

Seminarium Dyplomowe Semestr 7

Zajęcia nr 4

Projektowanie (konceptcja) rozwiązań

Mgr inż. Jerzy Stankiewicz

ZAJĘCIA NR 1 - ROZLICZENIE

- **Przygotować harmonogram prac** z wykorzystaniem MS Project (za okres październik 2018 : 31-03-2019r)
- **Utworzyć dokument pracy dyplomowej** (nazwisko imię v1.docx) ze stroną tytułową, proponowanymi rozdziałami (z wygenerowanym spisem treści)
- **Literatura** (na końcu dokumentu) – przedstawić propozycje literatury z dziedzin:
 - Projektowania systemów informatycznych
 - Modelowania systemów
 - Projektowania baz danych
 - Języków programowania
 - Dziedziny tematycznej pracy dyplomowej (normy prawne, dzienniki ustaw itp.)
 - Strony internetowe (ćwiczenia, opisy, przegląd produktów rynkowych o podobnej tematyce itp.)
- **Opracować rozdział wstępny** w zakresie: temat pracy, cel i zakres pracy, wprowadzenie do problemu (ogólne)

- ***Opracować część analityczną pracy dyplomowej w zakresie:***
 - *szczegółowy opis problemu*

Opracować część analityczną pracy dyplomowej w zakresie:

- Wymagania funkcjonalne systemu*
- Wymagania pozafunkcjonalne systemu*
- Użytkownicy systemu i dostępne im funkcje*

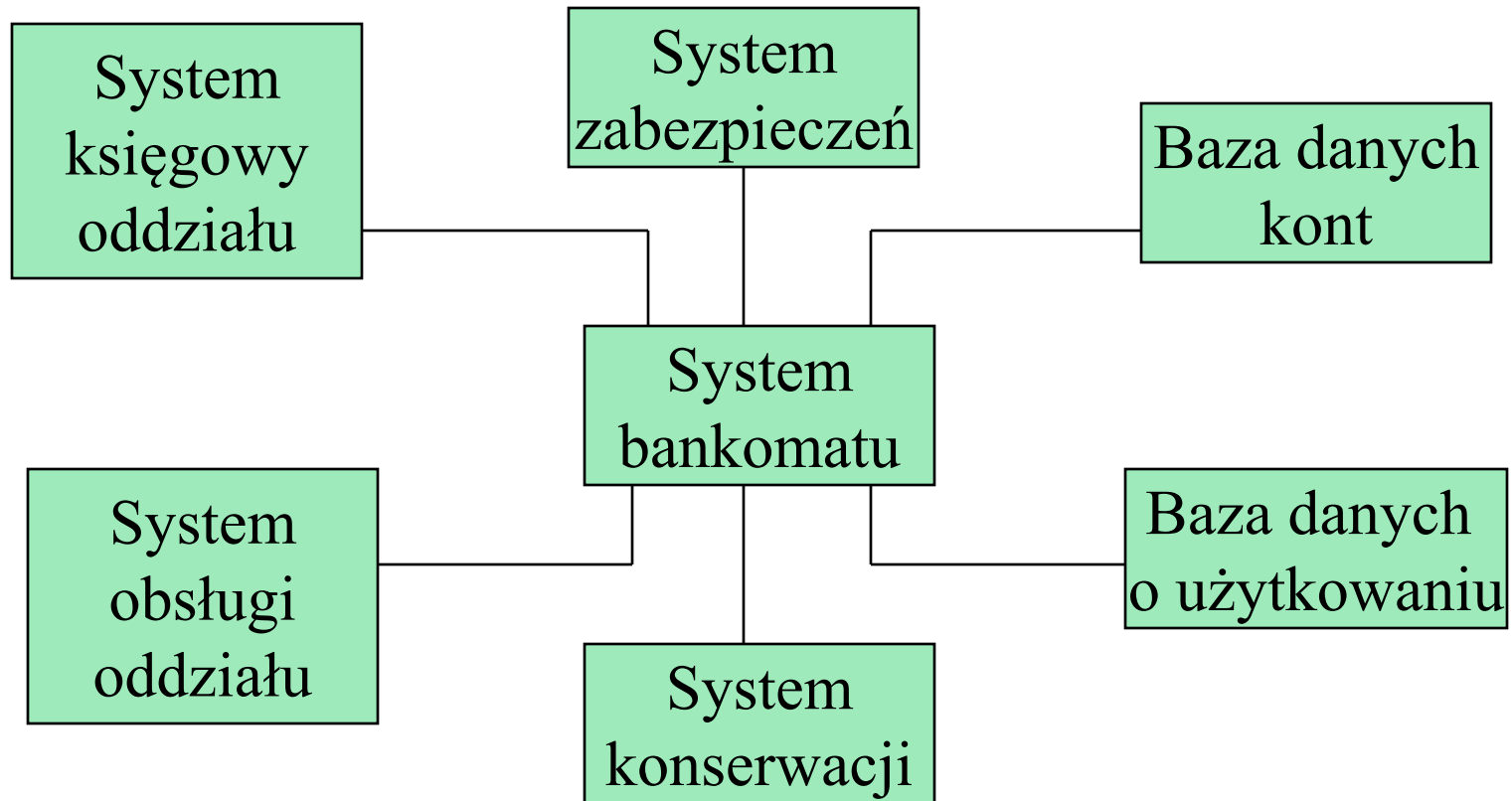
Zakres tematyczny

- Modele kontekstowe
- Techniki strukturalne /ogólnie/
- Techniki obiektowe

Modele kontekstowe

- Prezentują operacyjny kontekst: system-otoczenie
- We wczesnej fazie procesu określania i analizowania wymagań należy ustalić granice systemu
- Kwestie społeczne i organizacyjne mogą mieć wpływ na ustalenia:
 - co do systemu należy?
 - a co jest poza jego granicami?
- W niektórych wypadkach granica między systemem, a jego środowiskiem jest dość czytelna
- Granicę między systemem a jego środowiskiem należy ustalić w trakcie procesu inżynierii wymagań

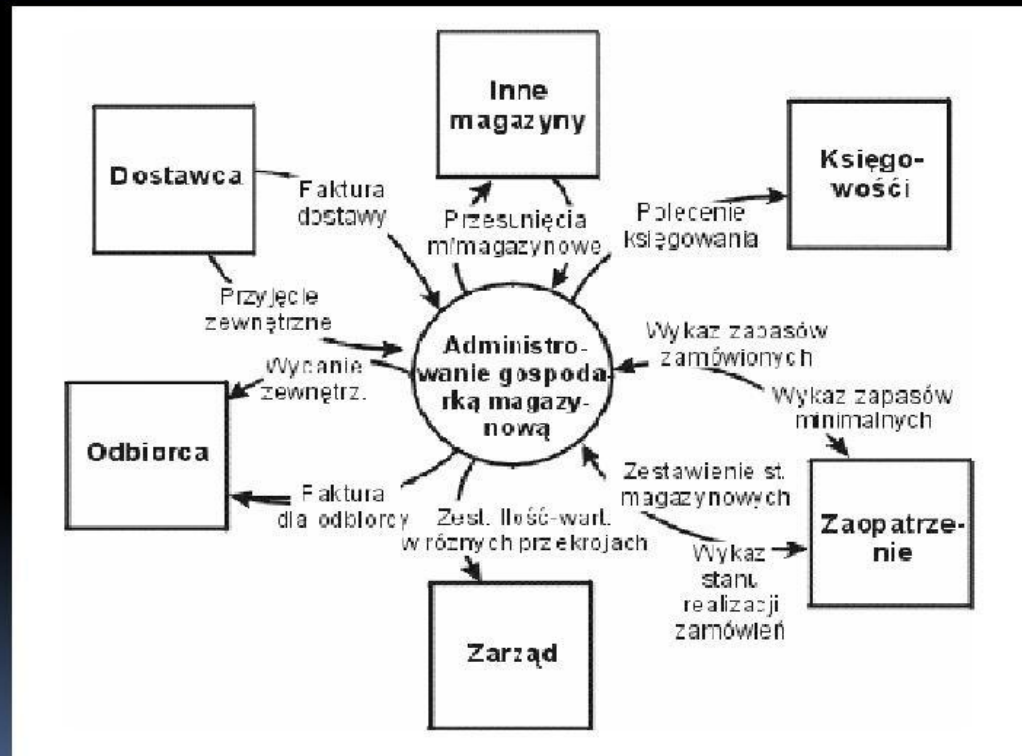
Kontekst Systemu Bankomatu



Kontekst Administrowanie Gospodarką Magazynową

DFD – Przykład

Diagram kontekstowy



Przeznaczenie

- Diagramy kontekstowe mają zastosowanie dla:
 - Projektantów
 - Programistów
 - Użytkowników systemu

Modelowanie strukturalne

Diagram Hierarchii Funkcji
Diagram Przepływu danych
Diagram Związków Encji

Modele obiektowe

Modele obiektowe

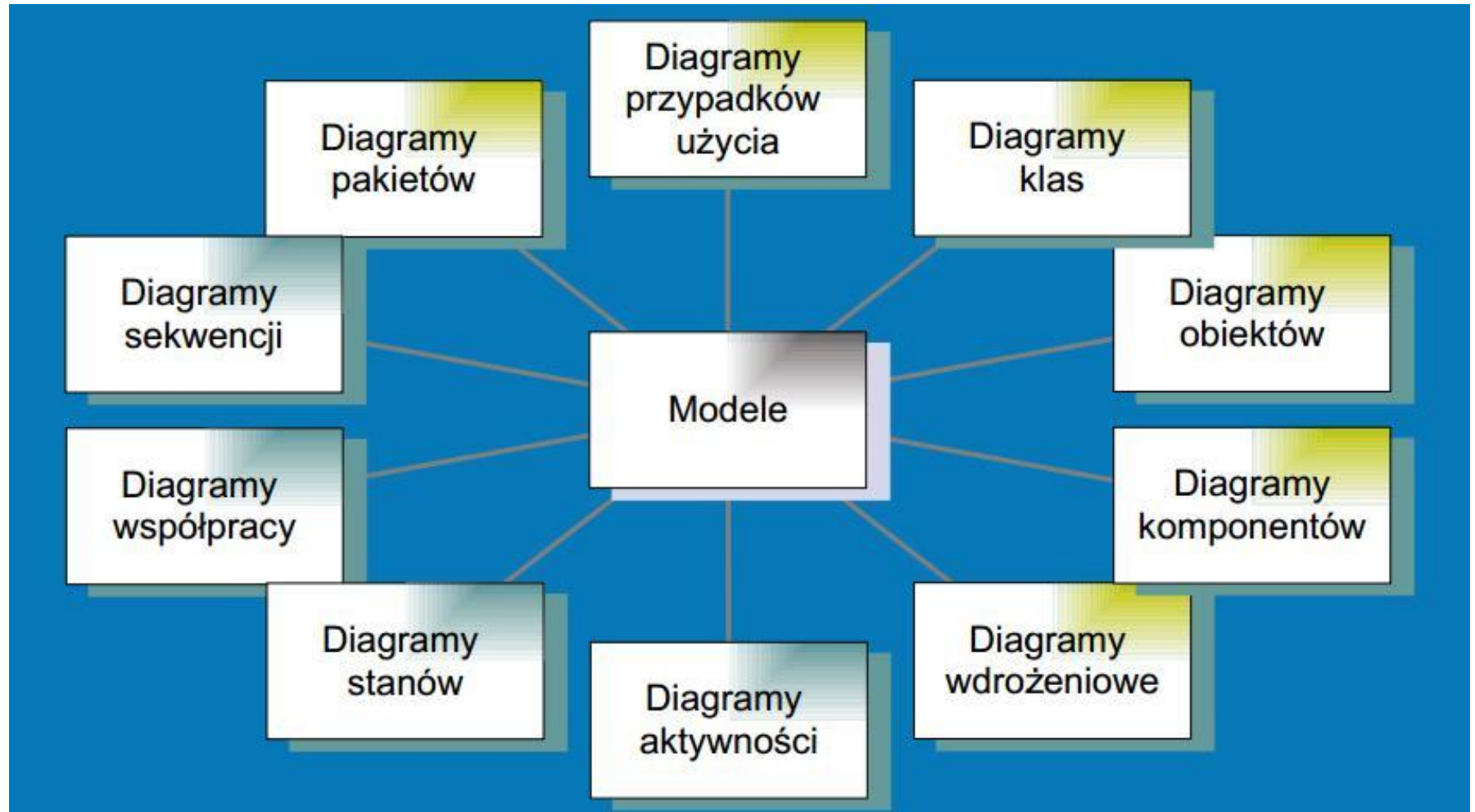
- **Naturalnie odzwierciedlają elementy rzeczywistości, którymi manipuluje system.**
- **Opisują system w terminach klas obiektów i relacji pomiędzy nimi.**
- **Obiekty mogą być rzeczywiste lub abstrakcyjne.**
- **Identyfikacja klas obiektów jest uważana za trudny proces (wymaga głębokie znajomości dziedziny aplikacji).**
- **Klasy opisujące obiekty z dziedziny problemowej mają duży potencjał ponownego użycia.**

- UML cieszy się aktualnie bardzo dużą popularnością
- Prawdopodobnie przez wiele najbliższych lat będzie dominował w obszarze analizy i projektowania
- UML nie jest metodyką projektowania
- Notacja UML, która opiera się o podstawowe pojęcia obiektowości może być wykorzystana w dowolnej metodyce
- Pojęcia UML, wynikające z doświadczenia jej twórców, mają w założeniu przykrywać większość istotnych aspektów modelowanych systemów

UML jest określany jako język modelowania z 4+1 perspektywą :

- **Perspektywa przypadków użycia** – opisuje funkcjonalność systemu widzianą przez użytkowników
- **Perspektywa logiczna** – sposób realizacji funkcjonalności, struktura systemu widziana przez projektanta
- **Perspektywa implementacyjna** – zawiera moduły i interfejsy, przeznaczona dla programisty
- **Perspektywa procesowa** – podział systemu na czynności i jednostki wykonawcze (wątki, procesy, współbieżność) – służy głównie programistom i instalatorom
- **Perspektywa wdrożeniowa** – fizyczny podział elementów systemu i ich rozmieszczenie w infrastrukturze, ważna dla instalatorów

Diagramy definiowane w UML



Model a diagram ?

- **Model** - pewna abstrakcja projektowanego systemu, widziana z określonej perspektywy, na określonym poziomie szczegółowości.
- **Diagram** - środek służący do opisu modelu. Dany model może być opisany przy pomocy wielu diagramów. Dany element modelu może pojawiać się na wielu diagramach jednego modelu.
- Dwa najważniejsze modele w UML, wykorzystywane w fazie analizy (modelowania), to:
 - **model przypadków użycia** opisujący system widziany z perspektywy jego przyszłego użytkownika (za pomocą diagramów przypadków użycia),
 - **model obiektowy** przedstawiający statyczną budowę, czyli strukturę systemu (za pomocą **diagramów klas** i **diagramów obiektów**). Diagram klas może zawierać obiekty. Diagram obiektów nie zawiera klas, ale wyłącznie obiekty.

Model a diagram ?

- **Notacja:**
 - Klasy obiektów reprezentowane są jako prostokąty Opis podzielony jest na trzy części:
 - Nazwa
 - Atrybuty
 - Operacje
 - Powiązania pomiędzy klasami (zwane *asocjacjami*) reprezentowane są przez linie łączące klasy
 - Dziedziczenie, określane mianem związku generalizacji modeluje związek „jest rodzaju”
 - Agregacja modelująca zależności typu całość - część

Diagram przypadków użycia

Diagram przypadków użycia (ang. *use-case diagram*) służy do modelowania aktorów (użytkowników systemu, odbiorców efektów pracy systemu, systemów zewnętrznych) i ich potrzeb w stosunku do tworzonego systemu.

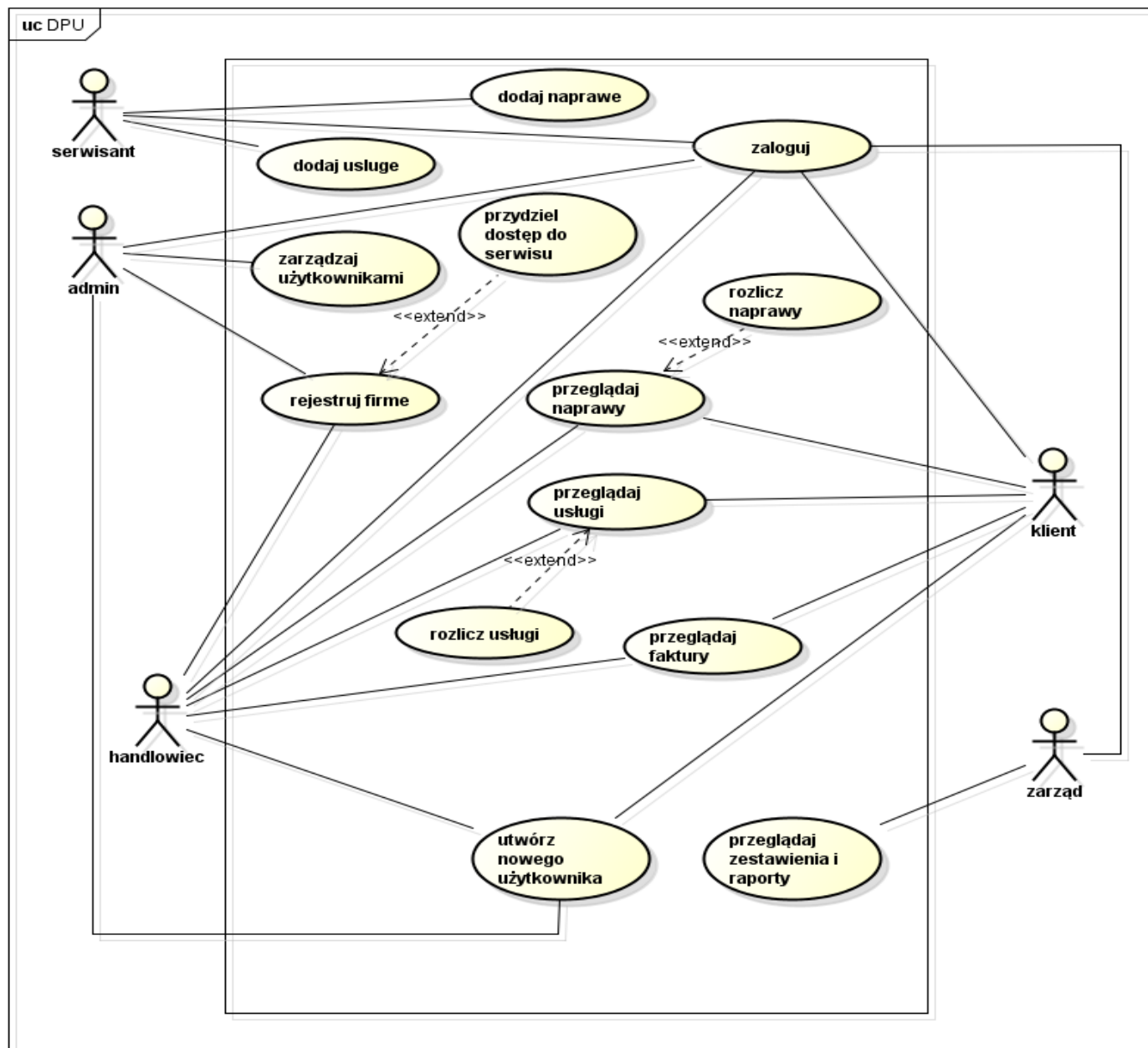
Przypadki użycia prezentowane na diagramie to sekwencje czynności, które prowadzą do spełnienia celu przypadku użycia (zaspokojenia pewnej potrzeby użytkownika).

- definiuje granice modelowanego systemu
- określa jego kontekst
- wymienia użytkowników systemu i jednostki zewnętrzne
- przedstawia funkcje dostępne dla użytkowników
- określa powiązania i zależności pomiędzy nimi

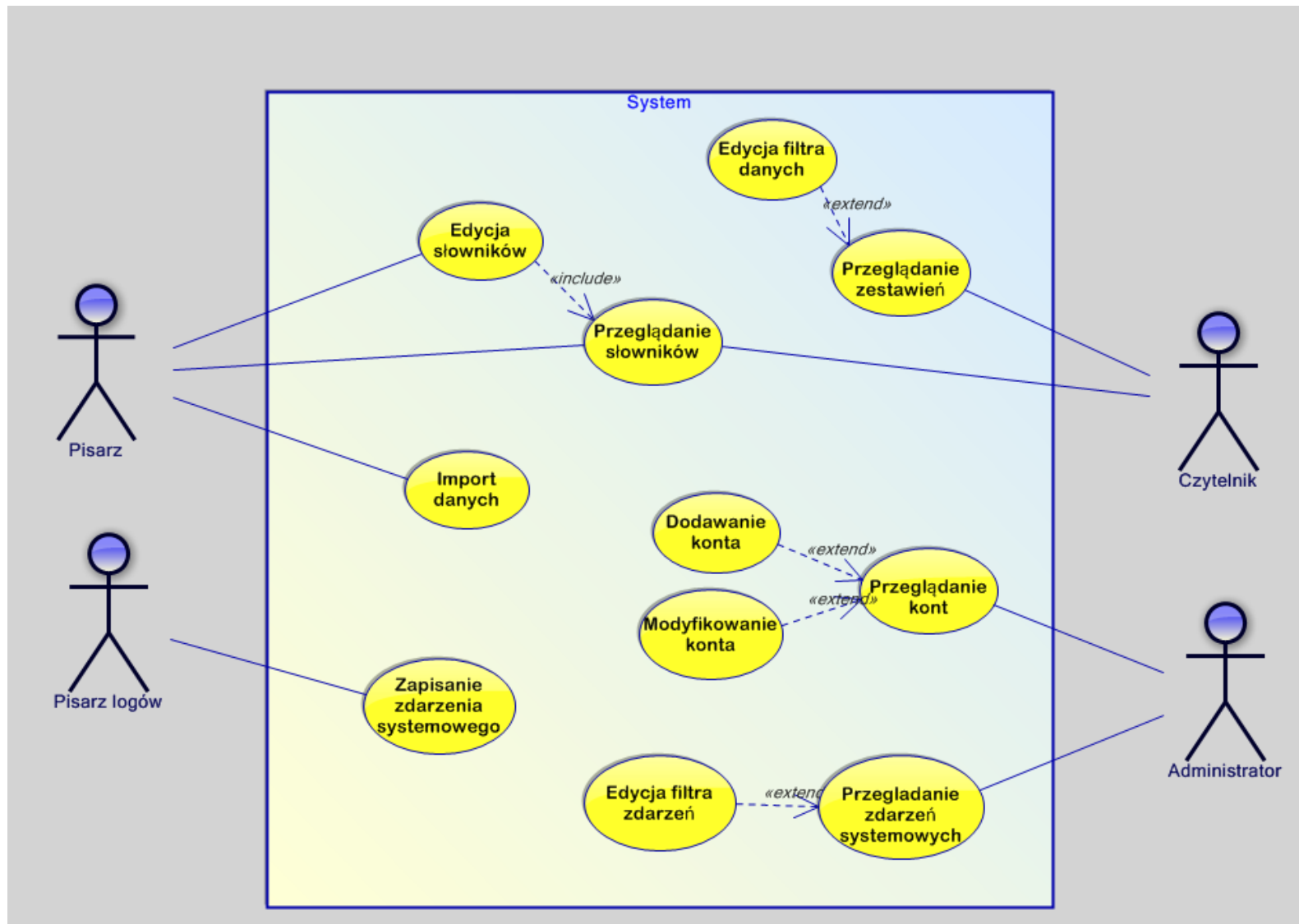
Przypadki użycia komunikują się z aktorami poprzez powiązania, pokazujące, który aktor ma dostęp do podanego przypadku użycia.

Ponadto mogą być powiązane pomiędzy sobą: *relacją uogólnienia*, *rozszerzenia* i *zawierania*.

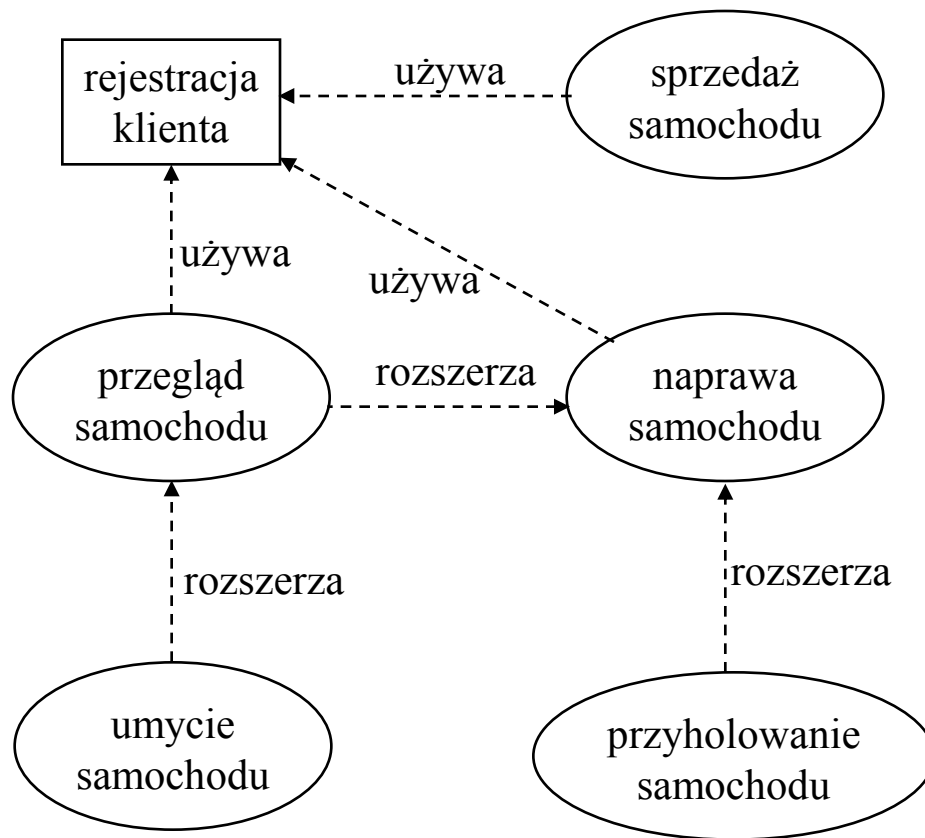
Przykład 1: Diagram przypadków użycia



Przykład 2: Diagram przypadków użycia



Przypadki użycia: Powiązania rozszerza i używa



Używa/zawieranie (include):

- wskazuje na wspólny fragment wielu przypadków użycia, wyłączony “przed nawias” (w tym sensie jest “abstrakcyjny”, podobieństwo do dziedziczenia)
- Wykonanie jednego przypadku użycia przez drugi

Rozszerza (extend): strzałka prowadzi od przypadku użycia, który czasami (nie zawsze) rozszerza przypadki użycia.
/Dodatkowa funkcjonalność przypadku użycia/

Opis przypadków użycia /szablon/

Nazwa:	Pełna nazwa przypadku użycia (PU)
Numer:	Numer identyfikacyjny przypadku użycia
Twórca:	Dane twórcy, np.: imię, nazwisko, stanowisko
Poziom ważności:	Poziom ważności z pkt. widzenia użytkownika, np. niski, średni, wysoki
Typ przypadku użycia:	Określenie typu PU z pkt. widzenia jego złożoności, oraz zaspokojenia potrzeb użytkownika, np.ogólny/szczegółowy; niezbędny/istotny/moło istotny
Aktorzy:	Lista aktorów będących w związku z PU
Krótki opis:	Ogólna charakterystyka PU
Warunki początkowe:	Ch-ka koniecznych warunków inicjujących przypadek
Warunki końcowe:	Ch-ka stanu systemu po realizacji PU
Główny przepływ zdarzeń:	Wypunktowana i scharakteryzowana lista przepływów zdarzeń (scenariusz główny)
Alternatywne przepływy zdarzeń:	Scenariusz alternatywny
Specjalne wymagania:	Wypunktowana i scharakteryzowana lista dodatkowych zidentyfikowanych wymagań нефункциональных
Notatki i kwestie:	Lista wszystkich komentarzy i otwartych kwestii

Przykład 1: Opis przypadków użycia

Nazwa przypadku użycia
Przeglądanie kont
Aktorzy
Administrator
Opis
Przypadek użycia dotyczy przeglądania przez administratora systemu listy kont użytkowników oraz ich właściwości.
Warunki początkowe
<ol style="list-style-type: none">1. Na ekranie wyświetlony jest graficzny interfejs użytkownika2. System czeka na aktywność użytkownika3. Użytkownik jest zalogowany w systemie z uprawnieniami administratora
Zdarzenie inicjujące
Użytkownik inicjuje akcję poprzez żądanie za pośrednictwem graficznego interfejsu użytkownika.
Podstawowa sekwencja akcji
<ol style="list-style-type: none">1. Na ekranie zostaje wyświetlona lista kont użytkowników zawierająca następujące składniki:<ul style="list-style-type: none">• Login• Hasło• Imię• Nazwisko• Data utworzenia konta• Termin wygaśnięcia konta• Informacja czy użytkownik posiada prawa administratoraPonadto wyświetlony zostaje przycisk służący do zamknięcia listy.2. Jeżeli administrator wskaże jeden z wyświetlonych składników, system sortuje listę na podstawie wskazanego składnika i następuje powrót do punktu 1.3. Jeżeli administrator wciśnie przycisk, lista zostaje zamknięta i przypadek użycia dobiega końca.
Alternatywna sekwencja akcji
Brak.
Warunki końcowe
Brak.

Przykład 2: Opis przypadków użycia

Nazwa przypadku użycia: Przeglądanie zdarzeń systemowych
Aktorzy: Administrator
Opis: Przypadek użycia dotyczy przeglądania przez administratora listy sesji oraz zdarzeń zapisanych w systemie. Przypadek jest rozszerzony przez przypadek użycia „Edycja filtra zdarzeń”.
Warunki początkowe
1.Na ekranie wyświetlony jest graficzny interfejs użytkownika 2.System czeka na aktywność użytkownika 3.Użytkownik jest zalogowany w systemie z uprawnieniami administratora 4.Użytkownik klika w menu głównym aplikacji pozycję „Panel administracyjny”
Podstawowa sekwencja akcji
<div>1. Na ekranie zostają wyświetlone:<ul style="list-style-type: none">• Tabela sesji - zawierająca następujące składniki:<ul style="list-style-type: none">• Nr sesji• Data i czas rozpoczęcia sesji• Data i czas zakończenia sesji• Adres IP komputera użytkownika• Nazwa aplikacji użytkownika• Login użytkownika• Imię użytkownika• Nazwisko użytkownika• Tabela zdarzeń - zawierająca poniższe składniki:<ul style="list-style-type: none">• Czas zdarzenia• Nazwa rodzaju zdarzenia• Opis zdarzenia• Pole tekstowe• Filtr danych - zawierający pola pozwalające zdefiniować zakres informacji prezentowanych w tabelach:<ul style="list-style-type: none">• Rodzaj prezentowanych sesji (otwarte, zamknięte lub wszystkie)• Adres IP• Nazwa aplikacji• Login• Imię• Nazwisko• Rodzaj zdarzenia• Jeżeli administrator zaznaczy wiersz w tabeli sesji, system aktualizuje dane prezentowane w tabeli zdarzeń ograniczając zakres prezentowanych danych do zdarzeń związanych z zaznaczoną sesją.• Jeżeli administrator zaznaczy wiersz w tabeli zdarzeń, system w celu ułatwienia odczytu wyświetla w polu tekstowym opis związany z zaznaczonym zdarzeniem.• Przypadek dobiega końca w sytuacji zamknięcia okna przez administratora.</div>
Alternatywna sekwencja akcji: Brak
Warunki końcowe: brak

Przykład 3: Opis przypadków użycia

Nazwa przypadku użycia
Edycja danych profilu użytkownika
Opis
Przypadek użycia opisuje funkcję modyfikacji danych profilu użytkownika
Aktorzy – użytkownicy
Użytkownik Uprawniony, Administrator
Warunki początkowe
Brak.
Sekwencja akcji – edytuj dane profilu użytkownika
1. Aktor wprowadza dane do profilu: <ul style="list-style-type: none">• Imię• Nazwisko• E-mail• Hasło• Potwierdzenie hasła 2. Aktor wybiera przycisk „Zapisz”.
Warunki końcowe

Przykład 4: Opis przypadków użycia

Nazwa przypadku użycia
Usunięcie placówki
Opis:
Przypadek użycia opisuje funkcję usunięcia placówki szkolnej z systemu.
Aktorzy – użytkownicy
Administrator
Warunki początkowe:
Placówka musi istnieć w systemie.
Sekwencja akcji – Dodaj klienta:
1. Aktor wybiera placówkę do usunięcia. 2. Aktor wybiera przycisk „Usuń”.
Warunki końcowe:
Placówka zostaje usunięta z systemu.

Diagram klas

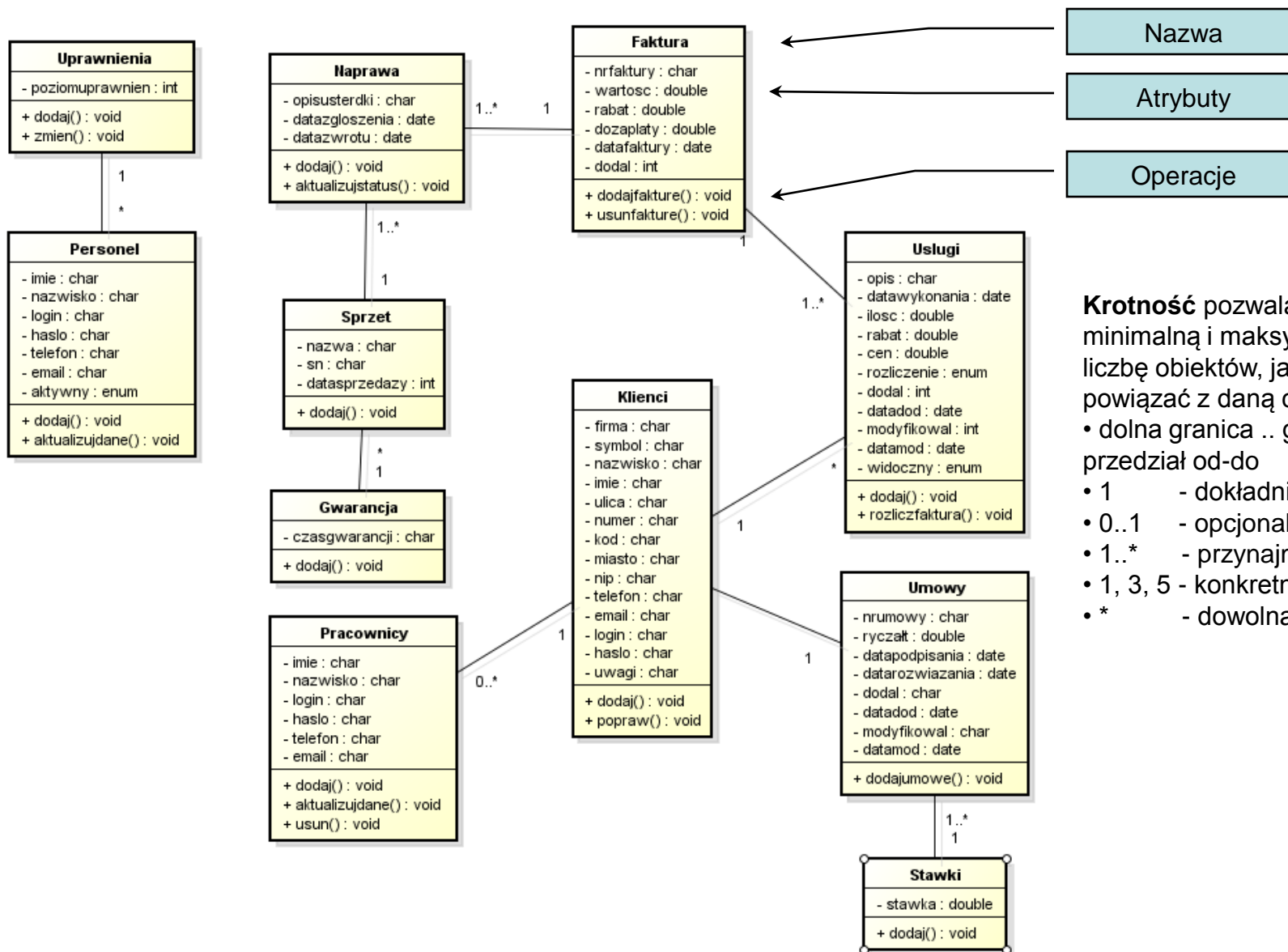
- **Diagram klas** (ang. *class diagram*) jest najczęściej używanym diagramem UML
- Z reguły zawiera także największą ilość informacji i stosuje największą liczbę symboli
- Przedstawia klasy występujące w systemie i statyczne relacje pomiędzy nimi wraz z ograniczeniami
- Jest podstawowym diagramem struktury logicznej systemu
- Na diagramie są prezentowane klasy, ich atrybuty i operacje, oraz powiązania między klasami.
- Diagram klas przedstawia więc podział odpowiedzialności pomiędzy klasy systemu i rodzaj wymienianych pomiędzy nimi komunikatów.
- Z uwagi na rodzaj i ilość zawartych na tym diagramie danych jest on najczęściej stosowany do generowania kodu na podstawie modelu

Klasa jest reprezentowana przez prostokąt z wydzielonymi przedziałami: nazwą, atrybutami i operacjami.

Cechy klasy reprezentują informację, jaką klasa przechowuje. Mogą zostać zapisane w postaci dwóch, w zasadzie równoważnych notacji: jako atrybuty klasy (umieszczane w przedziale atrybutów) lub jako relacje pomiędzy klasami (zapisywane w postaci linii łączącej klasy). Zwykle pierwsza notacja jest stosowana do typów prostych lub obiektów reprezentujących wartości, natomiast druga do typów złożonych.

Operacje reprezentują usługi, jakie klasa oferuje. Ich realizacje – metody – dostarczają implementacji tych usług.

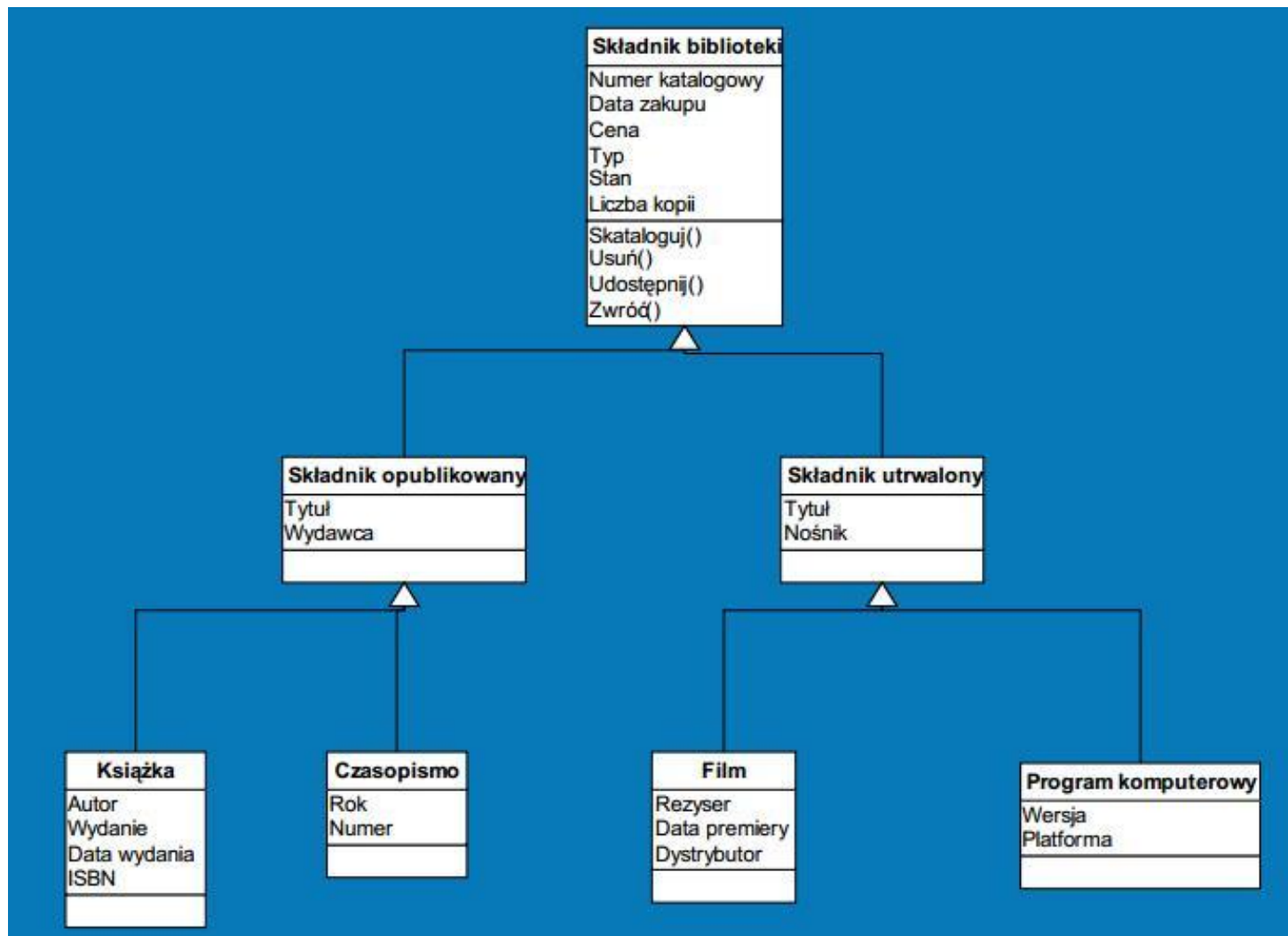
Diagram klas



- Krotność** pozwala określić minimalną i maksymalną liczbę obiektów, jakie można powiązać z daną cechą:
- dolna granica .. górna granica – przedział od-do
 - 1 - dokładnie jeden obiekt
 - 0..1 - opcjonalnie jeden obiekt
 - 1..* - przynajmniej jeden obiekt
 - 1, 3, 5 - konkretne liczby obiektów
 - * - dowolna liczba obiektów

Hierarchia klas w systemie biblioteka

Uogólnienie tworzy hierarchię klas, od ogólnych do bardziej szczegółowych. Pozwala wyłączyć części wspólne klas.



Hierarchia klas użytkowników w systemie biblioteka

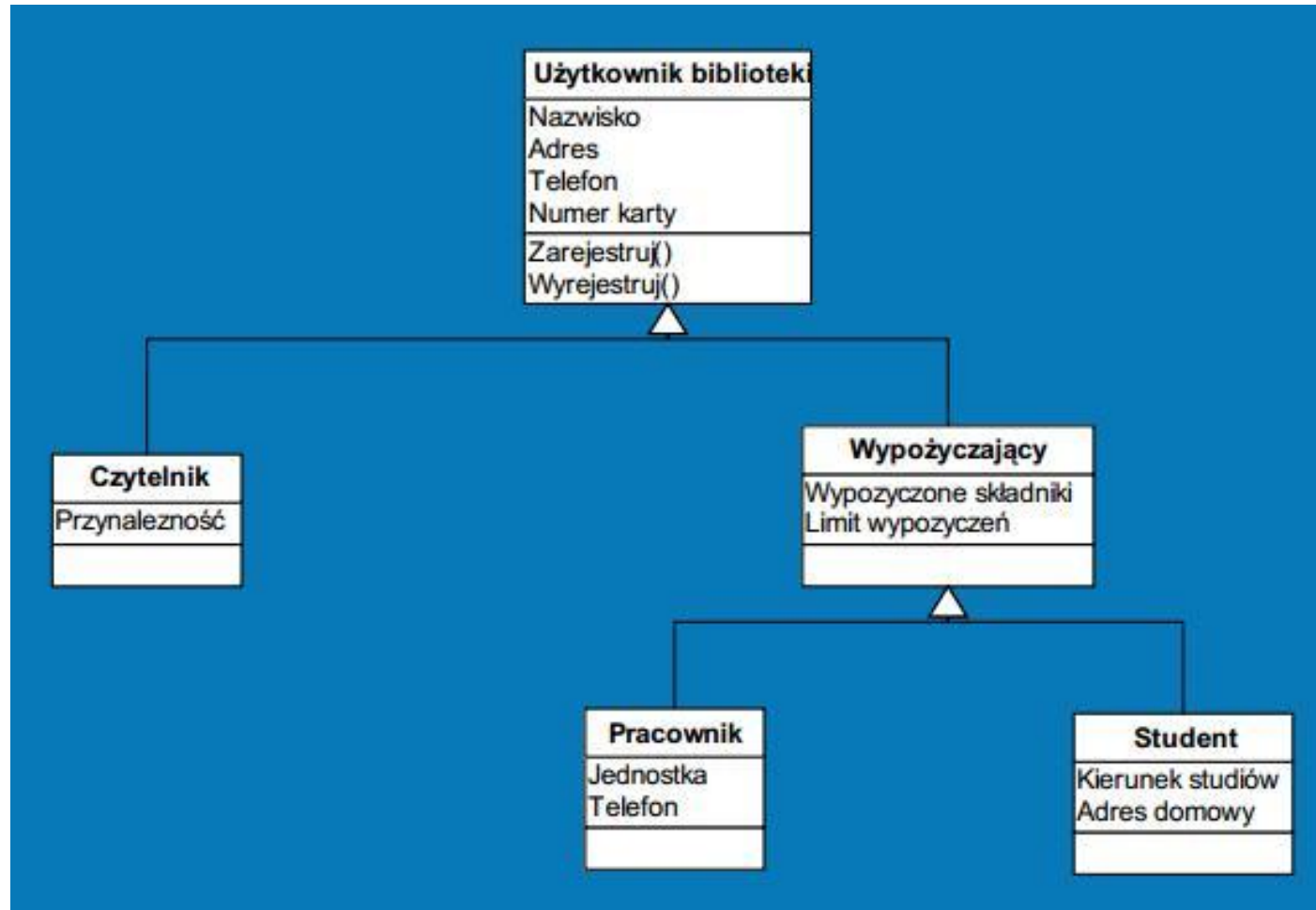
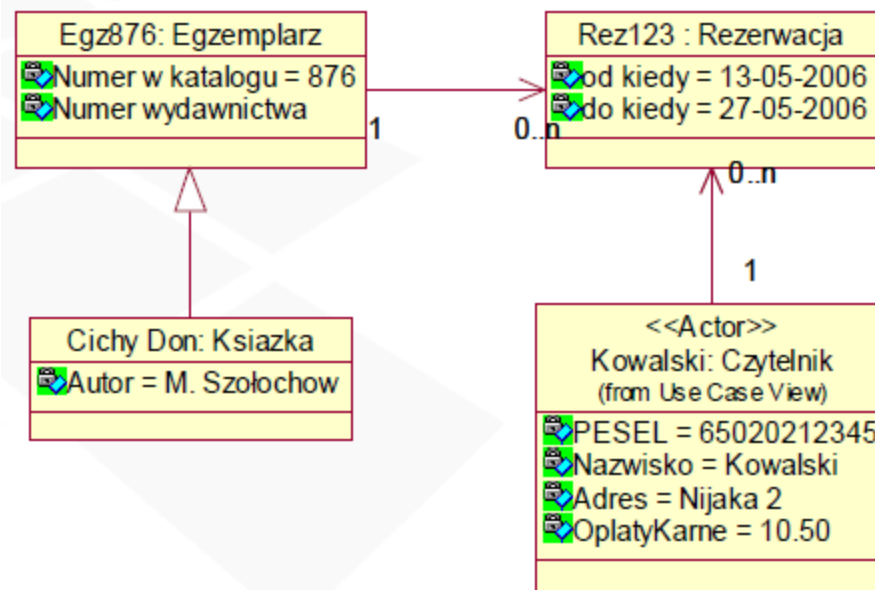


Diagram obiektów

- **Diagram obiektów** (ang. object diagram) prezentuje możliwą konfigurację obiektów w określonym momencie, jest pewnego rodzaju instancją diagramu klas, w której zamiast klas przedstawiono ich obiekty.
- Diagram ten posługuje się identycznymi symbolami co diagram klas, jednak, dla odróżnienia obiektów od klas, nazwy instancji są podkreślone.
- Ponadto, nazwa składa się z nazwy obiektu i nazwy klasy, oddzielonych dwukropkiem.
- Obie części nazwy można pominąć, więc aby uniknąć nieporozumień, jedna część nazwy oznacza nazwę obiektu, a sama nazwa klasy musi być zawsze poprzedzona dwukropkiem.
- Diagramy obiektów przydają się w przypadku szczególnie skomplikowanych zależności, których nie można przedstawić na diagramie klas. Wówczas przykładowe konfiguracje obiektów pomagają w zrozumieniu modelu.



- Głównym zadaniem pomocniczego **modelu dynamicznego (zachowania)** w UML jest wypełnienie diagramu klas metodami wynikającymi z analizy zachowania systemu w trakcie wykonywania zadań, gdzie zadaniem może być np. realizacja **przypadku użycia** czy też jednego konkretnego scenariusza danego przypadku użycia.
- W UML do modelowania zachowania wykorzystuje się głównie **diagramy sekwencji i współpracy** wspomagane **diagramami stanów**.

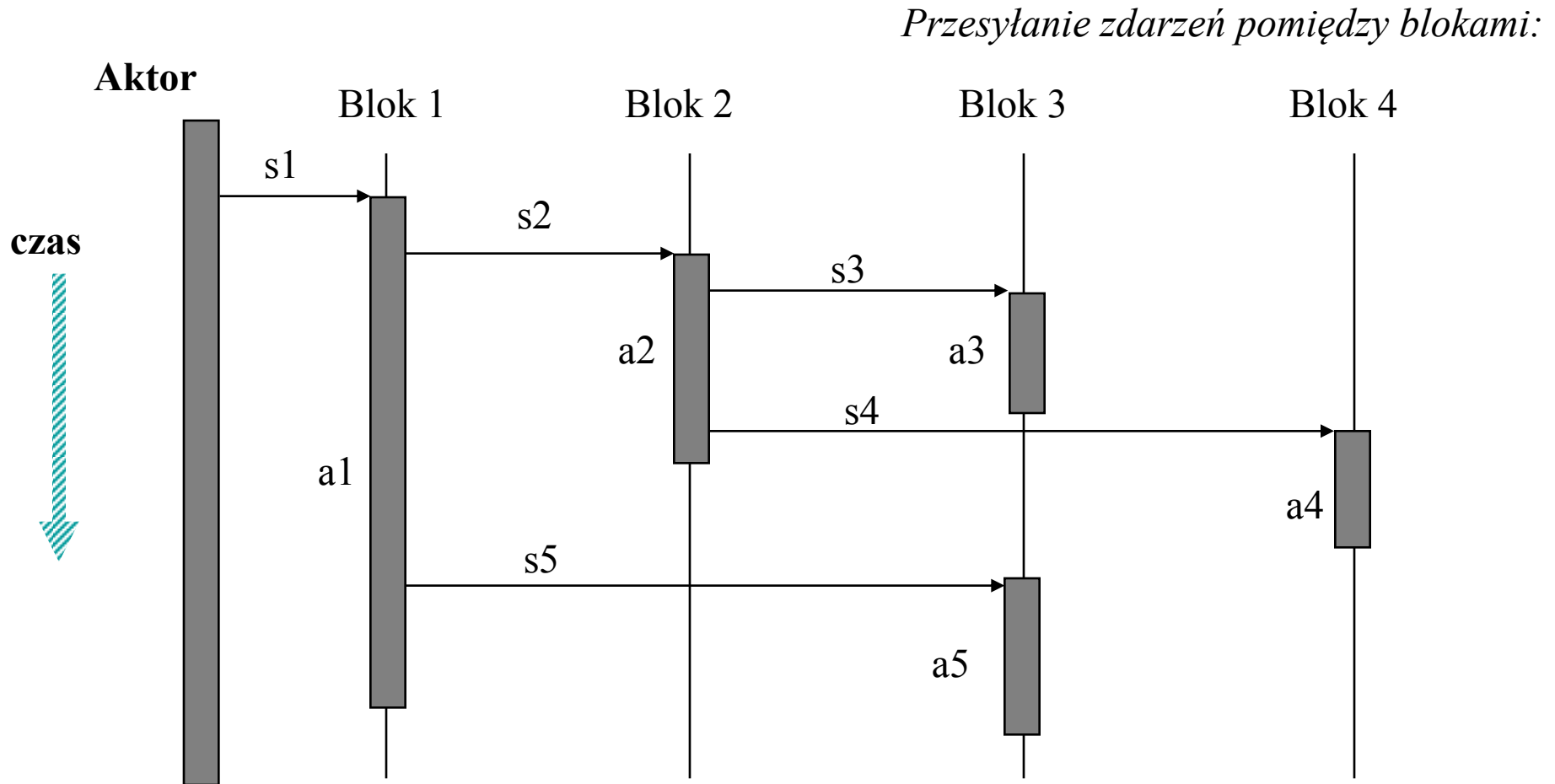
Diagram sekwencji

Diagramy sekwencji (ang. *sequence diagrams*) intuicyjnie prezentują kolejność wywołań operacji, przepływ sterowania pomiędzy obiektami oraz szablon realizowanego algorytmu. Pomijają natomiast całkowicie aspekt dostępu i operacji na danych, związany z komunikacją.

- Uczestnikami diagramów sekwencji są obiekty, opisane nazwą obiektu i jego klasą, które wymieniają między sobą komunikaty.
- Diagram sekwencji jest zapisany w prostokącie oznaczonym operatorem sd (od angielskiej nazwy diagramu) i składa się z pionowych linii życia (ang. *lifelines*) poszczególnych obiektów uczestniczących w interakcji oraz wymienianych między nimi komunikatów (wywołań operacji).
- Białe prostokąty umieszczone na linii życia obiektu oznaczają, że obiekt jest zajęty wykonywaniem pewnej czynności (natomiast nie mają bezpośredniego związku z istnieniem obiektu).
- Czas jest reprezentowany w postaci pionowej osi diagramu.
- Linia życia obiektu to czas, w którym konkretna instancja obiektu jest w stanie przyjmować komunikaty i wysyłać je. Innymi słowy, obejmuje ona czas istnienia obiektu w systemie.

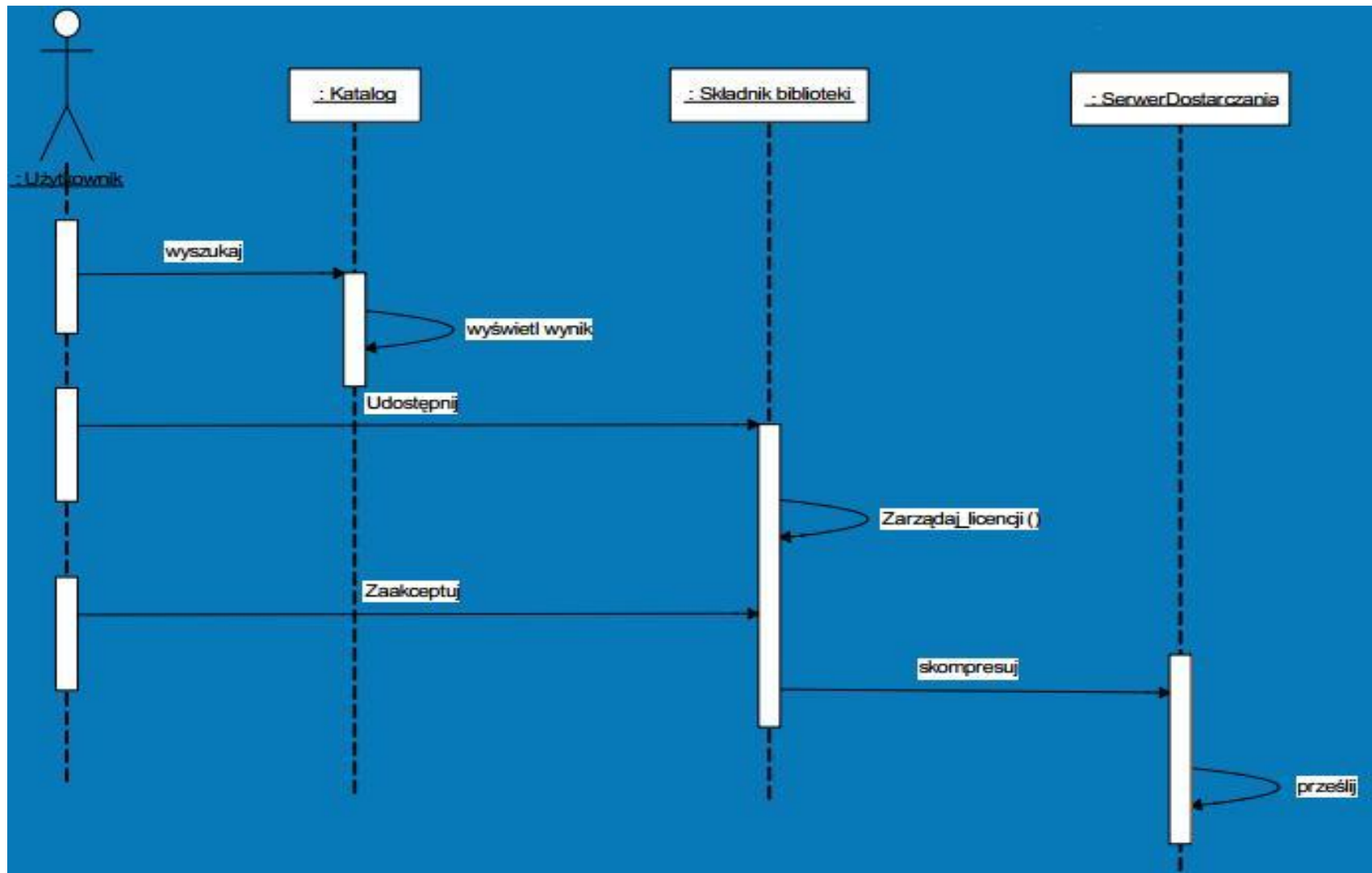
Przypadek użycia w postaci diagramu sekwencji

Diagram sekwencji przedstawia sposób wymiany komunikatów pomiędzy obiektami z zachowaniem ich kolejności

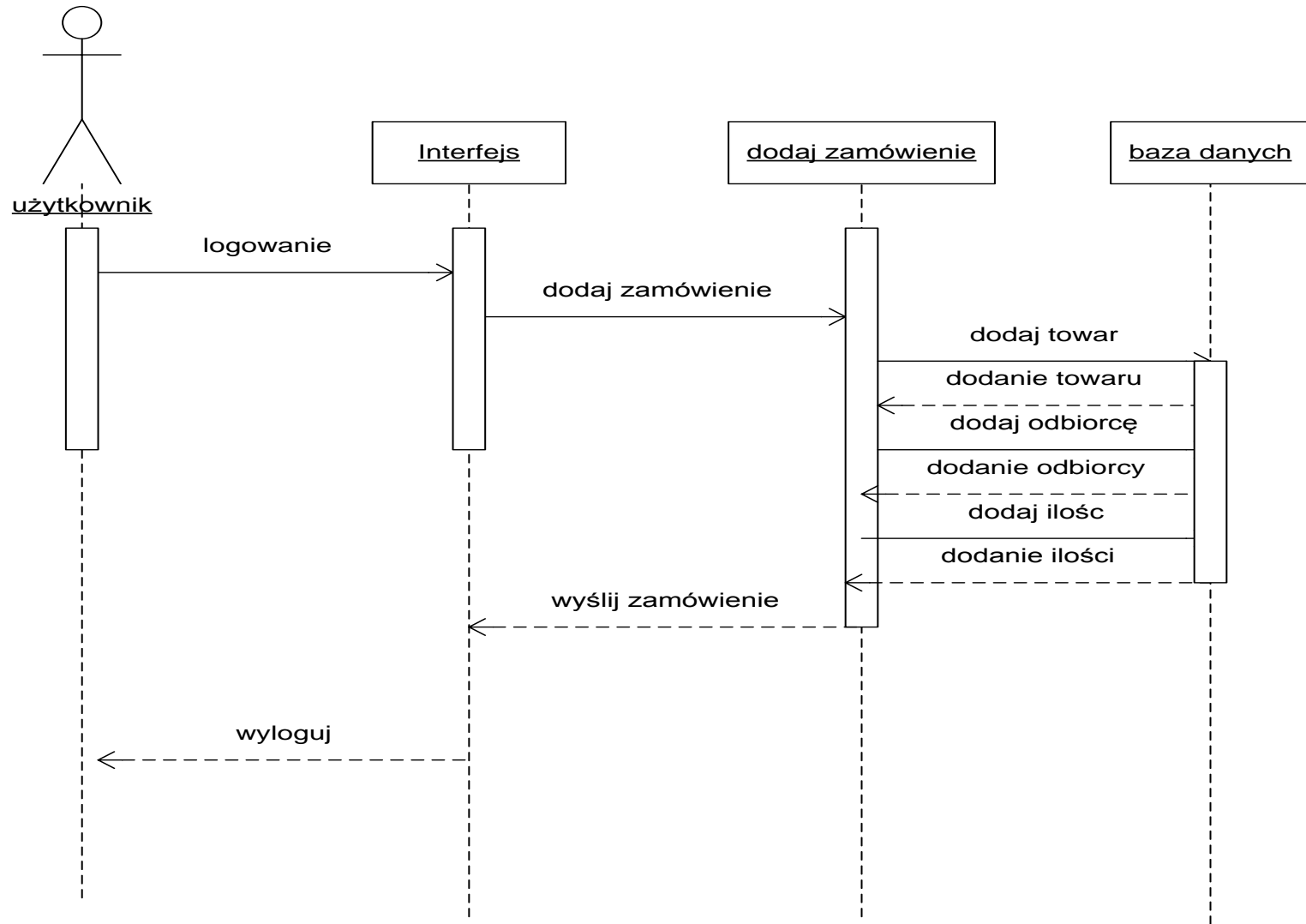


Blok: obiekt + pewna akcja podejmowana przez system s_i - zdarzenia

Pobranie oprogramowania z biblioteki



Zamówienie w restauracji



Przykład diagramu sekwencji

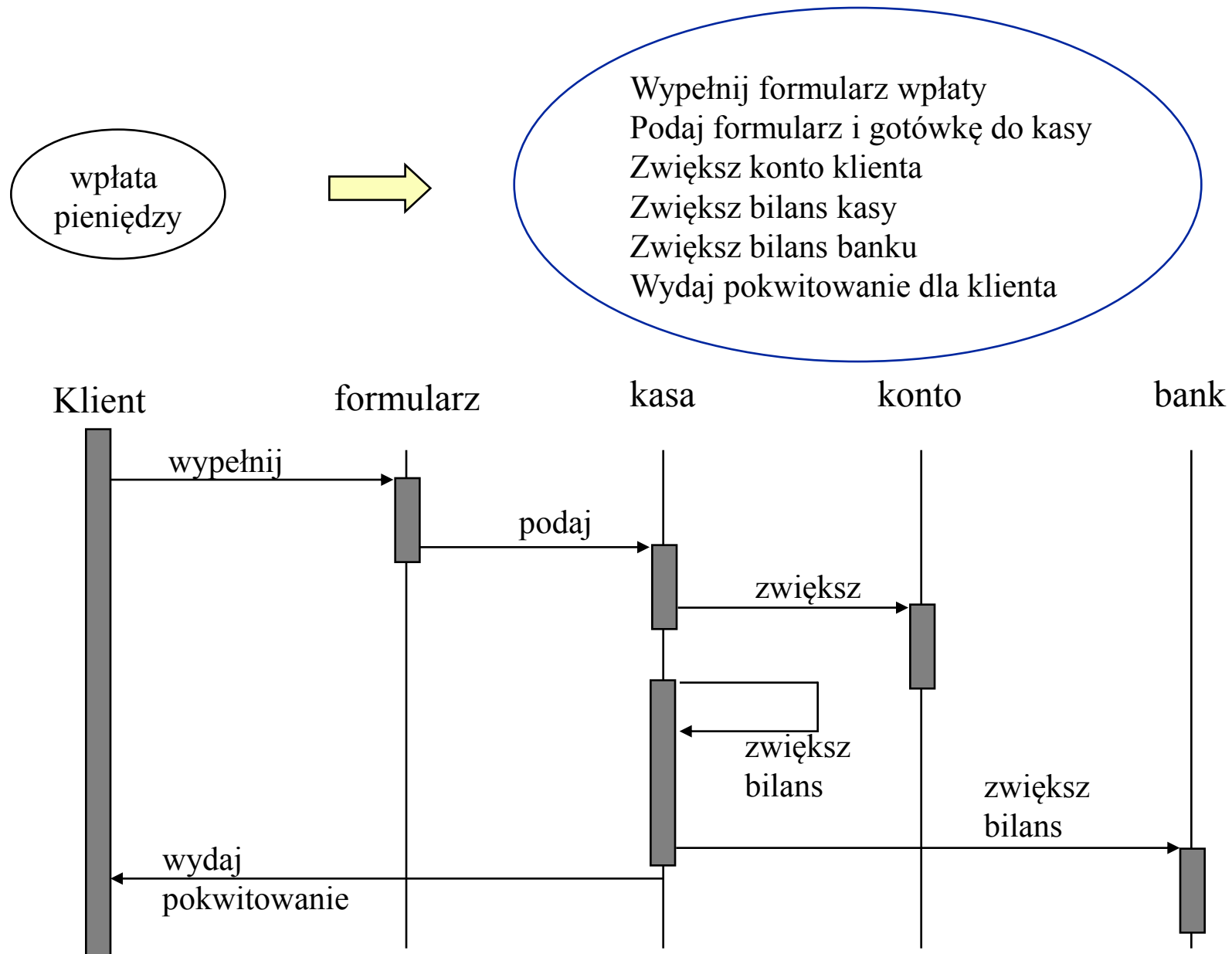
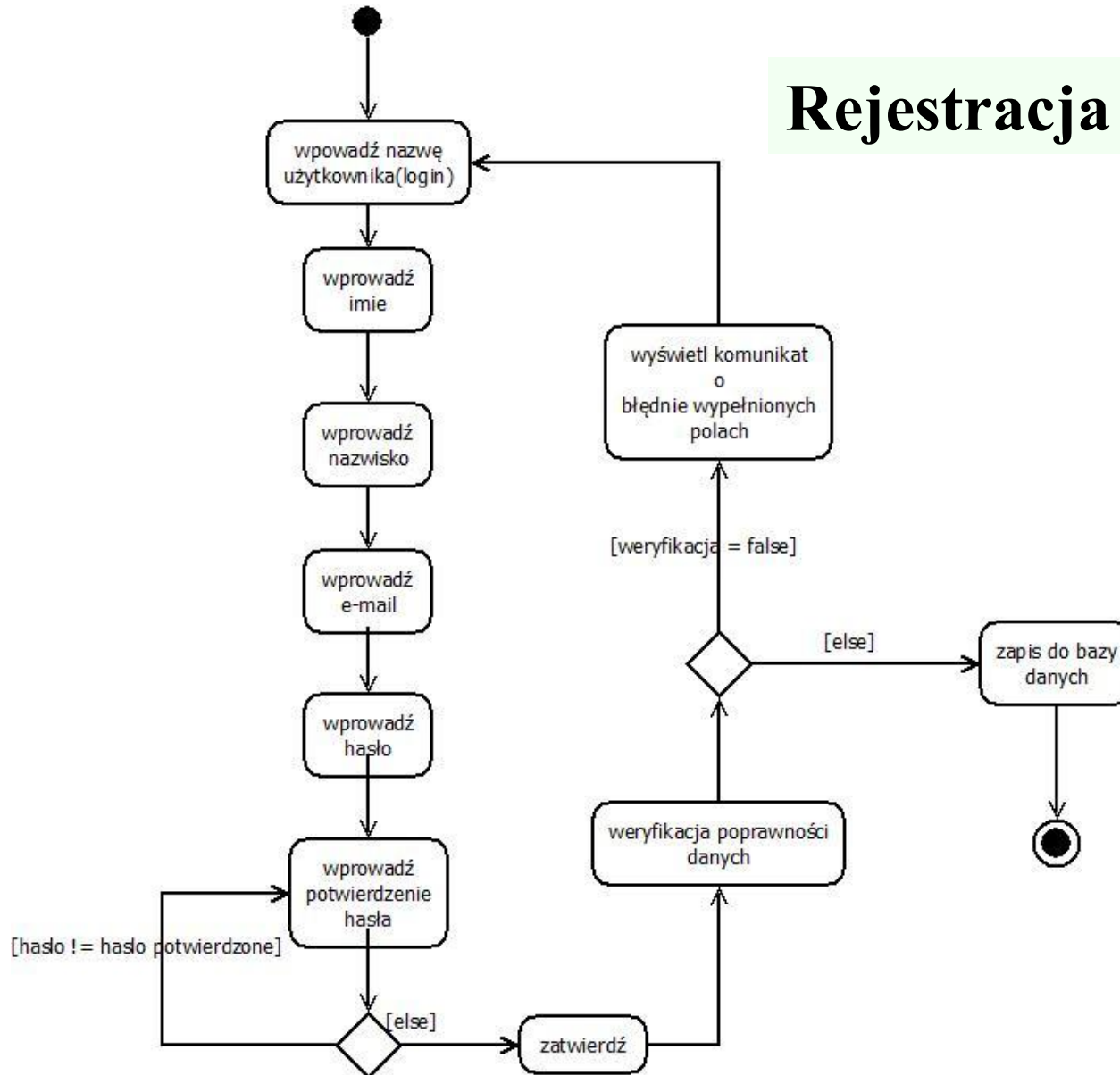


Diagram czynności

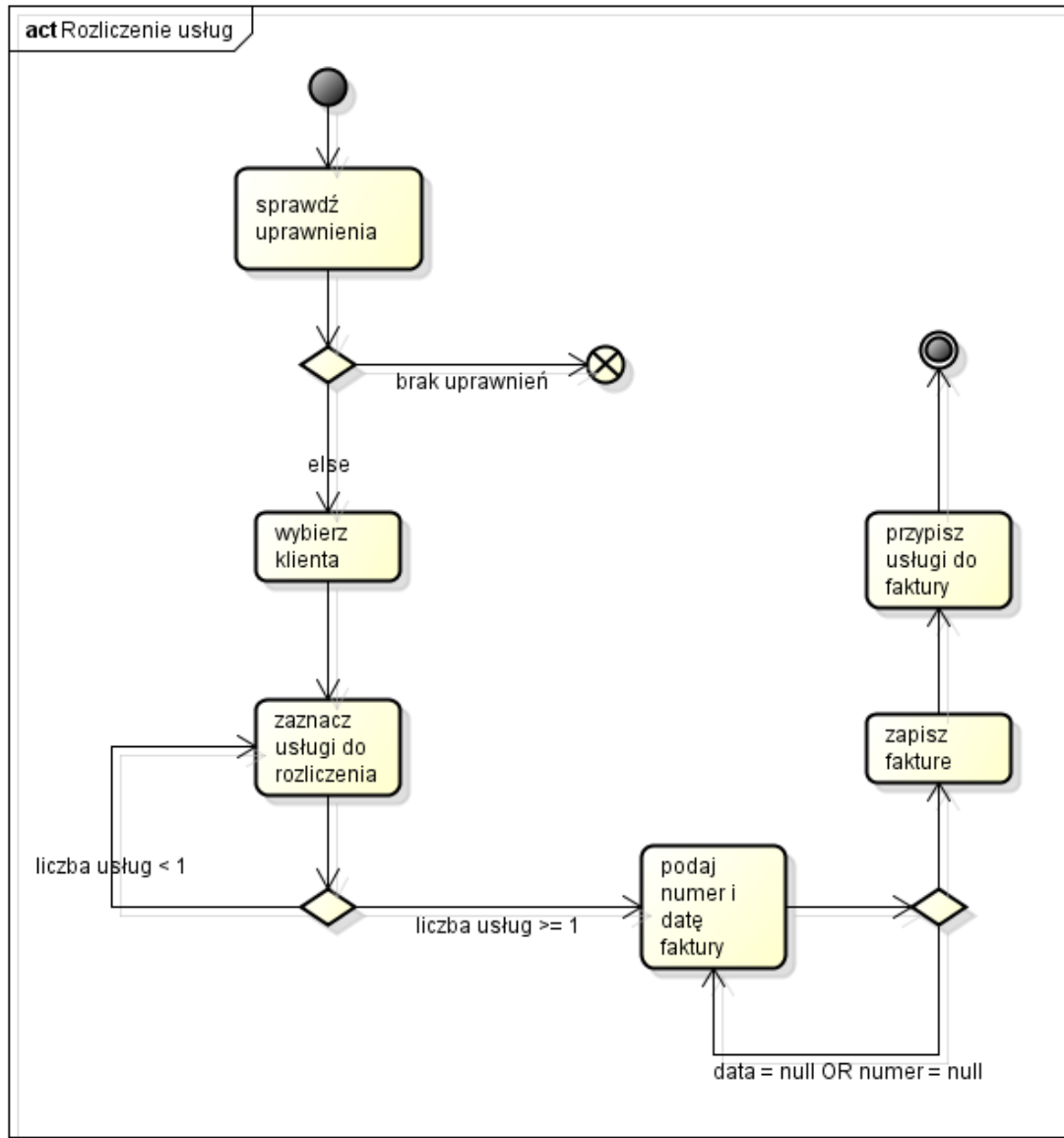
- Diagramy czynności** (ang. *activity diagrams*) prezentują przepływ sterowania w systemie związany z wykonaniem pewnej funkcji.
- Przepływy łączą czynności wykonywane przez poszczególne obiekty i stany obiektów, w których znajdują się po wykonaniu czynności.
 - Diagramy stanu skupiają się – jak nazwa wskazuje – na stanach, a akcje związane z ich zmianą są elementem dodatkowym.
 - W diagramach czynności jest odwrotnie: akcje są na pierwszym planie, a zmiany stanów są efektem ich wykonania.
 - Dlatego diagramy czynności dobrze nadają się do opisu przepływu sterowania pomiędzy obiektami (szczególnie w przypadku przetwarzania współbieżnego) oraz przepływu danych pomiędzy nimi.
 - Diagram, podobnie jak diagram stanu, może posiadać punkt startowy i dowolną liczbę stanów końcowych.
 - Najważniejszym jego elementem są akcje, reprezentowane przez prostokąty z zaokrąglonymi narożnikami oraz przejścia (łuki) przedstawiające przepływ sterowania.
 - Łuki mogą być opatrzone warunkami dozoru, które decydują o wykonaniu przejścia oraz zdarzeniami, które są generowane w momencie gdy przejście jest wykonywane.
 - Diagramy czynności zawierają także stany, w jakich może znaleźć się określony obiekt po wykonaniu akcji oraz elementy decyzyjne czy synchronizujące.

Przykład 1: Diagram czynności

Rejestracja użytkownika

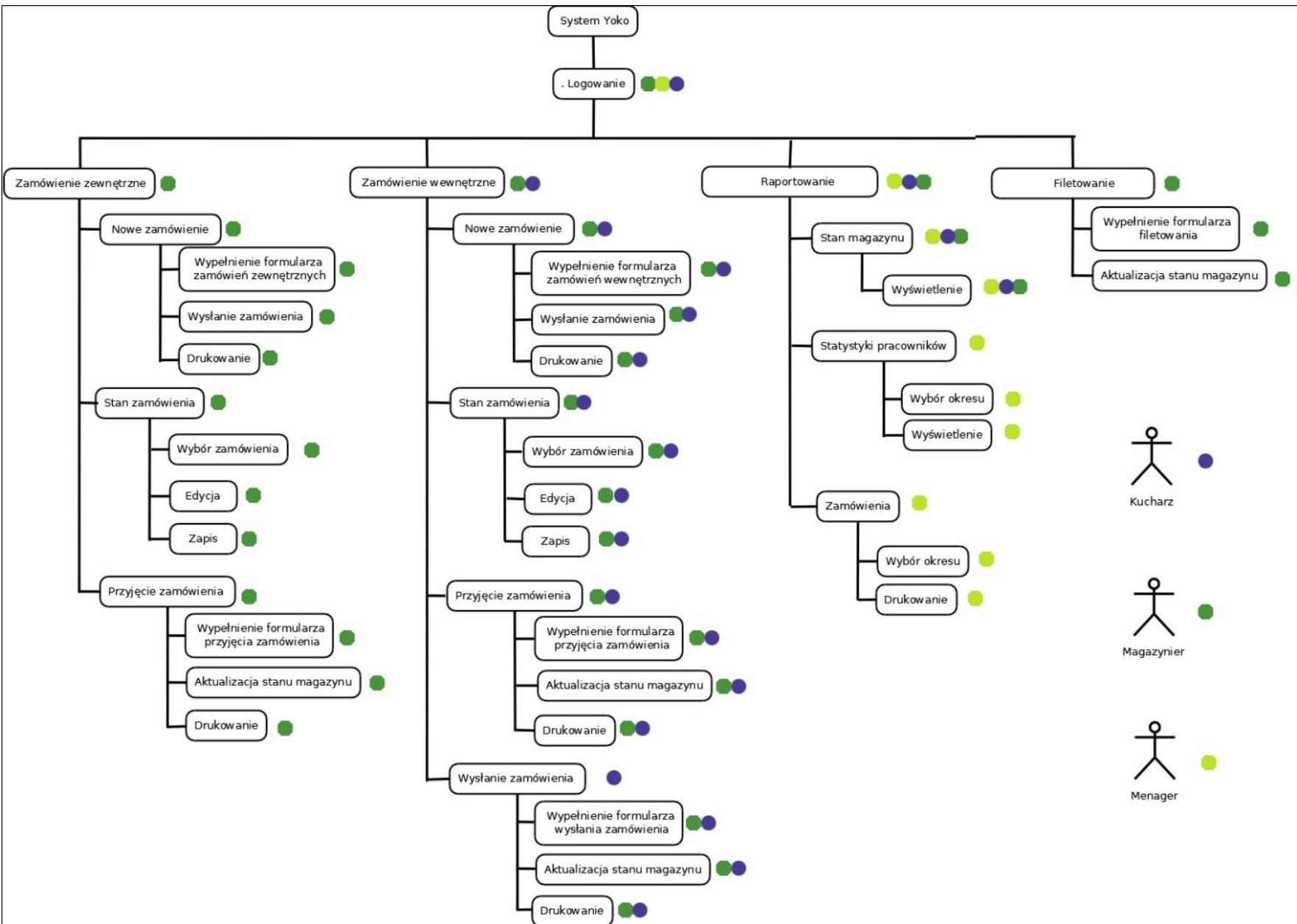


Przykład 2: Diagram czynności



Rozliczenie usługi

Diagram hierarchii funkcji



Zajęcia nr 2 - Analiza Zadania Dyplomowego

2 Przykład analizy – Port jachtowy:

1. Istnieje port jachtowy na jeziorze X
2. Port posiada miejsca do cumowania jachtów
3. Port posiada kilka pomostów < pomost A, B, C...>, przy których są miejsca do cumowania <miejsce 1, 2, ... 10.. > (uwaga: przy jednym pomoście może być wiele miejsc), miejsce ma określoną długość < 10m, 15 m, 20 m>, potrzebna jest również wiedza czy miejsce jest aktualnie wolne czy zajęte, niedostępne bo awaria, naprawa
4. W porcie cumują jachty, potrzebna wiedza <nazwa jachtu, długość jachtu, właściciel jachtu>
5. Potrzeba wiedzy o właścicielu jachtu < imię, nazwisko, adres, kraj> (uwaga: właściciel może mieć wiele jachtów, jeden jacht <cumujący> ma jednego właściciela).
6. Zajęcie miejsca cumowania <parkowanie> to: jaki jacht, jaki właściciel, które zajął miejsce na przystani, data przyplłynięcia, data odpłynięcia, łączna cena za cumowanie za dobę w tym miejscu.
7. Cumując można korzystać z pewnych (dodatkowo płatnych) usług <wymycie pokładu, doładowanie akumulatorów> jakie potrzebne informacje oprócz cennika?
8. Potrzeba ewidencjonowania tych usług, (uwaga jedno cumowanie może być wiele usług),
9. Cumując można odpłatnie pożyczyć na przystani narzędzia <młotek, siekiera, wiertarka> potrzeba ewidencjonowania tych płatnych pożyczek (uwaga jedno cumowanie może być wiele pożyczek sprzętu).

***Zamodelować system z wykorzystaniem statycznych
i dynamicznych diagramów UML***

Seminarium Dyplomowe

Dziękuję!!!!!! Tyle na dzisiaj

