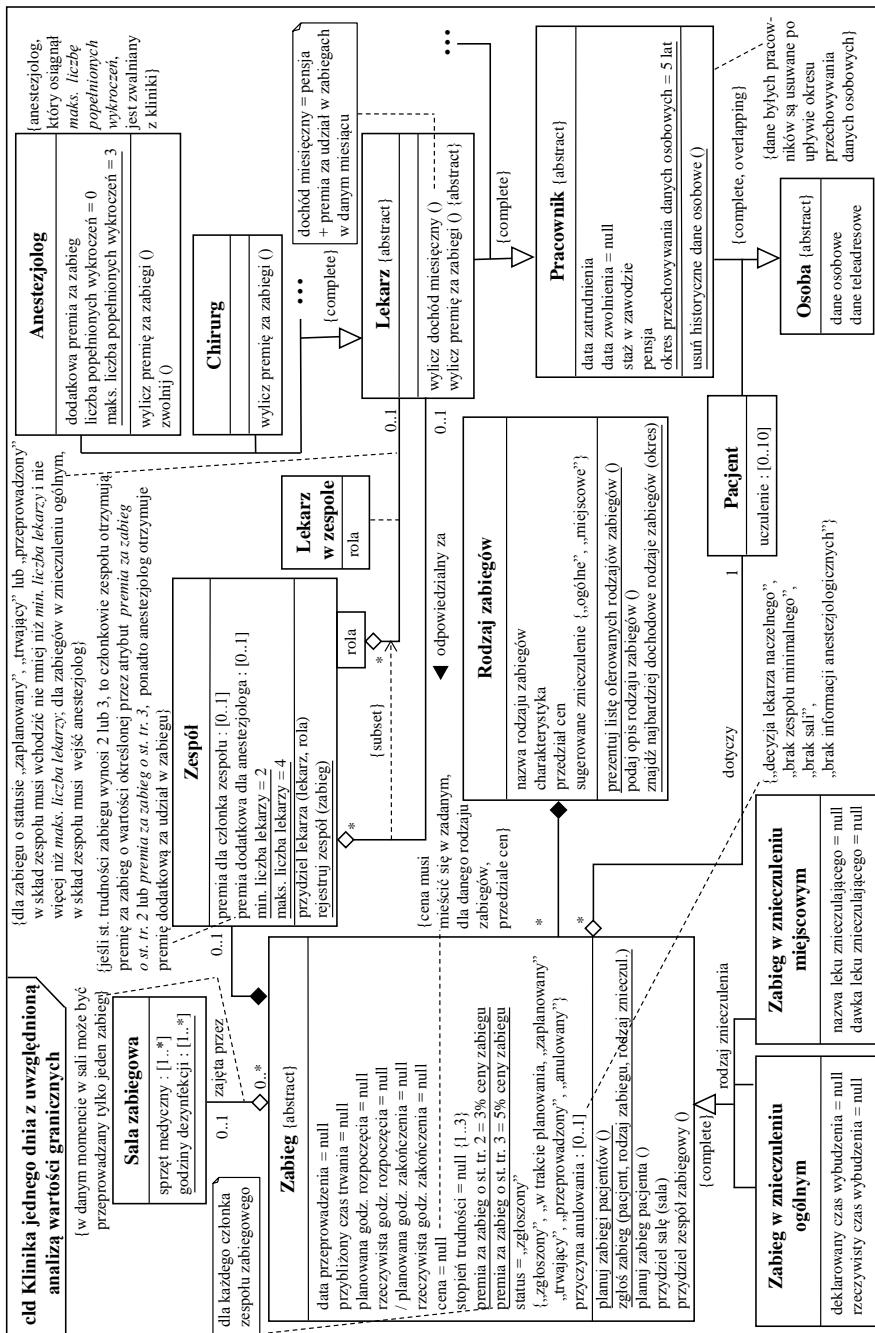
Tab. 10-3 Zestawienie wartości granicznych, ograniczeń i proponowanych zmian w schemacie pojęciowym

Wartość graniczna	Ograniczenie	Zmiany w schemacie pojęciowym
2-4 lekarzy	<p>{2..4 lekarzy wchodzi w skład zespołu dla zabiegu o statusie „zaplanowany”, „trwający” lub „przeprowadzony”}</p> <p>nałożone na agregację z klasą asocjacji <i>Lekarz w zespole</i> między klasą <i>Zespół</i> a klasą <i>Lekarz</i></p>	<p>dodatkowe atrybuty o zasięgu klasowym z zainicjalizowanymi wartościami w klasie <i>Zespół</i>:</p> <p><u>min. liczba lekarzy = 2</u></p> <p><u>maks. liczba lekarzy = 4</u></p> <p>zmiana ograniczenia do postaci</p> <p>{dla zabiegu o statusie „zaplanowany”, „trwający” lub „przeprowadzony” w skład zespołu musi wchodzić nie mniej niż min. liczba lekarzy i nie więcej niż maks. liczba lekarzy}</p>
3, liczba popelnionych wykroczeń	<p>{anestezjolog, który popełnił 3 wykroczenia, jest zwalniany z kliniki}</p> <p>nałożone na obiekt <i>Anestezjolog</i></p>	<p>dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością w klasie <i>Anestezjolog</i>:</p> <p><u>maks. liczba popełnionych wykroczeń = 3</u></p> <p>zmiana ograniczenia do postaci</p> <p>{anestezjolog, który osiągnął maks. liczbę popełnionych wykroczeń, jest zwalniany z kliniki}</p>
5 lat	<p>{dane byłych pracowników są usuwane po 5 latach od zwolnienia}</p> <p>jest zawarte w specy-</p>	<p>dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością w klasie <i>Pracownik</i>:</p> <p><u>okres przechowywania danych</u></p>

	fikacji metody <u><i>usuń zwolnionych ponad 5 lat temu ()</i></u> w klasie Pracownik  nałożone na obiekt klasy Pracownik	<u><i>osobowych = 5 lat</i></u>  zmiana specyfikacji metody <u><i>usuń historyczne dane osobowe ()</i></u> , z uwzględnieniem dodatkowego ograniczenia <i>{dane byłych pracowników są usuwane po upływie okresu przechowywania danych osobowych}</i> i uzależnienie jej implementacji od dodatkowego atrybutu
3%, wysokość % premii dla lekarza za udział w zabiegu o stopniu trudności 2	{członkowie zespołu otrzymują premię za zabieg o st. trudności 2 w wysokości 3%}  nałożone na atrybut premia na członka zespołu w klasie Zespół	dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością w klasie Zabieg: <u><i>premia za zabieg o st. tr. 2 = 3 % ceny zabiegu</i></u>  zmiana ograniczenia do postaci {jeśli st. trudności zabiegu wynosi 2, to członkowie zespołu otrzymują premię o wartości określonej przez atrybut premia za zabieg o st. tr. 2}
5%, wysokość % premii dla lekarza za udział w zabiegu o stopniu trudności 3	{członkowie zespołu otrzymują premię za zabieg o st. trudności 3 w wysokości 5%}  nałożone na atrybut premia na członka zespołu w klasie Zespół	dodatkowy atrybut o zasięgu klasowym z zainicjalizowaną wartością w klasie Zabieg: <u><i>premia za zabieg o st. tr. 3 = 5 % ceny zabiegu</i></u>  zmiana ograniczenia do postaci {jeśli st. trudności zabiegu wynosi 3, to członkowie zespołu otrzymują premię o wartości określonej przez atrybut premia za zabieg o st. tr. 3}

Schemat pojęciowy uwzględniający proponowane zmiany po analizie wartości granicznych został zaprezentowany na Rys. 10-14.

Przyjęte zostało założenie, że zgodnie z polityką kliniki, termin dla anulowania zabiegu o statusie „zgłoszony” lub „w trakcie planowania” *na trzy dni przed jego planowaną datą przeprowadzenia* nie ulegnie zmianie w przyszłości.

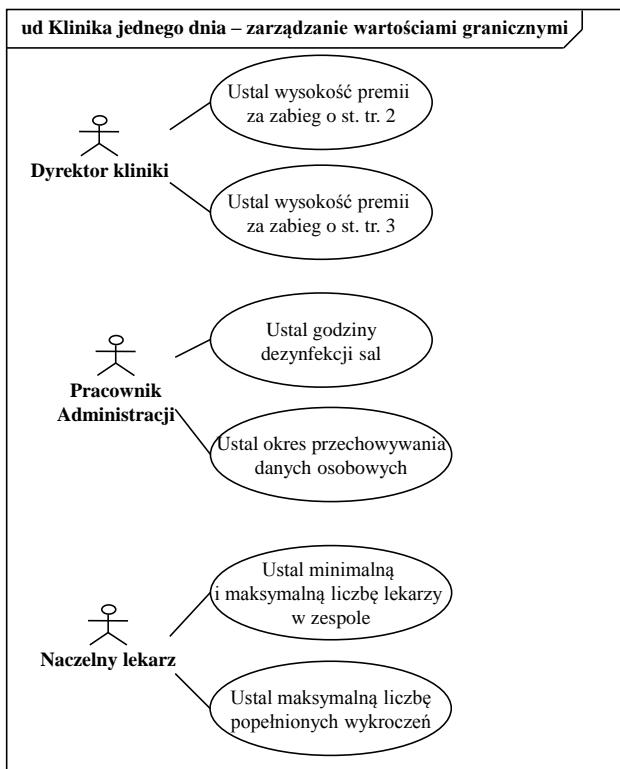


Rys. 10-14 Schemat pojęciowy po analizie elementów pochodnych, wartości początkowych oraz wartości granicznych

Aby w pełni można było wykorzystać zaproponowane wartości graniczne oraz nałożone na nie ograniczenia, należałoby dodatkowo wyspecyfikować wymagania użytkownika, które określałyby kto i za pomocą jakiej funkcjonalności byłby odpowiedzialny za zarządzanie tymi wartościami granicznymi. Rys. 10-15 przedstawia przykładowe przypadki użycia niezbędne dla zarządzania wartościami granicznymi określonymi w Tab. 10-3. Przypadek użycia *Ustal godziny dezynfekcji sal* został już wprowadzony w modelu przypadków użycia na Rys. 10-2, ale została powtórzony na Rys. 10-15 dla pełności rozważań. Na podstawie tej propozycji należałoby wprowadzić do schematu pojęciowego z Rys. 10-14 następujące metody:

- *ustal wysokość premii za zabieg o st. tr. 2 ()* oraz *ustal wysokość premii za zabieg o st. tr. 3 ()* w klasie *Zabieg*;
- *ustal godziny dezynfekcji sal ()* w klasie *Sala zabiegowa*;
- *ustal okres przechowywania danych osobowych ()* w klasie *Pracownik*;
- *ustal minimalną i maksymalną liczbę lekarzy w zespole* w klasie *Zespół*;
- oraz *ustal maksymalną liczbę popełnionych wykroczeń ()* w klasie *Anestezjolog*.

W dalszej części rozdziału nie będą rozważane przypadki użycia z Rys. 10-15.



Rys. 10-15 Diagram z przykładowymi przypadkami użycia dla zarządzania wartościami granicznymi z Tab. 10-3

### 10.3.5 Zadania podstawowe

Zadania omówione w tym rozdziale zostały zrealizowane w oparciu o schemat pojęciowy z Rys. 10-14.

**Zad. 1:** Podkreśl te rodzaje dziedziczenia (rozłączne, nierozłączne, kompletne, niekompletne, jednokrotne, wielokrotne, jednoaspektowe, wieloaspektowe, dynamiczne, elipsa), które uważaś za potrzebne do wykorzystania w schemacie pojęciowym. Dla każdego podkreślonego rodzaju dziedziczenia podaj hierarchię/aspekt, w której ten rodzaj dziedziczenia wystąpił.

- Rozłączne (ang. *disjoint*): hierarchia dla pracowników i hierarchia dla zabiegów;
- nierozłączne (ang. *overlapping*): hierarchia dla osób;
- kompletne: wszystkie hierarchie;
- jednokrotne: wszystkie hierarchie;
- jednoaspektowe: wszystkie hierarchie;
- elipsa: hierarchia dla pracowników i hierarchia dla lekarzy.

**Zad. 2:** Podaj przykłady metod abstrakcyjnych i zaimplementowanych.

- Metoda abstrakcyjna w klasie abstrakcyjnej: metoda *wylicz premię za zabiegi ()* w klasie *Lekarz*.  
*Uwaga:* przy założeniu, że dla wszystkich lekarzy poza anestezjologami ich dochód miesięczny jest wyliczany na podstawie pensji oraz premii za udział w zabiegach w danym miesiącu, lepiej byłoby, aby metoda *wylicz premię za zabiegi ()* w klasie *Lekarz* była metodą zaimplementowaną i przesłoniętą w klasie *Anestezjolog*;
- metoda abstrakcyjna w klasie konkretnej: nie można definiować takich metod;
- metoda zaimplementowana w klasie abstrakcyjnej: metoda *wylicz dochód miesięczny ()* w klasie *Lekarz*;
- metoda zaimplementowana w klasie konkretnej: *wylicz premię za zabiegi ()* w klasie *Chirurg*.

**Zad. 3:** Podaj przykłady metody obiektu i metody klasowej oraz atrybutu obiektu i atrybutu klasowego.

- Metoda obiektu: metoda *wylicz premię za zabiegi ()* w klasie *Lekarz*;
- metoda klasowa: *znajdź najbardziej dochodowe rodzaje zabiegów (okres)* – operuje na ekstensji klasy *Rodzaj Zabiegów*. Metodą klasową jest też metoda *zgłoś zabieg (pacjent, rodzaj zabiegu, rodzaj znieczul.)* – tworzy nowy obiekt klasy *Zabieg*;
- atrybut obiektu: *cena* w klasie *Zabieg*;
- atrybut klasowy: *godziny dezynfekcji* w klasie *Sala zabiegowa*.

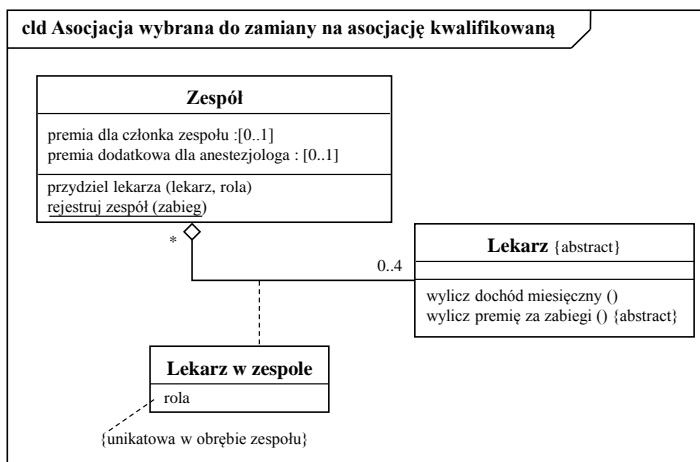
**Zad. 4:** Wyjaśnij pojęcie polimorfizmu metod i wskaż przykład jego zastosowania.

Metody polimorficzne to metody *wylicz premię za zabiegi ()* w klasach hierarchii zbudowanej dla lekarzy.

**Zad. 5:** Asocjację posiadającą niesymetryczne liczności oraz atrybut (lub klasę asocjacji) zamień na asocjację kwalifikowaną. Uzasadnij, dlaczego poddałeś zamianie właśnie tę asocjację.

Na diagramie umieszczono asocjację kwalifikowaną pomiędzy klasami *Zespół* i *Lekarz*. Wskazuje ona, że kluczem do przeszukiwania zbioru lekarzy przypisanych do danego zespołu może być rola, jaką każdy z lekarzy pełni w danym zespole. Rys. 10-16 przedstawia fragment diagramu sprzed zamiany asocjacji „zwykłej” na asocjację kwalifikowaną.

*Uwaga:* Asocjację kwalifikowaną wykorzystuje się do wskazania klucza użytecznego do przeszukiwania dużych zbiorów danych (a nie zbiorów o niewielkiej liczności). W tym konkretnie przypadku, przeszukiwany jest niezbyt duży zbiór (maksymalnie 4 elementów), ale sprawdzamy tu znajomość reguł transformacji. Wprowadzenie asocjacji kwalifikowanej zmienia licznosć tego końca asocjacji, który znajduje się naprzeciwko kwalifikatora. Nowa liczność bezpośrednio wskazuje „siłę” klucza, czyli np. liczność = 1 mówi o tym, że wartość kwalifikatora pozwala na jednoznaczną identyfikację lekarza w ramach danego zespołu. Nie ma sensu natomiast wprowadzenie asocjacji kwalifikowanej między klasami *Zabieg* i *Zespół*, ponieważ obie role tej asocjacji są zbiorami o liczności równej 1.



Rys. 10-16 Fragment diagramu bez wykorzystania asocjacji kwalifikowanej

**Zad. 6:** Wskaż asocjację, która jest agregacją lub kompozycją (lub może kwalifikować się do zamiany na takową). Objasnij, dlaczego zdecydowałeś się na wybór agregacji/kompozycji, a nie „zwykłej” asocjacji. Ponadto, dlaczego wybrałeś agregację, a nie kompozycję (czy też odwrotnie)?

Na schemacie pojęciowym występuje kilka agregacji: np. agregacja kwalifikowana (z nazwą domyślną *składa się/wchodzi w skład*) pomiędzy klasami: *Zespół* i *Lekarz*. Agregacja ta, lepiej niż zwykła asocjacja, wskazuje na związek część-całość występujący pomiędzy zespołem a lekarzami wchodzącymi w jego skład. Wykorzystano agregację, a nie kompozycję, ponieważ w tym przypadku cykl życiowy części nie zawiera się w cyklu życiowym całości, tzn. usunięcie z systemu obiektu klasy *Zespół* nie będzie skutkowało usunięciem informacji o lekarzach, którzy byli członkami danego zespołu. Ponadto, wraz z upływem czasu każdy z lekarzy mógł być członkiem wielu zespołów – stąd liczność \* dla roli asocjacji odnoszącej się do klasy *Zespół*.

**Zad. 7:** Wskaż przykładowe ograniczenie. Na jaki element zostało ono nałożone? Czy jest to ograniczenie statyczne czy dynamiczne? Zaproponuj, jaka metoda i w jakiej klasie mogłyby być odpowiedzialna za jego zapewnienie.

Na przykład ograniczenie, że *cena musi mieścić się w zadanych, dla danego rodzaju zabiegów, przedziale cen*. Jest to ograniczenie statyczne, nałożone na atrybut *cena* w klasie *Zabieg*. Ograniczenie to mogłyby być zrealizowane np. w metodzie *planuj zabieg pacjenta()* w klasie *Zabieg*. Z kolei ograniczenie brzmiące: *dla zabiegów w znieczuleniu ogólnym, w skład zespołu musi wchodzić anestezjolog*, jest ograniczeniem dynamicznym, nałożonym na asocjację pomiędzy klasami *Zespół* i *Lekarz*. Ograniczenie to mogłyby być zrealizowane np. w metodzie *przydziel lekarza(lekarz, rola)* w klasie *Zespół*.

*Dla przypomnienia:* w przypadku ograniczeń dynamicznych ważny jest poprzedni stan elementu (tu zbioru lekarzy powiązanych z danym zespołem), na który jest nakładane ograniczenie.

### **10.3.6 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się**

- Częstym błędem jest wprowadzanie do diagramów informacji, które ani bezpośrednio, ani pośrednio nie zostały umieszczone w tekście wymagań, jak np. *imiona* i *nazwiska* czy *telefony osób* itp. Należało, zgodnie z tekstem wymagań, wprowadzić do klasy *Osoba* atrybuty takie jak: *dane osobowe* i *dane teleadresowe*. Uszczegóławianie tych atrybutów to zadanie późniejsze. Za informację nadmiarową uważałyśmy też np. atrybut *stanowisko* wprowadzony do klasy *Pracownik*, o którym również nie było w ogóle mowy w tekście wymagań.
- Zapominanie o tym, że nazwa klasy to zazwyczaj rzeczownik w liczbie pojedynczej, czyli zgodnie z tekstem wymagań (w punkcie 10.1) klasa modelująca zabiegi powinna nazywać się *Zabieg*, a nie *Zabiegi*.
- Podobnie, nie *Znieczulenie ogólne*, ale *Zabieg w znieczuleniu ogólnym*. Ponadto, klasa nazwana *Znieczulenie ogólne* w ogóle nie powinna dziedziczyć po klasie *Zabieg*, ponieważ znieczulenie ogólne (jak również znieczulenie miejscowe) nie

jest szczególnym przypadkiem/rodzajem zabiegu, tylko jest szczególnym przypadkiem/rodzajem znieczulenia. Podobnie, klasa *Zabieg* nie może dziedziczyć po klasie *Rodzaj zabiegów* (ani nawet *Rodzaj zabiegu*), ponieważ zabieg nie jest szczególnym przypadkiem (specjalizacją) rodzaju zabiegów. Obie te klasy należało połączyć nie związkiem generalizacji/specjalizacji, ale związkiem kompozycji. To samo dotyczy nieprawidłowo wprowadzonego dziedziczenia klasy *Lekarz* po klasie *Pacjent* – prawdopodobnie w celu zapewnienia wymagania, że *lekarz kliniki może być także jej pacjentem*. Lekarz nie jest specjalnym przypadkiem/rodzajem pacjenta! Należało wykorzystać dziedziczenie nierożłączne, które pozwoliłoby na modelowanie informacji o tym, że dana osoba może pełnić obie te role: i lekarza, i pacjenta.

### **10.3.7 Schemat projektowy uwzględniający ograniczenia obiektowego środowiska implementacji**

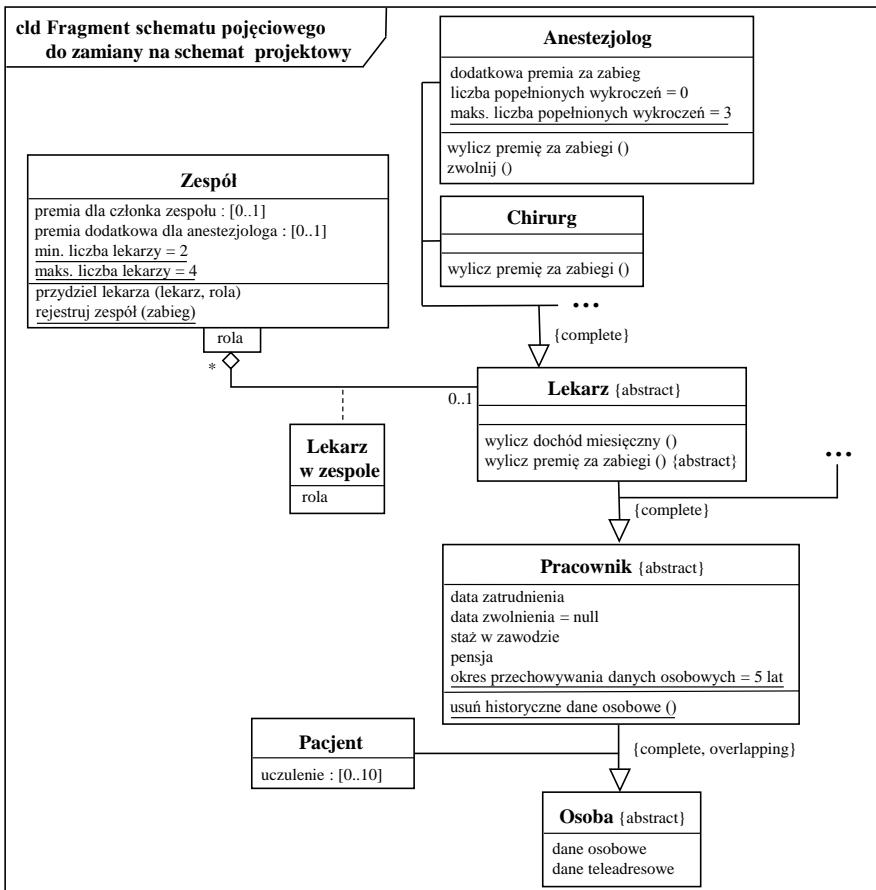
Fragment schematu pojęciowego (z pominięciem ograniczeń i komentarzy), który jest niezbędny do zamodelowania implementacji metody *Przydziel lekarza (lekarz, rola)* w klasie *Zespół*, jest przedstawiony na Rys. 10-17. Zawiera klasy dla hierarchii osób, klasę *Zespół* oraz agregację kwalifikowaną z klasą asocjacji *Lekarz w zespole*.

Zakładając, że środowiskiem implementacji jest popularny język programowania obiektowego (np. Java), koniecznym zadaniem do wykonania jest realizacja schematu projektowego [37, 40] zawierającego tylko takie elementy, które mają swoje bezpośrednie odpowiedniki w postaci natywnych konstrukcji języka obiektowego.

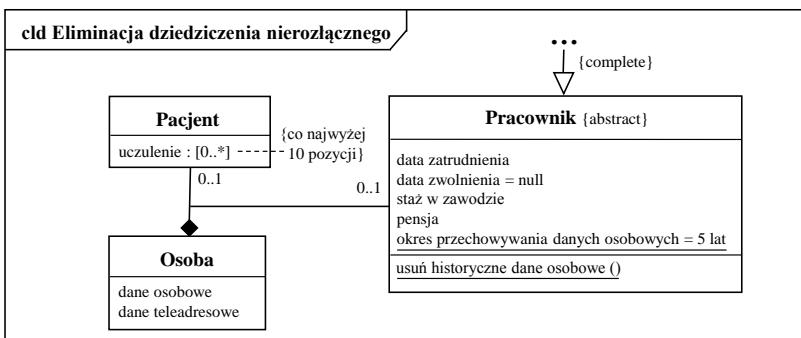
Pierwszym etapem transformacji fragmentu modelu pojęciowego jest eliminacja dziedziczenia nierożłącznego dla hierarchii osób. Dziedziczenie zostało zastąpione przez kompozycję jak na Rys. 10-18. Ponieważ atrybut powtarzalny *uczulenie* w klasie *Pacjent* zostanie prawdopodobnie zaimplementowany za pomocą dynamicznej kolekcji, dodano ograniczenie na liczbę jej elementów: *co najwyżej 10 pozycji*.

W drugim etapie agregacja kwalifikowana z klasą asocjacji *Lekarz w zespole* została zamieniona na dwie asocjacje i „zwykłą” klasę za pomocą mechanizmu nazywanego „wprowadzeniem klasy pośredniczącej” – jak na Rys. 10-19. Atrybuty opcjonalne *premia dla członka zespołu* i *premia dodatkowa dla anestezjologa* w klasie *Zespół* zostały zastąpione atrybutami zainicjowanymi wartością 0. Dla poprawy czytelności rozwiązania na diagramie pominięto klasy *Osoba* i *Pacjent*.

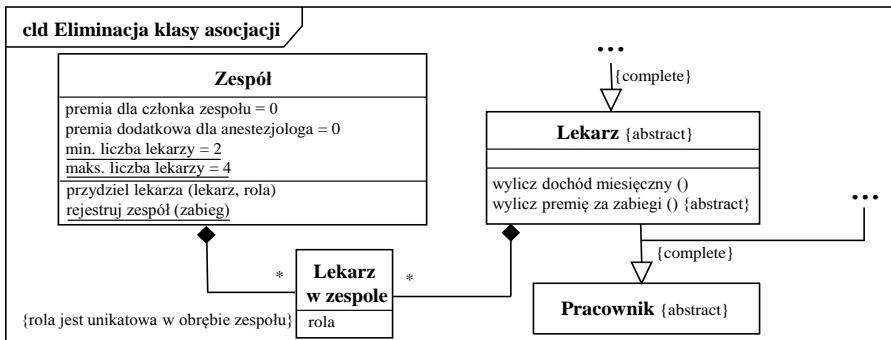
W trzecim etapie zaprezentowano sposób implementowania dwóch asocjacji, które powstały w wyniku eliminacji agregacji kwalifikowanej, poprzez wprowadzenie atrybutów w klasach połączonych asocjacjami – dla każdej asocjacji oddziennie – które będą przechowywać referencje do powiązanych obiektów klasy po przeciwniej stronie asocjacji. Funkcjonalność systemu wymaga nawigowania w obu kierunkach: od zespołu do jego członków – lekarzy (np. w celu weryfikacji składu zespołu zabiegowego) oraz od lekarzy do tych zespołów zabiegowych, których byli uczestnikami (dla obliczania premii oraz dochodu miesięcznego lekarzy).



Rys. 10-17 Fragment schematu pojęciowego przed zamianą na schemat projektowy

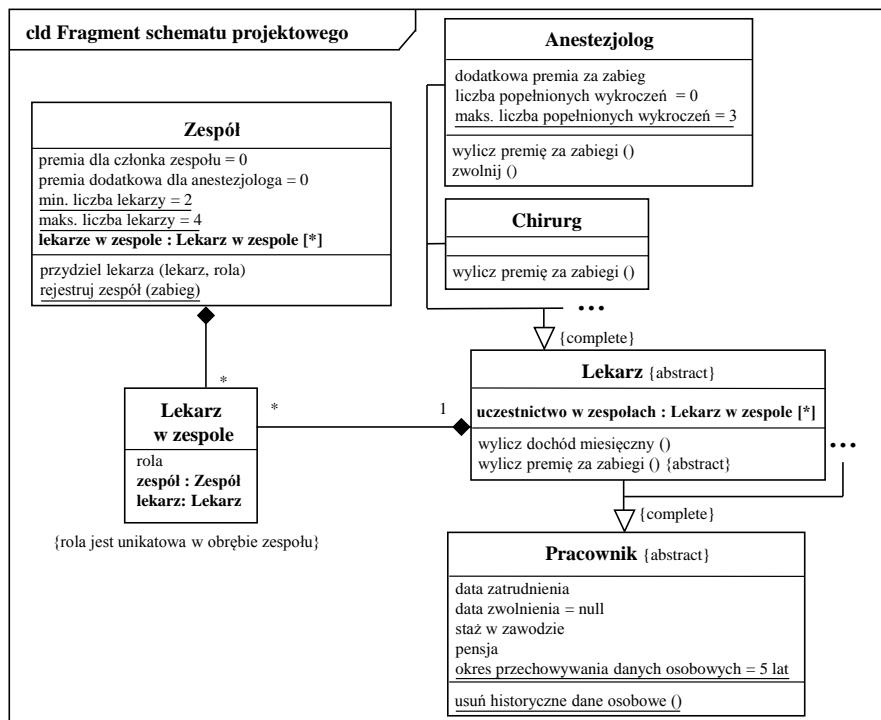


Rys. 10-18 Fragment schematu projektowego po eliminacji dziedziczenia nierożłącznego



Rys. 10-19 Fragment schematu projektowego po eliminacji klasy asocjacji

Ostateczny fragment schematu projektowego został przedstawiony na Rys. 10-20. Na tym etapie nie zajmujemy się jeszcze typami dla atrybutów i metod, a pogrubienie atrybutów implementujących asocjacje i wprowadzenie ich typów zostało zrealizowane w celu uzyskania diagramu o lepszej czytelności. Pominięto klasy *Osoba* i *Pacjent*.

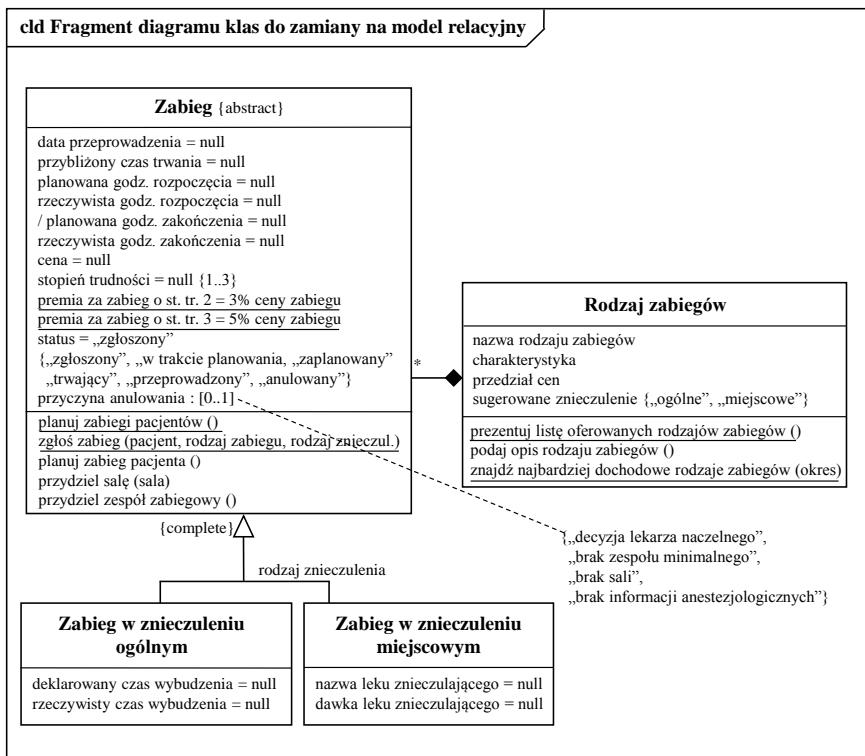


Rys. 10-20 Fragment schematu projektowego dostosowany do implementacji w języku obiektowym

### 10.3.8 Schemat relacyjny dla fragmentu schematu pojęciowego

**Polecenie:** Fragment schematu pojęciowego zawierający przynajmniej trzy klasy, związek dziedziczenia oraz asocjację przedstaw w modelu relacyjnym.

Fragment schematu pojęciowego podlegający zamianie na model relacyjny [37, 47] przedstawiono na Rys. 10-21.

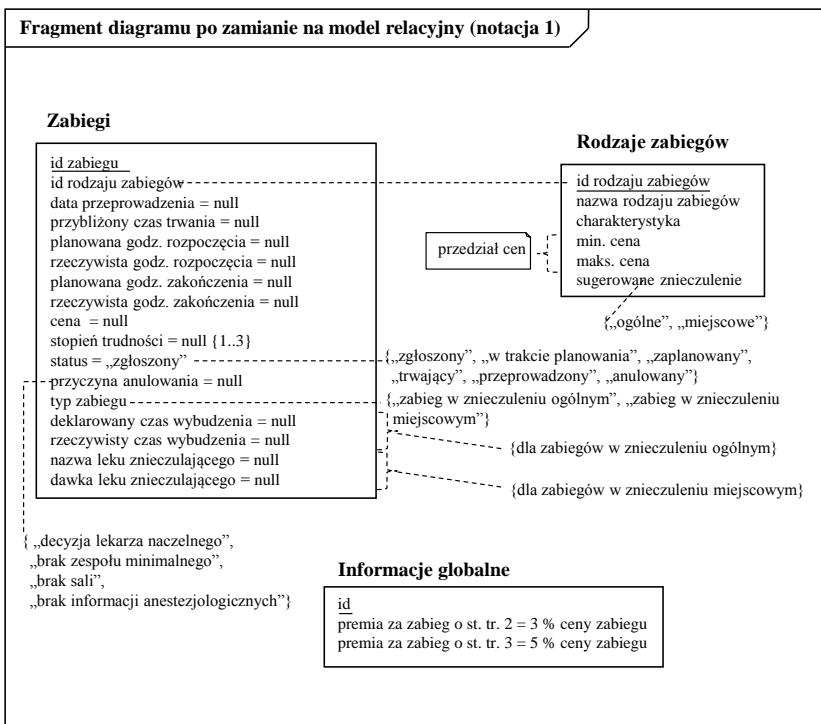


Rys. 10-21 Fragment schematu pojęciowego podlegający zamianie na model relacyjny

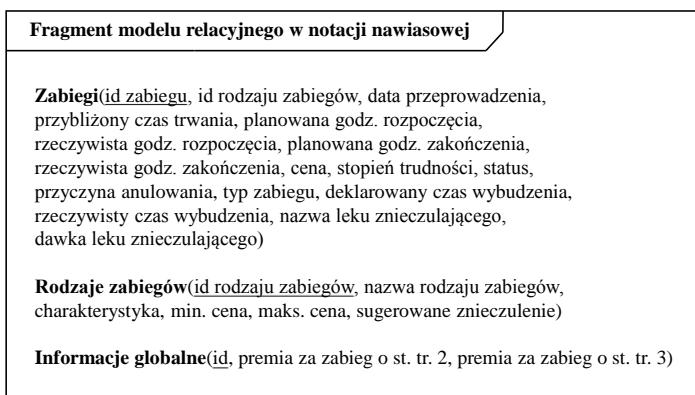
Rys. 10-22 zawiera model relacyjny tego fragmentu diagramu klas w notacji 1 [41].

Na Rys. 10-23 przedstawiono model relacyjny tego fragmentu diagramu klas w notacji nawiasowej.

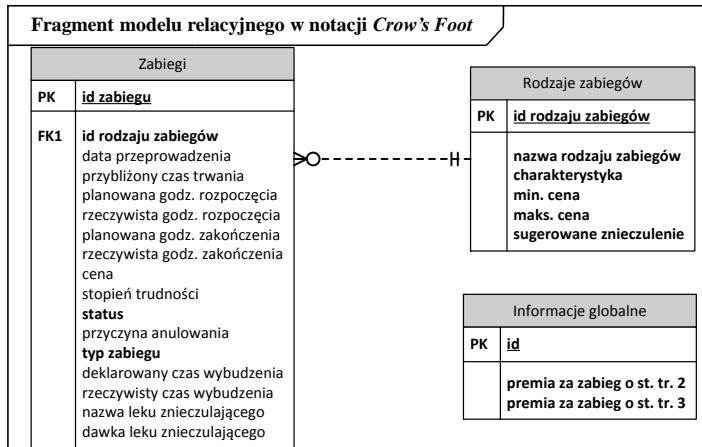
Model relacyjny rozważanego fragmentu klas w notacji *Crow's Foot* został na zaprezentowany na Rys. 10-24.



Rys. 10-22 Model relacyjny dla fragmentu diagramu klas przedstawionego na Rys. 10-21 (notacja 1)



Rys. 10-23 Model relacyjny dla fragmentu diagramu klas przedstawionego na Rys. 10-21 (notacja nawiasowa)



Rys. 10-24 Model relacyjny dla fragmentu diagramu klas przedstawionego na Rys. 10-21 (notacja *Crow's Foot*)

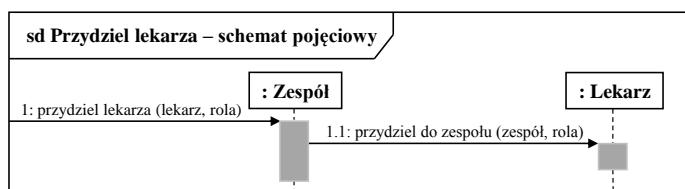
## 10.4 Analiza dynamiczna

### 10.4.1 Zadania podstawowe

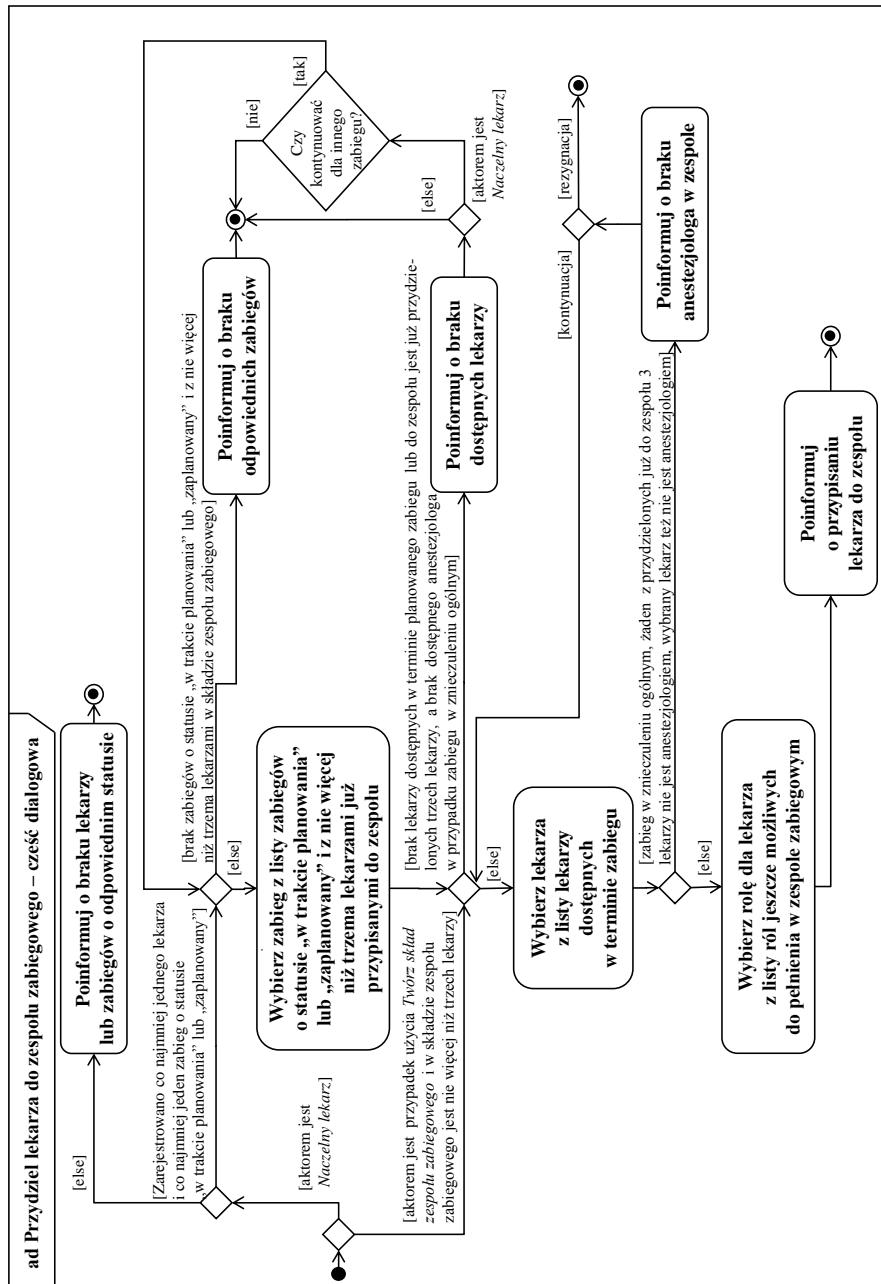
**Zad. 1:** Dla funkcjonalności *Przydziel lekarza do zespołu zabiegowego* (punkt 14.7 tekstu wymagań) skonstruuj diagram aktywności zgodny ze scenariuszem z Tab. 10-2.

Diagram aktywności zawarty na Rys. 10-26 ilustruje przykładową realizację części dialogowej przypadku dla scenariusza z Tab. 10-2.

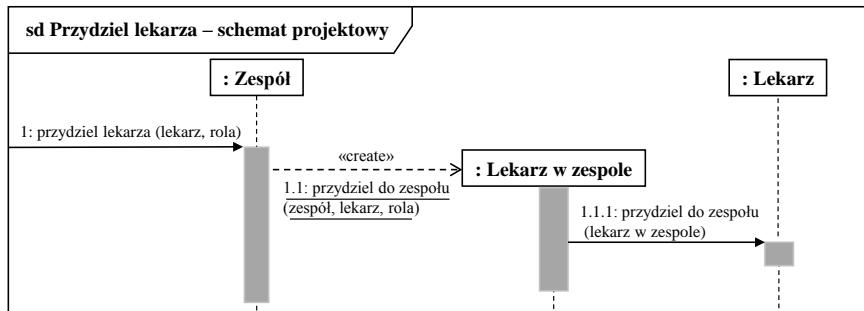
**Zad. 2:** Sporządź diagram interakcji dla części wykonawczej przypadku użycia *Przydziel lekarza do zespołu zabiegowego*. Diagram powinien przedstawić interakcje pomiędzy obiektami co najmniej dwóch klas.



Rys. 10-25 Diagram sekwencji zgodny z diagramem aktywności sporządzonym w zadaniu 2 dla schematu pojęciowego z Rys. 10-14.



Rys. 10-26 Diagram aktywności dla przypadku użycia *Przydziel lekarza do zespołu zabiegowego*



Rys. 10-27 Diagram sekwencji zgodny z diagramem aktywności sporządzonym w zadaniu 2 dla schematu projektowego z Rys. 10-20

Analiza dynamiczna spowodowała potrzebę zdefiniowania na schemacie pojęciowym metody *przydziel do zespołu (zespół, rola)* w klasie *Lekarz*, zaś na schemacie projektowym – metod: *przydziel do zespołu (zespół, lekarz, rola)* w klasie *Lekarz w zespole* oraz *przydziel do zespołu (lekacz w zespole)* w klasie *Lekarz*. Po zakończeniu analizy dynamicznej, oba schematy (pojęciowy i projektowy) powinny zostać uzupełnione o elementy modelu wynikające z tej analizy.

**Zad. 3:** Dla klasy *Zabieg* ze schematu pojęciowego wskaż kilka stanów (co najmniej dwa stany, nie licząc stanów początkowego i końcowego), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

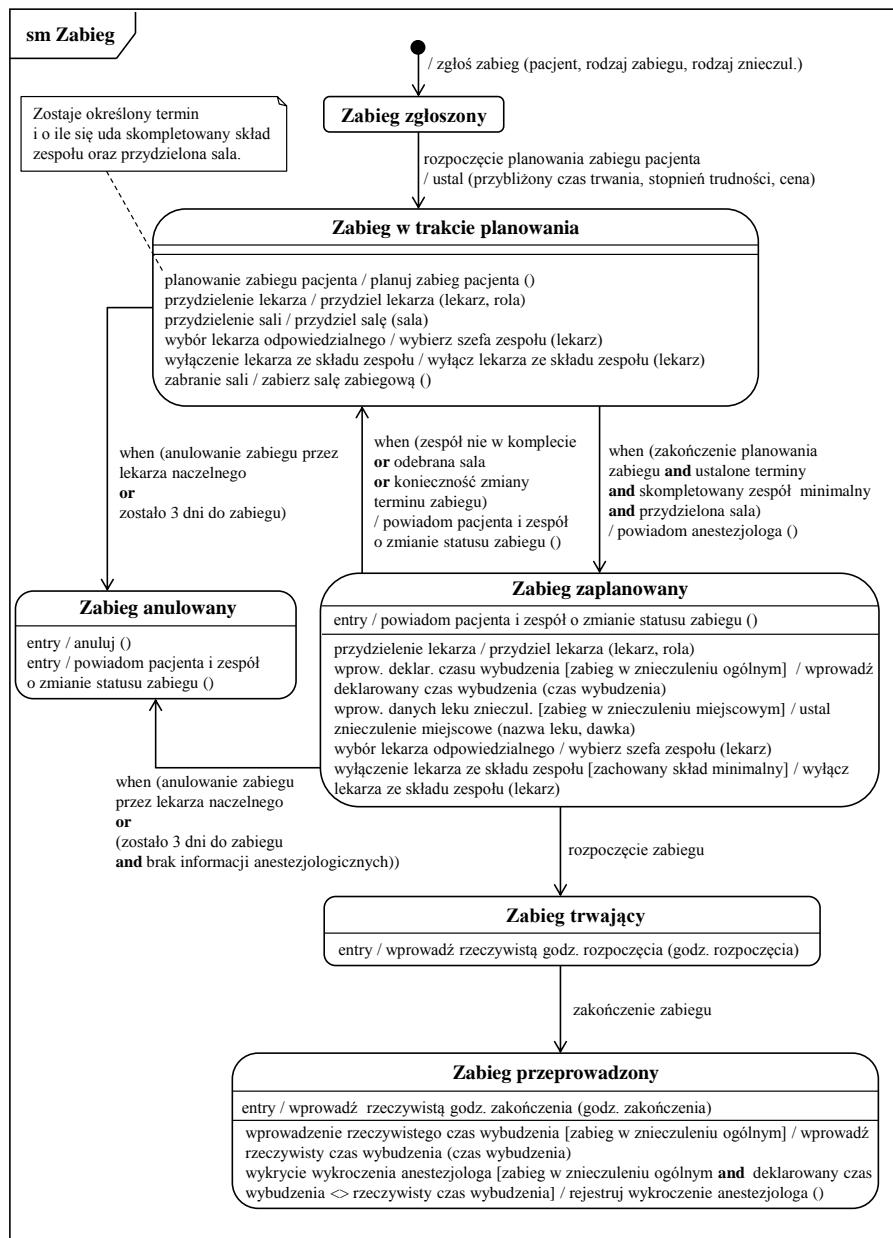
Maszyna stanowa dla klasy *Zabieg* została przedstawiona na Rys. 10-28. Ponieważ stany *Zabieg w trakcie planowania* i *Zabieg zaplanowany* posiadają kilka takich samych zdarzeń wewnętrznych, a także ze względu na to, że anulowanie zabiegu może być wykonane wyłącznie wówczas, gdy zabieg występuje w jednym z obu wymienionych stanów, można rozważyć wprowadzenie stanu złożonego sekwencyjnego *Zabieg w trakcie planowania / zaplanowany*, który obejmowałby oba stany, jak na Rys. 10-29.

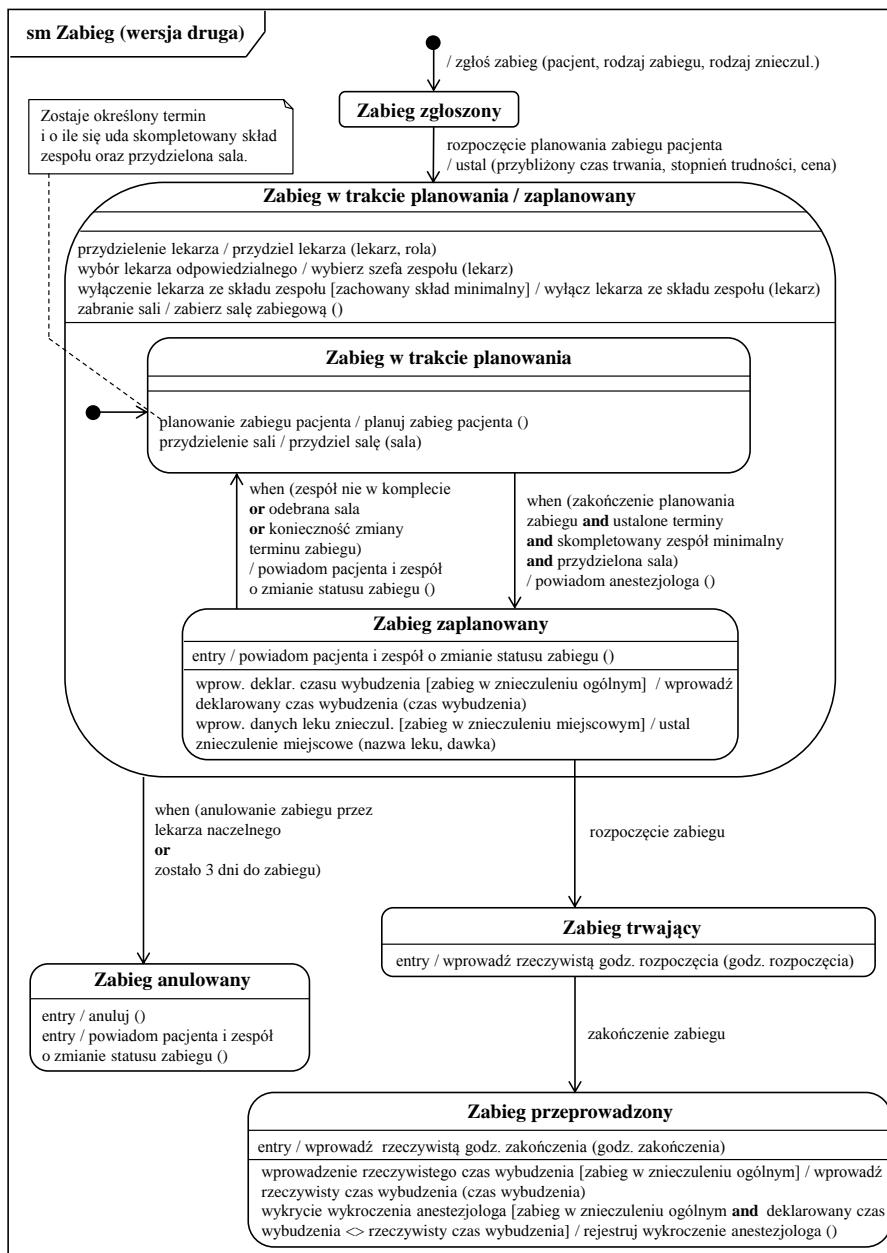
#### 10.4.2 Omówienie zmian na schemacie pojęciowym i projektowym wynikających z przeprowadzonej analizy dynamicznej oraz z poprzednich rozważań

W wyniku wykonanej analizy dynamicznej schemat pojęciowy i schemat projektowy należy uzupełnić o nowe elementy, które zostały ujęte na diagramach interakcji oraz diagramach stanów.

Lista zmian na schemacie pojęciowym została zamieszczona w Tab. 10-4.

Lista zmian na schemacie projektowym została przedstawiona w Tab. 10-5.

Rys. 10-28 Maszyna stanowa dla klasy *Zabieg* wykorzystująca wyłącznie stany proste



Rys. 10-29 Maszyna stanowa dla klasy *Zabieg* (wersja druga) z wykorzystaniem stanu złożonego sekwencyjnego

Tab. 10-4 Lista zmian na schemacie pojęciowym po analizie dynamicznej

Klasa	Zmiana
Anestezjolog	dodatkowa metoda: <i>rejestruj wykroczenie anestezjologa ()</i>
Lekarz	dodatkowa metoda: <i>przydziel do zespołu (zespół, rola)</i>
Zabieg	dodatkowe metody: <i>anuluj (przyczyna anulowania) powiadom anestezjologa () powiadom pacjenta i zespół o zmianie statusu zabiegu () ustal (przybliżony czas trwania, stopień trudności, cena) wprowadź rzeczywistą godz. rozpoczęcia (godz. rozpoczęcia) wprowadź rzeczywistą godz. zakończenia (godz. zakończenia) zabierz salę ()</i>
Zabieg w znieczuleniu ogólnym	dodatkowe metody: <i>wprowadź deklarowany czas wybudzenia (czas wybudzenia) wprowadź rzeczywisty czas wybudzenia (czas wybudzenia)</i>
Zabieg w znieczuleniu miejscowym	dodatkowa metoda: <i>ustal znieczulenie miejscowe (nazwa leku, dawka)</i>
Zespół	dodatkowe metody: <i>wybierz szefa zespołu (lekarz) wyłącz lekarza ze składu zespołu (lekarz)</i>

Tab. 10-5 Lista zmian na schemacie projektowym po analizie dynamicznej

Klasa	Zmiana
-------	--------

<i>Anestezjolog</i>	dodatkowa metoda: <i>rejestruj wykroczenie anestezjologa ()</i>
<i>Lekarz</i>	dodatkowa metoda: <i>przydziel do zespołu (lekarz w zespole)</i>
<i>Lekarz w zespole</i>	dodatkowa metoda: <i>przydziel do zespołu (zespół, lekarz, rola)</i>
<i>Zabieg</i>	dodatkowe metody: <i>anuluj (przyczyna anulowania) powiadom anestezjologa () powiadom pacjenta i zespół o zmianie statusu zabiegu () ustal (przybliżony czas trwania, stopień trudności, cena) wprowadź rzeczywistą godz. rozpoczęcia (godz. rozpoczęcia) wprowadź rzeczywistą godz. zakończenia (godz. zakończenia) zabierz salę ()</i>
<i>Zabieg w znieczuleniu ogólnym</i>	dodatkowe metody: <i>wprowadź deklarowany czas wybudzenia (czas wybudzenia) wprowadź rzeczywisty czas wybudzenia (czas wybudzenia)</i>
<i>Zabieg w znieczuleniu miejscowym</i>	dodatkowa metoda: <i>ustal znieczulenie miejscowe (nazwa leku, dawka)</i>
<i>Zespół</i>	dodatkowe metody: <i>wybierz szefa zespołu (lekarz) wyłącz lekarza ze składu zespołu (lekarz)</i>

## 10.5 Zadania do samodzielnej pracy

1. Czy system zewnętrzny lub podsystem wewnętrzny mogą być aktorami? Jeśli tak, odpowiedź uzupełnij przykładami spójnymi z tekstem wymagań.
2. Dla przypadku użycia *Zarządzaj salami zabiegowymi* sformułuj scenariusz oraz zilustruj podział tego przypadku na podprzypadki.

3. Podaj definicję *klasy abstrakcyjnej*. Czy klasa abstrakcyjna może być zdefiniowana wyłącznie w korzeniu hierarchii klas? Czy klasa w korzeniu hierarchii klas zawsze musi być klasą abstrakcyjną? Odpowiedź uzupełnij przykładami spójnymi ze schematem pojęciowym skonstruowanym na podstawie tekstu wymagań.
4. Wskaż na schemacie pojęciowym skonstruowanym na podstawie tekstu wymagań asocjację wiele-do-wielu (z niesymetrycznymi licznosciami dla obu końców asocjacji ze zdefiniowaną klasą lub atrybutem asocjacji). Na jakim etapie cyklu życia systemu powinno się zamienić klasę asocjacji na konstrukcję zawierającą klasę pośredniczącą? Narysuj odpowiedni przykład (z atrybutami i metodami) dla wskazanego uprzednio fragmentu schematu.
5. Wskaż na schemacie pojęciowym skonstruowanym na podstawie tekstu wymagań taki fragment, gdzie *role asocjacji* mogłyby być użyteczne. Zamień ten fragment (zachowując semantykę) z wykorzystaniem ról asocjacji.
6. Dla schematu pojęciowego skonstruowanego na podstawie tekstu wymagań podaj przykład ograniczenia na istnienie obiektu. Czy jest ograniczenie statyczne czy dynamiczne? Do jakiej metody i w jakiej klasie włożyłbyś implementację tego ograniczenia?
7. Dla schematu pojęciowego skonstruowanego na podstawie tekstu wymagań podaj przykład ograniczenia, które odnosi się do dwóch agregacji. Do jakiej metody i w jakiej klasie włożyłbyś implementację tego ograniczenia?
8. Dla wybranego przypadku użycia systemu (innego niż omówionego szczegółowo w analizie funkcjonalnej lub dynamicznej), sformułuj scenariusz i narysuj diagram aktywności (określ przynajmniej dwie aktywności, nie licząc początkowej i końcowej).
9. Dla wybranego przypadku użycia systemu (innego niż omówionego szczegółowo w analizie funkcjonalnej lub dynamicznej), sformułuj algorytm i narysuj odpowiedni diagram interakcji, który przedstawia interakcje pomiędzy obiektami co najmniej dwóch różnych klas.
10. Dla dowolnej klasy (innej niż omówionej szczegółowo w analizie dynamicznej) ze schematu pojęciowego wskaż dwa stany (wyłączając stan początkowy i stan końcowy), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

## 10.6 Podsumowanie zadania

W Tab. 10-6 przedstawiono ogólną charakterystykę omawianego zadania. Szczegółową charakterystykę zadania dla grup tematycznych: *Wprowadzenie do*

*przedmiotu, Analiza funkcjonalna, Analiza strukturalna, Analiza dynamiczna* zamieszczono odpowiednio w tabelach Tab. 10-7, Tab. 10-8, Tab. 10-9 i Tab. 10-10.

Tab. 10-6 Ogólna charakterystyka zadania

<b>Ogólna charakterystyka zadania</b>		
<i>Nazwa</i>	Klinika jednego dnia	
<i>Dziedzina problemowa</i>	system mógłby znaleźć zastosowanie w lecznictwie	
<i>Cel</i>	system ma ułatwić planowanie obsługi zabiegów w klinice	
<i>Zakres odpowiedzialności</i>	ewidencja pracowników kliniki i pacjentów, ewidencja i zarządzanie zabiegami, zespołami i salami zabiegowymi, wyliczanie miesięcznego wynagrodzenia, raporty dotyczące przeprowadzonych zabiegów	
<i>Stopień trudności</i>	bardzo trudne	
<i>Rozmiar</i>	12 klas	
<i>Użyteczność w grupach tematycznych</i>		
<i>Grupa tematyczna</i>	<i>Użyteczność w grupie tematycznej</i>	Użyteczność na etapie zgodnym z modelem efektywnego nauczania
<i>Wprowadzenie do przedmiotu</i>	niska	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> niska <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> niska
<i>Analiza funkcjonalna</i>	bardzo wysoka	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> bardzo wysoka <i>Podsumowanie dotychczasowych działań:</i> bardzo wysoka <i>Ewaluacja:</i> wysoka
<i>Analiza strukturalna</i>	wysoka	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> wysoka <i>Podsumowanie dotychczasowych działań:</i> wysoka <i>Ewaluacja:</i> wysoka
<i>Analiza dynamiczna</i>	wysoka	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> wysoka

		<i>Podsumowanie dotychczasowych działań: wysoka Ewaluacja: wysoka</i>
--	--	---

Tab. 10-7 Szczegółowa charakterystyka zadania dla grupy tematycznej *Wprowadzenie do przedmiotu*

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>	<i>Wprowadzenie do przedmiotu</i>	
<b>Stopień trudności grupy w zadaniu</b>	łatwe	
<b>Poruszane tematy</b>	<i>analiza a projektowanie, dekompozycja diagramów, mechanizmy rozszerzalności, reguły nazewnictwa, poprawa percepcji diagramu/diagramów</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>analiza a projektowanie</i>	łatwe	<i>model pojęciowy, model projektowy</i>
<i>mechanizmy rozszerzalności</i>	łatwe	<i>stereotyp, ograniczenie, komentarz</i>
<i>dekompozycja diagramów</i>	łatwe	podział modelu przypadków użycia na diagramy
<i>reguły nazewnictwa</i>	łatwe	reguły nazewnictwa przypadków użycia i klas
<i>poprawa percepcji diagramu/diagramów</i>	łatwe	reguły organizacji przypadków użycia, budowy oraz identyfikacji atrybutów

Tab. 10-8 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza funkcjonalna*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Analiza funkcjonalna</i>

<b>Stopień trudności grupy w zadaniu</b>	bardzo trudne	
<b>Poruszane tematy</b>	<i>aktorzy, przypadki użycia, scenariusze, dekompozycja modelu/diagramu przypadków użycia, mechanizmy rozszerzalności</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>aktorzy</i>	trudne	<i>aktor abstrakcyjny, aktor Podsystem czasu, hierarchia dla aktorów, stosunkowo duża liczba aktorów</i>
<i>przypadki użycia</i>	bardzo trudne	<i>przypadek użycia, stosunkowo duża liczba przypadków użycia oraz relacji pomiędzy przypadkami użycia</i>
<i>scenariusze</i>	bardzo trudne	<i>bardzo duża złożoność scenariuszy przypadków użycia, podprzepływy nazwane</i>
<i>dekompozycja modelu/diagramu przypadków użycia</i>	bardzo trudne	<i>dekompozycja pozioma, dekompozycja pionowa, przypadek użycia w roli aktora</i>
<i>mechanizmy rozszerzalności</i>	łatwe	<i>ograniczenie, komentarz</i>

Tab. 10-9 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza strukturalna*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Analiza strukturalna</i>
<b>Stopień trudności grupy w zadaniu</b>	trudne
<b>Poruszane tematy</b>	<i>atrybuty klasy, metody klasy, struktury generalizacji-spezjalizacji, klasa abstrakcyjna a klasa konkretna, asocjacje, analiza wartości, przejście na schemat projektowy, przejście na schemat relacyjny, wymagania na system, mechanizmy rozszerzalności</i>

<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>atrybuty klasy</i>	średnio trudne	<i>atrybut opcjonalny, powtarzalny, pochodny, obiektowy, klasowy, stosunkowo duża liczba atrybutów</i>
<i>metody klasy</i>	średnio trudne	<i>metoda abstrakcyjna, zaimplementowana, obiektowa, klasowa, polimorfizm metod, stosunkowo duża liczba metod klasowych</i>
<i>struktury generalizacji-specjalizacji</i>	trudne	dziedziczenie nierożłączne i elipsa, rozważona wersja z dziedziczeniem dynamicznym i wieloaspektowym, stosunkowo duża złożoność hierarchii dla osób
<i>klasa abstrakcyjna a klasa konkretna</i>	średnio trudne	<i>klasa abstrakcyjna, klasa konkretna, stosunkowo duża liczba klas abstrakcyjnych</i>
<i>asocjacje</i>	trudne	<i>klasa asocjacji, agregacja, kompozycja, asocjacja kwalifikowana, stosunkowo duża liczba agregacji i kompozycji, stosunkowo duża liczba i duża złożoność ograniczeń nałożonych na asocjacje (m.in. ograniczenie subset)</i>
<i>analiza wartości</i>	trudne	<i>wartość pochodna, wartość początkowa (domyślana), wartość graniczna, duża liczba wartości początkowych (domyślnych) oraz duża liczba wartości granicznych</i>
<i>przejście na schemat projektowy</i>	trudne	obejście dziedziczenia nierożłącznego, zamiana asocjacji z klasą asocjacji oraz agregacji kwalifikowanej
<i>przejście na schemat relacyjny</i>	trudne	<i>obejście dla dziedziczenia klas, zamiana atrybutów klasowych i atrybutów złożonych</i>
<i>wymagania na system</i>	średnio trudne	propozycja rozszerzenia wymagań i modelu funkcjonalnego o przypadki użycia do zarządzania wartościami granicznymi
<i>mechanizmy rozszerzalności</i>	średnio trudne	<i>ograniczenie, komentarz, stereotyp («dynamic»), ograniczenie statyczne a dynamiczne, stosunkowo duża liczba ograniczeń i komentarzy na schemacie pojęciowym</i>

		i projektowym
--	--	---------------

Tab. 10-10 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza dynamiczna*

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>	<i>Analiza dynamiczna</i>	
<b>Stopień trudności grupy w zadaniu</b>	trudne	
<b>Poruszane tematy</b>	<i>diagramy aktywności, diagramy interakcji, diagramy stanów, podsumowanie analizy dynamicznej, mechanizmy rozszerzalności</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>diagramy aktywności</i>	trudne	duża złożoność diagramu aktywności
<i>diagramy interakcji</i>	średnio trudne	
<i>diagramy stanów</i>	bardzo trudne	diagram stanów o bardzo dużej złożoności, akcja <i>entry</i> , zdarzenie <i>when</i> , duża liczba zdarzeń wewnętrznych, stosunkowo złożone warunki (określone dla przejść między stanami), stan zagnieżdżony
<i>podsumowanie analizy dynamicznej</i>	średnio trudne	stosunkowo duża liczba dodatkowych metod na schemacie pojęciowym i projektowym
<i>mechanizmy rozszerzalności</i>	łatwe	komunikat ze stereotypem « <i>create</i> » na diagramie interakcji

## Spis treści rozdziału

Rozdział 11 Hurtownia .....	462
11.1 Tekst wymagań .....	462
11.2 Analiza funkcjonalna .....	465
11.2.1 Diagram kontekstowy .....	465
11.2.2 Model przypadków użycia .....	466
11.2.3 Scenariusz wybranego przypadku wraz z podziałem przypadku na pod przypadki .....	472
11.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	473
11.2.5 Charakterystyka przypadków użycia – priorytety, złożoność .....	476
11.2.6 Interfejs użytkownika .....	482
11.3 Analiza strukturalna .....	490
11.3.1 Schemat pojęciowy .....	490
11.3.2 Analiza elementów pochodnych .....	492
11.3.3 Analiza wartości początkowych .....	493
11.3.4 Analiza wartości granicznych .....	493
11.3.5 Zadania podstawowe .....	495
11.3.6 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się .....	499
11.3.7 Schemat relacyjny dla fragmentu diagramu klas .....	499
11.4 Analiza dynamiczna .....	501
11.4.1 Maszyna stanowa dla klasy <i>Transakcja</i> .....	501
11.5 Podział schematu pojęciowego na pakiety .....	503
11.6 Architektura warstwy biznesowej .....	514
11.7 Wstępny, zgrubny plan iteracji realizujących założoną funkcjonalność .....	517
11.8 Podsumowanie zadania .....	519

# Rozdział 11

## Hurtownia

*Ewa Stemposz, Alina Stasiecka, Andrzej Jodłowski*

### 11.1 Tekst wymagań

Właściciel hurtowni postanowił zamówić system informatyczny, wspierający obsługę transakcji typu kupno (od dostawcy) i sprzedaż (dla klienta) produktów, których obrotem zajmuje się hurtownia.

1. Hurtownia posiada produkty różnego rodzaju, opisywane przez: unikatową nazwę handlową, jednostkę miary (np. sztuka lub kg), aktualną cenę jednostkową obowiązującą dla sprzedaży, wagę jednostkową oraz łączną liczbę jednostek zamówionych, ale jeszcze niedostarczonych przez kooperanta-dostawcę.
2. Jeśli stan danego produktu, wyrażany poprzez liczbę aktualnie posiadanych jego jednostek w magazynie, spadnie poniżej pewnego poziomu (ustalanego indywidualnie dla każdego produktu tzw. *min. stanu posiadania*), system ma alarmować, aby hurtownia uzupełniła zapasy u firmy-kooperanta (zwanej dalej kooperantem) – niekoniecznie tej samej, u której kupowano poprzednio.
3. Hurtownia przechowuje dane każdego z kooperantów (takie jak: nazwa, adres, numery telefonów – co najmniej 1) i to zarówno tych, u których już kupowała jakieś produkty, jak i tych potencjalnych. Możliwa jest sytuacja, w której ten sam kooperant zarówno kupuje od hurtowni, jak i sprzedaje jej swoje produkty. Firma, której hurtownia sprzedaje produkty, nazywana będzie dalej kooperantem-klientem, a ta, od której kupuje, kooperantem-dostawcą. Z kolei transakcje zawierane z firmami-kooperantami to kupno (transakcja zakupu przez hurtownię u kooperanta-dostawcy) i sprzedaż (transakcja sprzedaży kooperantowi-klientowi).
4. Jedna transakcja (kupna lub sprzedaży) może obejmować wiele produktów; dla każdego z nich musi być określona liczba zamówionych jednostek, liczba jednostek możliwych do dostarczenia (wstępnie zakłada się, że jest to liczba równa liczbie jednostek zamówionych), a także ile jednostek rzeczywiście dostarczono. Ponadto, należy pamiętać cenę jednostkową dla każdego produktu objętego daną transakcją. W przypadku sprzedaży przez hurtownię istotna jest informacja, czy wydano z magazynu produkty objęte daną transakcją (w celu dostarczenia do kooperanta-klienta).

5. Kooperant-klient określa ostateczny termin realizacji danej transakcji. Przyjmuje się, że po przekroczeniu tego terminu kooperant nie jest już zainteresowany realizowaniem transakcji, co oznacza, że ma ona zostać automatycznie anulowana.
6. Jeśli hurtownia nie posiada wystarczającej ilości zamawianego produktu, to może dostarczyć tyle, ile aktualnie ma na stanie (o ile kooperant-klient wyraził zgodę na mniejszą ilość). Zdarza się, że w zależności od daty ostatecznego terminu realizacji może brakować czasu na uzupełnienie zapasów. Taka sprzedaż uznawana jest za zrealizowaną pomimo tego, że ilości: *zamówiona* i *dostarczona* były różne.
7. Dla każdej transakcji (kupno, sprzedaż) mają być przechowywane następujące informacje: data zainicjalizowania, data akceptacji przez kupującego, data zrealizowania (tzn. data dostarczenia produktów do hurtowni lub do kooperanta-klienta) oraz status. Inicjalizowanie rozumiane jest jako złożenie przez hurtownię zamówienia na kupno od kooperanta-dostawcy lub złożenie przez kooperanta-klienta zamówienia na sprzedaż przez hurtownię.
8. Transakcje sprzedaży są uporządkowane dwojako (oddzielnie dla każdej firmy-kooperanta): według ostatecznych terminów realizacji oraz według sumarycznych wag transakcji (wyliczonych w oparciu o wagi produktów objętych transakcją).
9. Status transakcji może przyjmować następujące wartości:
  - (1) „w trakcie przetwarzania”, co oznacza, że transakcja:
    - najpierw „oczekuje na potwierdzenie przez sprzedającego”, przy czym potwierdzenie to oznacza ustalenie liczby jednostek produktu/produktów możliwych do dostarczenia przez hurtownię dla transakcji *sprzedaży* lub przez kooperanta-dostawcę dla transakcji *kupna*. Maksymalny termin ustalenia liczby jednostek możliwych do dostarczenia aktualnie wynosi 1 dzień, licząc od daty zainicjalizowania transakcji. W tym czasie strona sprzedająca powinna określić, jakie ilości każdego z produktów objętych transakcją jest w stanie dostarczyć. Jeśli tego nie zrobi, zakłada się, że obie liczby są równe. Data ustalenia liczby jednostek możliwych do dostarczenia będzie wtedy równa dacie upłynięcia maksymalnego terminu przewidzianego na ustalenie liczby tych jednostek;
    - następnie „oczekuje na akceptację przez kupującego”, czyli zaakceptowanie liczby jednostek produktu/produktów możliwych do dostarczenia. Maksymalny okres oczekiwania na akceptację transakcji wynosi aktualnie 3 dni. Po przekroczeniu tego okresu rozważane są dwie sytuacje: (a) co najmniej dla jednego produktu w obrębie transakcji liczba jednostek zamówionych różni się od liczby jednostek możliwych do dostarczenia – wtedy transakcja jest automatycznie anulowana, (b) dla wszystkich produktów w obrębie transakcji liczba

jednostek zamówionych jest równa liczbie jednostek możliwych do dostarczenia – transakcja uzyskuje status „w trakcie realizacji”;

- (2) „w trakcie realizacji”, co w przypadku transakcji sprzedaje kooperantowi-klientowi oznacza oczekiwanie na przydział transportu, a następnie na potwierdzenie przez kooperanta-klienta odbioru produktu/produktów objętych transakcją. W przypadku kupna od kooperanta-dostawcy transakcja oczekuje na zarejestrowanie przez magazyniera odbioru produktów;
  - (3) „zrealizowana” – transakcja osiągnie ten status w następujących sytuacjach: dla kupna od kooperanta-dostawcy – w momencie, gdy magazynier zarejestrował odbiór produktów objętych transakcją, a dla transakcji sprzedaje kooperantowi-klientowi, gdy kooperant-klient potwierdzi odbiór transportu, co skutkuje zmianą statusu na „zrealizowana” dla wszystkich transakcji objętych danym transportem;
  - (4) „anulowana” – anulowanie transakcji może nastąpić w następujących sytuacjach: (a) transakcji nie udało się zrealizować w zadany terminie, (b) kupujący nie zaakceptował liczby jednostek produktu/produktów możliwych do dostarczenia w określonym terminie, czyli nie zaakceptował transakcji – w przypadku, gdy co najmniej dla 1 produktu w obrębie transakcji liczba jednostek zamówionych jest różna od liczby jednostek możliwych do dostarczenia oraz (c) na życzenie kupującego/sprzedającego. Należy zapamiętać, kto i dlaczego anulował transakcję. Anulowanie może dotyczyć wyłącznie transakcji w trakcie przetwarzania.
10. Hurtownia nie zajmuje się obrotem produktami, których pojedyncze jednostki nie dają się przewieźć najbardziej ładownym samochodem hurtowni (ma być pamiętać, ile aktualnie wynosi maksymalna ładowność).
  11. Pojedynczy transport produktów z hurtowni (na obecnym etapie tylko taki kierunek dostawy jest rozważany) dotyczy realizacji pojedynczej lub kilku transakcji sprzedaje wyłącznie jednemu kooperantowi i jest realizowany jednym samochodem z jednym kierowcą i ewentualnie jednym ochroniarzem. W pierwszej kolejności realizowane są transakcje najpilniejsze, tzn. te o najbliższym terminie ostatecznej realizacji.
  12. Transport powinien wykorzystać nie mniej niż 75% ładowności samochodu (przy założeniu, że nie sąbrane pod uwagę gabaryty produktów, a jedynie ich wagi). Jeśli wartość transportu przekroczy ustalony limit, do transportu przydzielany jest ochroniarz. Ochroniarz musi mieć przynajmniej jeden dzień przerwy między kolejnymi transportami, które ochrania. Jest możliwe, że pracownik może pełnić zarówno rolę kierowcy, jak i ochroniarza (ale nie jednocześnie).
  13. Dla każdego transportu system ma przechowywać informacje takie jak: kto był odbiorcą (który z kooperantów-klientów hurtowni), kto obsługiwał (który kierowca, ochroniarz). System ma przechowywać: datę i godzinę rozpoczęcia

oraz datę i godzinę zakończenia transportu (zarówno planowane, jak i faktyczne), ponieważ stanowią one podstawę do algorytmicznego wyliczania wynagrodzenia miesięcznego pracowników.

14. Dla każdego pracownika powinny być przechowywane: imiona (co najwyżej dwa), nazwisko, data ur., data zatrudnienia, data zwolnienia, dane teleadresowe oraz stawka za godzinę pracy. Ponadto, dla ochroniarzy ma być przechowywany dodatek specjalny za godzinę pracy w trakcie transportu produktów. Wynagrodzenie miesięczne pracownika wyliczane jest na podstawie liczby przepracowanych godzin oraz stawki za godzinę pracy, a dla ochroniarzy dodatkowo uwzględniony jest dodatek specjalny za godzinę pracy.
15. System ma wspomóc w realizowaniu usług takich jak na przykład:
  - 15.1 Prezentowanie produktów posiadanych przez hurtownię z ewentualnym podaniem opisu wybranego produktu;
  - 15.2 Wyliczanie wynagrodzenia miesięcznego pracownika;
  - 15.3 Tworzenie, dla zadanego okresu, listy transakcji realizowanych w hurtowni, tzn. takich, które zostały zainicjalizowane w podanym okresie, z ewentualnym wyodrębnieniem transakcji anulowanych;
  - 15.4 Alarmowanie o osiągnięciu minimalnego stanu posiadania danego produktu.

## 11.2 Analiza funkcjonalna

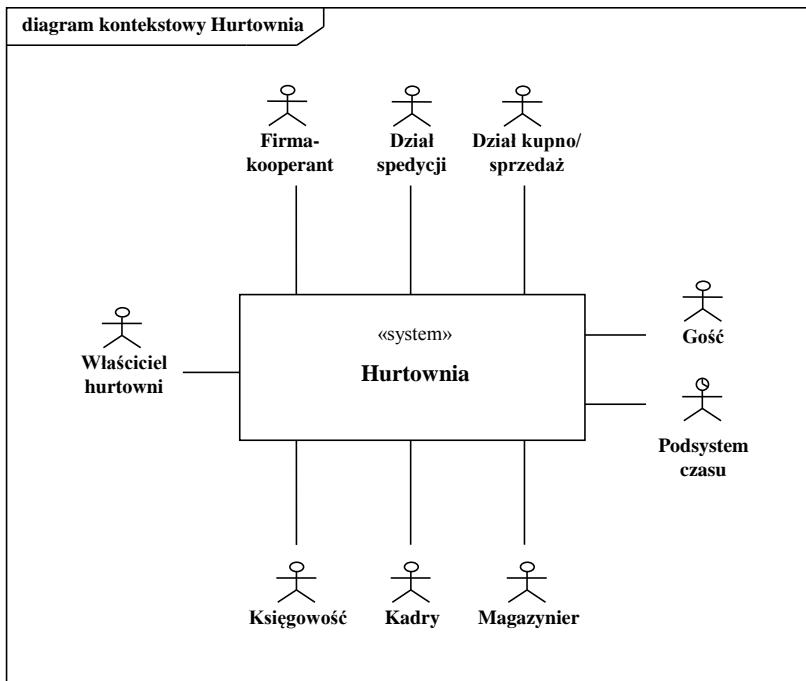
### 11.2.1 Diagram kontekstowy

*Polecenie:* Zbuduj diagram kontekstowy w oparciu o zamieszczony tekst wymagań.

Diagram kontekstowy, przedstawiający aktorów wybranych dla systemu *Hurtownia* (w oparciu o analizę tekstu wymagań) został zaprezentowany na Rys. 11-1. Należy zwrócić uwagę na to, że nie wszyscy aktorzy zostali wyraźnie wymienieni w tekście wymagań. Ich wyboru dokonał analityk, na podstawie znajomości dziedziny problemowej, biorąc pod uwagę możliwą ewolucję systemu w przyszłości.

W modelu funkcjonalności systemu należy zachować spójność kolejnych diagramów (diagramu kontekstowego i diagramu/ów przypadków użycia). Dotyczy to zarówno wyboru aktorów, jak również przypadków użycia.

*Uwaga:* Aktor *Podsystem czasu* wprowadzony do diagramu kontekstowego symbolizuje wszystkich aktorów tego rodzaju znajdujących się na diagramach przypadków użycia (np. aktora *Codziennie 23:59*).



Rys. 11-1 Diagram kontekstowy

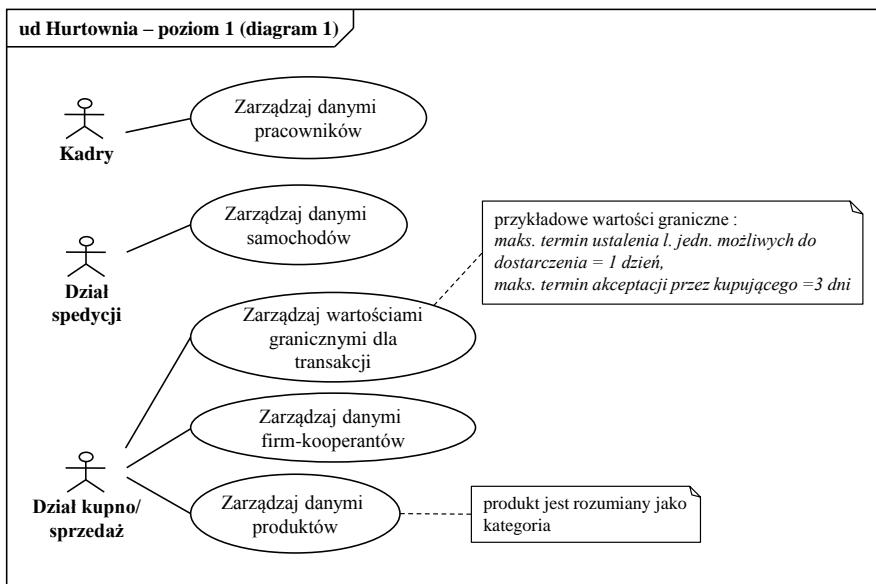
### 11.2.2 Model przypadków użycia

*Polecenie:* Zbuduj model przypadków użycia, uwzględniając między innymi funkcjonalność sugerowaną w ostatnim punkcie tekstu wymagań (tzn. w punkcie 15 tego tekstu – patrz 11.1 Tekst wymagań).

*Uwaga:* Diagram należy skonstruować z perspektywy aktorów z otoczenia systemu (najwyższy poziom abstrakcji modelu), biorąc pod uwagę możliwą ewolucję systemu w przyszłości. Należy uwzględnić hierarchie aktorów i relacje pomiędzy przypadkami (o ile mają/mogłyby mieć miejsce).

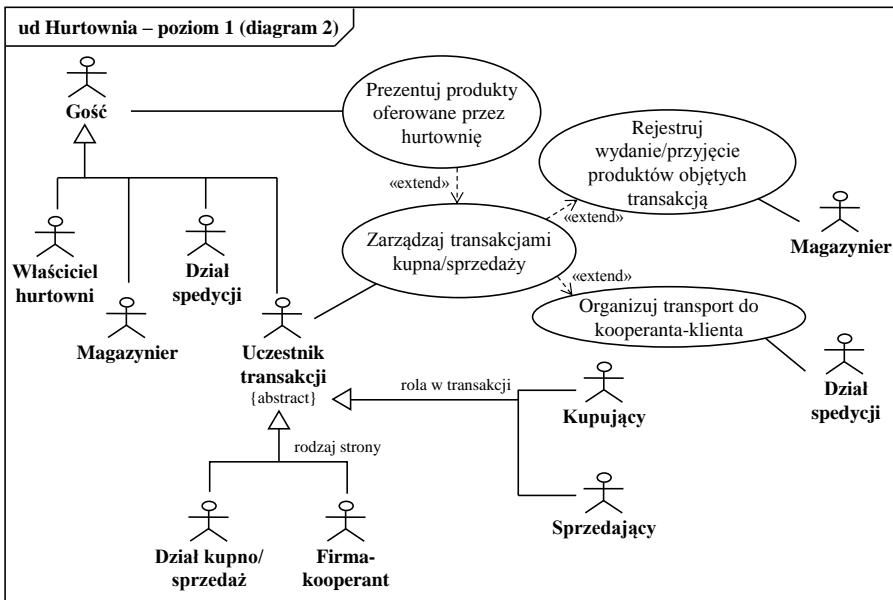
W celu zachowania przejrzystości modelu diagram przypadków [35, 40, 41, 46] użycia został on podzielony na trzy diagramy.

Na diagramie pierwszym (Rys. 11-2) zaprezentowano przypadki *Zarządzaj danymi*, rozumiane jako funkcje typu CRUD (ang. *Create, Read, Update, Delete*): utwórz, wyświetl, modyfikuj, usuń dane opisujące produkt, pracownika, samochód czy firmę-kooperanta, potrzebne do wypełnienia bazy danych. Często przypadki te są na wczesnym etapie analizy funkcjonalności pomijane, nie mniej jednak w końcowym modelu zawsze powinny być uwzględnione.



Rys. 11-2 Model przypadków użycia (diagram 1)

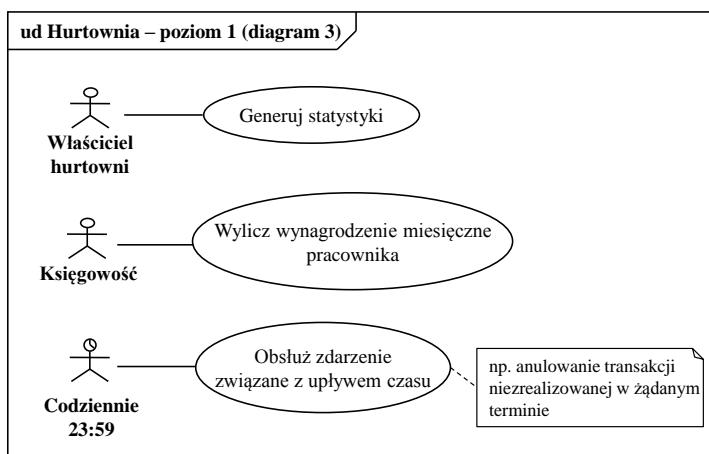
Diagram drugi modelu przypadków użycia (Rys. 11-3) opisuje hierarchię aktorów oraz funkcjonalność dostępną dla aktorów: *Gość*, *Uczestnik transakcji*, *Magazynier* i *Dział spedycji*.



Rys. 11-3 Model przypadków użycia (diagram 2)

*Uwaga:* W analizowanym tekście wymagań rozważa się wyłącznie transport produktów do kooperantów klientów, z wykorzystaniem transportu własnego hurtowni (przypadek użycia *Organizuj transport do kooperanta-klienta*). W przypadku rozszerzenia systemu o obsługę transportu dostaw produktów do hurtowni (od kooperantów-dostawców) należałoby uwzględnić w modelu m.in. taką funkcjonalność, jak *Organizuj transport produktów do hurtowni* (dla aktora – *Dział spedycji*).

Diagram trzeci (Rys. 11-4) prezentuje przypadki użycia przypisane do aktorów: *Właściciel hurtowni*, *Księgowość* oraz *Podsystem czasu*. W przypadku aktora *Podsystem czasu* na diagramie przypadków użycia została sprecyzowana jego nazwa: *Codziennie 23:59*.

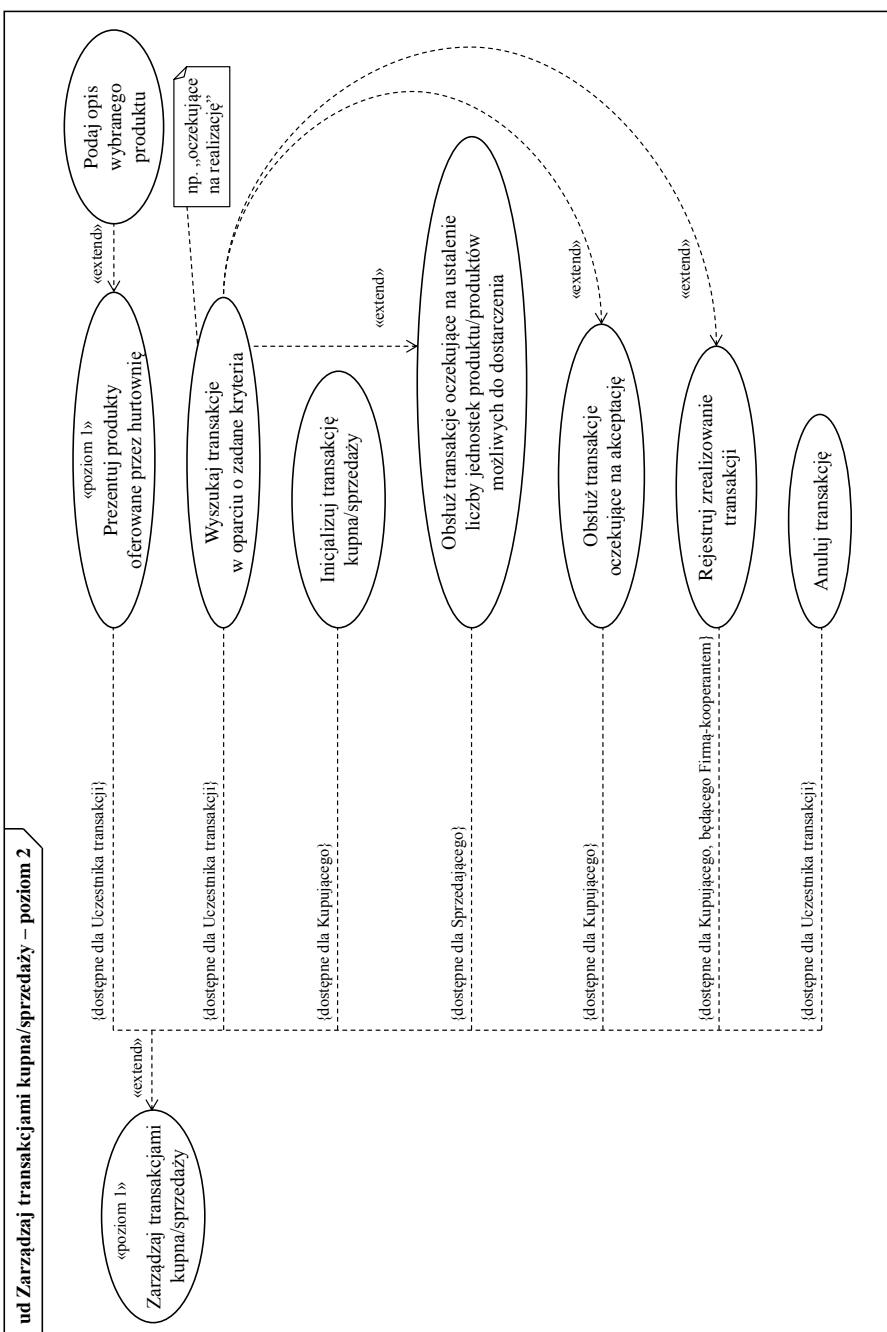


Rys. 11-4 Model przypadków użycia (diagram 3)

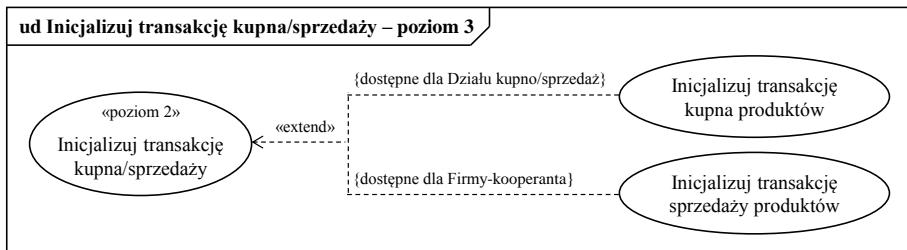
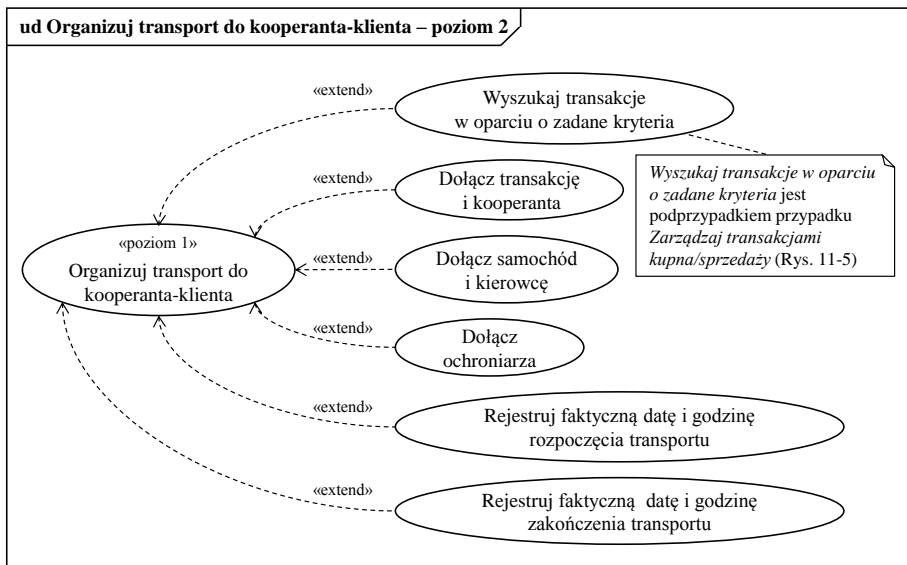
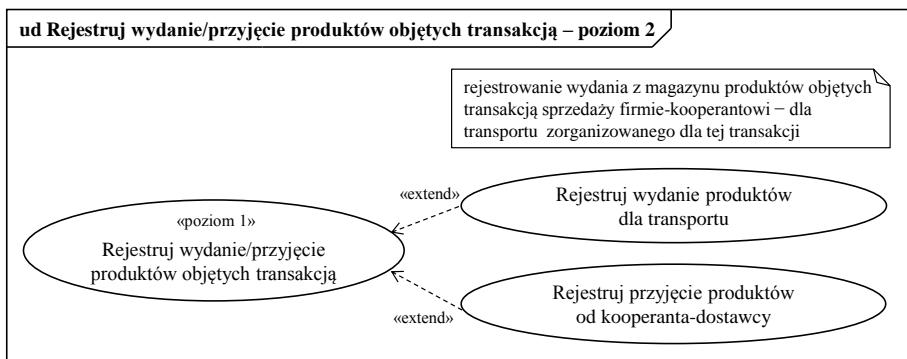
Model przypadków użycia zaprezentowany na diagramach z Rys. 11-2 – Rys. 11-4 został skonstruowany na najwyższym poziomie abstrakcji (oznaczonym numerem 1). W kolejnym kroku dla każdego złożonego przypadku użycia należałoby dokonać jego podziału na podprzypadki, co umożliwiłoby redukcję złożoności kodu i identyfikację bloków ponownego użycia. Przykładowe podprzypadki z kolejnych poziomów abstrakcji zostały przedstawione na Rys. 11-5 – Rys. 11-11.

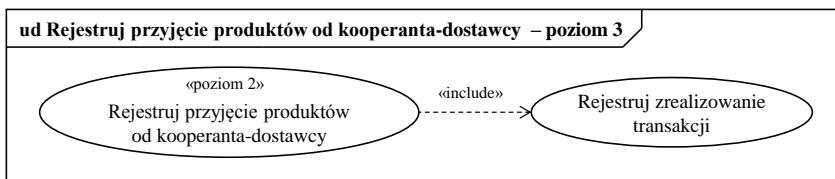
Rys. 11-5 określa podział przypadku użycia *Zarządzaj transakcjami kupna/sprzedaży*, przy czym dla jednego z jego podprzypadków *Inicjalizuj transakcję kupna/sprzedaży* pokazano podział na podprzypadki na poziomie 3 (Rys. 11-6).

Rys. 11-7 ilustruje podział przypadku *Organizuj transport do kooperanta-klienta*, Rys. 11-8 – przypadku *Rejestruj wydanie/przyjęcie produktów objętych transakcją*, a Rys. 11-9 prezentuje podprzypadek *Rejestruj zrealizowanie transakcji* przypadku użycia *Rejestruj przyjęcie produktów od kooperanta-dostawcy* (poziom 3).



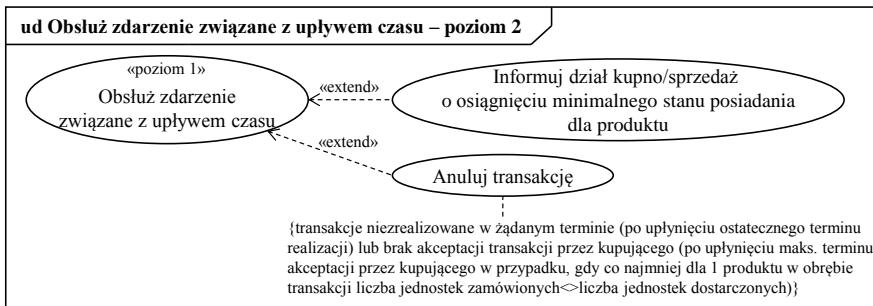
Rys. 11-5 Podział przypadku użycia *Zarządzaj transakcjami kupna/sprzedaży* na podprzypadki

Rys. 11-6 Podział przypadku użycia *Inicjalizuj transakcję kupna/sprzedaży* na pod przypadkiRys. 11-7 Podział przypadku użycia *Organizuj transport do kooperanta-klienta* na pod przypadkiRys. 11-8 Podział przypadku użycia *Rejestruj wydanie/przyjęcie produktów objętych transakcją* na pod przypadki



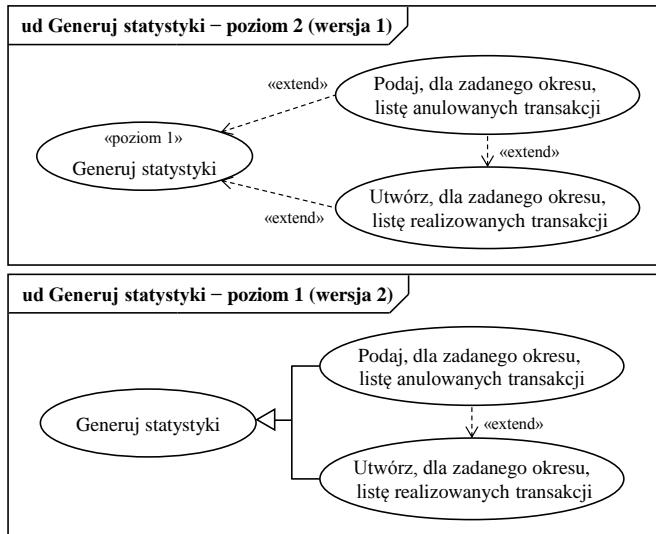
Rys. 11-9 Podprzypadek *Rejestruj zrealizowanie transakcji* przypadku użycia *Rejestruj przyjęcie produktów od kooperanta-dostawcy*

Podział przypadku *Obsłuż zdarzenie związane z upływem czasu* (z poziomu pierwszego) na podprzypadki przedstawiono na Rys. 11-10.



Rys. 11-10 Podział podprzypadku użycia *Obsłuż zdarzenie związane z upływem czasu* na podprzypadki

Rys. 11-11 ilustruje przykładowy podział dla przypadku *Generuj statystyki* (w dwóch wersjach). Statystyki są rozumiane jako raporty analityczno-statystyczne dotyczące działania hurtowni.



Rys. 11-11 Podział przypadku użycia *Generuj statystyki* na podprzypadki – dwie wersje

### 11.2.3 Scenariusz wybranego przypadku wraz z podziałem przypadku na podprzypadki

*Polecenie:* Dla przypadku użycia *Prezentuj produkty oferowane przez hurtownię*, z ewentualnym podawaniem opisu dla wybranego produktu (pkt 15.1 tekstu wymagań, patrz 11.1 Tekst wymagań):

- napisz scenariusz;
- zaproponuj podział tego przypadku na podprzypadki.

Przykładowy scenariusz dla wybranego przypadku użycia został przedstawiony w Tab. 11-1. Scenariusz dotyczy sytuacji, w której ten przypadek jest wywołany przez aktora *Gość* lub dowolnego z aktorów dziedziczących funkcjonalność po aktorze *Gość*.

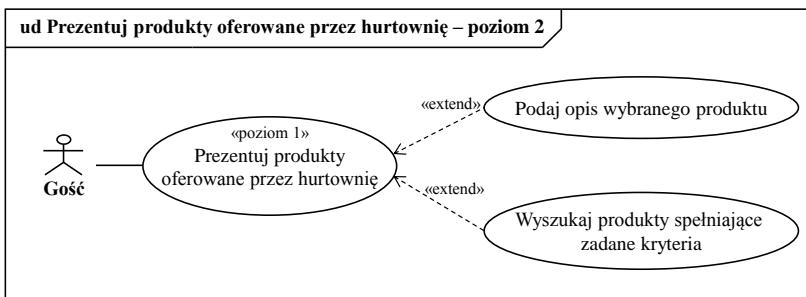
W rozważaniach pominięto sytuację, w której przypadek *Prezentuj produkty oferowane przez hurtownię* jest wywoływany przez inny przypadek użycia *Zarządzaj transakcjami kupna/sprzedaży*.

Tab. 11-1 Scenariusz dla przypadku użycia *Prezentuj produkty oferowane przez hurtownię* (z ewentualnym podawaniem opisu dla wybranego produktu)

<b>Nazwa przypadku użycia</b>	Prezentuj produkty oferowane przez hurtownię
<b>Warunek początkowy</b>	W systemie jest zarejestrowany co najmniej jeden produkt.
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Przypadek użycia jest wywoływany przez aktora <i>Gość</i> lub dowolnego z aktorów dziedziczących funkcjonalność po aktorze <i>Gość</i>.</li> <li>2. System odpytuje o kryteria wyszukiwania, np. nazwę handlową i/lub zakres cen jednostkowych sprzedaży. Aktor wprowadza kryteria.</li> <li>3. System wyświetla alfabetyczną listę produktów spełniających zadane kryteria.</li> <li>4. Aktor wybiera kolejno produkty, dla których chciałby zobaczyć opis. Po wybraniu produktu z listy system wyświetla jego opis.</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	<ol style="list-style-type: none"> <li value="2">2a. Nie zadano żadnego kryterium, system wyświetla pełną ofertę, z produktami uporządkowanymi alfabetycznie według nazwy handlowej.</li> </ol>

	2b. Nie odnaleziono produktów dla podanych kryteriów, system informuje o tym aktora i ponownie odpytuje o kryteria. 4a. Aktor rezygnuje z wyświetlania opisu dla wybranego produktu.
Zakończenie	W dowolnym momencie.
Warunek końcowy	Brak

Diagram z przykładowym podziałem przypadku *Prezentuj produkty oferowane przez hurtownię* (z ewentualnym podawaniem opisu dla wybranego produktu) przedstawiono na Rys. 11-12.



Rys. 11-12 Diagram z przykładowym podziałem przypadku *Prezentuj produkty oferowane przez hurtownię* (z ewentualnym podawaniem opisu dla wybranego produktu), z wyróżnionym nowym podprzypadkiem *Wyszukaj produkty spełniające zadane kryteria*

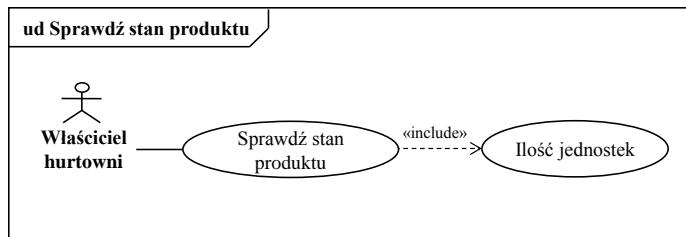
**Wniosek:** Analiza każdego ze scenariuszy może doprowadzić zarówno do wyróżnienia nowych, nieuwzględnionych do tej pory przypadków użycia, jak i identyfikacji nowych relacji pomiędzy już wyróżnionymi przypadkami, co ma miejsce w rozważanym przykładzie (wyróżniono nowy przypadek *Wyszukaj produkty spełniające zadane kryteria*).

#### 11.2.4 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się

- **Uwaga:** Dla przypomnienia, aktorem może być nie tylko osoba, ale też komórka organizacyjna, biuro, urządzenie zewnętrzne, inny system itd. Można więc nadać aktorowi nazwę *Dział spedycji* zamiast nazwy *Pracownik działu spedycji*, co oznacza każdego pracownika działu spedycji. Nie jest to błędem.

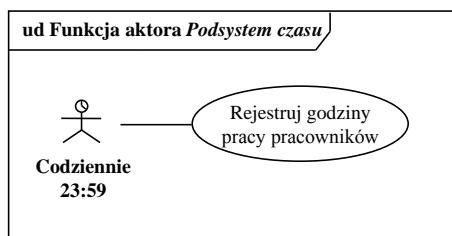
- Niejasne (lub w ogóle niepoprawne) nazwy przypadków, jak np.: *Nadaj dostawcę*, *Nadzoruj personel*, *Prowadź obowiązującą dokumentację*, *Zarządzaj firmą*, *Przechowywanie produktów* (przypadek przypisany do aktora *Magazyn*) czy *Ochroniaj towar* (przypadek przypisany do aktora *Ochroniarz*). Nazwa przypadku powinna sugerować rodzaj zadania zleconego systemowi przez aktora (a tak nie jest dla podprzypadku na Rys. 11-13, którego nazwa *Ilość jednostek* jest nazwą danych, a nie nazwą zadania). Tworząc nazwy przypadków, należy pamiętać, że przypadek jest częścią systemu, czyli oznacza (od strony implementacyjnej) pewną sekwencję instrukcji języka programowania.

Uwaga: Skuteczną pomoc w weryfikowaniu poprawności nazwy przypadku stanowi pytanie o obserwowalny rezultat przypadku.



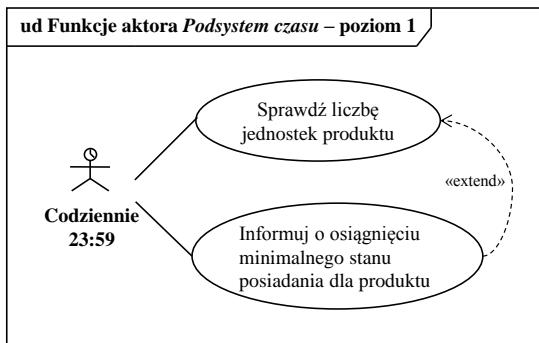
Rys. 11-13 Przykład błędnej nazwy przypadku użycia

- Błędne interakcje aktorów z systemem, jak np. na Rys. 11-14, gdzie nie wydaje się możliwe, aby *Podsystem czasu* miał dostęp do informacji umożliwiających rejestrację godzin pracy, np. kierowców czy ochroniarzy pracujących przy realizacji dostaw.

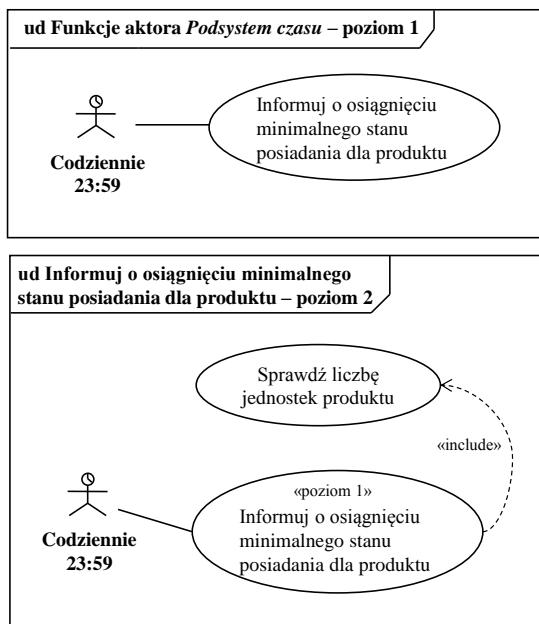


Rys. 11-14 Przykład błędnej interakcji aktora z systemem

- Niezrozumienie istoty zadania, widoczne poprzez niewłaściwe określenie relacji, jak na Rys. 11-15, gdzie zadaniem głównym (inaczej „przypadkiem bazowym”) powinno być poinformowanie o minimalnym stanie posiadania osiągniętym dla produktu, a zadaniem pobocznym – sprawdzenie jego ilości. Na diagramie rysowanym z perspektywy aktorów z otoczenia systemu, należało umieścić wyłącznie przypadek *Informuj o osiągnięciu minimalnego stanu posiadania dla produktu*. Przypadek rodzaju *Sprawdź liczbę jednostek produktu* mógłby wystąpić na diagramie ilustrującym podział przypadku pochodzącego z wyższego poziomu abstrakcji, jak na Rys. 11-16.



Rys. 11-15 Nieprawidłowo określone relacje dla przypadku użycia *Informuj o osiągnięciu minimalnego stanu posiadania dla produktu* (perspektywa aktorów systemu).



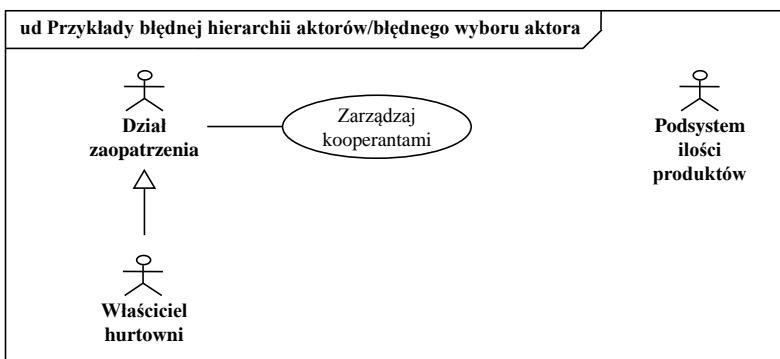
Rys. 11-16 Diagramy poprawne dla przykładu z Rys. 11-15

- Błędny wybór lub nazwa aktora, brak hierarchii dla aktorów lub błędne hierarchie, jak na Rys. 11-17. Aktor to byt z otoczenia systemu (dla systemu traktowanego jako całość), a nie jego część (częścią systemu jest np. *Podsystem ilości produktów*). Ponadto, trudno zrozumieć, jaki zbiór usług miałby być dostarczony tak nazwanemu aktorowi.

Następnie, jeśli już hurtownia posiada *Dział zaopatrzenia*, to właściciel nie musi/nie powinien realizować zadań należących do obowiązków pracowników tego działu. Jeśli Kowalski, będący właścicielem hurtowni, zechce (będzie

musiał ze względu na pewne okoliczności) zarządzać kooperantami, to realizowałby to zadanie z pozycji aktora *Dział zaopatrzenia*.

*Dla przypomnienia:* jeden konkretny byt z dziedziny problemowej (instancja aktora, np. właściciel hurtowni Kowalski) może pełnić wiele ról (identyfikowanych za pomocą mechanizmu aktorów). Podobnie, jedną rolę może pełnić wiele konkretnych bytów z dziedziny problemowej.



Rys. 11-17 Błędy w identyfikowaniu aktorów/hierarchii dla aktorów

### 11.2.5 Charakterystyka przypadków użycia – priorytety, złożoność

W Tab. 11-2 przedstawiono przypadki użycia ponumerowane (zgodnie z numeracją zagnieżdzoną, związaną z poziomami abstrakcji w następujący sposób: „nr przypadku na najwyższym poziomie abstrakcji, nr na kolejnym poziomie, itd.”), wraz ze wstępna propozycją określenia ich rodzajów, priorytetów oraz złożoności. Na dalszym etapie prac, w miarę uszczegóławiania modelu przypadków użycia, propozycje te mogą podlegać mniejszym lub większym zmianom.

Kolumna *Rodzaj* w tabeli zawiera informację o stopniu ogólności przypadków użycia w skali {pojedynczy, złożony}. Przyjęto, że przypadek jest pojedynczy, jeśli odpowiada pojedynczej funkcjonalności systemu, np. *Podaj opis wybranego produktu*. Natomiast przypadek złożony grupuje kilka funkcjonalności systemu, np. *Zarządzaj pracownikami* grupuje podprzypadki typu CRUD: *Utwórz/Wyświetl/Modyfikuj/Usuń pracownika*.

*Priorytet przypadku użycia* związany jest z ważnością danego przypadku użycia dla użytkownika systemu z perspektywy jego działalności w dziedzinie problemowej.

Priorytet został określony w skali: bardzo wysoki=4, wysoki=3, średni=2, niski=1.

*Złożoność* określa trudność implementacji danego przypadku dla danego zespołu projektowego, na który ma wpływ m.in. złożoność algorytmiczna przypadku, znajomość środowiska implementacji. Dla tej własności przyjęto skalę: bardzo

duża=4, duża=3, średnia=2, mała=1. Należy zwrócić uwagę na fakt, że ocena złożoności przypadku zależy od doświadczenia i umiejętności zespołu projektowego, tak więc nawet dla tego samego zespołu może różnić się w zależności od środowiska, w jakim implementacja ma być zrealizowana. Ze względu na fakt, że w dalszych rozważaniach nie zajmujemy się określaniem środowiska implementacji, ocenę trudności każdego z przypadków oparto wyłącznie na jego przewidywanej złożoności algorytmicznej.

Przypadki użycia umieszczone w Tab. 11-2 zostały uporządkowane według nadanych im numerów (z uwzględnieniem zagnieźdzonej numeracji).

Tab. 11-2 Lista przypadków użycia z rodzajami, priorytetami i złożonością, uporządkowana według zagnieździonej numeracji przypadków

Nr	Nazwa przypadku użycia	Rodzaj	Priorytet	Złożoność
1.	<i>Zarządzaj danymi pracowników (CRUD)</i>	złożony	2	1
2.	<i>Zarządzaj danymi firm-kooperantów (CRUD)</i>	złożony	3	1
3.	<i>Zarządzaj danymi produktów (CRUD)</i>	złożony	4	1
4.	<i>Zarządzaj danymi samochodów (CRUD)</i>	złożony	2	1
5.	<i>Zarządzaj wartościami granicznymi dla transakcji</i>	złożony	4	1
6.	<i>Prezentuj produkty oferowane przez hurtownię</i>	złożony	4	3
6.1	<i>Podaj opis wybranego produktu</i>	pojedynczy	4	1
6.2	<i>Wyszukaj produkty spełniające zadane kryteria</i>	pojedynczy	4	4
7.	<i>Zarządzaj transakcjami kupna/sprzedaży</i>	złożony	4	1
7.1	<i>Rejestruj zrealizowanie transakcji</i>	pojedynczy	3	1
7.2	<i>Inicjalizuj transakcję kupna/sprzedaży</i>	złożony	4	1
7.2.1	<i>Inicjalizuj transakcję kupna produktów</i>	pojedynczy	4	2
7.2.2	<i>Inicjalizuj transakcję sprzedaży produktów</i>	pojedynczy	4	2
7.3	<i>Obsłuż transakcje oczekujące na akceptację</i>	złożony	3	2
7.4	<i>Obsłuż transakcje oczekujące na ustalenie liczby jednostek</i>	złożony	3	2

	<i>produktu/produktów możliwych do dostarczenia</i>			
7.5	<i>Wyszukaj transakcje w oparciu o zadane kryteria</i>	pojedynczy	3	4
7.6	<i>Anuluj transakcję</i>	pojedynczy	3	2
8.	<i>Rejestruj wydanie/przyjęcie produktów objętych transakcją</i>	pojedynczy	3	2
8.1	<i>Rejestruj wydanie produktów dla transportu</i>	pojedynczy	3	2
8.2	<i>Rejestruj przyjęcie produktów od kooperanta-dostawcy</i>	pojedynczy	3	1
9.	<i>Organizuj transport do kooperanta-klienta</i>	złożony	3	4
9.1	<i>Rejestruj faktyczną datę i godzinę rozpoczęcia transportu</i>	pojedynczy	2	1
9.2	<i>Rejestruj faktyczną datę i godzinę zakończenia transportu</i>	pojedynczy	3	1
9.3	<i>Dolacz transakcję i kooperanta</i>	pojedynczy	3	3
9.4	<i>Dolacz samochód i kierowcę</i>	pojedynczy	3	2
9.5	<i>Dolacz ochroniarza</i>	pojedynczy	3	1
10.	<i>Generuj statystyki</i>	złożony	1	4
10.1	<i>Utwórz, dla zadanego okresu, listę realizowanych transakcji</i>	złożony	3	3
10.2	<i>Podaj, dla zadanego okresu, listę anulowanych transakcji</i>	złożony	2	2
11.	<i>Wylicz wynagrodzenie miesięczne pracownika</i>	złożony	2	2
12.	<i>Obsłuż zdarzenie związane z upływem czasu</i>	złożony	1	3
12.1	<i>Informuj dział kupno/sprzedaż o osiągnięciu minimalnego stanu posiadania dla produktów</i>	pojedynczy	1	1

W sytuacji, gdy chcemy odnaleźć konkretny przypadek, posługując się jego nazwą, przeglądanie tabeli uporządkowanej według numerów przypadków użycia nie jest wygodne. W związku z powyższym, została dołączona druga tabela, w której przypadki użycia z numerami, rodzajami, priorytetami i złożonością zostały uporządkowane alfabetycznie według nazwy przypadku (Tab. 11-3).

Tab. 11-3 Lista przypadków użycia z rodzajami, priorytetami i złożonością, uporządkowana alfabetycznie według nazw przypadków

Nazwa przypadku użycia	Nr	Rodzaj	Priorytet	Złożoność
Anuluj transakcję	7.6	pojedynczy	3	2
Dolacz ochroniarza	9.5	pojedynczy	3	1
Dolacz samochód i kierowcę	9.4	pojedynczy	3	2
Dolacz transakcję i kooperanta	9.3	pojedynczy	3	3
Generuj statystyki	10.	złożony	1	4
Informuj dział kupno/sprzedaż o osiągnięciu minimalnego stanu posiadania dla produktów	12.1	pojedynczy	1	1
Inicjalizuj transakcję kupna produktów	7.2.1	pojedynczy	4	2
Inicjalizuj transakcję kupna/sprzedaży	7.2	złożony	4	1
Inicjalizuj transakcję sprzedaży produktów	7.2.2	pojedynczy	4	2
Obsłuż transakcje oczekujące na akceptację	7.3	złożony	3	2
Obsłuż transakcje oczekujące na ustalenie liczby jednostek produktu/produktów możliwych do dostarczenia	7.4	złożony	3	2
Obsłuż zdarzenie związane z upływem czasu	12.	złożony	1	3
Organizuj transport do kooperanta-klienta	9.	złożony	3	4
Podaj opis wybranego produktu	6.1	pojedynczy	4	1
Podaj, dla zadanego okresu, listę anulowanych transakcji	10.2	złożony	2	2
Prezentuj produkty oferowane przez hurtownię	6.	złożony	4	3
Rejestruj faktyczną datę i godzinę rozpoczęcia transportu	9.1	pojedynczy	2	1
Rejestruj faktyczną datę i godzinę zakończenia transportu	9.2	pojedynczy	3	1
Rejestruj przyjęcie produktów od kooperanta-dostawcy	8.2	pojedynczy	3	1

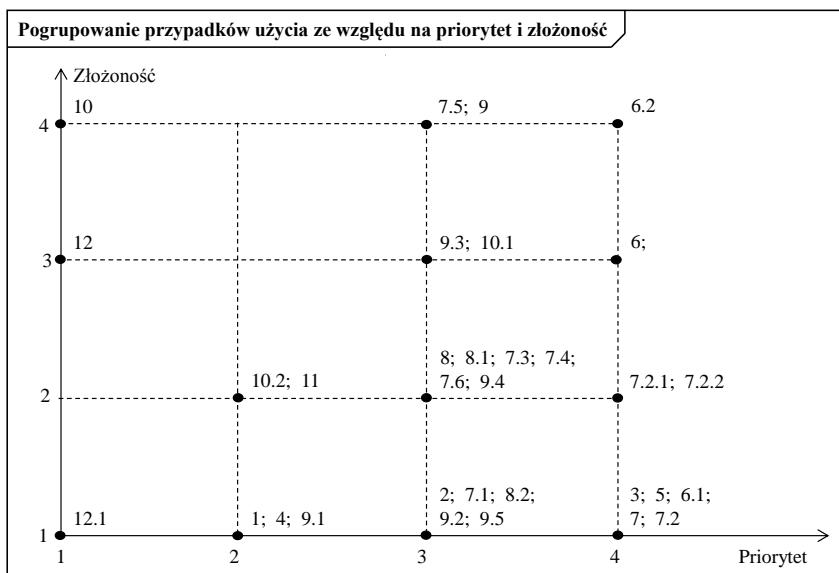
<i>Rejestruj wydanie produktów dla transportu</i>	8.1	pojedynczy	3	2
<i>Rejestruj wydanie/przyjęcie produktów objętych transakcją</i>	8.	pojedynczy	3	2
<i>Rejestruj zrealizowanie transakcji</i>	7.1	pojedynczy	3	1
<i>Utwórz, dla zadanego okresu, listę realizowanych transakcji</i>	10.1	złożony	3	3
<i>Wylicz wynagrodzenie miesięczne pracownika</i>	11.	złożony	2	2
<i>Wyszukaj produkty spełniające zadane kryteria</i>	6.2	pojedynczy	4	4
<i>Wyszukaj transakcje w oparciu o zadane kryteria</i>	7.5	pojedynczy	3	4
<i>Zarządzaj danymi firm-kooperantów (CRUD)</i>	2.	złożony	3	1
<i>Zarządzaj danymi pracowników (CRUD)</i>	1.	złożony	2	1
<i>Zarządzaj danymi produktów (CRUD)</i>	3.	złożony	4	1
<i>Zarządzaj danymi samochodów (CRUD)</i>	4.	złożony	2	1
<i>Zarządzaj transakcjami kupna/sprzedaży</i>	7.	złożony	4	1
<i>Zarządzaj wartościami granicznymi dla transakcji</i>	5.	złożony	4	1

Dzięki określeniu priorytetów oraz złożoności przypadków, można je pogrupować oraz uporządkować od najbardziej użytecznych (czyli tych o najwyższym priorytecie) i najbardziej złożonych, do najmniej użytecznych i najłatwiejszych. Pozwoli to na optymalne zaplanowanie kolejnych iteracji procesu wytwarzania oprogramowania (zgodnie z modelem iteracyjno-przyrostowym). Należy wyraźnie podkreślić, że dane zawarte w powyższej tabeli pozwolą na pierwsze podejście do utworzenia zgrubnego planu projektu, zawierającego plany na kolejne iteracje. Po każdej kolejnej iteracji, w miarę postępu prac, zawartość planu projektu, zawartość poszczególnych zgrubnych planów iteracji może ulegać zmianom.

*Dla przypomnienia:* Proces tworzenia systemu informatycznego, niezależnie od stosowanej metodyki, powinien zawierać zwykle te same etapy, wykonywane w ściśle określonej kolejności. Wydaje się, że spośród różnych modeli procesu wytwarzanego oprogramowania, najlepsze efekty przynosi wykorzystanie modelu iteracyjno-przyrostowego. Dzięki niemu po każdej iteracji powstaje pracujący, spójny fragment oprogramowania, z działającym interfejsem. Główne korzyści wynikające z zastosowania modelu iteracyjno-przyrostowego można określić następująco: (1) testując fragmenty systemu powstałe w kolejnych iteracjach klient będzie mógł sukcesywnie weryfikować swoje poglądy na wymagania, (2) realizacja

w pierwszej kolejności (w początkowych iteracjach) przypadków najbardziej potrzebnych klientowi i najtrudniejszych implementacyjnie spowoduje, że tego rodzaju przypadki zostaną w efekcie najlepiej przetestowane, (3) jeśli zespół projektowy będzie miał kłopoty z dotrzymaniem zaplanowanego terminu zakończenia realizacji tworzonego oprogramowania, to na końcu zostaną do wykonania najłatwiejsze i najmniej ważne dla klienta funkcje. Reasumując, przy wykorzystaniu modelu iteracyjno-przyrostowego, pozycja zespołu realizującego system informatyczny jest zdecydowanie lepsza, niż byłaby, gdyby projekt realizowano w oparciu o model kaskadowy.

Na podstawie danych zawartych w Tab. 11-3 został sporządzony wykres (Rys. 11-18) przedstawiający pogrupowanie przypadków użycia ze względu na priorytet oraz trudność implementacji (złożoność).



Rys. 11-18 Pogrupowanie przypadków ze względu na priorytet i trudność implementacji (złożoność)

Analiza wykresu z Rys. 11-18 ułatwia podjęcie decyzji dotyczących zawartości kolejnych iteracji przy wykorzystaniu modelu iteracyjno-przyrostowego wytwarzania oprogramowania dzięki m.in. wyodrębnieniu przypadków ważnych dla przyszłego użytkownika systemu oraz trudnych w realizacji dla zespołu projektowego, które powinny zostać zrealizowane w pierwszej kolejności.

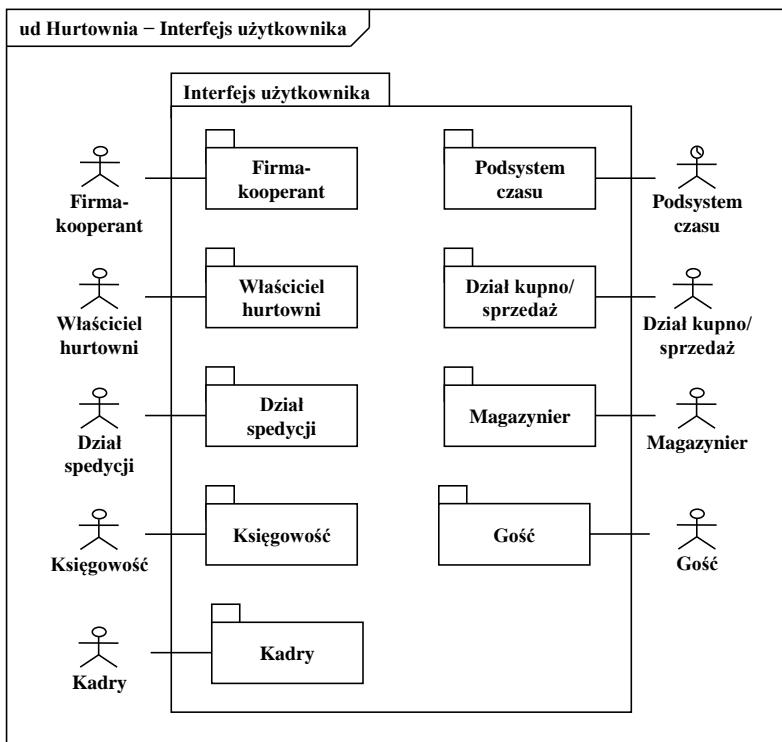
Warto wspomnieć, że plan rozwijania oprogramowania powinien uwzględniać scenariusze dla każdego z zidentyfikowanych przypadków użycia, które nie muszą być implementowane jednocześnie. W pierwszej kolejności zostają zaimplementowane ich przepływy główne, które właśnie z tego powodu są w scenariuszach wyodrębniane.

### 11.2.6 Interfejs użytkownika

*Polecenie 1:* Dla modelu przypadków użycia (patrz 11.2.2 i 11.2.3) zaproponuj pogrupowanie usług w pakiety.

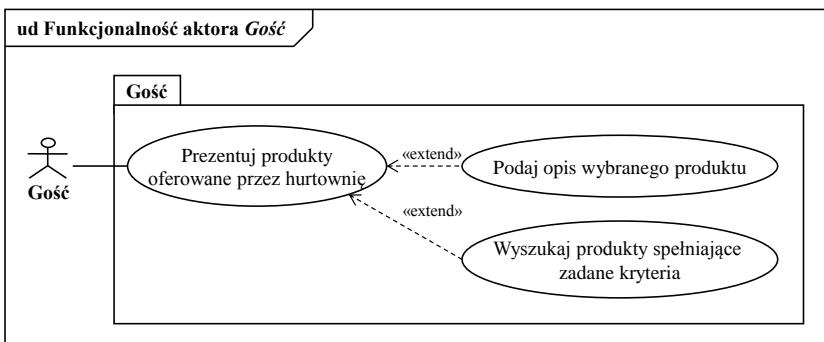
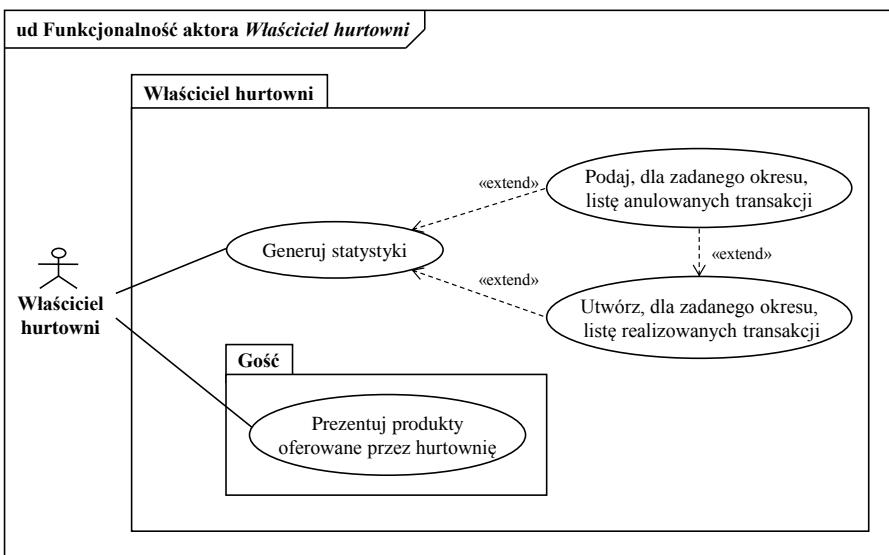
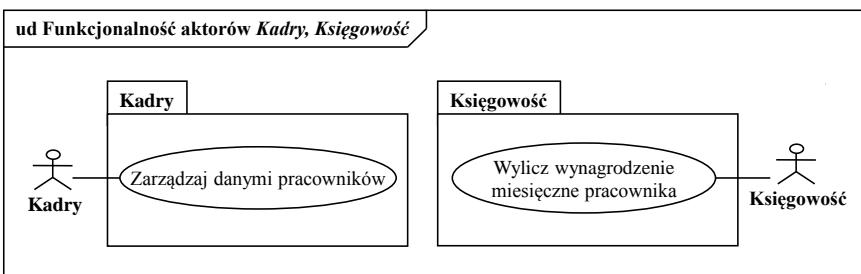
Zaprezentowany model służył do zidentyfikowania usług (przypadek użycia) i przypisania ich do aktorów. Podczas tworzenia modelu przypadków użycia można skorzystać z pakietów [38, 41], co ułatwia analizę modelu, jak również może znacząco wspomóc proces projektowania interfejsu użytkownika (GUI). Pogrupowanie usług w pakiety zostało przeprowadzone zgodnie z ich przydziałem do aktorów, który to przydział został przedstawiony wcześniej.

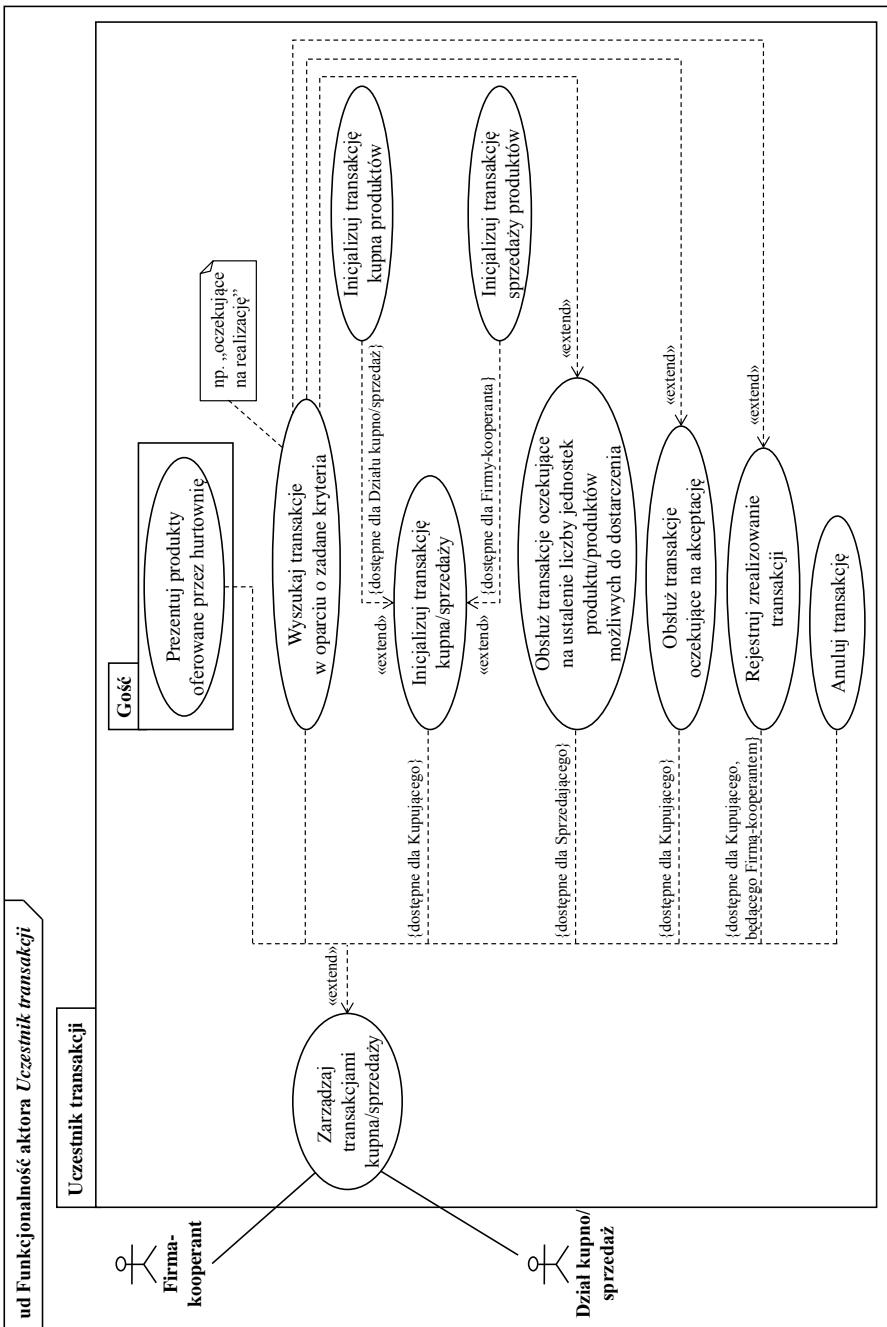
Rys. 11-19 przedstawia ogólny podział na pakiety dokonany w oparciu o diagram kontekstowy, tzn. ka demu aktorowi wprowadzonemu do diagramu kontekstowego przypisano pakiet modeluj cy okre on  dla niego grup e przypadków u ycia. Uwzgl dniono r wnie  pakiet funkcji abstrakcyjnego aktora *Uczestnik transakcji*, który zosta  wprowadzony do modelu zaproponowanego w 11.2.2.



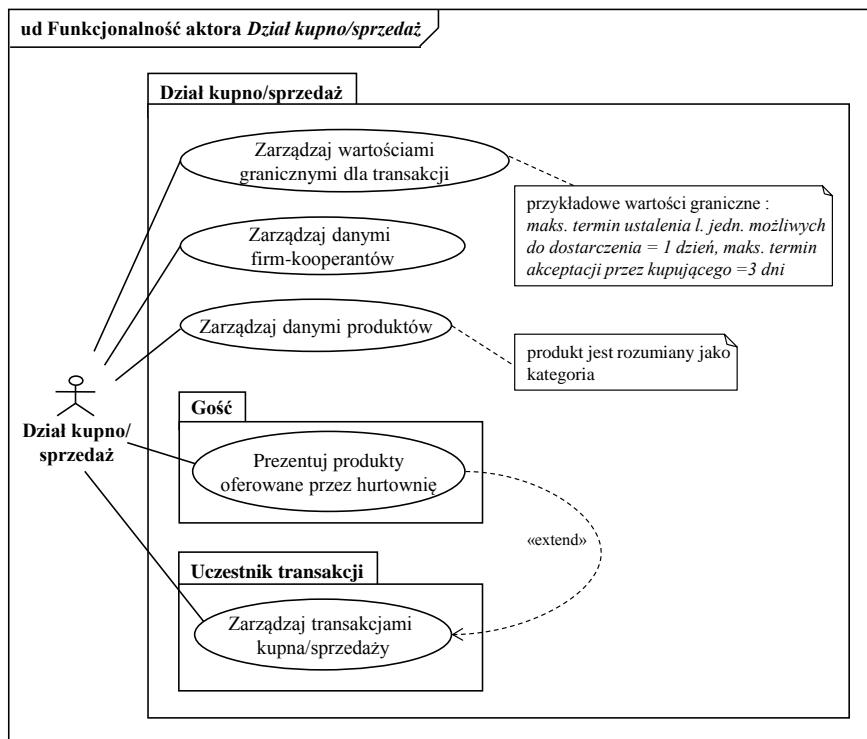
Rys. 11-19 Hurtownia – Interfejs użytkownika

Zawarto  poszczególnych pakietów ilustruj  Rys. 11-20 – Rys. 11-28. Na diagramach pokazano zarówno przypadki z poziomu 1, jak i ich podprzypadki z kolejnych poziomów, co u atwi zidentyfikowanie związków pomiędzy pakietami.

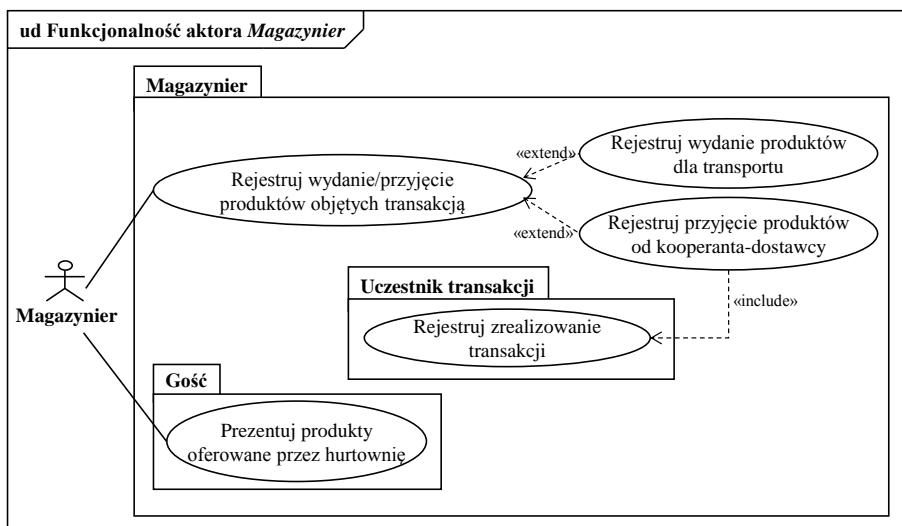
Rys. 11-20 Funkcjonalność aktora *Gość* z pogrupowaniem przypadków w pakietyRys. 11-21 Funkcjonalność aktora *Właściciel hurtowni* z pogrupowaniem przypadków w pakietyRys. 11-22 Funkcjonalność aktorów *Kadry, Księgowość* z pogrupowaniem przypadków w pakiety



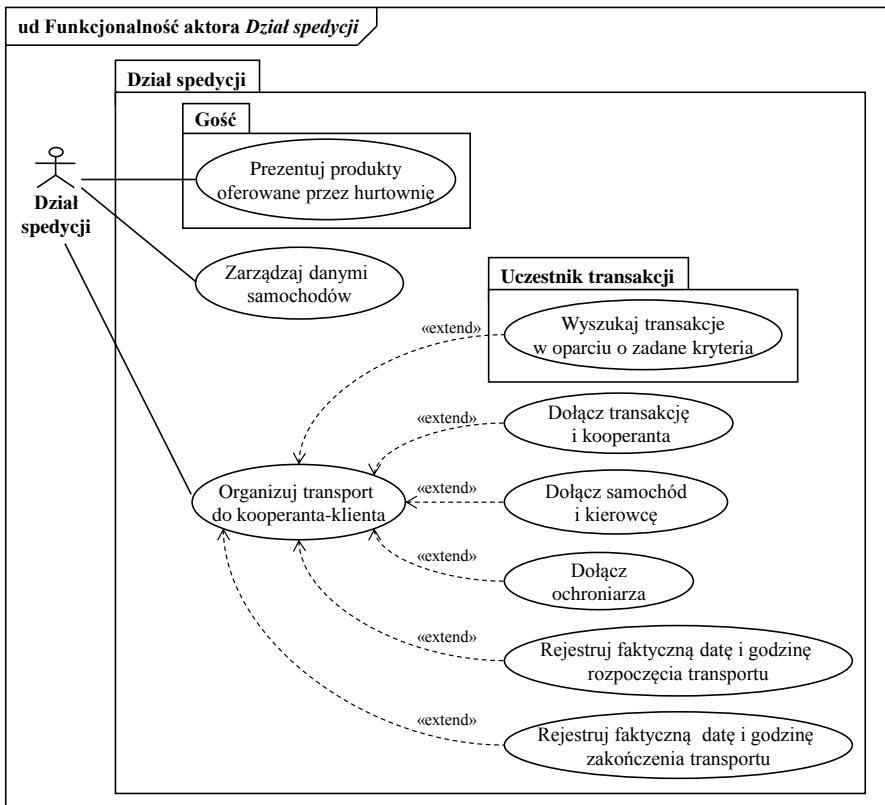
Rys. 11-23 Funkcjonalność aktora *Uczestnik transakcji* z pogrupowaniem przypadków w pakiety



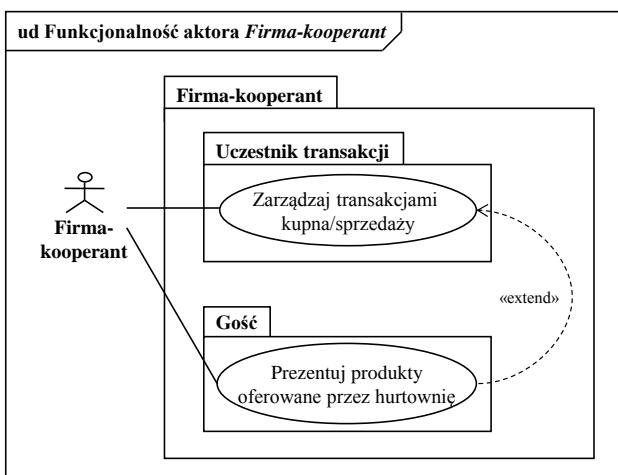
Rys. 11-24 Funkcjonalność aktora *Dział kupno/sprzedaż* z pogrupowaniem przypadków w pakiety



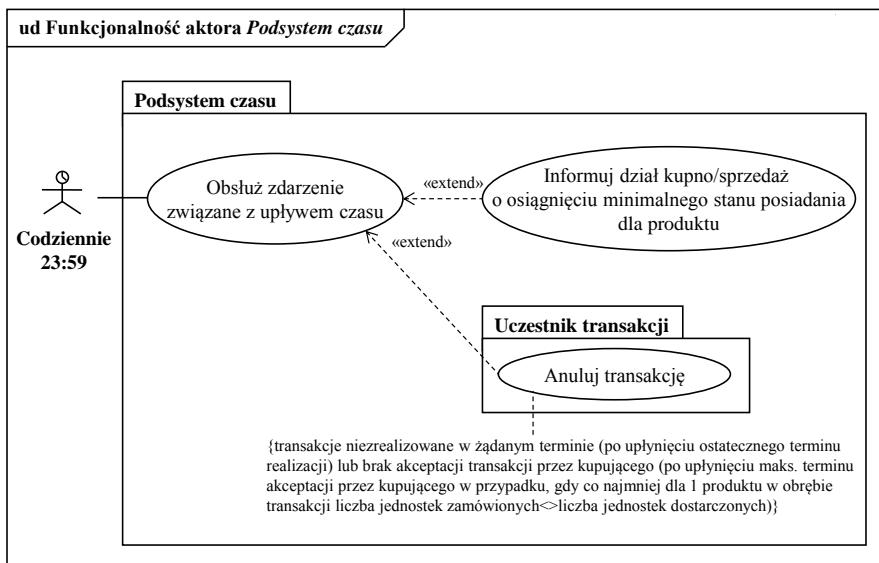
Rys. 11-25 Funkcjonalność aktora *Magazynier* z pogrupowaniem przypadków w pakiety



Rys. 11-26 Funkcjonalność aktora *Dział spedycji* z pogrupowaniem przypadków w pakiety



Rys. 11-27 Funkcjonalność aktora *Firma-kooperant* z pogrupowaniem przypadków w pakiety



Rys. 11-28 Funkcjonalność aktora *Podsystem czasu* z pogrupowaniem przypadków w pakiety

Dla przypomnienia: W dalszych rozważaniach należy uwzględnić hierarchię aktorów (dziedziczenie po aktorze *Gość*) zidentyfikowaną na etapie analizy funkcjonalnej.

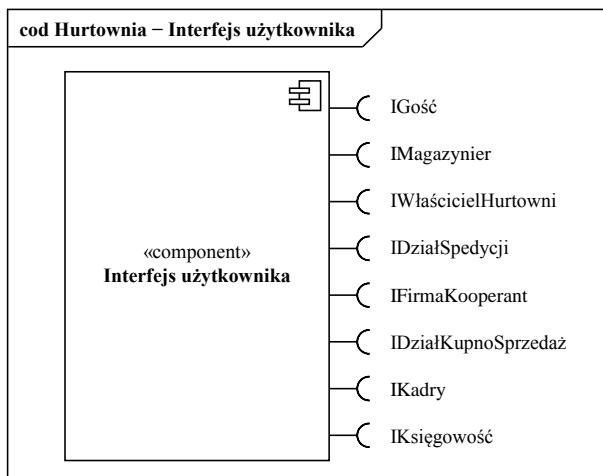
Polecenie 2: Zaprojektuj architekturę interfejsu

Uwaga: Komponenty [38, 40, 41] zostaną wyspecyfikowane w postaci *czarnej skrzynki*, tzn. z perspektywy zewnętrznej bez pokazywania zawartości komponentu; specyfikowane zostaną wyłącznie interfejsy i/lub operacje udostępnione przez komponent za pośrednictwem interfejsu (w postaci niekompletnych sygnatur, tylko nazwy operacji bez list argumentów i typów wartości zwracanych) – w notacji graficznej i notacji tekowej.

Na Rys. 11-29 zaprezentowano komponent *Interfejs użytkownika* w notacji graficznej, a na Rys. 11-30 – tekstowej.

Komponent *Interfejs użytkownika* zawiera w sobie komponenty interfejsów określone dla aktorów zidentyfikowanych podczas analizy funkcjonalnej. Jego strukturę, wynikającą z podziału na pakiety (Rys. 11-19), ilustruje Rys. 11-31.

Interfejsy przykładowych komponentów zagnieżdżonych w komponencie *Interfejs użytkownika* przedstawiono na Rys. 11-32. Zaprezentowano je w notacji tekowej.

Rys. 11-29 Komponent *Interfejs użytkownika* – notacja graficzna

**«component»**  
**Interfejs użytkownika**

**«required interfaces»**

IGośc  
prezentujProduktyOferowanePrzezHurtownię()

IMagazynier  
rejestrujWydanieOdbiórProduktówObjętychTransakcją()  
prezentujProduktyOferowanePrzezHurtownię()

IWłaścicielHurtowni  
generujStatystyki()  
prezentujProduktyOferowanePrzezHurtownię()

IDziałSpedycji  
zarządzajDanymiSamochodów()  
organizujTransportDoKooperantaKlienta()  
prezentujProduktyOferowanePrzezHurtownię()

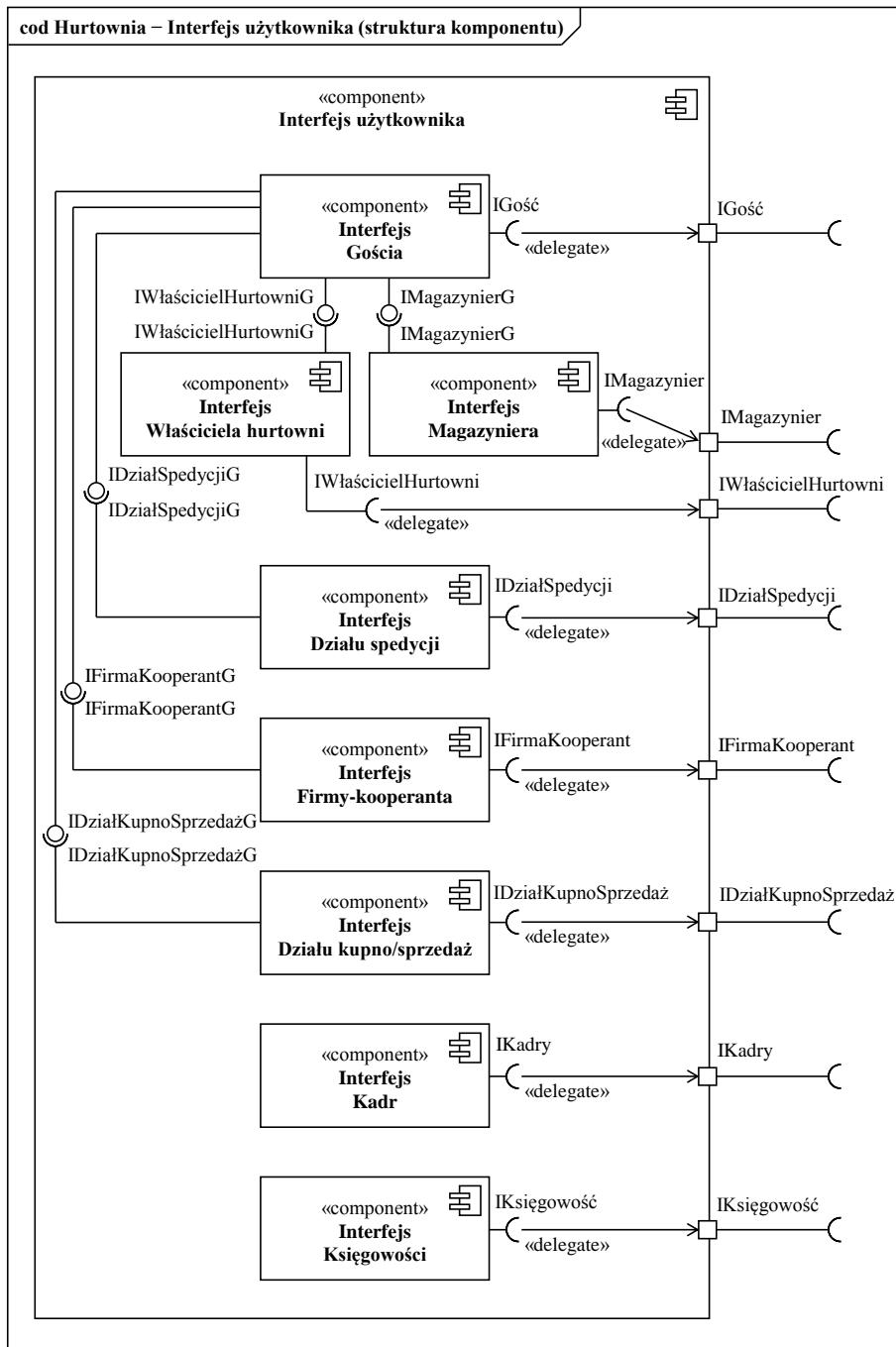
IFirmaKooperant  
zarządzajTransakcjamiKupnaSprzedaży()  
prezentujProduktyOferowanePrzezHurtownię()

IDziałKupnoSprzedaż  
zarządzajDanymiProduktów()  
zarządzajDanymiFirmKooperantów()  
zarządzajWartościamiGranicznymiDlaTransakcji()  
zarządzajTransakcjamiKupnaSprzedaży()  
prezentujProduktyOferowanePrzezHurtownię()

IKadry  
zarządzajDanymiPracowników()

IKsięgowość  
wyliczWynagrodzenieMiesięcznePracownika()

Rys. 11-30 Komponent *Interfejs użytkownika* – notacja tekstowa

Rys. 11-31 Struktura komponentu *Interfejs użytkownika* – notacja graficzna

<b>cod Hurtownia – interfejsy przykładowych komponentów zagnieźdzonych w komponencie <i>Interfejs użytkownika</i></b>	
<b>«component»</b> <b>Interfejs Gościa</b>	
«provided interfaces»	
IWłaścicielHurtowniG prezentujProduktyOferowanePrzezHurtownię()	
IFirmaKooperantG prezentujProduktyOferowanePrzezHurtownię()	
IDziałSpedycjiG prezentujProduktyOferowanePrzezHurtownię()	
IDziałKupnoSprzedażG prezentujProduktyOferowanePrzezHurtownię()	
IMagazynierG prezentujProduktyOferowanePrzezHurtownię()	
«required interfaces»	
IGość prezentujProduktyOferowanePrzezHurtownię()	
<b>«component»</b> <b>Interfejs Właściciela hurtowni</b>	
«required interfaces»	
IWłaścicielHurtowniG prezentujProduktyOferowanePrzezHurtownię()	
IWłaścicielHurtowniI generujStatystyki()	
<b>«component»</b> <b>Interfejs Firmy-kooperanta</b>	
«required interfaces»	
IFirmaKooperantG prezentujProduktyOferowanePrzezHurtownię()	
IFirmaKooperant zarządzajTransakcjamiKupnaSprzedaży()	
<b>«component»</b> <b>Interfejs Kadr</b>	
«required interfaces»	
IKadry zarządzajDanymiPracowników()	
<b>«component»</b> <b>Interfejs Księgowości</b>	
«required interfaces»	
IKsięgowość wyliczWynagrodzenieMiesięcznePracownika()	

Rys. 11-32 Interfejsy przykładowych komponentów zagnieźdzonych w komponencie *Interfejs użytkownika* – notacja tekstowa

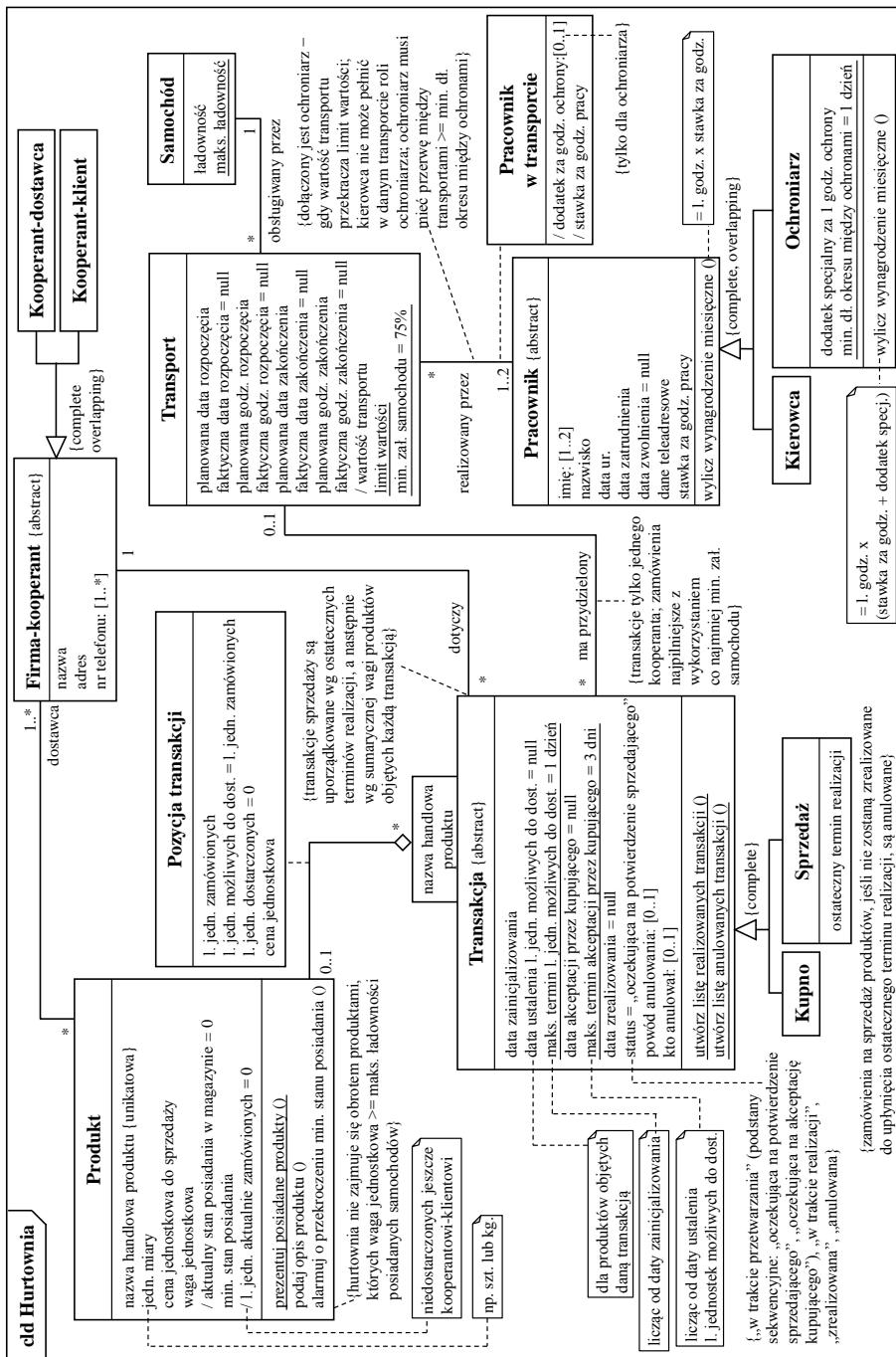
Powyżej zaprezentowane przykłady z jawną specyfikacją interfejsów pokazują zasadę, zgodnie z którą należałoby postępować, określając interfejsy dla wszystkich zidentyfikowanych komponentów.

## 11.3 Analiza strukturalna

### 11.3.1 Schemat pojęciowy

*Polecenie:* Dla podanego tekstu wymagań skonstruuj schemat pojęciowy (nie redukując liczności).

Schemat pojęciowy [36, 41, 42, 45] przedstawiono na Rys. 11-33. *Uwaga:* Dla zachowania przejrzystości modelu wprowadzono do klas jedynie te metody, które wynikały z 15 punktu tekstu wymagań (patrz 11.1 Tekst wymagań).

Rys. 11-33 Schemat pojęciowy dla systemu *Hurtownia*

### 11.3.2 Analiza elementów pochodnych

Na schemacie pojęciowym przedstawionym na Rys. 11-33 oznaczono następujące atrybuty jako elementy pochodne: *aktualny stan posiadania w magazynie i l. jedn. aktualnie zamówionych* w klasie *Produkt*, *dodatek za godz. ochrony i stawka za godz. pracy* w klasie *Pracownik w transporcie*, oraz *wartość transportu* w klasie *Transport*.

1. Atrybuty: *aktualny stan posiadania w magazynie i l. jedn. aktualnie zamówionych* w klasie *Produkt*

W przypadku atrybutów *aktualny stan posiadania w magazynie i l. jedn. aktualnie zamówionych* interesuje nas wyłącznie ich aktualna wartość wyliczana na podstawie transakcji. Wydaje się więc, że wystarczy wprowadzić do schematu wyłącznie metody *obl. aktualny stan posiadania w magazynie()* oraz *obl. l. jedn. aktualnie zamówionych()*. Należy jednak zwrócić uwagę na fakt, że wartość aktualnego stanu posiadania danego produktu w magazynie jest potrzebna użytkownikowi na bieżąco i wyliczanie jej za każdym razem na podstawie wszystkich transakcji nie wydaje się dobrym rozwiązaniem.

Może się zdarzyć, że na początku wdrożenia systemu mogą już być produkty na stanie magazynu (bez informacji o transakcjach zakupu tych produktów) i wówczas atrybut ten nie powinien być pochodny. Jego wartość mogłaby być w takiej sytuacji aktualizowana podczas wprowadzania informacji o transakcjach, co umożliwiłoby zarządzanie tą wartością. Analogiczne rozwiązania można przeprowadzić dla atrybutu *l. jedn. aktualnie zamówionych*.

Po analizie tych wartości podjęto decyzję o pozostawieniu w tym momencie atrybutów na schemacie oznaczonych jako pochodne, co nie zakazuje zmiany sposobu ich obsługiwanego w kolejnych etapach prac.

2. Atrybuty: *dodatek za godz. ochrony i stawka za godz. pracy* w klasie *Pracownik w transporcie*

Z analizy zmian wartości na przestrzeni czasu wynika, że atrybuty *dodatek za godz. ochrony i stawka za godz. pracy* powinny być przechowywane jako atrybuty bazowe, a nie pochodne. Wynika to z faktu, że dla danego pracownika biorącego udział w transporcie produktów, stawki mogą ulegać zmianom na przestrzeni czasu (w efekcie zmiany wartości stawki za godzinę pracy oraz dodatku specjalnego za godzinę ochrony). Mogłoby to spowodować błędne obliczenie wynagrodzenia pracownika w miesiącu, w którym nastąpiła zmiana stawek.

3. Atrybut *wartość transportu* w klasie *Transport*

W przypadku atrybutu *wartość transportu* nie widać istotnej przesłanki dla przechowywania jego wartości. Wartość ta jest w prosty sposób wyliczana na podstawie cen produktów w momencie realizacji danej transakcji (atribut *cena jednostkowa* w klasie asocjacji *Pozycja transakcji*) i nie ulega zmianom na

przestrzeni czasu. Wydaje się więc, że wystarczy wprowadzić do schematu wyłącznie metodę *obl. wartość transportu()*.

### 11.3.3 Analiza wartości początkowych

Analiza wartości początkowych została przeprowadzona już w trakcie konstruowania schematu pojęciowego. Lista domyślnych wartości początkowych została zaprezentowana poniżej:

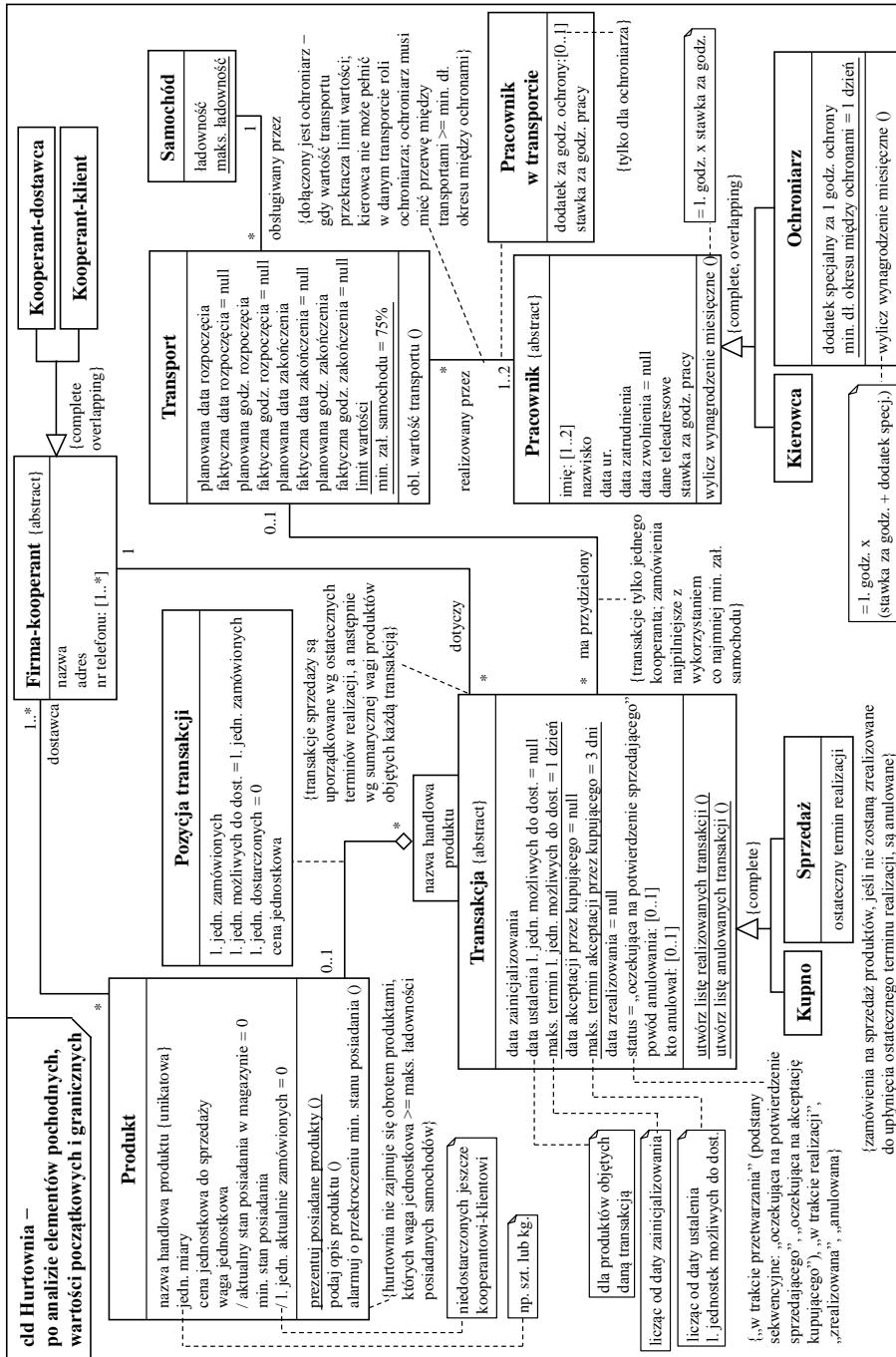
- Klasa *Transakcja* – atrybuty: *data ustalenia l. jedn. możliwych do dost.*, *data akceptacji przez kupującego*, *data zrealizowania* zainicjalizowane wartością *null*;
- klasa *Transakcja* – atrybut *status* zainicjalizowany wartością „*oczekująca na potwierdzenie sprzedającego*”;
- klasa *Pozycja transakcji* – atrybut *l. jedn. możliwych do dost.* zainicjalizowany wartością atrybutu *l. jedn. zamówionych*;
- klasa *Pozycja transakcji* – atrybut *l. jedn. dostarczonych*. zainicjalizowany wartością *0* (wartość ta zostanie zmieniona w momencie zrealizowania transakcji);
- klasa *Transport* – atrybuty: *faktyczna data rozpoczęcia*, *faktyczna godz. rozpoczęcia*, *faktyczna data zakończenia*, *faktyczna godz. zakończenia* zainicjalizowane wartością *null*;
- klasa *Pracownik* – atrybut *data zwolnienia* zainicjalizowany wartością *null*.

### 11.3.4 Analiza wartości granicznych

W trakcie konstruowania schematu pojęciowego uwzględniono rozważania dotyczące tych wartości granicznych, które na przestrzeni czasu mogłyby ulegać zmianom. W związku z powyższym, wprowadzono do schematu atrybuty klasowe, zainicjalizowane wartościami wynikającymi z tekstu wymagań:

- klasa *Transakcja* – atrybuty: *maks. termin ustalenia l. jedn. możliwych do dost.* zainicjalizowany wartością *1 dzień* oraz *maks. termin dla akceptacji przez kupującego* zainicjalizowany wartością *3 dni*;
- klasa *Transport* – atrybut *min. zal. samochodu* zainicjalizowany wartością *75%*;
- klasa *Ochroniarz* – atrybut *min. dł. okresu między ochronami* zainicjalizowany wartością *1 dzień*.

Schemat pojęciowy po modyfikacji uwzględniającej analizę elementów pochodnych, wartości początkowych oraz wartości granicznych został przedstawiony na Rys. 11-34. Dalsze rozważania będą prowadzone w oparciu o ten schemat.



Rys. 11-34 Schemat pojęciowy po analizie el. pochodnych, wart. początkowych i granicznych

### 11.3.5 Zadania podstawowe

Zaprezentowane poniżej rozwiązania zadań odnoszą się do schematu pojęciowego z uwzględnioną analizą elementów pochodnych, wartości początkowych i granicznych (przedstawionego na Rys. 11-34).

**Zad. 1:** Podkreśl te rodzaje dziedziczenia (rozłączne, nierozłączne, kompletne, niekompletne, jednokrotne, wielokrotne, jednoaspektowe, wieloaspektowe, dynamiczne, elipsa), które uważaś za potrzebne do wykorzystania na schemacie pojęciowym dla systemu *Hurtownia*. Dla każdego podkreślonego rodzaju dziedziczenia podaj hierarchię/aspekt, w której ten rodzaj dziedziczenia wystąpił.

- Dziedziczenie rozłączne (ang. *disjoint*): hierarchia transakcji;
- dziedziczenie nierozłączne (ang. *overlapping*): hierarchia pracowników, hierarchia firm-kooperantów;
- dziedziczenie kompletne: wszystkie hierarchie;
- dziedziczenie jednokrotne: wszystkie hierarchie;
- dziedziczenie jednoaspektowe: wszystkie hierarchie.

**Zad. 2:** Podaj przykłady metody klasowej, metody obiektowej, atrybutu klasowego i atrybutu obiektowego.

- Metoda obiektowa: *wylicz wynagrodzenie miesięczne ()* w klasie *Ochroniarz*;
- metoda klasowa: *utwórz listę realizowanych transakcji (okres)* – operuje na ekstensji klasy *Transakcja*;
- atrybut obiektowy: *nazwisko* w klasie *Pracownik*;
- atrybut klasowy: *limit wartości* w klasie *Transport*.

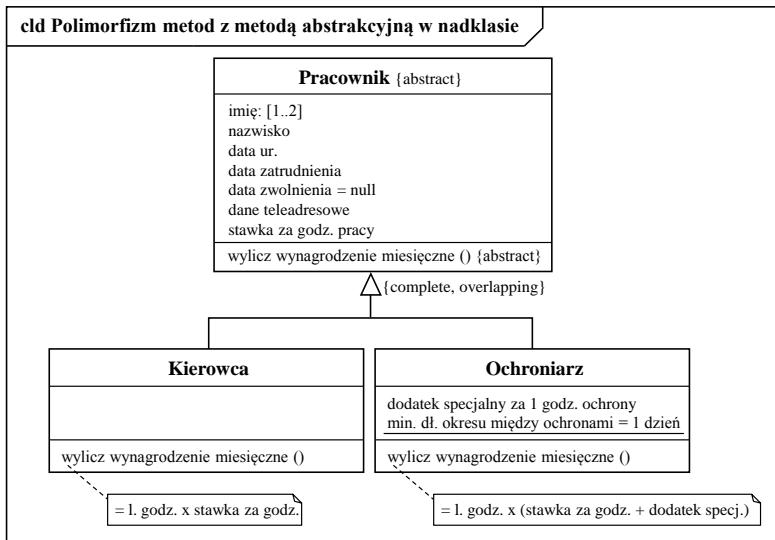
**Zad. 3:** Jakie rodzaje metod (abstrakcyjne, zaimplementowane) może zawierać klasa abstrakcyjna, a jakie klasa konkretna? Podaj przykłady.

Klasa abstrakcyjna może zawierać oba rodzaje metod, a klasa konkretna tylko metody zaimplementowane.

- Metoda abstrakcyjna w klasie abstrakcyjnej: na diagramie nie ma metody abstrakcyjnej, ale można by umieścić taką metodę w klasie abstrakcyjnej *Pracownik* (*wylicz wynagrodzenie miesięczne ()*) – wtedy należałoby umieścić też odpowiednią metodę zaimplementowaną w klasie *Kierowca*, co ilustruje Rys. 11-35;
- metoda zaimplementowana w klasie abstrakcyjnej: *utwórz listę realizowanych transakcji (okres)* w klasie *Transakcja*;
- metoda zaimplementowana w klasie konkretnej: *wylicz wynagrodzenie miesięczne ()* w klasie *Ochroniarz*.

**Uwaga:** W odpowiedzi na to polecenie (w przypadku sprawdzianu/egzaminu) często brakuje przykładu dla metody zaimplementowanej w klasie abstrakcyjnej. Zdarza się, że osoby nauczane/uczące się piszą: „na moim diagramie nie ma takiej metody”, ale w takiej sytuacji należy do diagramu wprowadzić przykładową metodę, co nie

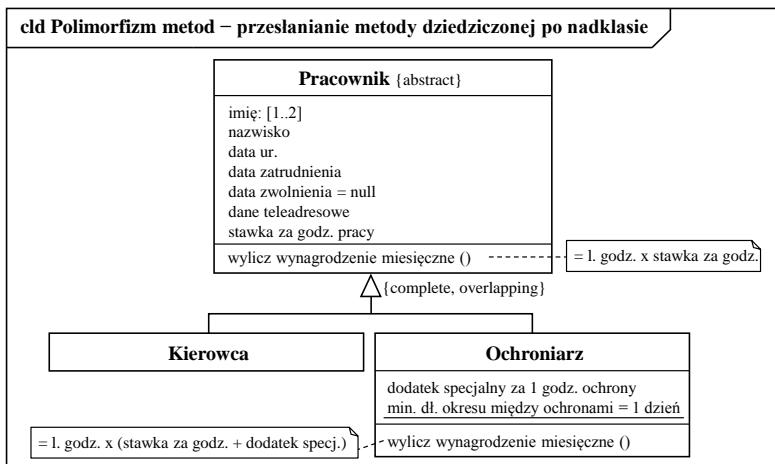
jest zadaniem trudnym (może to być np. metoda typu *setter* albo *getter* dla atrybutu zdefiniowanego w klasie abstrakcyjnej).



Rys. 11-35 Polimorfizm metod z metodą abstrakcyjną umieszczoną w klasie abstrakcyjnej

**Zad. 4:** Wskaż przykład zastosowania polimorfizmu metod.

Metody polimorficzne to metody *wylicz wynagrodzenie miesięczne ()* w klasach hierarchii zbudowanej dla pracowników. Jest to przykład przesłaniania metod. W klasie *Ochroniarz* metoda *wylicz wynagrodzenie miesięczne ()* dziedziczona po klasie *Pracownik* jest przesłonięta własną metodą opartą na innym algorytmie, przy czym „własna” oznacza zdefiniowaną w klasie *Ochroniarz* (patrz Rys. 11-36).



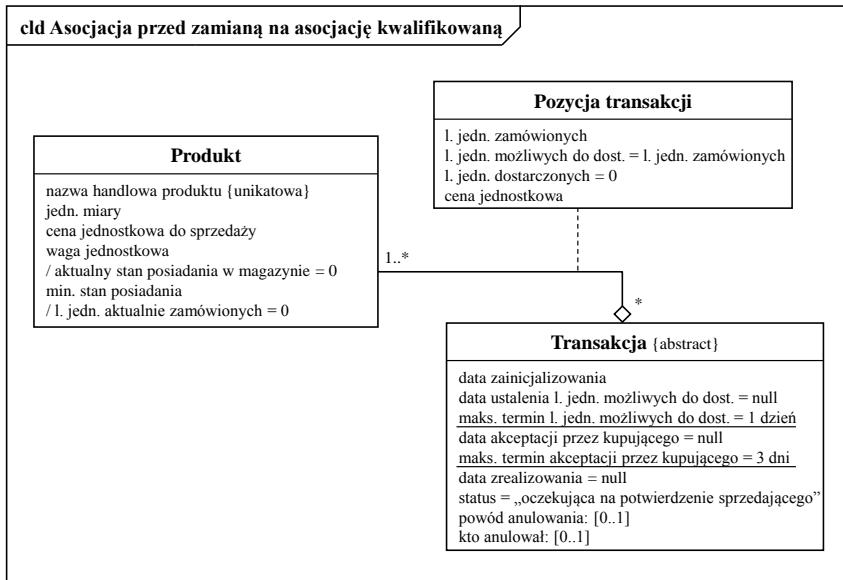
Rys. 11-36 Polimorfizm metod z przesłanianiem metody dziedziczonej z nadklasy

**Zad. 5:** Asocjację posiadającą niesymetryczne liczności oraz atrybut (lub klasę asocjacji) zamień na asocjację kwalifikowaną. Uzasadnij, dlaczego poddałeś zamianie właśnie tę asocjację.

Na schemacie pojęciowym umieszczono asocjację kwalifikowaną łączącą klasy *Produkt* i *Transakcja*. Wskazuje ona, że kluczem do przeszukiwania zbioru produktów przypisanych do danej transakcji może być unikatowa nazwa handlowa produktu.

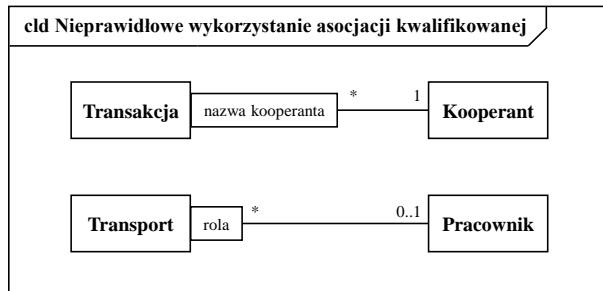
*Dla przypomnienia:* Po zamianie asocjacji „zwykłej” na asocjację kwalifikowaną nie usuwamy kwalifikatora z klasy, w której był pierwotnie umieszczony, ponieważ pozostawienie tego atrybutu wydaje się bardziej sprzyjać prawidłowej percepcej modelu (takie rozwiązanie można spotkać w niektórych publikacjach dotyczących analizy obiektowej).

Rys. 11-37 przedstawia fragment schematu pojęciowego sprzed zamiany asocjacji „zwykłej” na asocjację kwalifikowaną. Dla zachowania przejrzystości rysunku pominięto metody, ograniczenia i komentarze nieistotne dla przeprowadzanych rozważań.



Rys. 11-37 Fragment schematu pojęciowego bez wykorzystania asocjacji kwalifikowanej

*Uwaga:* Niewłaściwym miejscem dla wykorzystania asocjacji kwalifikowanej jest asocjacja pomiędzy klasami *Transakcja* i *Firma-kooperant* (kluczem miałaby tu być nazwa kooperanta) czy asocjacja *Transport-Pracownik* (klucz – rola pracownika w transporcie) – Rys. 11-38.



Rys. 11-38 Nieprawidłowe wykorzystanie asocjacji kwalifikowanej

W obu przypadkach wykorzystano klucze do przeszukiwania bardzo małych zbiorów: jednoelementowego (dla pierwszego przypadku) i co najwyżej dwuelementowego (dla drugiego przypadku), co nie ma większego sensu, ponieważ klucze mają wspomóc przeszukiwanie zbiorów dużych.

**Zad. 6:** Wskaż asocjację, która jest agregacją lub kompozycją (lub może kwalifikować się do zamiany na takową). Objaśnij, dlaczego zdecydowałeś się na wybór agregacji/kompozycji, a nie „zwykłej” asocjacji? Ponadto, dlaczego wybrałeś agregację, a nie kompozycję (czy też odwrotnie)?

Na schemacie pojęciowym aggregacja występuje pomiędzy klasami: *Produkt* i *Transakcja*. Agregacja lepiej niż zwykła asocjacja wskazuje na związek części-całości określony pomiędzy transakcją a objętymi przez nią produktami. Wykorzystano aggregację, a nie kompozycję, ponieważ usunięcie z systemu obiektu klasy *Transakcja* nie będzie skutkowało usunięciem związkanych z nią obiektów klasy *Produkt*, czyli cykl życia części, którym jest produkt, nie zawiera się w cyklu życia całości – transakcji. Ponadto, na przestrzeni czasu każdy z produktów mógł być elementem składowym wielu transakcji.

**Zad. 7:** Wskaż przykładowe ograniczenie. Na jaki element modelu zostało ono nałożone? Czy jest to ograniczenie statyczne czy dynamiczne? Zaproponuj, jaka metoda i w jakiej klasie mogłaby być odpowiedzialna za jego zapewnienie.

Można wskazać ograniczenie, że ochroniarz musi być dołączony do transportu, gdy wartość transportu przekracza limit wartości – jest to ograniczenie statyczne, implementowane np. w metodzie *przydziel ochroniarza (ochroniarz)* (w klasie *Transport*). Inny przykład to ograniczenie, że ochroniarz musi mieć jeden dzień przerwy pomiędzy transportami, które ochrania (ograniczenie dynamiczne), implementowane w tej samej metodzie co poprzednie. Oba ograniczenia dotyczą asocjacji pomiędzy klasami *Transport* i *Pracownik*. W przypadku ograniczeń dynamicznych ważny jest poprzedni stan tego elementu modelu (tu asocjacji), na który jest nakładane ograniczenie.

*Uwaga:* Ograniczenie, że hurtownia nie zajmuje się obrotom produktami, których jedna jednostka nie da się przewieźć najbardziej ładownym samochodem hurtowni,

nie jest ograniczeniem na wagę jednostkową produktu, ale na istnienie obiektu klasy *Produkt* i powinno być implementowane w metodzie tworzącej nowy obiekt tej klasy.

### **11.3.6 Omówienie najczęściej popełnianych błędów – na podstawie modeli tworzonych przez osoby nauczane/uczące się**

- Wprowadzanie do diagramów informacji, które ani bezpośrednio, ani pośrednio nie zostały umieszczone w tekście wymagań, jak np. *id* dla transakcji czy też *elipsa* dla dziedziczenia po klasie *Pracownik* (w tekście wymagań wyróżniono wyłącznie dwa rodzaje pracowników: kierowców i ochroniarzy).
- Zapominanie o regułach nazewnictwa klas: (1) nazwa klasy to zazwyczaj rzeczownik w liczbie pojedynczej, (2) wybór nazwy dla klasy jest zależny od rodzaju informacji przechowywanej w jej obiektach, tzn., jeżeli pojedynczy obiekt przechowuje informację o jednym produkcie, to klasa nazywa się *Produkt*, a jeśli o zbiorze produktów, to klasa nazywałaby się *Produkty*.

Ponadto, należy też dobrze zrozumieć, co opisuje jeden obiekt danej klasy: czy opisuje ją pełniącą rolę klasyfikatora, czy może jedną instancję (egzemplarz) danego klasyfikatora. W tym konkretnie zadaniu jeden obiekt klasy *Produkt* opisuje rodzaj (kategorię) produktów (czyli klasyfikator), a nie jeden egzemplarz produktu danego rodzaju. Dlatego klasa *Produkt* musi posiadać atrybut związany ze stanem posiadania produktu (z liczbą jednostek danego produktu będącego na stanie magazynu hurtowni).

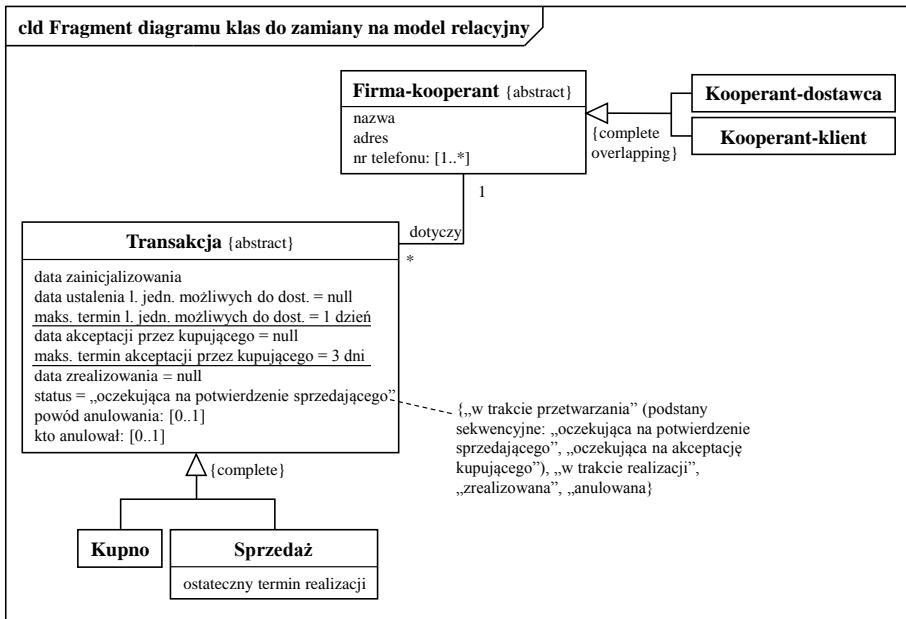
### **11.3.7 Schemat relacyjny dla fragmentu diagramu klas**

*Polecenie:* Fragment diagramu klas zawierający dwa związki dziedziczenia i asocjację przedstaw w modelu relacyjnym.

Fragment diagramu klas ze schematu pojęciowego podlegający zamianie na model relacyjny [44, 48] przedstawiono na Rys. 11-39. Dla przejrzystości rysunku pominięto metody oraz nieistotne ograniczenia i komentarze.

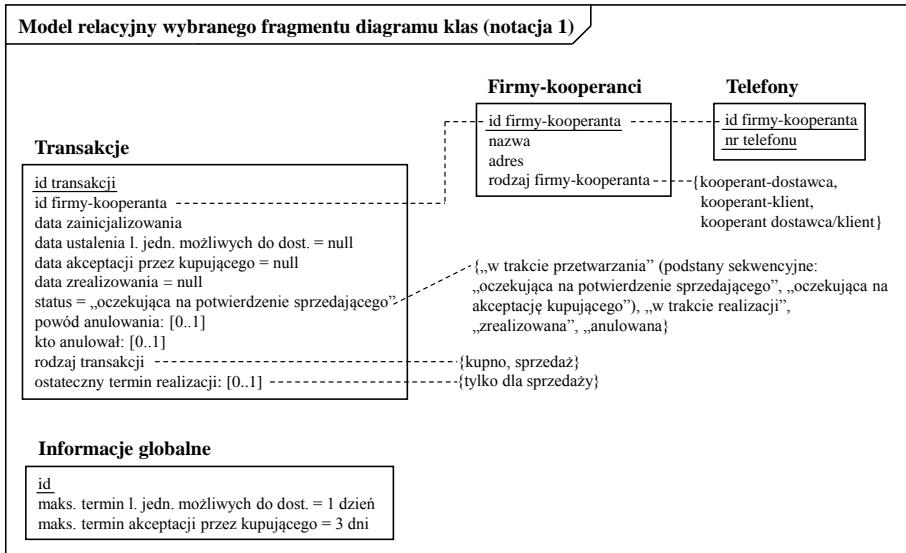
*Obejście dziedziczenia:* Wybrany fragment schematu pojęciowego zawierał dwie hierarchie, które zostały przekształcone w następujący sposób: dla każdej z nich utworzono jedną tabelę poprzez przeniesienie do nadklasy wszystkich właściwości wynikających z podziału oraz dodanie dodatkowego atrybutu – dyskryminatora wariantu.

Model relacyjny został przedstawiony w trzech przykładowych notacjach, dwie pierwsze zostały omówione m.in. w: [41], a trzecią (ang. *Crow's Foot*) można uznać za jedną z częściej wykorzystywanych [44].

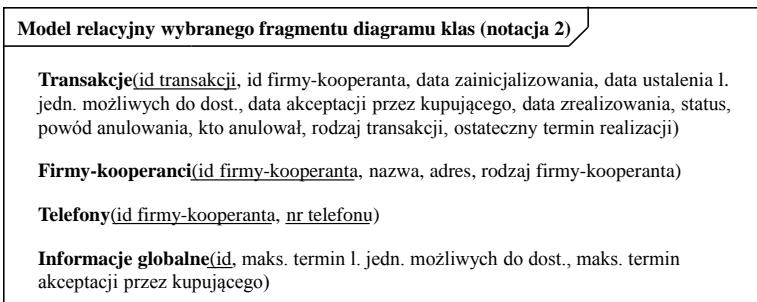


Rys. 11-39 Fragment diagramu klas podlegający zamianie na model relacyjny

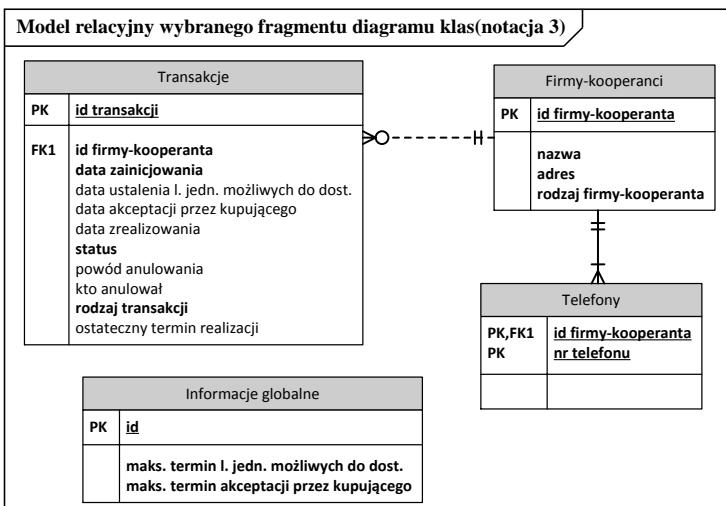
Model relacyjny tego fragmentu diagramu klas pokazano w trzech notacjach, odpowiednio na Rys. 11-40 – Rys. 11-42.



Rys. 11-40 Model relacyjny wybranego fragmentu diagramu klas (notacja 1)



Rys. 11-41 Model relacyjny wybranego fragmentu diagramu klas (notacja 2)



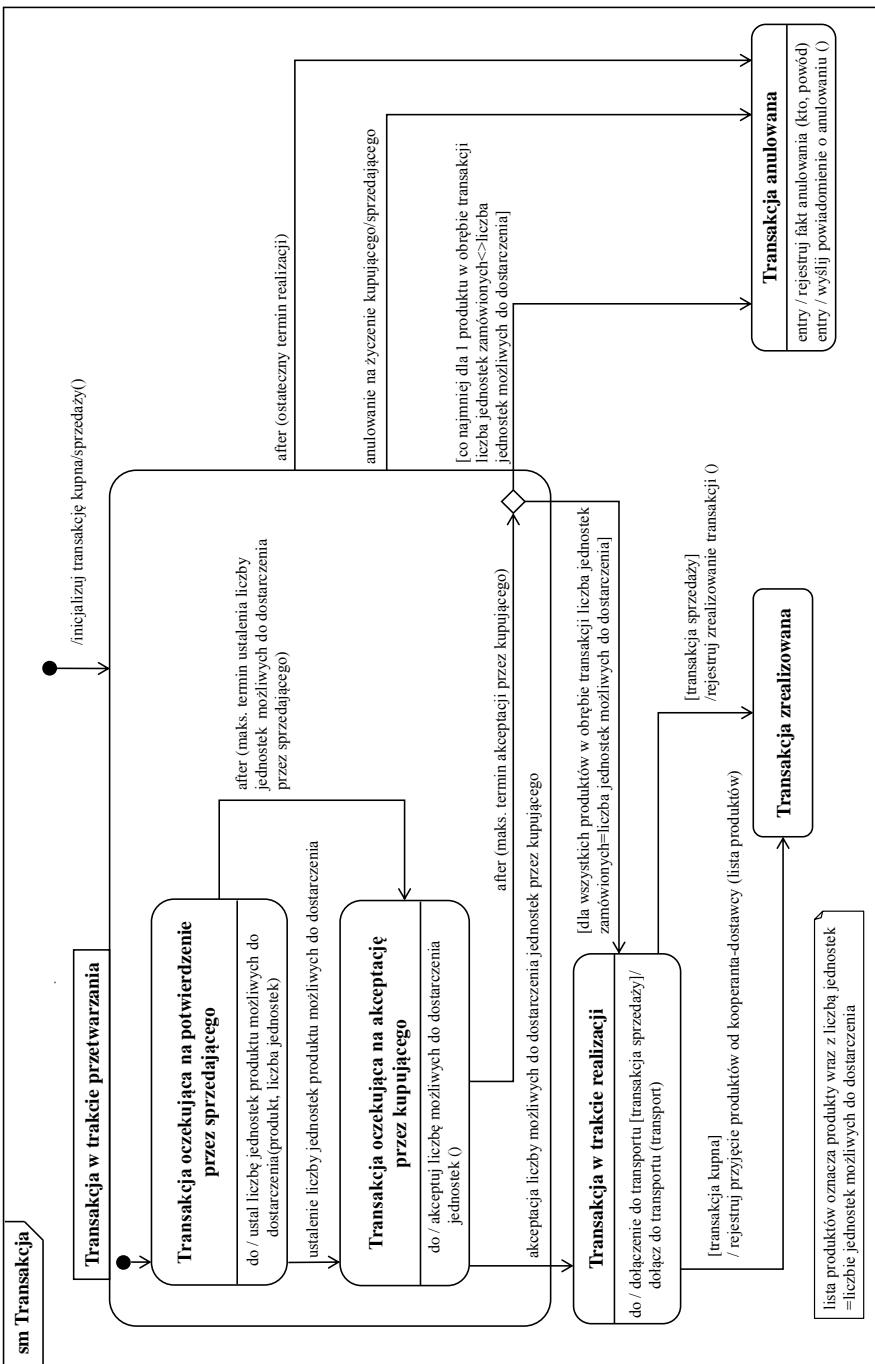
Rys. 11-42 Model relacyjny wybranego fragmentu diagramu klas (notacja 3)

#### 11.4 Analiza dynamiczna

#### 11.4.1 Maszyna stanowa dla klasy *Transakcja*

**Zad. 1:** Dla klasy *Transakcja* wskaż kilka stanów (co najmniej dwa stany, nie licząc stanów początkowego i końcowego), w jakich mogą znajdować się obiekty tej klasy. Przedstaw diagram stanów zawierający te stany wraz ze specyfikacją przejścia/przejść pomiędzy nimi.

Maszyna stanowa [40, 41] dla klasy *Transakcja* została przedstawiona na Rys. 11-43. Należy zwrócić uwagę na brak informacji w tekście wymagań o warunkach usuwania transakcji. W związku z powyższym, do diagramu na Rys. 11-43 nie został wprowadzony stan końcowy.

Rys. 11-43 Maszyna stanowa dla klasy *Transakcja*

## 11.5 Podział schematu pojęciowego na pakiety

*Polecenie:* Zaproponuj podział schematu pojęciowego (z Rys. 11-34) na pakiety

Przed podziałem schematu pojęciowego na pakiety [41] należy przeprowadzić analizę danych, które są potrzebne do wykonania każdego ze zidentyfikowanych przypadków użycia. W wyniku tej analizy zostanie zaproponowany podział struktury i próba rozłożenia przypadków w tej strukturze.

*Uwaga:* Rozważane będą przypadki w ich aspekcie wykonawczym.

Na wstępie, dla każdego przypadku użycia rozważono, w których klasach znajdują się dane potrzebne do jego realizacji. Wyniki tych rozważań zaprezentowano w Tab. 11-4. Przypadki użycia zamieszczone w tabeli zostały uporządkowane według nazwy.

W kolumnie *Klasa startowa* podano nazwy klas, gdzie została zdefiniowana metoda, od której rozpocznie się realizacja części wykonawczej danego przypadku. W kolumnie *Klasy pomocnicze* zamieszczono nazwy klas, których obiekty będą przechowywały potrzebne dane.

Tab. 11-4 Lista przypadków użycia z przypisaniem klas, w których znajdują się dane potrzebne do ich realizacji

Nazwa przypadku użycia	Nr	Klasa startowa	Klasy pomocnicze
Anuluj transakcję	7.6	Transakcja	Produkt, Firma-kooperant, Pozycja transakcji
Dolacz ochroniarza	9.5	Transport	Samochód, Ochroniarz, Pracownik w transporcie, Produkt, Pozycja transakcji
Dolacz samochód i kierowcę	9.4	Transport	Samochód, Kierowca, Pracownik w transporcie
Dolacz transakcję i kooperanta	9.3	Transport	Transakcja, Firma-kooperant, Produkt, Pozycja transakcji
Generuj statystyki	10.	Transakcja	
Informuj dział kupno/sprzedaż o osiągnięciu minimalnego stanu posiadania dla produktów	12.1	Produkt	

<i>Inicjalizuj transakcję kupna produktów</i>	7.2.1	Transakcja	Produkt, Firma-kooperant, Pozycja transakcji
<i>Inicjalizuj transakcję kupna/sprzedaży</i>	7.2	Transakcja	Produkt, Firma-kooperant, Pozycja transakcji
<i>Inicjalizuj transakcję sprzedaży produktów</i>	7.2.2	Transakcja	Produkt, Firma-kooperant, Pozycja transakcji
<i>Obsłuż zdarzenie związane z upływem czasu</i>	12.	Transakcja	Produkt, Pozycja transakcji
<i>Obsłuż transakcje oczekujące na akceptację</i>	7.3	Transakcja	Produkt, Pozycja transakcji
<i>Obsłuż transakcje oczekujące na ustalenie liczby jednostek produktu/produktów możliwych do dostarczenia</i>	7.4	Transakcja	Produkt, Pozycja transakcji
<i>Organizuj transport do kooperanta-klienta</i>	9.	Transport	Transakcja, Firma-kooperant, Produkt
<i>Podaj opis wybranego produktu</i>	6.1	Produkt	
<i>Podaj, dla zadanego okresu, listę anulowanych transakcji</i>	10.2	Transakcja	
<i>Prezentuj produkty oferowane przez hurtownię</i>	6.	Produkt	
<i>Rejestruj faktyczną datę i godzinę rozpoczęcia transportu</i>	9.1	Transport	
<i>Rejestruj faktyczną datę i godzinę zakończenia transportu</i>	9.2	Transport	
<i>Rejestruj przyjęcie produktów od kooperanta-dostawcy</i>	8.2	Kupno	Produkt, Firma-kooperant, Pozycja transakcji
<i>Rejestruj wydanie produktów dla transportu</i>	8.1	Sprzedaż	Transport, Produkt, Pozycja transakcji
<i>Rejestruj wydanie/przyjęcie produktów objętych transakcją</i>	8.	Transakcja	Produkt, Pozycja transakcji
<i>Rejestruj zrealizowanie transakcji</i>	7.1	Transakcja	

<i>Utwórz, dla zadanego okresu, listę realizowanych transakcji</i>	10.1	Transakcja	
<i>Wylicz wynagrodzenie miesięczne pracownika</i>	11.	Pracownik	Transport, Pracownik w transporcie
<i>Wyszukaj produkty spełniające zadane kryteria</i>	6.2	Produkt	
<i>Wyszukaj transakcje w oparciu o zadane kryteria</i>	7.5	Transakcja	Produkt, Firma-kooperant, Pozycja transakcji
<i>Zarządzaj danymi firm-kooperantów (CRUD)</i>	2.	Firma-kooperant	
<i>Zarządzaj danymi pracowników (CRUD)</i>	1.	Pracownik	
<i>Zarządzaj danymi produktów (CRUD)</i>	3.	Produkt	
<i>Zarządzaj danymi samochodów (CRUD)</i>	4.	Samochód	
<i>Zarządzaj transakcjami kupna/sprzedaży</i>	7.	Transakcja	Produkt, Firma-kooperant
<i>Zarządzaj wartościami granicznymi dla transakcji</i>	5.	Transakcja	

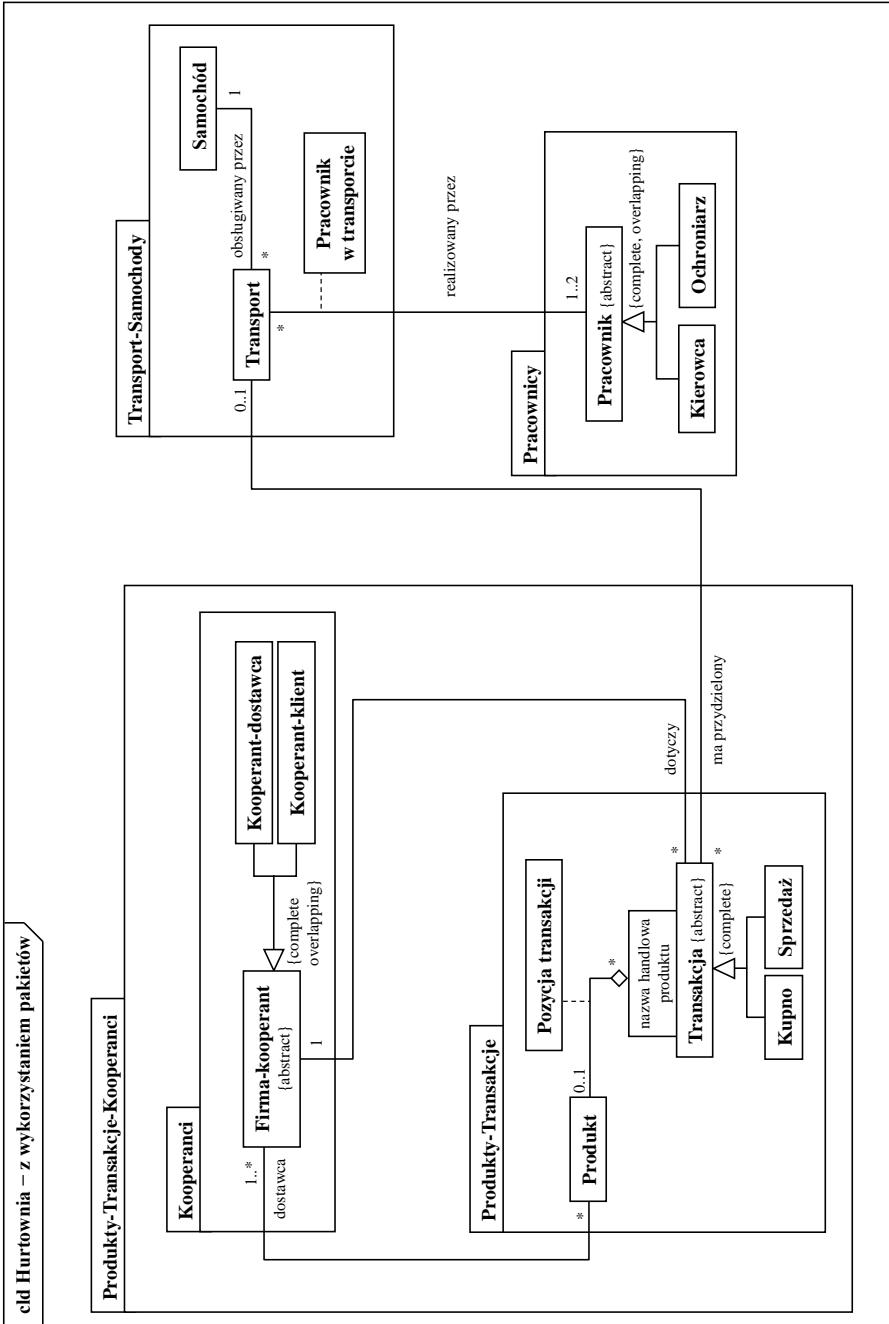
Na podstawie analizy schematu pojęciowego (Rys. 11-34) oraz informacji zawartych w Tab. 11-4 zaproponowano podział struktury systemu (warstwy biznesowej) na następujące pakiety:

- *Pracownicy*,
- *Transport-Samochody*,
- *Produkty-Transakcje-Kooperanci* (w tym *Produkty-Transakcje* oraz *Kooperanci*).

Podział ten przedstawiono na Rys. 11-44 (dla uproszczenia pominięto pola atrybutów i metod oraz komentarze i ograniczenia dotyczące metod/atributów).

W kolejnym kroku każdemu przypadkowi użycia przyporządkowano pakiety w oparciu o dane zawarte w Tab. 11-4 i podział struktury na pakiety (Rys. 11-44). Przypadki wraz z odpowiednimi pakietami zaprezentowano w Tab. 11-5.

W kolumnie *Pakiet klasy startowej* (tabela Tab. 11-5) podano nazwę pakietu, w którym znajduje się określona wcześniej dla każdego z przypadków *klasa startowa*. W kolumnie *Pakiety klas pomocniczych* umieszczone nazwy pakietów, w których znajdują się odpowiednie *klasy pomocnicze*, związane w danym przypadkiem.



Rys. 11-44 Podział struktury systemu na pakiety

Tab. 11-5 Lista przypadków użycia z przypisaniem pakietów

Nazwa przypadku użycia	Nr	Pakiet klasy startowej	Pakiety klas pomocniczych
Anuluj transakcję	7.6	Produkty-Transakcje	Produkty-Transakcje, Kooperanci
Dolacz ochroniarza	9.5	Transport-Samochody	Transport-Samochody, Pracownicy
Dolacz samochód i kierowcę	9.4	Transport-Samochody	
Dolacz transakcję i kooperanta	9.3	Transport-Samochody	Produkty-Transakcje, Kooperanci
Generuj statystyki	10.	Produkty-Transakcje	Produkty-Transakcje
Informuj dział kupno/sprzedaż o osiągnięciu minimalnego stanu posiadania dla produktów	12.1	Produkty-Transakcje	Produkty-Transakcje
Inicjalizuj transakcję kupna produktów	7.2.1	Produkty-Transakcje	Produkty-Transakcje, Kooperanci
Inicjalizuj transakcję kupna/sprzedaży	7.2	Produkty-Transakcje	Produkty-Transakcje, Kooperanci
Inicjalizuj transakcję sprzedaży produktów	7.2.2	Produkty-Transakcje	Produkty-Transakcje, Kooperanci
Obsłuż zdarzenie związanego z upływem czasu	12.	Produkty-Transakcje	Produkty-Transakcje
Obsłuż transakcje oczekujące na akceptację	7.3	Produkty-Transakcje	Produkty-Transakcje
Obsłuż transakcje oczekujące na ustalenie liczby jednostek produktu/produktów możliwych do dostarczenia	7.4	Produkty-Transakcje	Produkty-Transakcje
Organizuj transport do kooperanta-klienta	9.	Transport-Samochody	Produkty-Transakcje, Kooperanci
Podaj opis wybranego produktu	6.1	Produkty-Transakcje	
Podaj, dla zadanego okresu, listę anulowanych transakcji	10.2	Produkty-Transakcje	

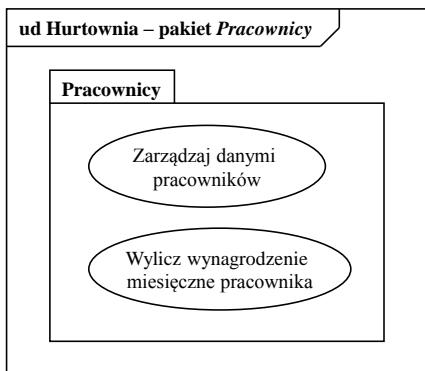
<i>Prezentuj produkty oferowane przez hurtownię</i>	6.	Produkty-Transakcje	
<i>Rejestruj faktyczną datę i godzinę rozpoczęcia transportu</i>	9.1	Transport-Samochody	
<i>Rejestruj faktyczną datę i godzinę zakończenia transportu</i>	9.2	Transport-Samochody	
<i>Rejestruj przyjęcie produktów od kooperanta-dostawcy</i>	8.2	Produkty-Transakcje	Produkty-Transakcje, Kooperanci
<i>Rejestruj wydanie produktów dla transportu</i>	8.1	Produkty-Transakcje	Produkty-Transakcje, Transport-Samochody
<i>Rejestruj wydanie/przyjęcie produktów objętych transakcją</i>	8.	Produkty-Transakcje-Kooperanci	Produkty-Transakcje
<i>Rejestruj zrealizowanie transakcji</i>	7.1	Produkty-Transakcje	
<i>Utwórz, dla zadanego okresu, listę realizowanych transakcji</i>	10.1	Produkty-Transakcje	Produkty-Transakcje-Kooperanci, Transport-Samochody, Pracownicy
<i>Wylicz wynagrodzenie miesięczne pracownika</i>	11.	Pracownicy	Transport-Samochody
<i>Wyszukaj produkty spełniające zadane kryteria</i>	6.2	Produkty-Transakcje	
<i>Wyszukaj transakcje w oparciu o zadane kryteria</i>	7.5	Produkty-Transakcje	Produkty-Transakcje, Kooperanci
<i>Zarządzaj danymi firm-kooperantów (CRUD)</i>	2.	Kooperanci	
<i>Zarządzaj danymi pracowników (CRUD)</i>	1.	Pracownicy	
<i>Zarządzaj danymi produktów (CRUD)</i>	3.	Produkty-Transakcje	
<i>Zarządzaj danymi samochodów (CRUD)</i>	4.	Transport-Samochody	

<i>Zarządzaj transakcjami kupna/sprzedaży</i>	7.	Produkty-Transakcje	Produkty-Transakcje, Kooperanci
<i>Zarządzaj wartościami granicznymi dla transakcji</i>	5.	Produkty-Transakcje	

Na podstawie danych zawartych w Tab. 11-5 zaproponowano rozmieszczenie przypadków użycia w pakietach określonych dla schematu pojęciowego.

Należy zwrócić uwagę na dwa przypadki użycia: *Obsłuż zdarzenie związane z upływem czasu* oraz *Generuj statystyki*. Wspomniane przypadki odwołują się do danych dotyczących transakcji i produktów, dlatego też znalazły się w pakiecie *Transakcje-produkty*. W przypadku, gdyby te usługi wymagały dodatkowo do swojej realizacji innych danych, np. pracowników czy kooperantów, należałoby zaproponować dla nich odrębne pakiety: *Statystyki* i *Obsługa zdarzeń związanych z upływem czasu*.

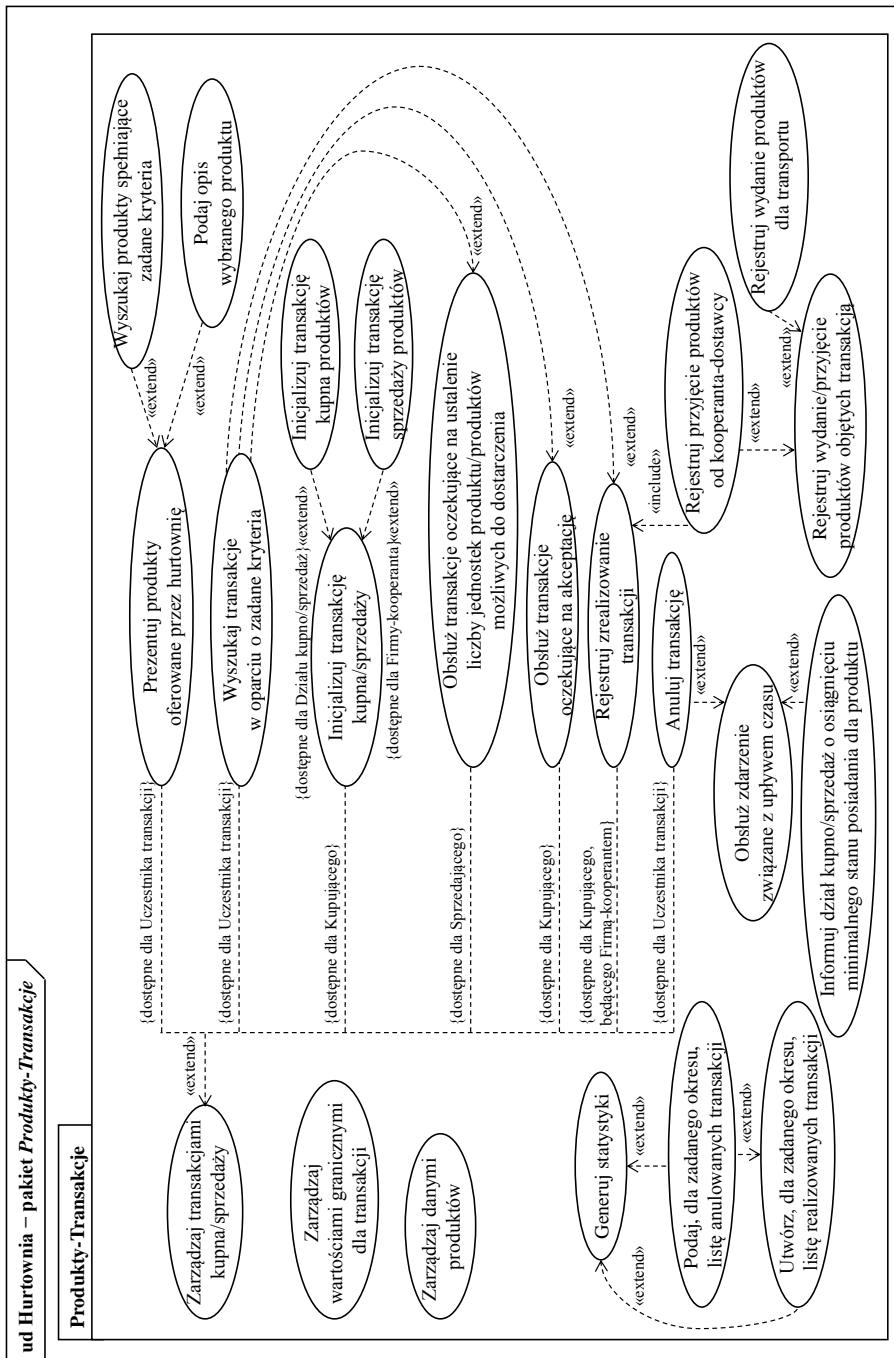
Ze względu na zachowanie przejrzystości diagramów, zawartość pakietów została przedstawiona na odrębnych rysunkach: *Pracownicy* – Rys. 11-45, *Kooperanci* – Rys. 11-46, *Produkty-Transakcje* – Rys. 11-47 oraz *Transport-Samochody* – Rys. 11-48.

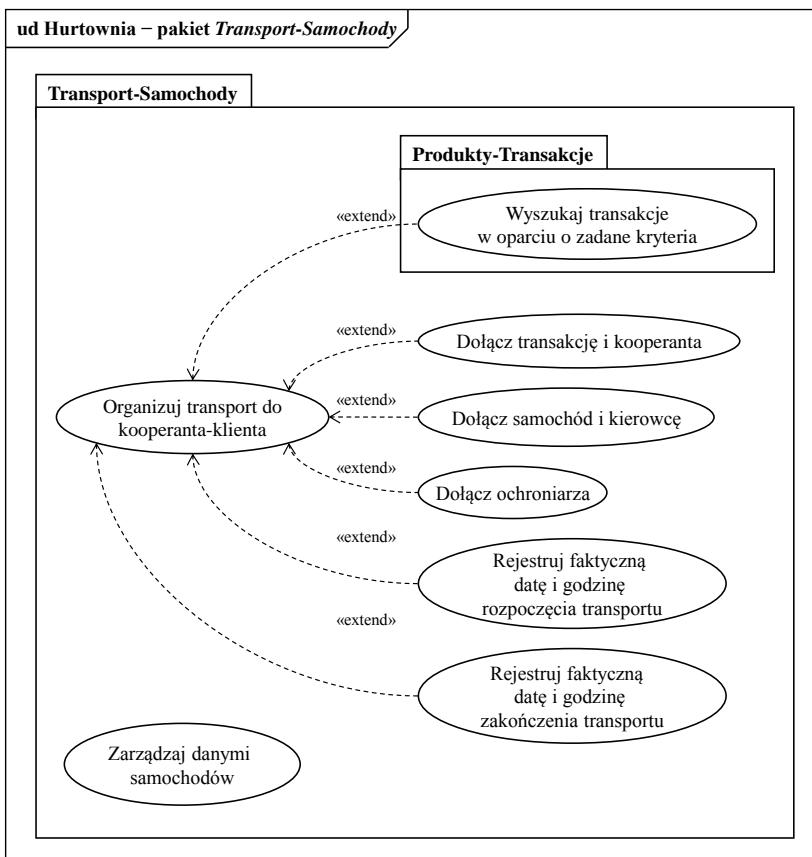


Rys. 11-45 Przypadki użycia umieszczone w pakiecie *Pracownicy*



Rys. 11-46 Przypadki użycia umieszczone w pakiecie *Kooperanci*

Rys. 11-47 Przypadki użycia umieszczone w pakiecie *Produkty-Transakcje*



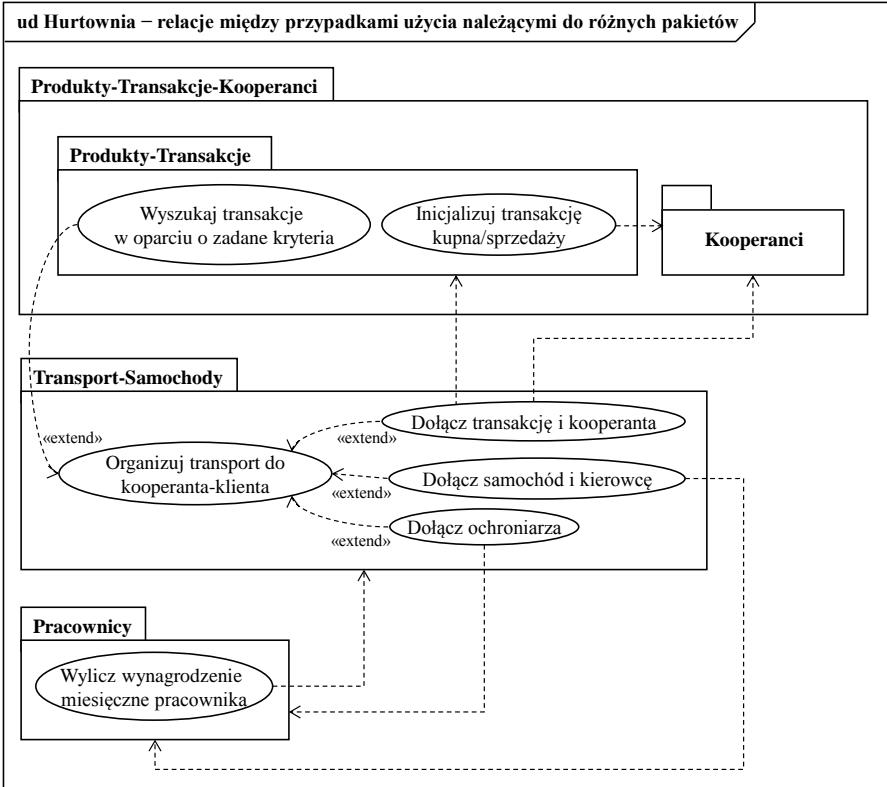
Rys. 11-48 Przypadki użycia umieszczone w pakiecie *Transport-Samochody*

Należy zwrócić uwagę, że usługa *Wyszukaj transakcje w oparciu o zadane kryteria* została zdefiniowana na innym diagramie (pakiet *Produkty-Transakcje* przedstawiony na Rys. 11-47), co uwzględniono na Rys. 11-48.

Dla przypomnienia: Rozważane są przypadki w ich aspekcie wykonawczym, nie dialogowym.

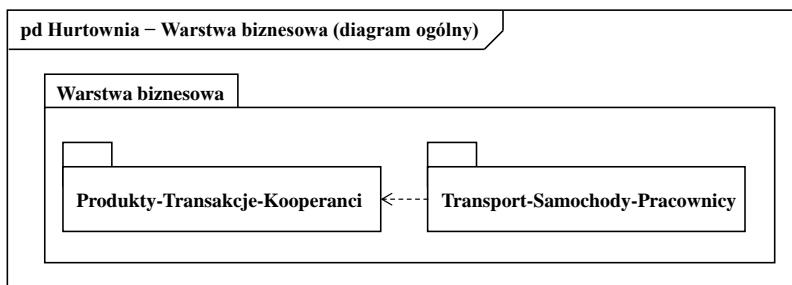
Relacje między przypadkami użycia umieszczonymi w różnych pakietach przedstawiono na Rys. 11-49. W pakietach zaprezentowano wyłącznie te przypadki użycia, które łączą się relacjami z przypadkami z innych pakietów. Pokazano również, w którym pakiecie/pakietach znajdują się dane potrzebne do realizacji wybranego przypadku użycia.

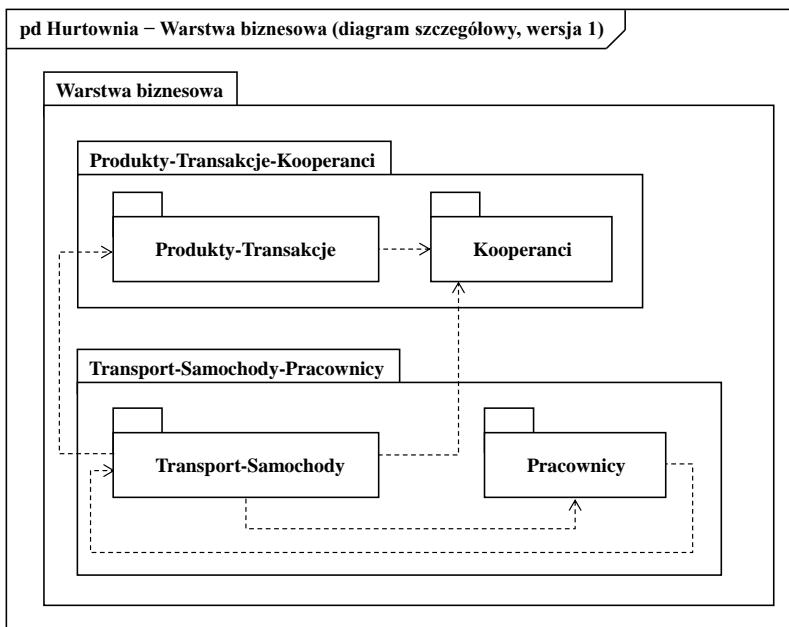
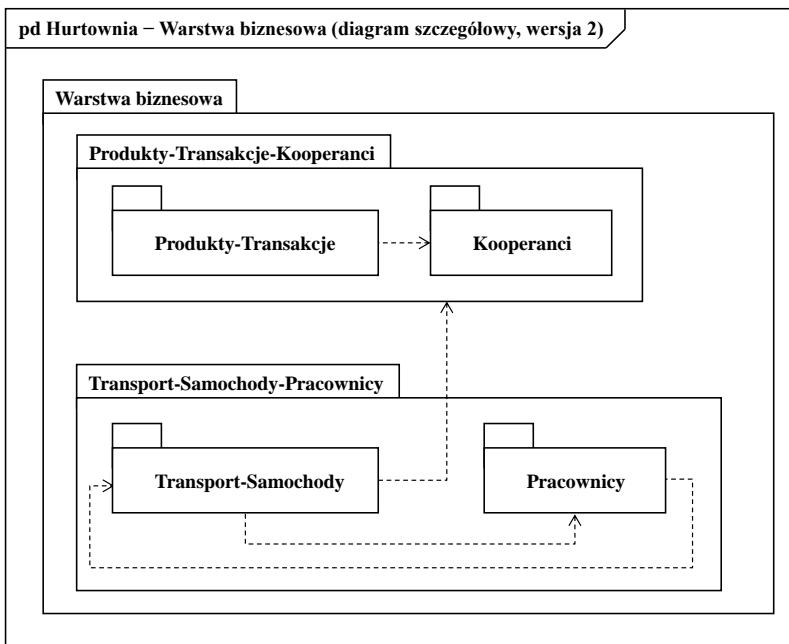
Ze względu na wzajemną zależność pakietów *Pracownicy* i *Transport-Samochody* zaproponowano wprowadzenie nowego pakietu *Pracownicy-Transport-Samochody*, zawierającego ww. pakiety.



Rys. 11-49 Relacje między przypadkami użycia należącymi do różnych pakietów

Warstwa biznesowa dla systemu *Hurtownia* została przedstawiona na Rys. 11-50 – Rys. 11-52. Rys. 11-50 prezentuje diagram ogólny – z dwoma zagnieżdzonymi pakietami *Produkty-Transakcje-Kooperanci* i *Transport-Samochody-Pracownicy*, a Rys. 11-51 i Rys. 11-52 – diagram szczegółowy z pakietami zagnieżdzonymi w pakietach *Produkty-Transakcje-Kooperanci* i *Transport-Samochody-Pracownicy* w dwóch równorzędnych wersjach.

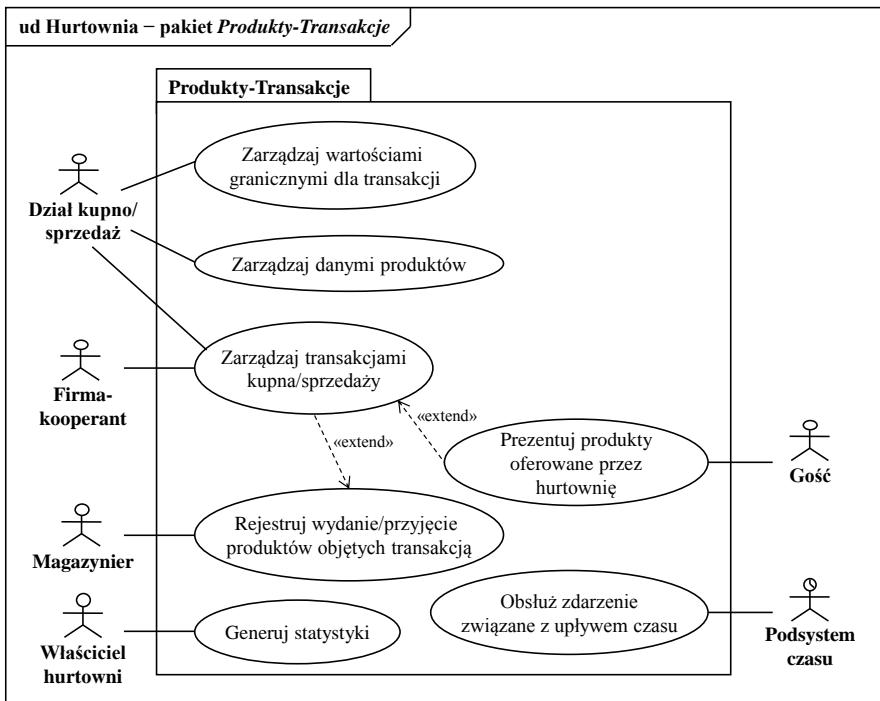
Rys. 11-50 Warstwa biznesowa dla systemu *Hurtownia* – diagram ogólny

Rys. 11-51 Warstwa biznesowa dla systemu *Hurtownia* – diagram szczegółowy (wersja 1)Rys. 11-52 Warstwa biznesowa dla systemu *Hurtownia* – diagram szczegółowy (wersja 2)

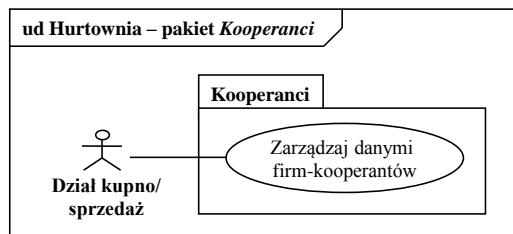
## 11.6 Architektura warstwy biznesowej

**Polecenie:** Zaprojektuj architekturę warstwy biznesowej.

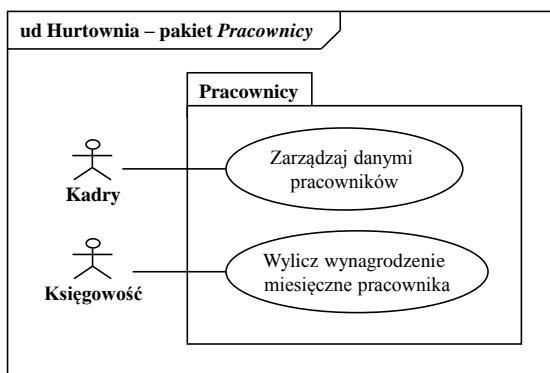
W celu ułatwienia dokonania podziału na komponenty i wyodrębnienia interfejsów [38, 41, 43], co jest jedną z czynności potrzebnych do zaproponowania interfejsu i zaimplementowania prototypu, przypadki użycia zawarte w pakietach przedstawionych na Rys. 11-45 – Rys. 11-48 zaprezentowano wraz z aktorami, którzy je wywołują. Diagramy przedstawione na Rys. 11-53 – Rys. 11-56 prezentują kolejno pakiety: *Produkty-Transakcje*, *Kooperanci*, *Pracownicy* i *Transport-Samochody*.



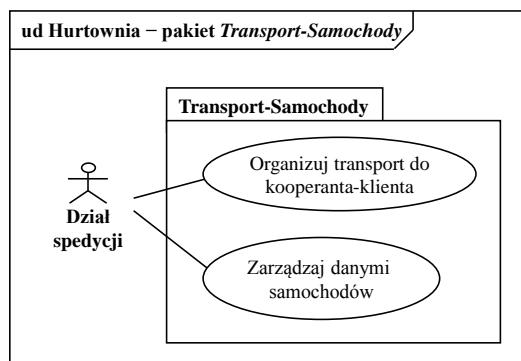
Rys. 11-53 Aktorzy wywołujący przypadki użycia zawarte w pakiecie *Produkty-Transakcje*



Rys. 11-54 Aktor wywołujący przypadek użycia zawarty w pakiecie *Kooperanci*



Rys. 11-55 Aktorzy wywołujący przypadki użycia zawarte w pakiecie *Pracownicy*



Rys. 11-56 Aktor wywołujący przypadki użycia zawarte w pakiecie *Transport-Samochody*

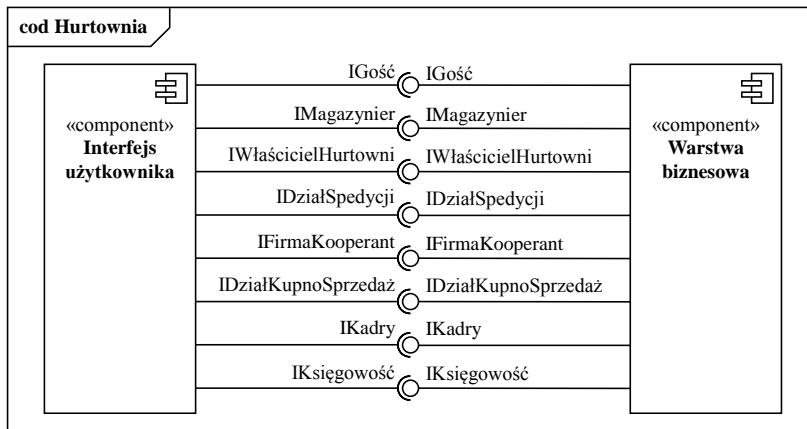
W rozważaniach należy uwzględnić hierarchię aktorów (dziedziczenie po aktorze *Gość*) zidentyfikowaną na etapie analizy funkcjonalnej.

### Diagramy komponentów Warstwy biznesowej

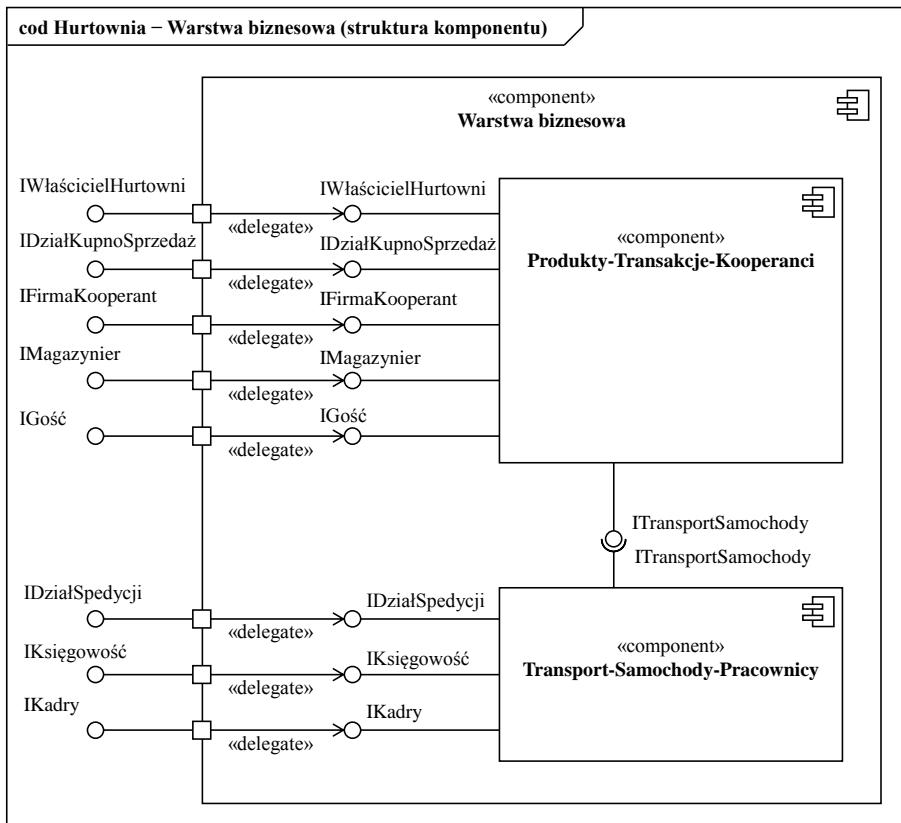
*Uwaga:* Tak jak poprzednio, komponenty zostaną wyspecyfikowane w postaci czarnej skrzynki.

Pierwszy z diagramów (Rys. 11-57) przedstawia architekturę systemu *Hurtownia*, uwzględniającą wyłącznie dwa komponenty: *Interfejs użytkownika* (zdefiniowany i omówiony w 11.2.6.) i *Warstwa biznesowa*.

Drugi diagram ( Rys. 11-58) prezentuje strukturę komponentu *Warstwa biznesowa* z wyspecyfikowanymi interfejsami dwóch zagnieżdżonych komponentów: *Produkty-Transakcje-Kooperanci* oraz *Transport-Samochody-Pracownicy*.



Rys. 11-57 Architektura systemu *Hurtownia* uwzględniająca wyłącznie dwa komponenty: *Interfejs użytkownika* i *Warstwę biznesową*



Rys. 11-58 Struktura komponentu *Warstwa biznesowa*

**Komentarz:** Przykład przedstawiony na Rys. 11-58 pokazuje regułę, zgodnie z którą należałoby postępować, określając strukturę komponentów złożonych, takich jak np. *Produkty-Transakcje-Kooperanci* czy *Transport-Samochody-Pracownicy*.

## 11.7 Wstępny, zgrubny plan iteracji realizujących założoną funkcjonalność

**Polecenie:** Zaproponuj wstępny, zgrubny plan iteracji realizujący założoną funkcjonalność.

Bazując na analizie rozmieszczenia przypadków użycia w pakietach oraz na ustalonych wcześniej ich priorytetach i złożoności, można przedstawić wstępny, zgrubny plan iteracji [39, 49] (Tab. 11-6). W kolejnych iteracjach opisano ogólnie cele określone dla każdej nich, bez posługiwania się nazwami zidentyfikowanych (wyróżnionych) przypadków użycia.

Przedstawiony wstępny plan iteracji powinien zostać zweryfikowany po przeprowadzeniu szczegółowej analizy funkcjonalnej, czyli po utworzeniu scenariuszy dla wszystkich przypadków, łącznie z ich podziałem na przepływy główne i przepływy alternatywne oraz podziału przypadków na podprzypadki. Na zmianę planu może mieć również wpływ przeprowadzenie analizy dynamicznej dla wszystkich przypadków.

**Uwaga:** Plan iteracji wspomaga planowanie prac zespołu projektowego: przydziela zadań do grup osób, kontrolę realizacji i monitorowanie postępu prac. Plan ten powinien być tak ułożony, aby w danej iteracji żaden z zespołów realizujących wybrane scenariusze/przypadki użycia/grupy przypadków użycia nie musiał czekać na zakończenie prac wykonywanych przez inny zespół lub, co gorsze, aby prace wykorzystujące wzajemne swoje artefakty były realizowane w tym samym czasie.

W początkowych iteracjach należy realizować przypadki użycia o najwyższym priorytecie i największej złożoności oraz te z przypadków użycia CRUD, które pozwalają na zautomatyzowanie wprowadzania do systemu danych niezbędnych do prawidłowej implementacji i testowania. W kolejnych iteracjach należy zwrócić uwagę na relacje pomiędzy przypadkami – we wcześniejszych iteracjach powinny znaleźć się te przypadki, które będą wykorzystywane przez przypadki zaplanowane w późniejszych iteracjach.

Tab. 11-6 Wstępny, zgrubny plan iteracji

Numer iteracji	Realizowane cele
1.	Realizacja dodawania (tworzenia) obiektów z klas <i>Produkt</i> , <i>Firma-kooperant</i> (funkcje typu CRUD), implementacja wyszukiwania

	produktów według zadanych kryteriów, implementacja mechanizmu aktorów (pozwalającego na logowanie do systemu i testowanie jego funkcjonalności z punktu widzenia różnych użytkowników).
2.	Zapewnienie obsługi transakcji typu kupno, co pozwoli na wprowadzenie danych zakupionych produktów oraz zapewnienie obsługi całego procesu realizacji transakcji z perspektywy kupowania produktów przez hurtownię.  Ze względu na fakt, że przy kupnie produktów kooperant-dostawca dostarcza je do hurtowni własnym transportem, a organizacja transportu w hurtowni dotyczy wyłącznie sprzedaży produktów kooperantowi-klientowi, obsługa transportu (samochodów, kierowców, ochroniarzy) została przewidziana dopiero w następnej iteracji.
3.	Realizacja wprowadzania do systemu informacji o samochodach, kierowcach, ochroniarzach (tzn. tworzenia obiektów odpowiednich klas; funkcje typu CRUD). Zapewnienie obsługi transakcji typu sprzedaż (co wiąże się z usługą transportu).  W tej iteracji można nie uwzględniać realizacji następujących usług: potwierdzenia przez sprzedającego liczby jednostek produktu możliwych do dostarczenia, akceptacji transakcji przez kupującego oraz anulowania transakcji.
4.	Zapewnienie obsługi funkcjonalności pominiętej w poprzedniej iteracji, dotyczącej następujących sytuacji: (1) ilość jednostek produktu/produktów możliwych do dostarczenia jest mniejsza niż zamówiona, (2) anulowanie transakcji – w tym brak akceptacji transakcji przez kupującego oraz obsługa zdarzenia związanego z upływem ostatecznego terminu realizacji transakcji przez sprzedającego.  Obsługa pozostałych zdarzeń związanych z upływem czasu została zaplanowana w iteracji 6.
5.	Realizacja wszystkich statystyk.
6.	Obsługa pozostałych zdarzeń związanych z upływem czasu.
7.	Uwzględnienie tych alternatywnych przepływow zdarzeń, które nie zostały uwzględnione w dotychczasowych iteracjach.

## 11.8 Podsumowanie zadania

W Tab. 11-7 przedstawiono ogólną charakterystykę omawianego zadania. Szczegółową charakterystykę zadania dla grup tematycznych: *Wprowadzenie do przedmiotu*, *Analiza funkcjonalna*, *Analiza strukturalna*, *Analiza dynamiczna* zamieszczono odpowiednio w tabelach Tab. 11-8 – Tab. 11-11.

Tab. 11-7 Ogólna charakterystyka zadania

<b>Ogólna charakterystyka zadania</b>	
<i>Nazwa</i>	Hurtownia
<i>Dziedzina problemowa</i>	system może zostać wykorzystany dla obsługi hurtowni
<i>Cel</i>	wsparcie obsługi transakcji typu kupno (od dostawcy) i sprzedaż (dla klienta) produktów, których obrotem zajmuje się hurtownia
<i>Zakres odpowiedzialności</i>	ewidencja produktów, kooperantów, pracowników, zarządzanie transakcjami i organizacja transportu produktów
<i>Stopień trudności</i>	trudne
<i>Rozmiar</i>	12 klas
<i>Użyteczność w grupach tematycznych</i>	
<i>Grupa tematyczna</i>	<i>Użyteczność w grupie tematycznej</i> <i>Użyteczność na etapie zgodnym z modelem efektywnego nauczania</i>
<i>Wprowadzenie do przedmiotu</i>	niska  <i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> niska <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> niska
<i>Analiza funkcjonalna</i>	wysoka  <i>Przygotowanie do nauki:</i> wysoka <i>Nauka:</i> wysoka <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> niska
<i>Analiza strukturalna</i>	wysoka  <i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> wysoka <i>Podsumowanie dotychczasowych działań:</i> niska

		<i>Ewaluacja:</i> średnia
<i>Analiza dynamiczna</i>	niska	<i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> niska <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> niska

Tab. 11-8 Szczegółowa charakterystyka zadania dla grupy tematycznej *Wprowadzenie do przedmiotu*

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>	<i>Wprowadzenie do przedmiotu</i>	
<b>Stopień trudności grupy w zadaniu</b>	łatwe	
<b>Poruszane tematy</b>	<i>mechanizmy rozszerzalności, dekompozycja diagramów, reguły nazewnictwa</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>mechanizmy rozszerzalności</i>	łatwe	<i>stereotyp, ograniczenie, komentarz</i>
<i>dekompozycja diagramów</i>	łatwe	<i>dekompozycja diagramów z wykorzystaniem diagramów pakietów, podziału modelu na diagramy, zagnieżdżanie</i>
<i>reguły nazewnictwa</i>	łatwe	<i>reguły nazewnictwa przypadków użycia i klas</i>

Tab. 11-9 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza funkcjonalna*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Analiza funkcjonalna</i>

<b>Stopień trudności grupy w zadaniu</b>	trudne	
<b>Poruszane tematy</b>	<i>wymagania na system, aktorzy, przypadki użycia, scenariusze, dekompozycja modelu/diagramu przypadków użycia, plan iteracji, diagram komponentów, mechanizmy rozszerzalności</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>wymagania na system</i>	średnio trudne	
<i>aktorzy</i>	trudne	<i>aktor abstrakcyjny, aktor Podsystem czasu, hierarchia dla aktorów – dziedziczenie wieloaspektowe, stosunkowo duża liczba aktorów</i>
<i>przypadki użycia</i>	trudne	<i>przypadki użycia typu CRUD, hierarchia przypadków użycia, stosunkowo duża liczba przypadków użycia oraz relacji pomiędzy przypadkami użycia, charakterystyka przypadków użycia – priorytety, złożoność (trudność implementacji)</i>
<i>scenariusze</i>	łatwe	
<i>dekompozycja modelu/diagramu przypadków użycia</i>	średnio trudne	<i>dekompozycja pozioma, dekompozycja pionowa</i> <i>podział na pakiety (interfejs użytkownika)</i>
<i>plan iteracji</i>	średnio trudne	<i>wstępny, zgrubny plan iteracji realizujących założoną funkcjonalność</i>
<i>diagram komponentów</i>	średnio trudne	<i>diagramy komponentów – struktura komponentu, komponenty zagnieżdzone, interfejs użytkownika</i>

<i>mechanizmy rozszerzalności</i>	łatwe	<i>ograniczenie, komentarz ograniczenia nałożone na relacje między przypadkami użycia</i>
-----------------------------------	-------	---

Tab. 11-10 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza strukturalna*

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>	<i>Analiza strukturalna</i>	
<b>Stopień trudności grupy w zadaniu</b>	trudne	
<b>Poruszane tematy</b>	<i>atrybuty klasy, metody klasy, struktury generalizacji-specjalizacji, klasa abstrakcyjna a klasa konkretna, asocjacje, analiza wartości, przejście na schemat relacyjny, dekompozycja modelu/diagramu klas, diagram komponentów, mechanizmy rozszerzalności</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>atrybuty klasy</i>	łatwe	<i>atrybut opcjonalny, powtarzalny, pochodny, klasowy</i>
<i>metody klasy</i>	łatwe	<i>metoda zaimplementowana, obiektowa, klasowa, polimorfizm metod, przesłanianie metod</i>
<i>struktury generalizacji-specjalizacji</i>	łatwe	<i>rodzaje dziedziczenia – nieroziączne</i>
<i>klasa abstrakcyjna a klasa konkretna</i>	łatwe	<i>klasa abstrakcyjna, klasa konkretna, kilka klas abstrakcyjnych</i>
<i>asocjacje</i>	łatwe	<i>nazwa asocjacji a rola asocjacji, agregacja, asocjacja kwalifikowana</i>
<i>analiza wartości</i>	łatwe	<i>wartość pochodna, wartość początkowa, wartość graniczna</i>

<i>przejście na model relacyjny</i>	łatwe	obejście dziedziczenia nierożłącznego
<i>dekompozycja modelu/diagramu klas</i>	trudne	podział na pakiety, zależności między pakietami, pakiety zagnieździone
<i>diagram komponentów</i>	trudne	<i>diagramy komponentów</i> – struktura komponentu, interfejsy dla komponentów zagnieźdzonych architektura systemu zawierająca dwa główne komponenty: interfejs użytkownika i warstwę biznesową
<i>mechanizmy rozszerzalności</i>	łatwe	<i>ograniczenie, komentarz</i>

Tab. 11-11 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza dynamiczna*

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>	<i>Analiza dynamiczna</i>	
<b>Stopień trudności grupy w zadaniu</b>	łatwe	
<b>Poruszane tematy</b>	<i>diagramy stanów</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>diagramy stanów</i>	łatwe	<i>maszyna stanowa behawioralna ze stanem złożonym</i>

## Spis treści rozdziału

Rozdział 12 Klub Judo .....	526
12.1 Wstępny opis projektu .....	526
12.1.1 Dziedzina problemowa .....	526
12.1.2 Cel .....	526
12.1.3 Zakres odpowiedzialności systemu .....	526
12.1.4 Użytkownicy systemu.....	527
12.2 Wymagania użytkownika .....	527
12.3 Wymagania na system .....	531
12.3.1 Wymagania funkcjonalne na system .....	531
12.3.2 Opis struktury systemu (schemat pojęciowy) .....	535
12.3.3 Wymagania niefunkcjonalne – miary .....	539
12.4 Opis przyszłej ewolucji systemu .....	542
12.5 Słownik pojęć z dziedziny problemowej .....	542
12.6 Podsumowanie zadania.....	551

## Rozdział 12

# Klub Judo

*Ewa Stemposz, Alina Stasiecka, Andrzej Jodłowski*

### 12.1 Wstępny opis projektu

#### 12.1.1 Dziedzina problemowa

Projektowany system został zamówiony przez prezesa niewielkiego klubu judo, nie mniej jednak może zostać wykorzystany jako narzędzie wspierające zarządzanie niewielkimi klubami związanymi z innymi sportami walki.

#### 12.1.2 Cel

System ma ułatwić dostęp do danych dotyczących m.in. trenerów i zawodników, terminarza treningów oraz turniejów, czy też do informacji o osiągnięciach najlepszych zawodników, dzięki czemu zarząd chce wypromować klub i zachęcić młodzież do trenowania judo.

#### 12.1.3 Zakres odpowiedzialności systemu

System powinien umożliwiać zarządzanie informacjami dotyczącymi:

- członków klubu (przede wszystkim zawodników i trenerów), jak również osób związanych z klubem, ale niebędących jego członkami;
- treningów;
- oraz turniejów judo.

Należy zapewnić właściwy dostęp do przechowywanych danych oraz wykonanych zestawień statystycznych, w tym także do zestawień (statystyk) dotyczących byłych członków klubu.

W systemie nie mają być przechowywane bardziej szczegółowe dane takie jak np. lista wszystkich zawodników biorących udział w danym turnieju (obejmująca także zawodników spoza klubu), czy lista wszystkich walk turnieju. Są to informacje zbędne z punktu widzenia potrzeb potencjalnych użytkowników projektowanego systemu.

Zarządzanie finansami klubu ma być wspierane przez odrębny system i dlatego nie wchodzi w zakres projektowanego systemu.

### 12.1.4 Użytkownicy systemu

Potencjalnymi użytkownikami systemu będą członkowie klubu (w tym zarząd, trenerzy i zawodnicy klubu), użytkownicy niezarejestrowani (goście) oraz podsystem czasu.

## 12.2 Wymagania użytkownika

### *Wymagania wstępne*

1. W systemie należy przechowywać informacje o osobach związanych z klubem. Zostały one podzielone ze względu na pełioną funkcję, m.in. na trenerów i zawodników (zdarza się, że trener może być równocześnie zawodnikiem) oraz ze względu na przynależność klubową – na członków klubu oraz osoby spoza klubu. Oba podziały są kompletne.
2. Dla każdej osoby należy przechowywać imię (co najmniej jedno), nazwisko oraz płeć (opcjonalnie – tylko dla trenerów i zawodników).
3. Członkowie klubu powinni być opisani przez datę urodzenia, PESEL, dane kontaktowe oraz obie daty: wstąpienia i wystąpienia z klubu. Dla osób spoza klubu mogą być pamiętane dane tego klubu, do którego należą.
4. System powinien umożliwić przechowywanie informacji o stopniach judo zdobytych przez członka klubu, o ile trenuje/trenował judo, takie jak: kto i kiedy zdobył określony stopień. Zdobycie stopnia judo ma być potwierdzone przez dołączenie informacji o unikatowym numerze odpowiedniego certyfikatu.
5. Dla stopnia judo należy pamiętać: nazwę potoczną, nazwę pełną (nazwy te są niepowtarzalne), rodzaj stopnia („szkolny”, „mistrzowski”), kolor pasa („biały”, „żółty”, „pomarańczowy”, „zielony”, „niebieski”, „brązowy”, „czarny”, „biało-czerwony”, „czerwony”). Ma być znana lista kryteriów (z co najmniej jedną pozycją), które muszą być spełnione dla uzyskania każdego ze stopni.
6. Trener zatrudniony w klubie staje się w ten sposób jednocześnie jego członkiem. Dla trenera klubu należy pamiętać datę zatrudnienia i datę zwolnienia z klubu, staż pracy w klubie oraz o ile nie jest to jego pierwsza praca – staż pracy w zawodzie trenera przed zatrudnieniem w klubie.
7. Trenerzy mogą posiadać licencje na wykonywanie zawodu. Każda taka licencja jest opisana za pomocą klasy trenerskiej („trener II klasy”, „trener I klasy”, „trener klasy mistrzowskiej”) oraz listy wymagań (z co najmniej jedną pozycją), które należy spełnić w celu jej uzyskania. Klasa trenerska określa stopień kwalifikacji zawodowej, przy czym najniższy stopień to „trener II klasy”. Jeśli trener klubu posiada jedną lub więcej licencji na wykonywanie

zawodu, to należy pamiętać, kiedy każdą z nich uzyskał. Numer licencji jest unikatowy.

8. Dla zawodnika klubu należy przechowywać informacje takie jak: wiek, dane kontaktowe rodzica (tylko dla osoby niepełnoletniej) oraz aktualne (tj. ostatnie) przydzielony zawodnika do kategorii wiekowej i wagowej.
9. Kategoria wiekowa może być definiowana na różne sposoby: (1) poprzez określenie roku urodzenia (tzw. *rocznik*) najmłodszego oraz najstarszego zawodnika, którzy mogą być do niej przypisani, (2) za pomocą tylko jednego z roczników: najmłodszego lub najstarszego. Co najmniej jeden z roczników musi zostać określony. Kategoria wiekowa niekiedy jest dodatkowo opisywana za pomocą unikatowej nazwy, np. *młodzicy*.
10. Kategoria wagowa z kolei jest definiowana za pomocą: (1) dwóch limitów wagowych określających najniższą i najwyższą dopuszczalną wagę zawodnika, który może zostać do niej przydzielony, (2) tylko jednej z wartości granicznych – najniższej albo najwyższej dopuszczalnej wagi zawodnika. Analogicznie jak w przypadku kategorii wiekowej, co najmniej jedna z ww. wartości musi zostać określona. Kategoria wagowa może dodatkowo posiadać unikatową nazwę.
11. W systemie mają też być przechowywane informacje o klubowych grupach treningowych, złożonych z trenerów i zawodników będących członkami klubu. Każda z nich powinna mieć nadany niepowtarzalny identyfikator. Ma być określony jej poziom zaawansowania („początkująca”, „średniozaawansowana”, „zaawansowana”).
12. Grupę treningową może prowadzić jednocześnie kilku trenerów (co najmniej 1). Ponieważ trener może być również zawodnikiem, należy pamiętać rolę, jaką pełni w danej grupie (zawodnika czy też trenera). Grupa nie może liczyć mniej niż 8 i więcej niż 24 osób w roli zawodników (w danym momencie). Wartości tych limitów mogą w przyszłości ulegać zmianie.
13. Należy przechowywać informacje o okresach przynależności danej osoby do grup treningowych, tj. daty początku i końca przynależności wraz z rolami, jakie pełniła. Zawodnik/trener może w kilku (różnych) okresach należeć do tej samej grupy. W danym momencie zawodnik, w przeciwieństwie do trenera, może być przydzielony co najwyższej do jednej grupy.
14. Grupa treningowa uczestniczy zwykle w kilku treningach tygodniowo, przy czym w danym dniu tygodnia może nie uczestniczyć w żadnym treningu lub uczestniczyć w więcej niż jednym. Dla każdego treningu system powinien przechowywać informacje takie jak: dzień tygodnia, godziny rozpoczęcia i zakończenia, miejsce („mata”, „siłownia”, „basen”) oraz grupa/grupy biorące w nim udział. Grafik treningów ustalany jest na 7 dni w tygodniu i powtarza się co tydzień. np. przez cały rok szkolny. Lista obecności zawodników na treningu oraz informacja o tym, czy dany trening się odbył, nie są istotne i dlatego nie mają być pamiętane.

15. System ma też przechowywać informacje o turniejach judo, takie jak: nazwa turnieju, jego ranga, lista organizatorów (z co najmniej jedną pozycją), termin, miejscowości i ewentualnie adres WWW strony turnieju. Ma być wiadomo, dla jakich kategorii wiekowych i wagowych oraz dla osób jakiej płci został zorganizowany. Organizatorzy turnieju muszą określić co najmniej jedną kategorię wiekową i jedną wagową.
16. Niektóre z turniejów są brane pod uwagę podczas tworzenia klubowego rankingu zawodników i dla nich należy pamiętać punktację pozwalającą na ustalenie, ile punktów rankingowych uzyskał zawodnik klubu po zdobyciu określonego miejsca w danym turnieju.
17. Ma być rejestrowany udział zawodnika klubu w turnieju, w tym kategorie (wiekowa i wagowa), w których startował, listy walk, które stoczył, a także miejsce, które zajął (przy czym brane są pod uwagę wyłącznie miejsca o numerze od 1 do 7). Dla każdej stoczonej walki zawodnika klubu należy przechowywać informacje o tym, czy ją wygrał, ile punktów zdobył oraz można też dodać komentarz. W każdej walce uczestniczy dwóch judoków i jeżeli istotne są dane przeciwnika zawodnika klubu, informacja taka ma być dodatkowo pamiętana.

System ma wspierać potencjalnych użytkowników m.in. w realizowaniu zadań, których listę przedstawiono poniżej. Została ona uporządkowana alfabetycznie według nazw funkcji, a w nawiasach podano byty uprawnione do ich wywołania.

- Aktualizowanie roczników dla wszystkich kategorii wiekowych (automatycznie 1 stycznia);
- Przydzielanie trenera do grupy treningowej oraz usuwanie go z niej (zarząd klubu);
- Przydzielanie zawodnika klubu do grupy treningowej oraz usuwanie go z niej (trener);
- Rejestrowanie udziału zawodnika klubu w turnieju judo (trener);
- Sporządzenie oraz zmiana grafiku treningów (zarząd klubu);
- Wprowadzenie informacji o zdobyciu stopnia judo przez członka klubu (zarząd klubu);
- Wprowadzenie informacji o licencji trenera (zarząd klubu);
- Wprowadzenie nowego turnieju (zarząd klubu);
- Wyświetlenie grafiku treningów – dla wszystkich grup treningowych (gość i każdy zarejestrowany użytkownik);
- Wyświetlenie grafiku treningów dla grupy treningowej (zawodnik klubu, trener);
- Wyświetlenie informacji o stopniu judo o zadanej nazwie potocznej (zawodnik klubu, trener);
- Wyświetlenie informacji o zawodniku klubu (zawodnik klubu, trener);
- Wyświetlenie listy aktualnie zatrudnionych trenerów (gość i każdy zarejestrowany użytkownik);

- Wyświetlenie listy aktualnych członków klubu (każdy członek klubu);
- Wyświetlenie listy turniejów rozgrywanych w zadanym okresie (gość i każdy zarejestrowany użytkownik);
- Wyświetlenie listy zawodników klubu o podanej płci i dla zadanej kategorii wiekowej (trener);
- Wyświetlenie rankingu zawodników klubu w zadanym okresie (gość i każdy zarejestrowany użytkownik);
- Wyświetlenie składu grupy treningowej (zawodnik klubu, trener);
- Zarządzanie osobami, grupami treningowymi, kategoriami wiekowymi, kategoriami wagowymi, licencjami, stopniami judo (zarząd klubu);
- Zmiana aktualnej kategorii wagowej zawodnika (trener);
- Znalezienie wolnych miejsc w grupach treningowych o zadanym poziomie zaawansowania (trener).

*Dla przypomnienia:* przypadki *Zarządzaj danymi* są rozumiane jako funkcje typu CRUD (ang. *Create, Read, Update, Delete*): utwórz, wyświetl, modyfikuj, usuń dane potrzebne do wypełnienia bazy danych.

#### *Wymagania niefunkcjonalne*

*Dla przypomnienia:* Wymagania niefunkcjonalne opisują ograniczenia, przy których system powinien realizować swoje funkcje, np. wymagania dotyczące koniecznych zasobów, ograniczeń czasowych, niezawodności, bezpieczeństwa, przenośności, współpracy z określonymi narzędziami i środowiskami, zgodności z normami i standardami, przepisami prawnymi itp. [38].

Poniżej zamieszczono listę przykładowych wymagań niefunkcjonalnych, które można wziąć pod uwagę, rozważając ograniczenia, przy których system *Klub judo* powinien pracować:

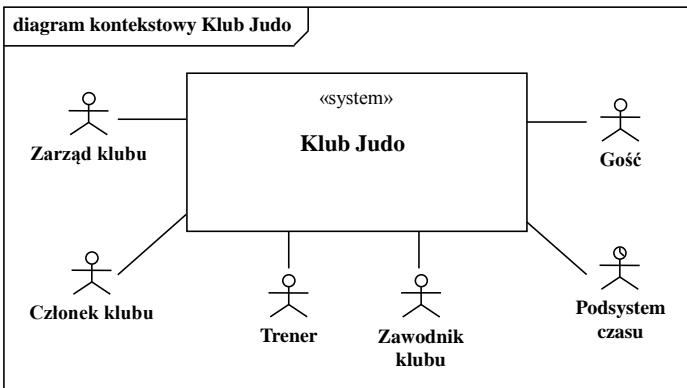
- dostępność;
- intuicyjność, czytelność, ergonomiczność interfejsu użytkownika;
- łatwość użytkowania;
- możliwość równoczesnej pracy wielu użytkowników;
- niezawodność;
- poprawna obsługa funkcji systemu poprzez różne urządzenia zewnętrzne;
- poprawne działanie na typowym współczesnym komputerze osobistym;
- prawidłowa współpraca z powszechnie stosowanymi drukarkami;
- przechowywanie danych w bazie danych;
- spełnienie przez system typowych wymagań czasowych/wydajnościowych;
- szybka reakcja systemu na akcje użytkownika;
- wygląd interfejsu niezależny od rodzaju przeglądarki internetowej;
- zabezpieczenie poprawności danych (walidacja danych);
- zapewnienie bezpieczeństwa danych.

Miary pozwalające na weryfikację wymienionych wymagań niefunkcjonalnych przedstawiono w Tab. 12-1, w pkt 11.3.3.

## 12.3 Wymagania na system

### 12.3.1 Wymagania funkcjonalne na system

Potencjalni użytkownicy systemu zostali przedstawieni na diagramie kontekstowym na Rys. 12-1.



Rys. 12-1 Diagram kontekstowy

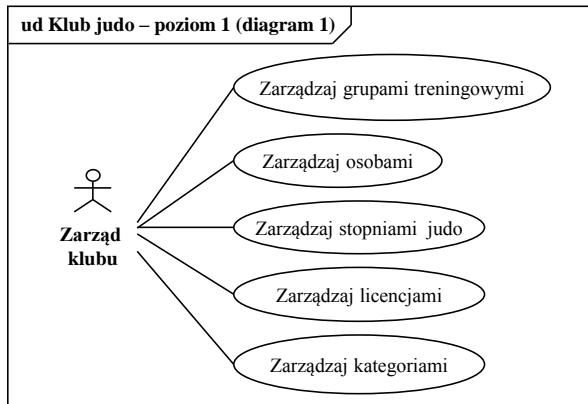
*Dla przypomnienia:* Aktor *Podsystem czasu* symbolizuje wszystkich aktorów związanych ze zdarzeniami wywoływanymi przez upływ czasu, takich jak np.: *1 stycznia*.

W celu ułatwienia skonstruowania modelu przypadków użycia [36, 41, 45], funkcjonalność projektowanego systemu, wymienioną w pkt 11.2, uporządkowano alfabetycznie według nazw aktorów określonych na diagramie kontekstowym. Dla aktora *Podsystem czasu* określono nazwę *1 stycznia*. Każdemu z aktorów przyporządkowano funkcje, do których wywołania powinien być uprawniony.

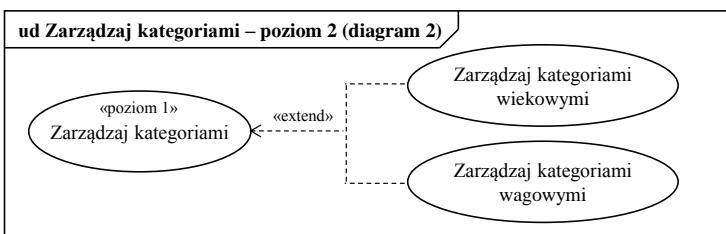
- 1 stycznia (Podsystem czasu):
  - Aktualizowanie roczników dla wszystkich kategorii wiekowych;
- Członek klubu:
  - Wyświetlenie grafiku treningów (dla wszystkich grup treningowych);
  - Wyświetlenie listy aktualnie zatrudnionych trenerów;
  - Wyświetlenie listy aktualnych członków klubu;
  - Wyświetlenie listy turniejów rozgrywanych w zadanym okresie;
  - Wyświetlenie rankingu zawodników klubu w zadanym okresie;
- Gość – użytkownik niezarejestrowany:
  - Wyświetlenie grafiku treningów;
  - Wyświetlenie listy aktualnie zatrudnionych trenerów;
  - Wyświetlenie listy turniejów rozgrywanych w zadanym okresie;

- Wyświetlenie rankingu zawodników klubu w zadanym okresie;
- Trener:
  - Przydzielanie zawodnika klubu do grupy treningowej oraz usuwanie go z niej;
  - Rejestrowanie udziału zawodnika klubu w turnieju judo;
  - Wyświetlenie grafiku treningów dla grupy treningowej;
  - Wyświetlenie grafiku treningów (dla wszystkich grup treningowych);
  - Wyświetlenie informacji o stopniu judo o zadanej nazwie potocznej;
  - Wyświetlenie informacji o zawodniku klubu;
  - Wyświetlenie listy aktualnie zatrudnionych trenerów;
  - Wyświetlenie listy aktualnych członków klubu;
  - Wyświetlenie listy turniejów rozgrywanych w zadanym okresie;
  - Wyświetlenie listy zawodników klubu o podanej płci, dla zadanej kategorii wiekowej;
  - Wyświetlenie rankingu zawodników klubu w zadanym okresie;
  - Wyświetlenie składu grupy treningowej;
  - Zmiana aktualnej kategorii wagowej zawodnika klubu;
  - Znalezienie wolnych miejsc w grupach treningowych o zadanym poziomie zaawansowania;
- Zarząd klubu:
  - Przydzielanie trenera do grupy treningowej oraz usuwanie go z niej;
  - Sporządzenie oraz zmiana grafiku treningów;
  - Wprowadzenie informacji o zdobyciu stopnia judo przez członka klubu;
  - Wprowadzenie informacji o licencji trenera;
  - Wprowadzenie nowego turnieju;
  - Wyświetlenie grafiku treningów (dla wszystkich grup treningowych);
  - Wyświetlenie listy aktualnie zatrudnionych trenerów;
  - Wyświetlenie listy aktualnych członków klubu;
  - Wyświetlenie listy turniejów rozgrywanych w zadanym okresie;
  - Wyświetlenie rankingu zawodników klubu w zadanym okresie;
  - Zarządzanie osobami, grupami treningowymi, kategoriami wiekowymi, kategoriami wagowymi, licencjami, stopniami judo;
- Zawodnik klubu:
  - Wyświetlenie grafiku treningów dla grupy treningowej;
  - Wyświetlenie grafiku treningów (dla wszystkich grup treningowych);
  - Wyświetlenie informacji o stopniu judo o zadanej nazwie potocznej;
  - Wyświetlenie informacji o zawodniku klubu;
  - Wyświetlenie listy aktualnie zatrudnionych trenerów;
  - Wyświetlenie listy aktualnych członków klubu;
  - Wyświetlenie listy turniejów rozgrywanych w zadanym okresie;
  - Wyświetlenie rankingu zawodników klubu w zadanym okresie;
  - Wyświetlenie składu grupy treningowej.

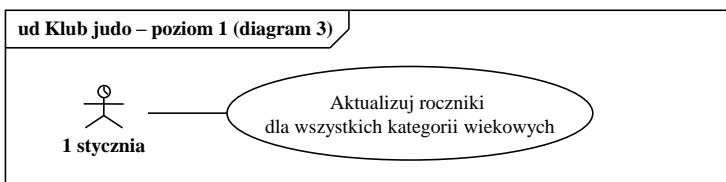
Dla przejrzystości, model przypadków użycia stworzony w oparciu o przedstawioną funkcjonalność został podzielony na pięć diagramów prezentujących kolejno: przypadki użycia typu CRUD (Rys. 12-2), podział przypadku *Zarządzaj kategoriami* na podprzypadki *Zarządzaj kategoriami wiekowymi*, *Zarządzaj kategoriami wagowymi* (Rys. 12-3), funkcjonalność aktora *Podsystem czasu (1 stycznia)* (Rys. 12-4), hierarchię aktorów wraz z przypadkami przypisanyimi do aktorów: *Osoba* (aktor abstrakcyjny z przypisanymi przypadkami wspólnymi dla aktorów *Gość*, *Członek klubu*), *Członek klubu*, *Zarząd klubu* (Rys. 12-5), funkcjonalność *Trenera* oraz wspólne przypadki użycia dla *Trenera i Zawodnika klubu*.



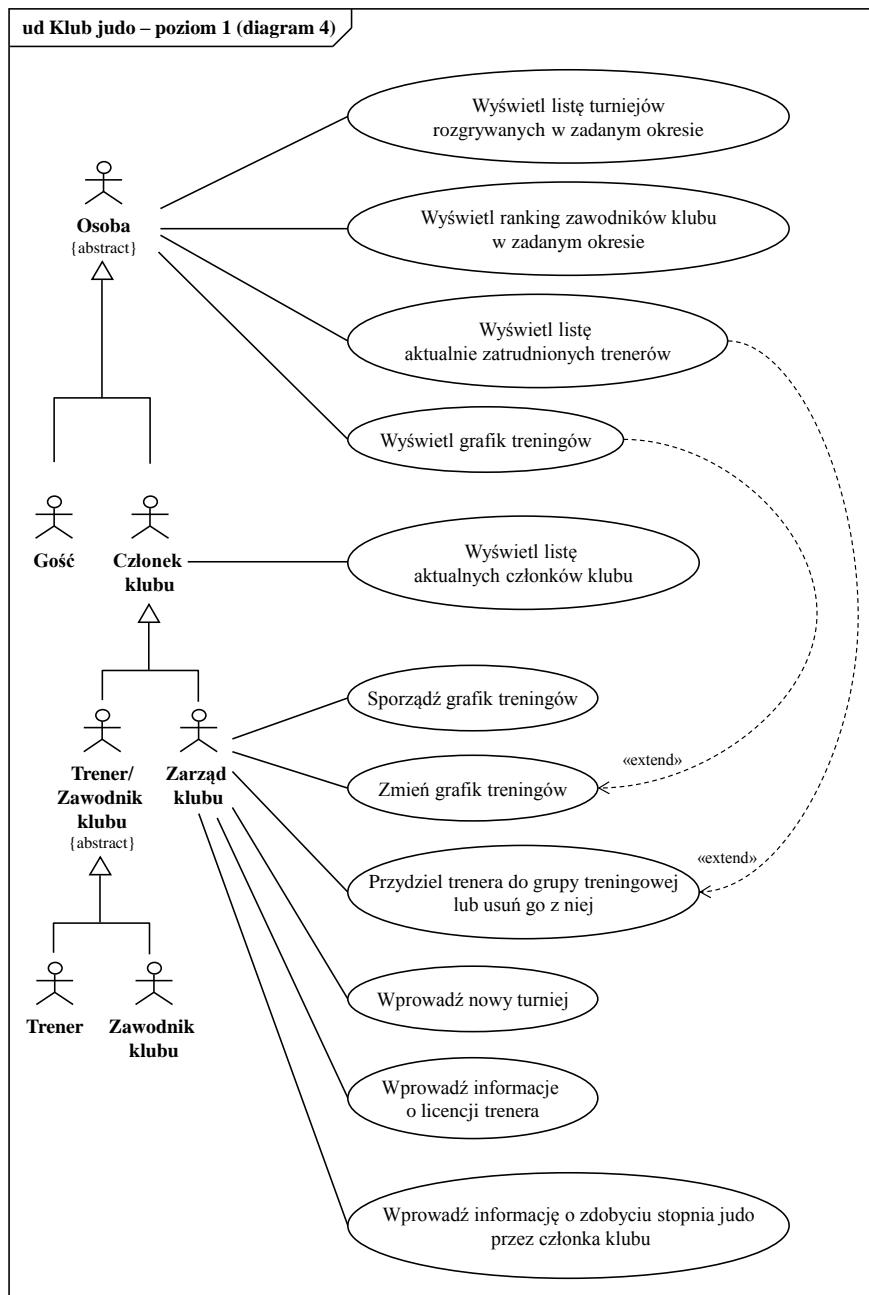
Rys. 12-2 Przypadki użycia typu CRUD (diagram 1)



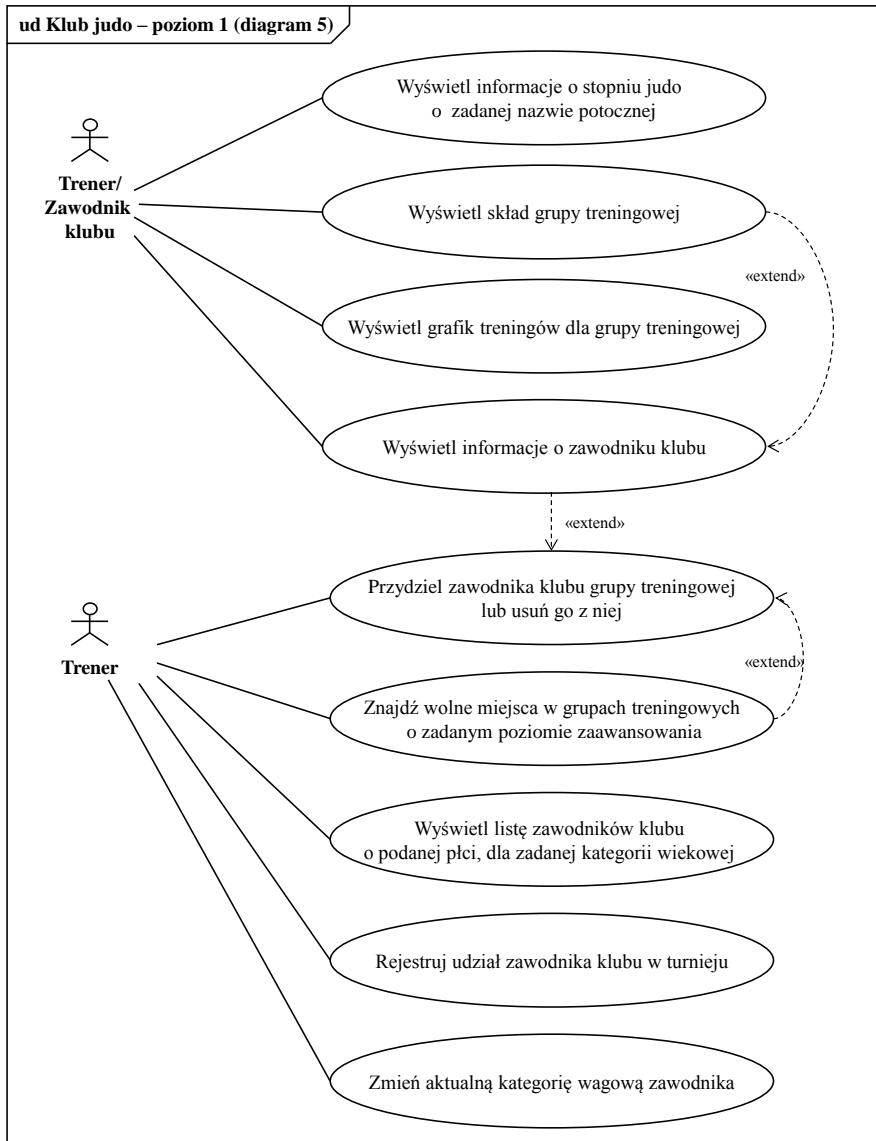
Rys. 12-3 Podział przypadku *Zarządzaj kategoriami* (diagram 2)



Rys. 12-4 Funkcjonalność aktora *Podsystem czasu* (diagram 3)



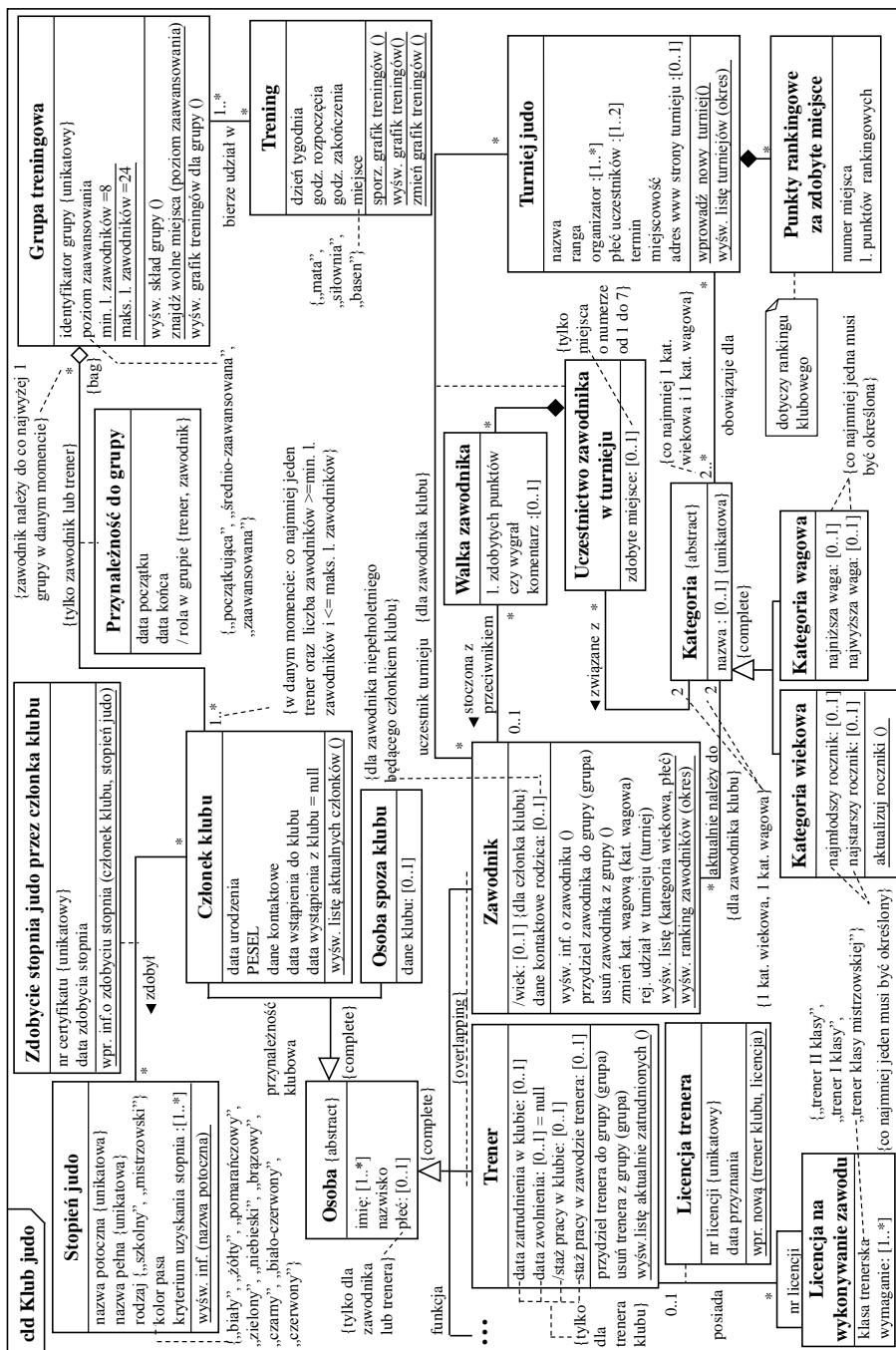
Rys. 12-5 Hierarchia aktorów wraz z przypadkami przypisanyimi do aktorów: *Osoba*, *Członek klubu*, *Zarząd klubu* (diagram 4)



Rys. 12-6 Funkcjonalność aktora *Trener* oraz wspólne przypadki użycia dla aktorów *Trener* i *Zawodnik klubu* (diagram 5)

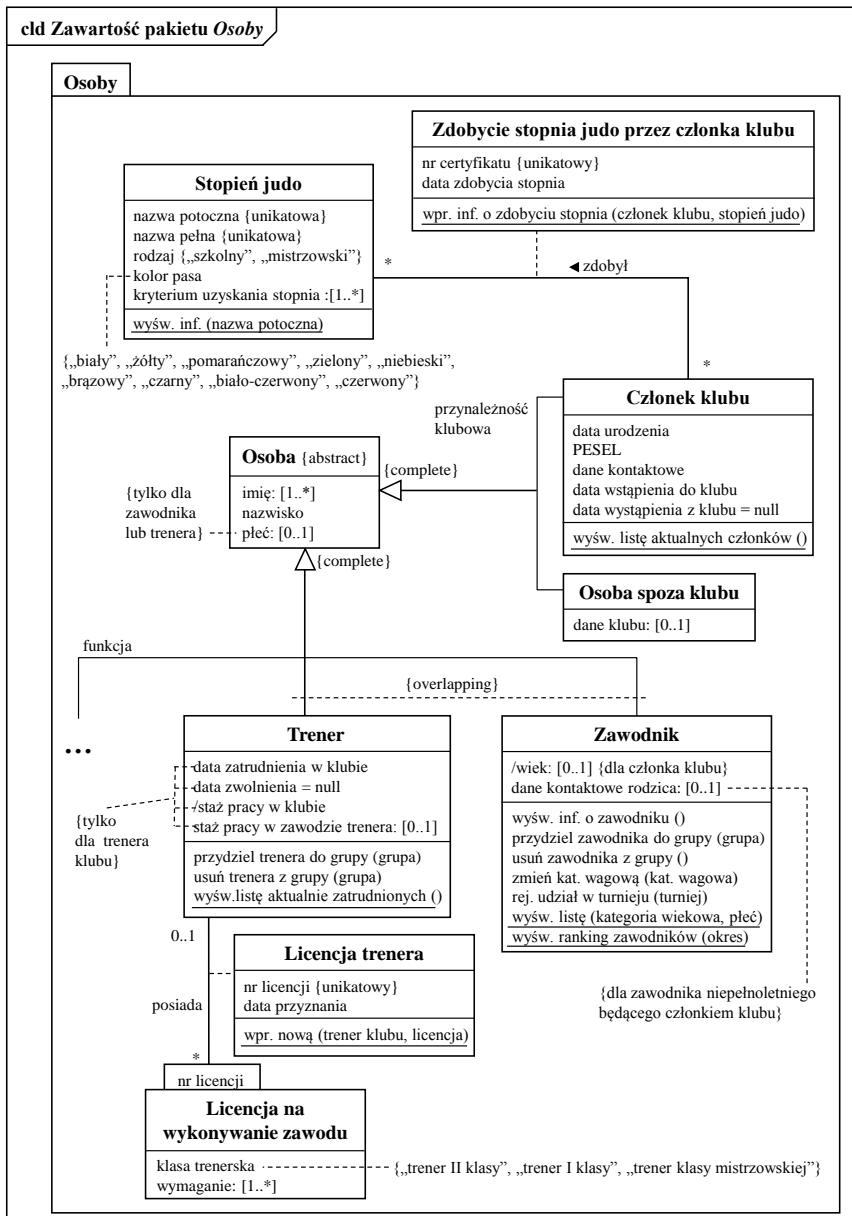
### 12.3.2 Opis struktury systemu (schemat pojęciowy)

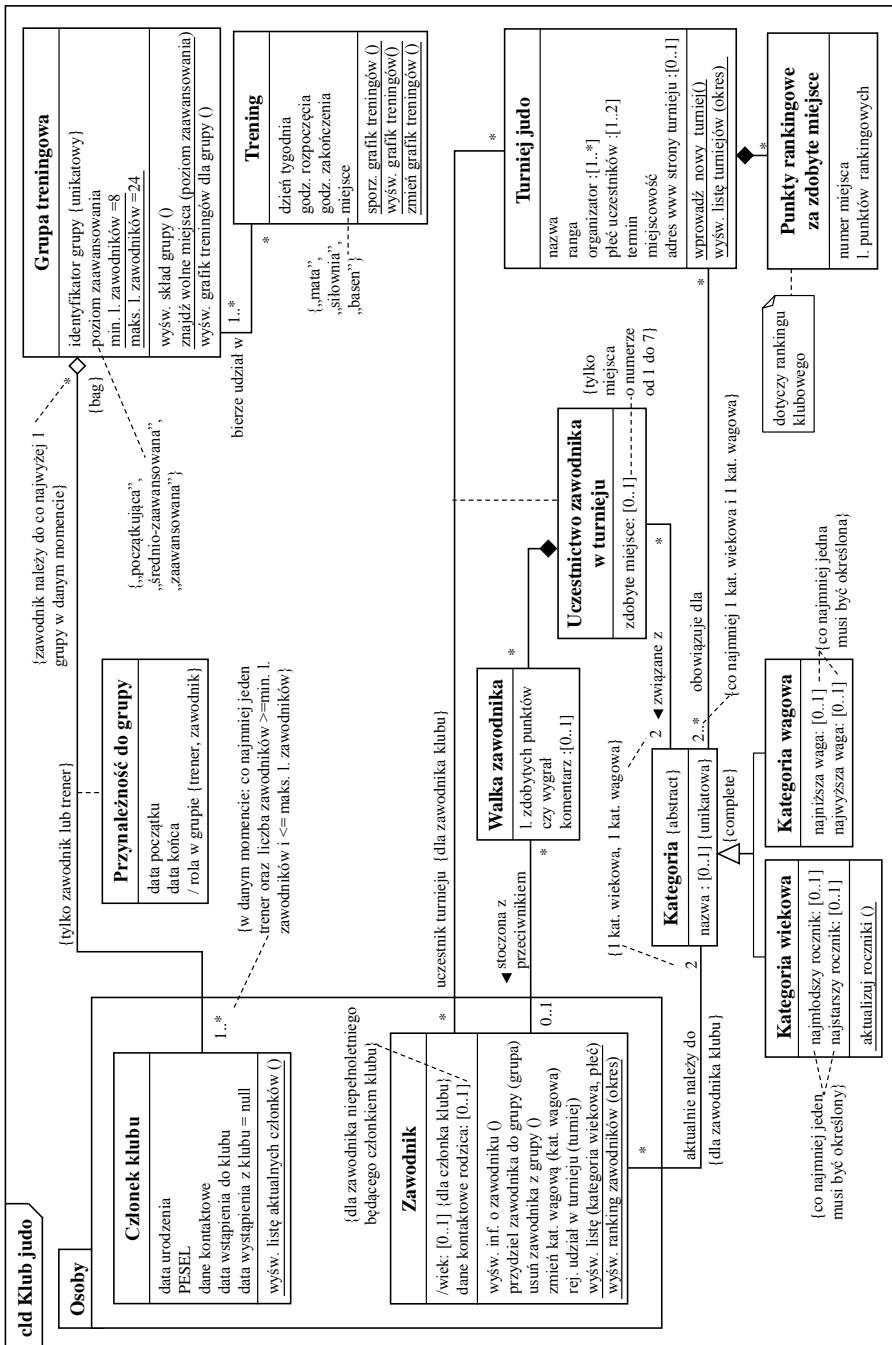
Schemat pojęciowy [35, 41, 43] przedstawiono na Rys. 12-7.



Rys. 12-7 Schemat pojęciowy systemu *Klub judo*

Dla poprawy percepcji schematu pojęciowego zaproponowano wyodrębnienie pakietu *Osoby* (Rys. 12-8), który następnie został wykorzystany na schemacie zaprezentowanym na Rys. 12-9.

Rys. 12-8 Zawartość pakietu *Osoby*

Rys. 12-9 Schemat pojęciowy z wykorzystanym pakietem *Osoby*

Należy zwrócić uwagę, że na Rys. 12-9 uwzględniono tylko te klasy z pakietu *Osoba*, które zostały połączone asocjacjami z klasami nieobejmowanymi przez ten pakiet.

### 12.3.3 Wymagania niefunkcjonalne – miary

*Dla przypomnienia:* Wymagania niefunkcjonalne [38, 41] powinny być możliwe do zweryfikowania, dlatego też należy zdefiniować kryteria ich oceny (tzw. *miary*).

W Tab. 12-1 przedstawiono te wymagania niefunkcjonalne, które można zweryfikować w oparciu o kryteria ilościowe wraz z propozycjami miar pozwalających na ich weryfikację.

Tab. 12-1 Wymagania niefunkcjonalne – miary „ilościowe”

Ograniczenie	Miara
dostępność	- czas, w jakim system powinien być dostępny dla użytkowników: 24 godziny przez 7 dni w tygodniu przez 365 dni w roku (dostępność w trybie 24/7/365)
intuicyjność interfejsu użytkownika	- system powinien zapewnić polskojęzyczny interfejs użytkownika oraz polskojęzyczne wartości danych przechowywanych w systemie (sortowanie danych z uwzględnieniem polskich liter, reprezentacja dat i liczb zgodna z polskim standardem)
łatwość użytkowania	- typowy użytkownik powinien mieć umiejętność korzystania ze wszystkich funkcji systemu po szkoleniu około 3 godzin - czas dla przeszkolenia dla osoby niedoświadczonej w obsłudze komputera – około 8-10 godzin - liczba ekranów pomocy nie powinna być mniejsza niż 5% wszystkich ekranów interfejsu
możliwość równoczesnej pracy wielu użytkowników	- dostęp i poprawna obsługa dla 500 użytkowników jednocześnie korzystających z systemu, w tym dla 100 użytkowników jednocześnie zalogowanych (zarejestrowanych użytkowników) oraz 400 użytkowników niezarejestrowanych (gości)
niezawodność	- średni czas pracy systemu bez awarii nie powinien być krótszy niż 6 miesięcy
poprawna obsługa funkcji systemu poprzez różne urządzenia zewnętrzne	- dostęp do wszystkich funkcjonalności oferowanych przez system powinien być możliwy zarówno przy użyciu myszki, jak również klawiatury

poprawne działanie na typowym współczesnym komputerze osobistym	<ul style="list-style-type: none"> <li>- system powinien zostać zaprojektowany na komputery PC</li> <li>- system powinien działać poprawnie pod systemami operacyjnymi: Windows XP/ Windows Vista/ Windows 7 i późniejsze, zarówno w wersji 32, jak i 64 bitowej</li> <li>- system powinien działać poprawnie na komputerach spełniających następujące wymagania sprzętowe: HDD: 12 G, RAM: 1 GB RAM, Pentium 200MHz</li> <li>- system powinien działać prawidłowo na komputerach z monitorami o rozdzielcości 800x600, 1024x768 lub wyższej</li> </ul>
prawidłowa współpraca z powszechnie stosowanymi drukarkami	<ul style="list-style-type: none"> <li>- system powinien współpracować z drukarkami firm: Canon, Epson, HP (Hewlett-Packard)</li> </ul>
przechowywanie danych w bazie danych	<ul style="list-style-type: none"> <li>- serwerem bazy danych powinien być Microsoft SQL Server 2000 lub nowszy</li> </ul>
spełnienie przez system typowych wymagań czasowych/ wydajnościowych	<p>wymagania czasowe/wydajnościowe:</p> <ul style="list-style-type: none"> <li>- opóźnienia w wyszukiwaniu danych w bazie do 15 sek, zgodne z metryką: <ul style="list-style-type: none"> <li>○ 5 sek - praca bardzo wydajna</li> <li>○ 10 sek - praca wydajna</li> <li>○ 15 sek - praca poprawna</li> <li>○ powyżej 15 sek - praca niepoprawna</li> </ul> </li> <li>- czas wygenerowania strony powinien wynosić maks. nie dłużej niż 3 sekundy dla 95 % operacji</li> <li>- czas wykonania kopii zapasowej nie powinien być dłuższy niż 30 min.</li> <li>- użytkownicy powinni być połączeni z systemem poprzez Internet, łączem o przepustowości co najmniej 128 kbps</li> </ul>
szybka reakcja systemu na akcje użytkownika	<ul style="list-style-type: none"> <li>- czas reakcji systemu na jakąkolwiek akcję użytkownika nie powinien przekraczać trzech sekund</li> <li>- operacje długotrwałe (przekraczające trzy sekundy) powinny być sygnalizowane wskaźnikiem postępu wykonania danej operacji, a użytkownik powinien mieć możliwość przerwania wykonywanej czynności</li> <li>- czas odświeżania ekranu nie powinien przekraczać trzech sekund</li> </ul>

wygląd interfejsu niezależny od rodzaju przeglądarki internetowej	<ul style="list-style-type: none"> <li>- interfejs powinien działać poprawnie z zaleconymi przeglądarkami internetowymi (przy zachowaniu takiego samego wyglądu): Google Chrome 22.0.1229, Internet Explorer 10, K-Meleon 1.5.4, Mozilla Firefox 20.0, Opera 12.10, Safari 5.1.2, Slim Brower 6.0 oraz z ich starszymi wersjami</li> </ul>
zabezpieczenie poprawności danych (walidacja danych)	<ul style="list-style-type: none"> <li>- wszystkie pola do wprowadzania danych powinny zostać zabezpieczone przed wprowadzaniem niepoprawnych danych</li> </ul>
zapewnienie bezpieczeństwa i ochrony danych	<p>system powinien:</p> <ul style="list-style-type: none"> <li>- przetwarzanie i przechowywanie informacji osobowe w zgodzie z Ustawą o Ochronie Danych Osobowych (Dz.U. 1997 Nr 133 poz. 883 USTAWA z dnia 29 sierpnia 1997 o ochronie danych osobowych); każdy członek klubu musi zgodzić się na przetwarzanie jego danych osobowych, zgodnie z ww. ustawą</li> <li>- zapewnić ochronę przed utratą danych, umożliwiając tworzenie (<i>backup</i>) i odtwarzanie kopii zapasowych bazy danych (kopii całościowych), w tym: automatyzacja wykonywania kopii periodycznych, definiowanie harmonogramu wykonywania kopii periodycznych, automatyczne utrzymywanie określonej liczby kopii bezpieczeństwa oraz prowadzenie dziennika wykonanych kopii bezpieczeństwa; powiadamianie o zaistniałych błędach i awariach wykonania kopii na podany adres email</li> <li>- zapewnić bezpieczeństwo gromadzenia danych przez transakcyjne przetwarzanie danych gwarantujące poprawność i spójność zapisów bazy danych w przypadku normalnej pracy systemu, jak i jego awarii</li> </ul>

W Tab. 12-2 przedstawiono wymagania niefunkcjonalne [38] dotyczące interfejsu, których nie da się zweryfikować w oparciu o kryteria ilościowe.

Tab. 12-2 Przykładowe wymagania niefunkcjonalne dotyczące interfejsu

Ograniczenie	Miara „jakościowa”
intuicyjność, czytelność, ergonomiczność interfejsu użytkownika	<ul style="list-style-type: none"> <li>- system powinien posiadać jednolity graficzny interfejs użytkownika (wygląd ekranu, obsługa klawiszy funkcyjnych, menu)</li> <li>- interfejs powinien być zgodny z projektem</li> </ul>

	<p>zatwierdzonym przez przyszłego użytkownika</p> <ul style="list-style-type: none"> <li>- wszystkie informacje potrzebne użytkownikowi w danym momencie powinny być w miarę możliwości wyświetlane na jednym ekranie, tak aby nie było konieczne jego przewijanie</li> </ul>
--	---

## 12.4 Opis przyszłej ewolucji systemu

W przyszłości planowana jest rozbudowa systemu o wprowadzenie danych sponsorów i rejestrację darowizn, a także dołączenie możliwości rejestrowania akcji promujących klub (np. pokazy judo w szkołach) oraz zgłaszanie zawodników na turnieje.

## 12.5 Słownik pojęć z dziedziny problemowej

Słownik pojęć z dziedziny problemowej został przedstawiony w Tab. 12-3.

Tab. 12-3 Słownik pojęć z dziedziny problemowej

Pojęcie (termin)	Synonim pojęcia	Objaśnienie
certyfikat nadania stopnia judo	certyfikat	Zawodnik otrzymuje certyfikat (zaświadczenie) nadania stopnia judo po spełnieniu określonych kryteriów wymaganych dla uzyskania tego stopnia (patrz <i>stopień judo</i> ). Certyfikat opisany jest unikatowym numerem; zawiera informacje: kto i kiedy uzyskał dany stopień.
członek klubu		Członkiem klubu jest osoba, która należy do klubu; może, ale nie musi być zawodnikiem czy też trenerem. Dla członka klubu należy pamiętać imiona (co najmniej jedno), nazwisko, płeć, datę urodzenia, PESEL, dane kontaktowe, datę wstąpienia do klubu oraz datę wystąpienia z klubu. Członek klubu może posiadać stopień/stopnie judo (patrz <i>stopień judo</i> ), przy czym zdobycie każdego stopnia jest potwierdzane certyfikatem (patrz <i>certyfikat nadania</i> )

		<p><i>stopnia judo).</i></p> <p>System powinien umożliwiać wyświetlenie listy wszystkich aktualnych członków klubu. Dostęp do tej funkcjonalności ma każdy członek klubu. Członek klubu ma także możliwość wywołania funkcjonalności zdefiniowanej dla gościa (patrz gość).</p>
gość	niezarejestrowany użytkownik	<p>Użytkownik, którego dane nie są rejestrowane w systemie.</p> <p>Ma on dostęp do następujących usług systemu:</p> <ul style="list-style-type: none"> <li>- wyświetlenie listy turniejów rozgrywanych w zadanym okresie,</li> <li>- wyświetlenie grafiku treningów,</li> <li>- wyświetlenie rankingu zawodników w zadanym okresie,</li> <li>- wyświetlenie listy aktualnie zatrudnionych trenerów.</li> </ul>
grupa	grupa treningowa	patrz <i>grupa treningowa</i>
grupa treningowa	grupa	<p>Grupa treningowa opisana jest unikatowym identyfikatorem oraz poziomem zaawansowania („początkująca”, „średniozaawansowana”, „zaawansowana”). Grupa bierze udział w treningach (patrz <i>trening</i>). W skład grupy treningowej wchodzą zawodnicy klubu i trenerzy (co najmniej 1), przy czym ustalona jest minimalna i maksymalna liczba zawodników, którzy mogą być w danym momencie przypisani do grupy. Aktualnie są to odpowiednio: minimalna liczbeowość grupy równa 8 zawodników i maksymalna liczbeowość grupy równa 24 zawodników. Wartości te mogą w przyszłości ulec zmianie. Każdą grupę prowadzi co najmniej jeden trener. Nie ma ograniczenia na maksymalną liczbę grup przydzielonych jednemu trenerowi.</p> <p>Należy archiwizować wszystkie okresy przynależności danej osoby do grupy</p>

		<p>treninguowej, przy czym może zdarzyć się, że zawodnik/trener w różnych okresach należał do tej samej grupy.</p> <p>System powinien umożliwiać:</p> <ul style="list-style-type: none"> <li>- wyświetlenie składu grupy treningowej,</li> <li>- wyświetlenie grafiku jej treningów,</li> <li>- znalezienie wolnych miejsc w grupach o zadany poziomie zaawansowania.</li> </ul>
judoka	zawodnik judo, zawodnik	patrz <i>zawodnik judo</i>
kategoria		Kategoria może oznaczać kategorię wiekową lub wagową (patrz <i>kategoria wagowa</i> , <i>kategoria wiekowa</i> ).
kategoria wiekowa		<p>Kategoria wiekowa określa przedział wiekowy zawodników, którzy mogą brać udział w turniejach judo zorganizowanych dla tej kategorii wiekowej.</p> <p>Istnieją kategorie wiekowe ustalone przez Międzynarodową Federację Judo (MFJ) (aktualnie kategorie nazywane są następująco: <i>dzieci</i>, <i>młodzicy/młodziczki</i>, <i>juniorzy/juniorki młodszego</i>, <i>juniorzy/juniorki</i>, <i>młodzież</i>, <i>seniorzy/seniorki</i>), obowiązujące dla turniejów judo organizowanych zgodnie z regulaminem MFJ. W projektowanym systemie podjęto decyzję, aby nie ograniczać kategorii wiekowych do ww., gdyż organizatorzy turniejów mogą ustalać własne kategorie. Kategoria wiekowa może być opisywana unikatową nazwą (ustalaną przez organizatora/ organizatorów turnieju) i rocznikami zawodników należących do tej kategorii – najmłodszy i najstarszy rocznik (np. <i>młodzicy</i> – roczniki 1995-1996). <i>Rocznik</i> oznacza rok, w którym urodził się zawodnik. Zdarzają się kategorie wiekowe, dla których określa się tylko jedną wartość (najmłodszy lub najstarszy rocznik), ale co najmniej jedna z tych nich musi być</p>

		<p>określona.</p> <p>System powinien umożliwiać aktualizację przedziałów wiekowych dla wszystkich kategorii tego rodzaju (pamiętane są tylko ich aktualne wartości) wraz z aktualizacją przynależności zawodników klubu do odpowiednich kategorii. Przykładowo, w roku kalendarzowym 2012 kategoria wiekowa <i>młodzież</i> obejmowała zawodników urodzonych w latach 1991-1992. W roku 2013 należy zaktualizować kategorie wiekowe i zmienić przedział wiekowy dla kategorii <i>młodzież</i> na roczniki 1992-1993 oraz zaktualizować przypisanie zawodników klubu do odpowiednich kategorii.</p>
kategoria wagowa		<p>Kategoria wagowa określa limity wagowe zawodników, którzy mogą ze sobą walczyć w turnieju judo (w tej samej walce). Istnieją kategorie wagowe ustalone przez Międzynarodową Federację Judo, odrębnie dla każdej kategorii wiekowej, z uwzględnieniem płci uczestników zawodów (przykładowo dla seniorek są to aktualnie: poniżej 48kg, od 48kg do 52kg, od 52kg do 57kg, od 57kg do 63kg, od 63kg do 70kg, od 70kg do 78kg, powyżej 78kg). Organizatorzy turniejów mogą ustalać własne kategorie wagowe uczestników, co powinno zostać uwzględnione w projektowanym systemie.</p> <p>Kategoria wagowa może mieć unikatową nazwę (może jej nie mieć) oraz limity wagowe: najniższą i najwyższą wagę zawodnika w tej kategorii. Podobnie jak w przypadku kategorii wiekowej, dla kategorii wagowej może zostać określona tylko jedna z wartości granicznych – najniższa lub najwyższa waga zawodnika, ale zawsze co najmniej jedna z nich musi być określona.</p> <p>System powinien umożliwiać zmianę aktualnej kategorii wagowej danego zawodnika klubu.</p>

klasa trenerska		Klasa trenerska określa stopień wykwalifikowania kadry trenereskiej w danej dyscyplinie sportowej. Aktualnie, zgodnie klasyfikacją zawodów i specjalności, są to: <i>trener II klasy</i> (najniższy stopień wykwalifikowania), <i>trener I klasy</i> , <i>trener klasy mistrzowskiej</i> .
klubowy ranking zawodników	ranking zawodników	Klubowy ranking zawodników powstaje w wyniku uporządkowania zawodników klubu według łącznej liczby punktów rankingowych uzyskanych przez zawodników na turniejach rozegranych w zadanym okresie (patrz <i>punktacja do klubowego rankingu zawodników</i> ).
licencja	licencja na wykonywanie zawodu trenera	Licencja na wykonywanie zawodu trenera dotyczy konkretnej klasy trenereskiej (np. „trener II klasy”). Określona jest lista wymagań, jakie trzeba spełnić w celu zdobycia danej licencji.
licencja trenera		Licencja trenera jest dokumentem wydanym trenerowi, który spełnił wymagania pozwalające na zdobycie odpowiedniej klasy trenereskiej. Oprócz danych trenera, pamiętaana jest również data przyznania licencji i jej unikatowy numer.
osoba spoza klubu		Osoba związana z klubem, niebędąca jego członkiem, dla której należy pamiętać dane klubu, do którego należy (o ile należy do jakiegoś klubu).
punktacja do klubowego rankingu zawodników	punktacja rankingowa	Dla turnieju powinna istnieć możliwość zapamiętania listy miejsc, które są punktowane w rankingu klubowym zawodników (patrz <i>ranking klubowy</i> ) wraz z liczbą punktów przyznawanych zawodnikowi za zajęcie danego punktowanego miejsca. Przykładowo, każdy zawodnik, klubu, który w danym turnieju zajmie pierwsze miejsce, otrzyma 10 punktów rankingowych, za zajęcie drugiego miejsce – 7 punktów itp. W turnieju judo miejsca zawodników nie są

		unikatowe w ramach turnieju. Przykładowo, jeśli w turnieju walki rozgrywane są w ośmiu kategoriach wagowych w kategorii wiekowej <i>dzieci</i> , to będzie przyznanych osiem pierwszych miejsc w tej kategorii wiekowej.
ranking zawodników	klubowy ranking zawodników	patrz <i>klubowy ranking zawodników</i>
stopień	stopień judo	patrz <i>stopień judo</i>
stopień judo	stopień	Zawodnicy trenujący judo zdobywają kolejne stopnie świadczące o nabytych przez nich umiejętnościach. O posiadanym stopniu judo świadczy kolor pasa służącego do przewiązywania stroju do trenowania judo (tzw. <i>judogi</i> ). Kolory pasów są ściśle określone, uporządkowane rosnąco według coraz wyższych stopni, którym odpowiadają: biały, żółty, pomarańczowy, zielony, niebieski, brązowy, czarny, biało-czerwony i czerwony. Dla każdego stopnia judo, oprócz odpowiadającego mu koloru pasa, należy zapamiętać: unikatową nazwę potoczną (np. 6 kyu), unikatową nazwę pełną (w języku japońskim, np. rokyu), rodzaj stopnia (szkolny lub mistrzowski) oraz kryteria nadawania stopnia (w postaci listy z co najmniej jedną pozycją). System ma umożliwiać wyświetlenie informacji o stopniu judo o podanej nazwie potocznej.
trener	trener judo, trener klubu	patrz <i>trener klubu</i>
trener judo	trener, trener klubu	patrz <i>trener klubu</i>
trener klubu	trener, trener judo	Klub zatrudnia trenerów judo, dla których oprócz danych takich jak dla każdego innego członka klubu, należy dodatkowo pamiętać datę zatrudnienia w klubie, datę zwolnienia, staż pracy w klubie oraz staż pracy

		<p>w zawodzie, o ile nie jest to jego pierwsza praca w charakterze trenera judo. Trener może być zawodnikiem klubu (patrz <i>zawodnik klubu</i>). Trenerzy zatrudnieni w klubie są równocześnie jego członkami.</p> <p>Każdy z trenerów może prowadzić kilka grup treningowych jednocześnie (patrz <i>grupa treningowa</i>). Należy przechowywać daty początku i końca prowadzenia przez niego grupy/grup treningowych.</p> <p>Jeśli trener posiada licencję/licencje na wykonywanie zawodu (patrz <i>licencja</i>), należy pamiętać odpowiednie dane (patrz <i>licencja trenera</i>).</p> <p>System powinien umożliwiać:</p> <ul style="list-style-type: none"> <li>- przydzielenie trenera do grupy treningowej,</li> <li>- usunięcie go z wybranej grupy treningowej (równoznaczne z odebraniem mu opieki nad grupą),</li> <li>- wyświetlenie listy aktualnie zatrudnionych trenerów.</li> </ul> <p>Trener ma dostęp do funkcjonalności takiej, jaką ma zawodnik klubu (patrz <i>zawodnik klubu</i>), a dodatkowo może:</p> <ul style="list-style-type: none"> <li>- rejestrować udział zawodnika w turnieju,</li> <li>- przydzielać zawodnika do grupy treningowej,</li> <li>- wyświetlać listę zawodników klubu o podanej płci – dla zadanej kategorii wiekowej,</li> <li>- znajdować wolne miejsca w grupach treningowych o zadanym poziomie zaawansowania.</li> </ul>
trening	trening judo	<p>W treningu może wziąć udział jedna lub kilka grup treningowych (patrz <i>grupa treningowa</i>). System powinien przechowywać informacje o zaplanowanym na trening dniu tygodnia, godzinach rozpoczęcia i zakończenia, miejscu („mata”, „siłownia”, „basen”), grupie/grupach biorących w nim udział. Grafik treningów ustalany jest na 7 dni w tygodniu i powtarza</p>

		<p>się co tydzień. np. przez cały rok szkolny. Lista obecności zawodników na treningu oraz informacja, czy dany trening się odbył, nie są istotne i nie mają być pamiętane.</p> <p>System powinien umożliwiać sporządzanie/ wyświetlanie/ zmianę grafiku treningów.</p>
turniej	turniej judo, zawody judo	<p>patrz <i>turniej judo</i></p>
turniej judo	turniej, zawody judo	<p>Dla każdego turnieju judo przechowuję się informacje takie jak: nazwa turnieju, jego ranga, lista organizatorów (z co najmniej jedną pozycją), termin, miejscowością i ewentualnie adres WWW strony turnieju. Ma być wiadomo, dla jakich kategorii wiekowych i wagowych oraz dla osób jakiej płci został zorganizowany. Organizatorzy turnieju muszą określić co najmniej jedną kategorię wiekową i jedną wagową.</p> <p>Dla turnieju może być ustalona punktacja służąca do tworzenia rankingów zawodników klubu (patrz <i>punktacja do klubowego rankingu zawodników</i>).</p> <p>Ma być rejestrowany udział zawodnika klubu w turnieju, w tym kategorie (wiekowa i wagowa), w których startował, listy walk, które stoczył, a także miejsce, które zajął (przy czym brane są pod uwagę wyłącznie miejsca o numerze od 1 do 7) – patrz <i>walka</i>. System rozgrywania walk w danej kategorii wagowej zależy od liczby zgłoszonych zawodników do tej kategorii. Nie ma on jednak znaczenia dla projektowanego systemu i dlatego został pominięty w rozważaniach.</p> <p>System ma umożliwiać wyświetlenie listy turniejów zorganizowanych w zadanym okresie.</p>
udział zawodnika	udział zawodnika w turnieju	Zawodnicy mogą brać udział w turniejach judo. System powinien umożliwiać

klubu w turnieju		rejestrowanie faktu udziału zawodnika w turnieju, w tym kategorie (wiekowa i wagowa), w których startował, listy walk, które stoczył, a także miejsce, które zajął (przy czym brane są pod uwagę wyłącznie miejsca o numerze od 1 do 7).
udział zawodnika w walce		Dla każdej stoczonej walki zawodnika klubu należy przechowywać informacje o tym, czy ją wygrał, ile punktów zdobył oraz można też dołączyć komentarz.  W każdej walce uczestniczy dwóch judoków i jeżeli istotne są dane przeciwnika zawodnika klubu, informacja taka ma być dodatkowo pamiętana.
walka	walka judo	W czasie trwania turnieju zawodnicy uczestniczą w walkach (w jednej walce – dwóch zawodników). Walka musi zakończyć się zwycięstwem jednego z zawodników.  Dla każdej walki należy przechowywać informacje o zawodnikach klubu biorących w niej udział (patrz <i>zawodnik, udział zawodnika w walce</i> ).
zarząd klubu		Osoby będące w zarządzie klubu są równocześnie członkami klubu.  Zarząd klubu ma dostęp do takiej samej funkcjonalności, jaką ma każdy członek klubu (patrz <i>członek klubu</i> ), a ponadto może m.in. :  <ul style="list-style-type: none"> <li>- przydzielać trenera do grupy treningowej lub usuwać go z niej ,</li> <li>- sporządzać/ zmieniać grafik treningów,</li> <li>- zarządzać danymi członków klubu, turniejów judo, treningów, grup treningowych, stopni judo, licencji oraz kategorii wiekowych i wagowych.</li> </ul>
zawodnik	judoka, zawodnik judo	patrz <i>zawodnik judo</i>

zawodnik judo	judoka, zawodnik	Zawodnik to osoba trenująca judo. Zawodnik może być jednocześnie trenerem judo. Zawodnicy zarejestrowani w systemie mogą być zarówno członkami klubu, jak i osobami spoza klubu.  Dla zawodników biorących udział w turnieju pamiętać należy informacje związane z tym faktem (patrz <i>udział zawodnika w turnieju</i> ).
zawodnik klubu		Zawodnik klubu to członek klubu trenujący judo; może być jednocześnie trenerem judo w tym klubie. Dla zawodnika będącego członkiem klubu należy dodatkowo przechowywać: wiek oraz dane kontaktowe rodzica (tylko dla osoby niepełnoletniej).  Każdy zawodnik klubu przypisany jest do konkretnej kategorii wiekowej i wagowej, przy czym mają być przechowywane dane aktualne.  Zawodnik klubu ma dostęp do takiej samej funkcjonalności, jaką ma każdy członek klubu (patrz <i>członek klubu</i> ), a dodatkowo może m.in. wyświetlać: informacje o stopniu judo o zadanej nazwie potocznej, informacje o zawodniku, grafik treningów dla grupy treningowej oraz skład grupy treningowej.
zawody judo	turniej, turniej judo	patrz <i>turniej judo</i>

## 12.6 Podsumowanie zadania

W Tab. 12-4 przedstawiono ogólną charakterystykę omawianego zadania. Szczegółową charakterystykę zadania dla grup tematycznych: *Wprowadzenie do przedmiotu*, *Analiza funkcjonalna*, *Analiza strukturalna* zamieszczono odpowiednio w tabelach Tab. 12-5 – Tab. 12-7.

Tab. 12-4 Ogólna charakterystyka zadania

<b>Ogólna charakterystyka zadania</b>	
<i>Nazwa</i>	Klub Judo
<i>Dziedzina problemowa</i>	system może zostać wykorzystany jako narzędzie wspierające zarządzanie niewielkimi klubami związanymi ze sportami walki
<i>Cel</i>	wypromowanie klubu poprzez ułatwienie dostępu do danych dotyczących m.in. trenerów i zawodników, do terminarza treningów oraz turniejów, czy też do informacji o osiągnięciach najlepszych zawodników (aktualnych i byłych)
<i>Zakres odpowiedzialności</i>	zarządzanie informacjami dotyczącymi: członków klubu, jak również osób związanych z klubem, ale niebędących jego członkami, treningów, turniejów judo oraz wykonywanie zestawień statystycznych
<i>Stopień trudności</i>	trudne
<i>Rozmiar</i>	15 klas
<i>Użyteczność w grupach tematycznych</i>	
<i>Grupa tematyczna</i>	Użyteczność w grupie tematycznej
<i>Wprowadzenie do przedmiotu</i>	niska <i>Przygotowanie do nauki:</i> niska <i>Nauka:</i> średnia <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> niska
<i>Analiza funkcjonalna</i>	średnia <i>Przygotowanie do nauki:</i> średnia <i>Nauka:</i> średnia <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> niska
<i>Analiza strukturalna</i>	wysoka <i>Przygotowanie do nauki:</i> średnia <i>Nauka:</i> wysoka <i>Podsumowanie dotychczasowych działań:</i> niska <i>Ewaluacja:</i> średnia

Tab. 12-5 Szczegółowa charakterystyka zadania dla grupy tematycznej *Wprowadzenie do przedmiotu*

<b>Szczegółowa charakterystyka zadania</b>		
<b>Grupa tematyczna</b>	<i>Wprowadzenie do przedmiotu</i>	
<b>Stopień trudności grupy w zadaniu</b>	łatwe	
<b>Poruszane tematy</b>	<i>mechanizmy rozszerzalności, dekompozycja diagramów, reguły nazewnictwa, poprawa percepcji diagramu/diagramów</i>	
<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>mechanizmy rozszerzalności</i>	łatwe	<i>stereotyp, ograniczenie, komentarz</i>
<i>dekompozycja diagramów</i>	łatwe	podział modelu przypadków użycia na diagramy
<i>reguły nazewnictwa</i>	łatwe	reguły nazewnictwa dla aktorów, przypadków użycia, klas
<i>poprawa percepcji diagramu/diagramów</i>	łatwe	organizacja diagramu przypadków użycia, wyodrębnienie pakietu w celu poprawy percepcji schematu pojęciowego

Tab. 12-6 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza funkcjonalna*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Analiza funkcjonalna</i>
<b>Stopień trudności grupy w zadaniu</b>	średnio trudne
<b>Poruszane tematy</b>	<i>wymagania na system, aktorzy, przypadki użycia, słownik pojęć, dekompozycja modelu/diagramu przypadków użycia</i>

<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
wymagania na system	średnio trudne	analiza wymagań na system – określenie przykładowych wymagań niefunkcjonalnych wraz z określeniem miar ilościowych służących do ich weryfikacji
aktorzy	średnio trudne	aktor Podsystem czasu, rozbudowana hierarchia aktorów z aktorem abstrakcyjnym,
przypadki użycia	łatwe	przypadki użycia typu CRUD, relacje pomiędzy przypadkami użycia «include» i «extend»
słownik pojęć	łatwe	rozbudowany słownik pojęć z dziedziny problemowej
dekompozycja modelu/diagramu przypadków użycia	łatwe	dekompozycja pozioma – przedstawienie modelu przypadków użycia w postaci kilku diagramów

Tab. 12-7 Szczegółowa charakterystyka zadania dla grupy tematycznej *Analiza strukturalna*

<b>Szczegółowa charakterystyka zadania</b>	
<b>Grupa tematyczna</b>	<i>Analiza strukturalna</i>
<b>Stopień trudności grupy w zadaniu</b>	trudne
<b>Poruszane tematy</b>	atrybuty klas, metody klas, struktury generalizacji-specjalizacji, klasa abstrakcyjna a klasa konkretna, asocjacje, dekompozycja modelu/diagramu klas, mechanizmy rozszerzalności

<b>Tematy, na których skupiono większą uwagę w zadaniu</b>	<b>Stopień trudności tematu w grupie tematycznej</b>	<b>Pojęcia, zagadnienia omawiane w temacie</b>
<i>atrybuty klasy</i>	łatwe	<i>atrybut opcjonalny, pochodny, powtarzalny, klasowy</i>
<i>metody klasy</i>	łatwe	<i>metoda zaimplementowana, obiektowa, klasowa, stosunkowo duża liczba metod</i>
<i>struktury generalizacji-specjalizacji</i>	trudne	<i>rodzaje dziedziczenia – wieloaspektowe, nierożłączne (dla wybranych podklas), typu elipsa</i>
<i>klasa abstrakcyjna a klasa konkretna</i>	łatwe	<i>klasa abstrakcyjna, klasa konkretna</i>
<i>asocjacje</i>	trudne	<i>nazwa asocjacji a rola asocjacji, klasa asocjacji, agregacja, kompozycja, asocjacja kwalifikowana, klasa asocjacji powiązana asocjacjami z innymi klasami</i>
<i>dekompozycja modelu/diagramu klas</i>	średnio trudne	wykorzystanie pakietu
<i>mechanizmy rozszerzalności</i>	łatwe	<i>ograniczenie, komentarz, stosunkowo duża liczba ograniczeń</i>

## Bibliografia

1. Kupisiewicz C. (2012), *Dydaktyka. Podręcznik akademicki*, Oficyna Wydawnicza IMPULS
2. Siemak-Tylikowska A., Dębska E., Widota E., Pierścieniak K. (2011), *Dydaktyka Szkoły Wyższej*, podręcznik akademicki, Uniwersytet Otwarty Uniwersytetu Warszawskiego
3. Siemak-Tylikowska A., Słomczyński M. (2011), *Czy kształcenie zdalne potrzebuje nowej dydaktyki?*, Technologie informacyjne w warsztacie nauczyciela. Nowe wyzwania edukacyjne, str. 31-41, Wydawnictwo Naukowe Uniwersytetu Pedagogicznego
4. Siemak-Tylikowska A. (2011), *Nowoczesna dydaktyka szkoły wyższej w nowoczesnym uniwersytecie? „Czy współczesny uniwersytet potrzebuje nowoczesnej dydaktyki szkoły wyższej?”* Publikacja pokonferencyjna 17 grudnia 2010 roku, str. 16-28, Uniwersytet Warszawski
5. Kupisiewicz C., Kupisiewicz M. (2009), *Słownik Pedagogiczny*, Warszawa, PWN
6. Kupisiewicz C. (2005), *Podstawy dydaktyki*, Warszawa, WSiP
7. Okoń W. (2004), *Nowy słownik pedagogiczny*, Warszawa, Wyd. Akademickie Źak
8. Okoń W. (2003), *Wprowadzenie do dydaktyki ogólnej*, Warszawa, Wyd. Akademickie Źak
9. Allessi S.M. and Trollip S.R. (2001), *Multimedia for Learning: Methods and Development*, Needham Heights: Allyn and Bacon
10. Joyce B., Calhoun E. and Hopkins D. (1999), *Przykłady modeli uczenia się i nauczania*, Warszawa, WSiP
11. Smith L. P., Ragan T. J. (1999), *Instructional Design*, Upper Saddle River, NJ: Prentice-Hall
12. Driscoll M.P. (1994), *Psychology of learning for instruction*, Needham Heights, MA: Allyn & Bacon
13. Mazur B., Siemak-Tylikowska A. (1993), *Sposób formułowania celów a efekty dydaktyczne*, TORUŃSKIE STUDIA DYDAKTYCZNE, Tom 1993r., str. 160-168
14. Gagné R. M. (1972), *The conditions of learning*, New York, Holt, Rinehart & Winston
15. Jodłowski A., Stasiecka A., Stemposz E. (2012), *The comparison of the influence of didactic and non-didactic features on the quality of materials used in distance*

- learning*, Monographs in Applied Informatics, Computing in Science and Technology 2011, ISBN 978-83-7583-378-2, Section 4, pp.51-65
16. Stemposz E., Stasiecka A. (2008), *Determining a set of measures for quality estimation of e-resources conformant to the model/models defined in traditional education*, (rozszerzona wersja Proc. of 6th WSEAS "E-ACTIVITIES '07"), journal WSEAS TRANSACTIONS ON ADVANCES in ENGINEERING EDUCATION, ISSN: 1790-1979, Issue 5, Volume 5, May 2008
  17. Stasiecka A., Stemposz E., Jodłowski A. (2008), *E-resources vs. traditional teaching models*, Journal of Telecommunications and Information Technology" (JTIT), no. 3
  18. Stasiecka A., Stemposz E. (2007), *Analiza jakości e-zasobów z wykorzystaniem programu GradeStat*, Prace IPI PAN: 1006, ISSN 0138-0648, grudzień 2007, 39 stron (na prawach rękopisu)
  19. Stemposz E., Stasiecka A. (2007), *Determining a set of measures for quality estimation of e-resources conformant to the model/models defined in traditional education*, Proc. of 6th WSEAS Int. Conf. on E-ACTIVITIES (E-Learning, E-Communities, E-Commerce, E-Management, E-Marketing, E-Governance, Tele-Working) (E-ACTIVITIES '07), Puerto de la Cruz, Tenerife, Spain, 14-16 December 2007, ISBN: 978-960-6766-22-8, ISSN: 1790-5117, pp 320-325
  20. Stemposz E., Stasiecka A., Jodłowski A. (2007), *The proposal of metadata for defining the quality of e-learning*, Proc. of ENMA 2007 (Engineering and Mathematics), July 9-11, Bilbao, Spain
  21. Stasiecka A., Płodzień J. and Stemposz E. (2006), *Introducing new elements into the SCORM standard metadata*, Proc. of Web Information Systems and Technologies (WEBIST 2006), April, 11-13, Lisbone, Portugal
  22. Stasiecka A., Płodzień J. and Stemposz E. (2006), *The Classification of e-Learning Resources Metadata: A Proposal*, Proc. of Web Information Systems and Technologies (WEBIST 2006), April, 11-13, Lisbone, Portugal, pp. 332-338, INSTICC Press, Setubal, Portugalia
  23. Stasiecka A., Płodzień J. and Stemposz E. (2006), *Measures for Estimating the Quality of e-learning materials in the didactic aspect*, Proc. of Web Information Systems and Technologies (WEBIST 2006), April, 11-13, Lisbone, Portugal, pp. 204-212, INSTICC Press, Setubal, Portugalia
  24. Stemposz E., Stasiecka A., Płodzień J. (2006), *Identyfikacja zbioru miar dla przyblizonego szacowania jakości zasobów e-learningowych*, Proc. of VI Conference "Virtual University: VU'2006", Warszawa, Poland
  25. Płodzień J., Stemposz E., Stasiecka A. (2006), *An approach to the quality and reusability of metadata specification for e-learning objects*, The international journal of digital information research and use (OIR), Bradford, United Kingdom, Vol. 30 Number 3, pp. 238-251

26. Stasiecka A., Płodzień J. and Stemposz E. (2005), *New Categories for Learning Object Metadata*, Proc. of the 1st on-line Conference on Metadata and Semantics Research (MTSR'05), LINK: <http://www.metadata-semantics.org/>
27. Stasiecka A., Płodzień J. and Stemposz E. (2005), *Towards Improving the Quality of E-learning Resources*, Proc. of the 1st on-line Conference on Metadata and Semantics Research (MTSR'05), LINK: <http://www.metadata-semantics.org/>
28. Stasiecka A., Stemposz E. and Dąbrowski W. (2005), *Didactic Aspects Influence on Quality of e-Learning Resources*, Proc. of the WSEAS International Conference on Circuits, Systems, Communications and Computers, Athens, Greece, WSEAS Transactions on Information Science and Applications, Issue 7, Volume 2, ISSN 1790-0832, pp. 1002-1008
29. Stasiecka A., Bednaruk W., Dąbrowski W. and Stemposz E. (2004), *Didactic Aspects of e-Learning Resources*, Proc. of the International Conference on Interactive Computer-Aided Learning, Villach, Austria, (CD-ROM)
30. Stasiecka A., Bednaruk W., Dąbrowski W., Stemposz E. (2004), *Budowa materiałów e-learningowych z uwzględnieniem aspektów dydaktycznych*, IV Konferencja i Warsztaty Politechniki Warszawskiej "Uniwersytet Wirtualny: model, narzędzia i praktyka", Warszawa, 3-5 czerwca 2004
31. Stasiecka A., Dąbrowski W., Płodzień J., Stemposz E. (2003), *Systemowe wsparcie zdalnej edukacji*, materiały V Krajowej Konferencji Inżynierii Oprogramowania, WNT, Szklarska Poręba, str. 675-684, październik 2003
32. Stasiecka A., Płodzień J., Stemposz E. (2003), *ObAn – an Application Supporting e-Learning*, materiały konferencji: The WSEAS International Conference on Applied Mathematics, Malta, WSEAS Transactions on Computers, Issue 2, Volume 2, ISSN 1109-2750, str. 305-310
33. Habela P., Płodzień J., Stasiecka A., Stemposz E. (2002), *Propozycja metodyki nauczania inżynierii oprogramowania w oparciu o podejście obiektywne*, IV Krajowa Konferencja Inżynierii Oprogramowania KKIO 2002, Tarnowo Podgórne k.Poznania, KKIO 2002
34. Płodzień J., Stasiecka A., Stemposz E. (2002), *Telepraca: aspekty organizacyjne procesu dydaktycznego w nauczaniu na odległość*, materiały V Konferencji z cyklu: Problemy Spoleczeństwa Globalnej Informacji wydane jako monografia, 142-147, Szczecin, Polska
35. [www.uml.org](http://www.uml.org) (październik 2013)
36. <http://www.omg.org/spec/UML/2.4.1/> (październik 2013)
37. Trzaska M. (2008), *Modelowanie i implementacja systemów informatycznych*, Wydawnictwo PWSTK

38. Dąbrowski W., Stasiak A., Wolski M. (2007), *Modelowanie systemów informatycznych w języku UML 2.1*, Mikom
39. Kroll P., Kruchten P. (2007), *Rational Unified Process od Strony Teoretycznej/ Rational Unified Process od Strony Praktycznej*, Wydawnictwa Naukowo-Techniczne
40. Maciaszek L. (2007), *Requirements Analysis and Systems Design (3<sup>rd</sup> Edition)*, Pearson Education Canada
41. Płodzień J., Stemposz E. (2005), *Analiza i projektowanie systemów informatycznych*, Wydawnictwo PJWSTK
42. Śmiałek M. (2005), *Zrozumieć UML 2.0, Metody modelowania obiektowego*, Helion
43. Wrycza S., Marcinkowski B., Wyrzykowski K. (2005), *Język UML 2.0 w modelowaniu systemów informatycznych*, Helion
44. Banachowski L., Chądzyńska A., Matejewski K. (2004), *Relacyjne bazy danych. Wykłady i ćwiczenia*, Wydawnictwo PJWSTK
45. Pender T. (2003), *UML Bible*, Wilby Publishing, Inc.
46. Cockburn A. (2001), *Writing Effective Use Cases*, Addison-Wesley
47. Ullman J.D., Widom J. (2001), *Podstawowy wykład z systemów baz danych*, WNT
48. Date C.J. (2000), *Wprowadzenie do systemów baz danych*, Wydawnictwa Naukowo-Techniczne
49. Kruchten P. (1999), *The Rational Unified Process An Introduction*, Addison-Wesley
50. Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorensen W. (1991), *Object-Oriented Modeling and Design*, Englewood Cliffs, New Jersey, Prentice Hall International, Inc.

# ポーランド日本情報工科大学



## POLSKO-JAPONSKA WYZSZA SZKOŁA TECHNIK KOMPUTEROWYCH

Jedna z najlepszych uczelni w Polsce – wyróżniana przez pracodawców, studentów i media. Od początku swojej działalności zajmuje czołowe miejsce w prestiżowych rankingach uczelni wyższych – wielokrotnie zdobywała pierwsze miejsce w rankingach tygodników „Polityka”, „Wprost” i „Newsweek” oraz Perspektyw/Rzeczpospolitej w kategoriach uczelni technicznych, jak i niepublicznych.

PJWSTK jest uczelnią akademicką – Wydział Informatyki posiada uprawnienia do nadawania stopnia doktora oraz doktora habilitowanego w dziedzinie nauk technicznych.

**Uczelnia prowadzi studia na kierunkach:**

**Architektura Wnętrz – Wydział Sztuki Nowych Mediów**  
studia I stopnia

**Grafika – Wydział Sztuki Nowych Mediów**  
studia I i II stopnia oraz magisterskie jednolite

**Informatyka – Wydział Informatyki**  
studia I, II i III stopnia oraz studia podyplomowe

**Kulturoznawstwo – Wydział Kultury Japonii**  
studia I i II stopnia

**Zarządzanie – Wydział Zarządzania Informacją**  
studia I stopnia

**Przy PJWSTK działają także:**

Akademickie Centrum Szkoleniowe  
Akademickie Liceum Ogólnokształcące  
Niepubliczne Liceum Plastyczne

**Główna siedziba znajduje się w samym centrum Warszawy:**

ul. Koszykowa 86  
02-008 Warszawa  
tel.: 22 584 45 00  
[www.pjwstk.edu.pl](http://www.pjwstk.edu.pl)  
e-mail: [pjwstk@pjwstk.edu.pl](mailto:pjwstk@pjwstk.edu.pl)

**Ośrodkowie Bytomiu i w Gdańsku dopełniają oferty edukacyjnej:**

**Wydział Zamiejscowy Informatyki w Bytomiu**  
Aleja Legionów 2  
41-902 Bytom  
tel.: 32 387 16 60  
[www.bytom.pjwstk.edu.pl](http://www.bytom.pjwstk.edu.pl)  
e-mail: [bytom@pjwstk.edu.pl](mailto:bytom@pjwstk.edu.pl)  
kierunki: informatyka, grafika

**Wydział Zamiejscowy Informatyki w Gdańskim**  
**Wydział Zamiejscowy Sztuki Nowych Mediów w Gdańskim**  
ul. Brzegi 55  
80-045 Gdańsk  
tel.: 58 683 59 75  
[www.gdansk.pjwstk.edu.pl](http://www.gdansk.pjwstk.edu.pl)  
e-mail: [gdansk@pjwstk.edu.pl](mailto:gdansk@pjwstk.edu.pl)  
kierunki: informatyka, grafika



02-008 Warszawa, ul. Koszykowa 86  
tel.: 22 58 44 526, fax: 22 58 44 503  
e-mail: [oficyna@pjwstk.edu.pl](mailto:oficyna@pjwstk.edu.pl)  
[www.wydawnictwo.pjwstk.edu.pl](http://www.wydawnictwo.pjwstk.edu.pl)

ISBN 978-83-63103-39-2

Monografie naukowe, tom 12

