



UNIwersytet
Przyrodniczy
we Wrocławiu

Diagram klas

Wprowadzenie do Inżynierii Oprogramowania
2018/2019

Radosław Zajdel
Instytut Geodezji i Geoinformatyki, pok. 329a
radoslaw.zajdel@upwr.edu.pl

UML

- Problem: Przełożenie skomplikowanych reguł biznesowych z języka biznesu na język informatyki.
 - Jednakowo interpretowany i rozumiały dla wszystkich członków zespołu projektowego system komunikacji. Standardem w tej dziedzinie stał się język **UML (Unified Modelling Language)** – graficzny system wizualizacji, specyfikowania oraz dokumentowania składników systemów informatycznych
 - UML to notacja umożliwiająca **zaprezentowanie systemu w sposób graficzny**, w postaci diagramów. Modele zapisane w języku UML prezentują system od ogółu do szczegółu, umożliwiając oglądanie modelu systemu z wybraną w danym momencie szczegółowością.
-

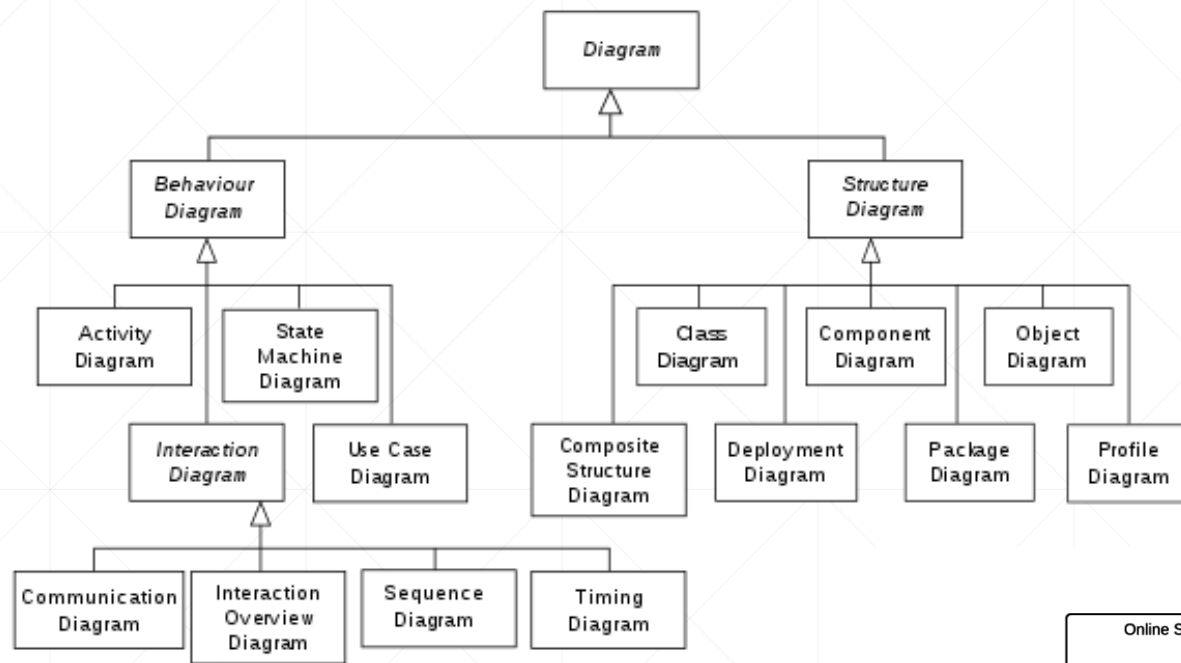


Diagram przypadków użycia

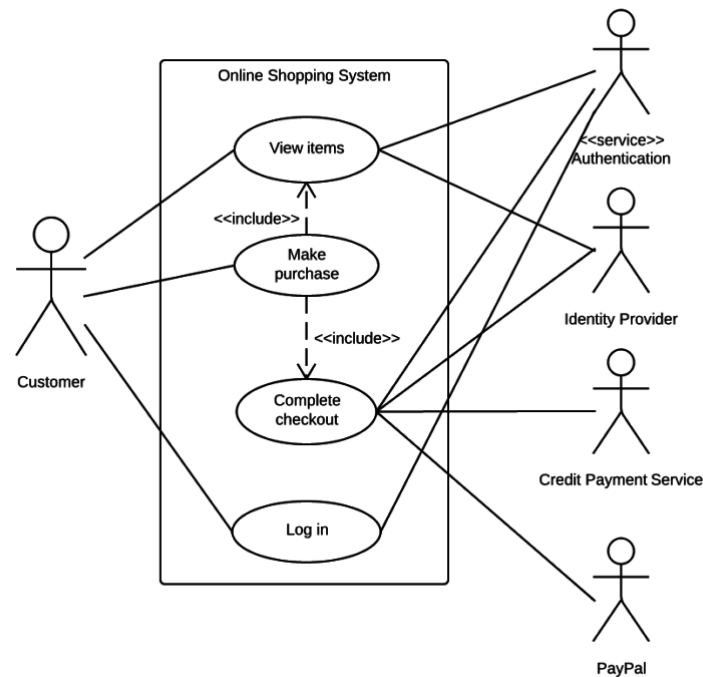


Diagram klas

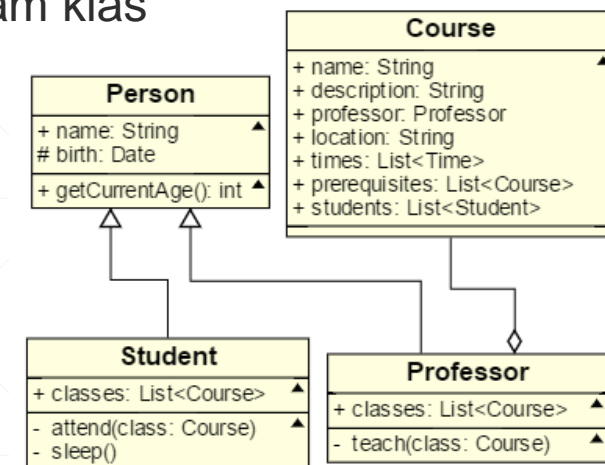
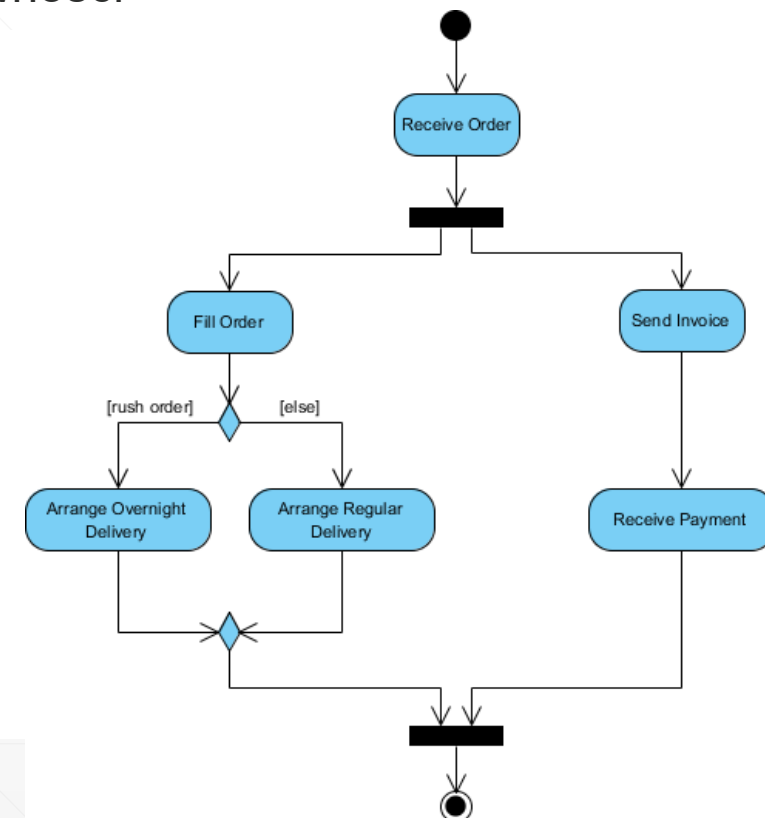


Diagram aktywności



Entity Relationship Diagram - Internet Sales Model

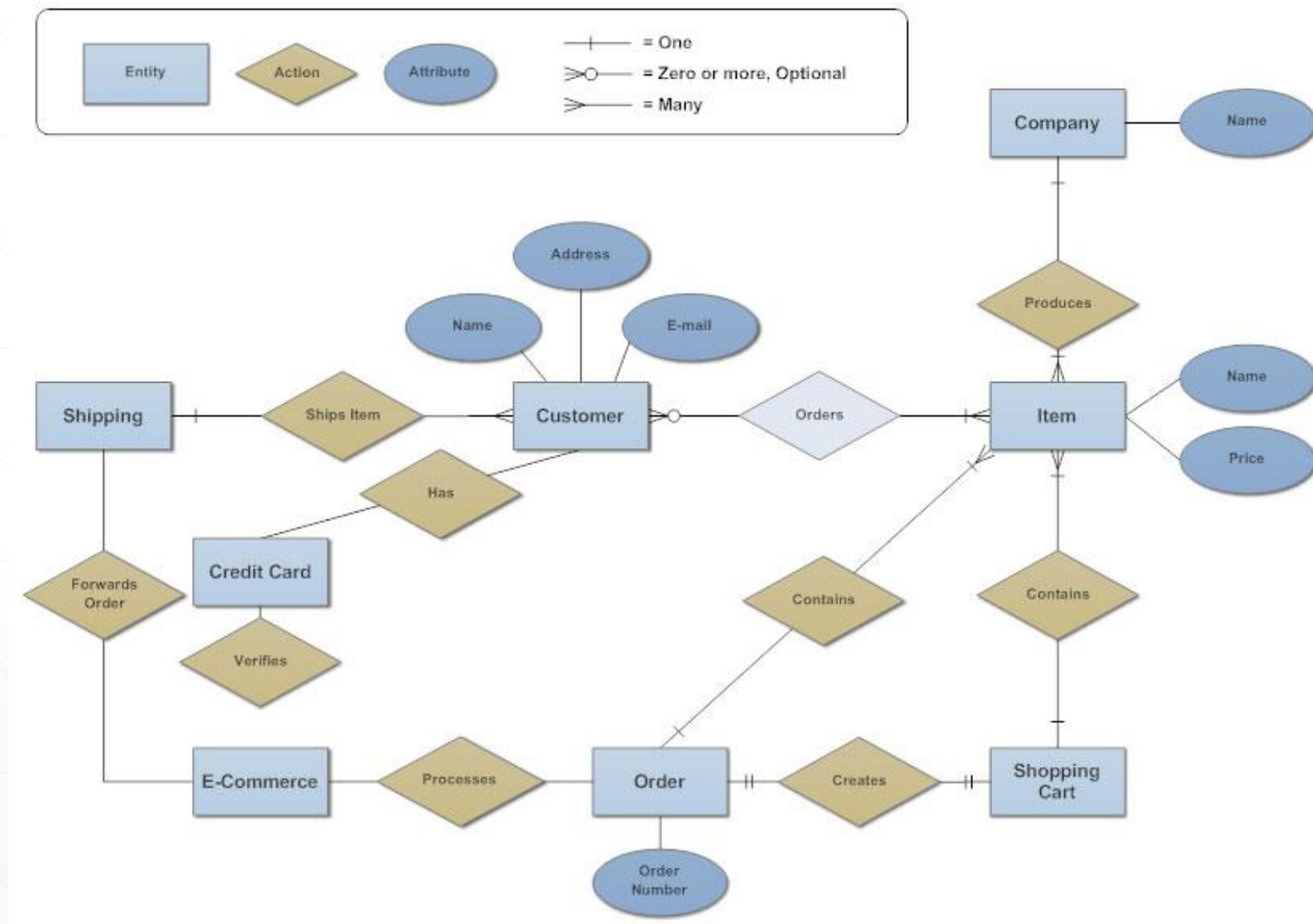


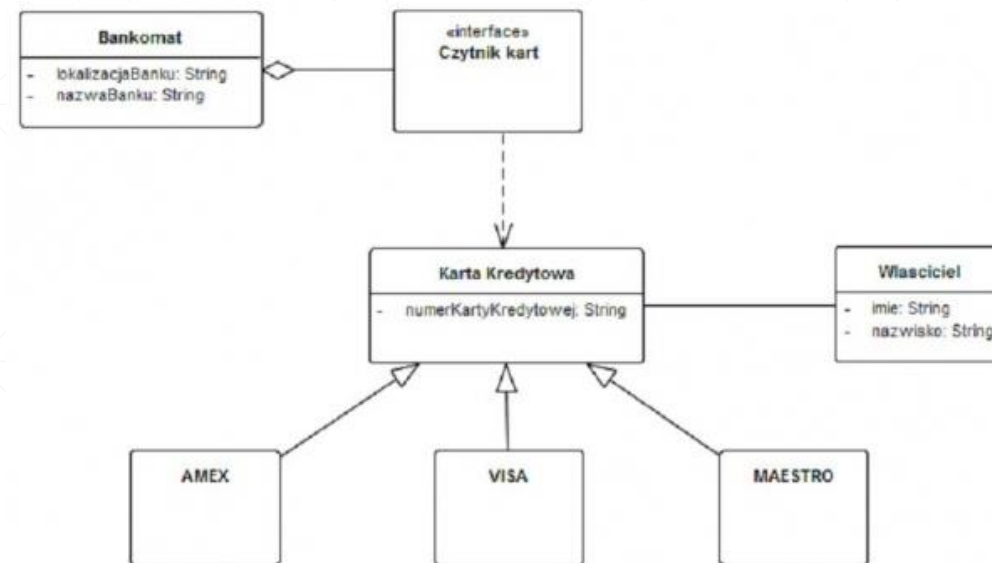
Diagram klas UML

Def: Diagram klas to graficzne przedstawienie statycznych, deklaratywnych elementów dziedziny przedmiotowej oraz związków między nimi.

- Najczęściej stosowany obok diagramy przypadków użycia (następny wykład)
 - Diagram przedstawiający strukturę klas w systemie i ich wzajemne związki (statyka systemu).
 - Główne elementy diagramu klas mają znaczny wpływ na układ i zawartość innych diagramów UML
 - Wykorzystywany do opisywania struktury danych (INSPIRE)
 - Podstawowym klasyfikatorem jest Klasa
-

Cel diagramów klas

- definiują granice modelowanego systemu
- określają jego kontekst
- wymieniają użytkowników systemu i jednostki zewnętrzne
- przedstawiają funkcje dostępne dla użytkowników
- określają powiązania i zależności pomiędzy nimi



Modelowanie obiektowe

- Jedno z podejść modelowania rzeczywistości spopularyzowane razem z rozwojem języków programowania zorientowanych obiektowo
 - „Świat rzeczywisty może być dokładnie opisany za pomocą zbioru obiektów, które oddziałują pomiędzy sobą”
 - Podstawowe pojęcia:
 - Obiekt – element posiadający stan i zachowanie
 - Klasa obiektu – definicja zbioru obiektów. Określa ich dopuszczalny stan i zachowanie
 - Instancjonowanie – utworzenie nowego obiektu na podstawie klasy (Obiekt=Instancja Klasy)
-

Klasa „Samochód”

Atrybuty:

- Marka
- Model
- Rok produkcji
- Kolor

Operacje:

- Jedź
- Hamuj

instancjonowanie

Obiekt „Mój Bugatti Veyron” *instancja klasy „Samochód”*



Pola:

- Marka - Bugatti
- Model - Veyron 16.4
- Rok produkcji - 2010
- Kolor - czarny

Metody:

- Jedź
- Hamuj

Paradygmat podejścia obiektowego

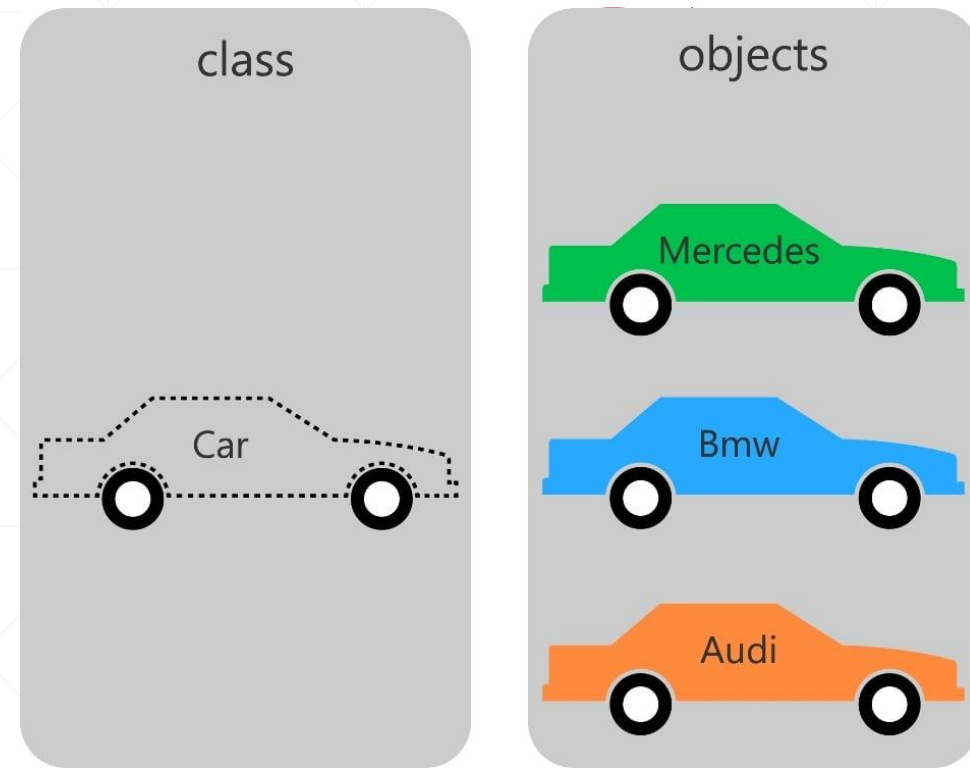
- Obiektem jest każdy byt – pojęcie lub rzecz – mający znaczenie w kontekście rozwiązania problemu w danej dziedzinie przedmiotowej.
 - Wszystko, co wiadomo o obiekcie jest reprezentowane przez wartości atrybutów – czyli cech obiektu
 - Zachowanie obiektu wyrażone jest w metodach, które oferuje obiekt
 - Obiekt charakteryzuje się unikatową tożsamością, jest więc możliwe wyodrębnienie każdego obiektu nawet, gdy każdy opisany jest przez te same atrybuty. Obiekt jest wyróżniany przez samo istnienie a nie cechy opisowe
-

Klasa

- Klasa to grupa obiektów charakteryzujących się:
 - Identyčną strukturą danych tj. takimi samymi atrybutami
 - Identyčnym zachowaniem tj. takimi samymi operacjami
 - Identyčnymi związaniami
 - Identyčnym znaczeniem w określonym kontekście

Def: Klasa jest uogólnieniem zbioru obiektów, które mają takie same atrybuty, operacje, związki i znaczenie

Klasa to wzorzec, według którego tworzy się obiekty,
Klasa to zbiór obiektów stworzonych w oparciu o pewien wzorzec.



Wskazówki przy identyfikacji klas

- Typowe wskazówki pomocne w identyfikacji klas przy analizie opisu dziedziny zastosowań – w postaci rodzajów bytów będących kandydatami na modelowanie w postaci obiektów:
 - przedmioty materialne i ich grupy (urządzenia, wyposażenie)
 - role pełnione przez osoby (pracownik, student, kierownik)
 - zdarzenia (zakup, dostawa, wypadek)
 - interakcje między obiektami (spotkanie, pożyczka, zapłata)
 - procedury (zatwierdzanie, weryfikacja, pobieranie, naprawa)
 - lokalizacje (pracownia, lotnisko)
 - instytucje i organizacje (dział, wydział, bank)
 - dokumenty (faktura, zamówienie, lista obecności)
 - konkretne byty niematerialne z dziedziny zastosowań (np. w matematyce – figura, macierz, równanie)
-

Wykrywanie klas

- Klient idzie do Działu Zamówień i składa zamówienie na towar. Dział Zamówień sprawdza czy towar jest w magazynie. Jeśli tak, to przesyła zamówienie do Działu Spedycji i Działu Księgowości. Dział Księgowości przygotowuje fakturę i przeprowadza transakcję otrzymując zapłatę za pośrednictwem banku. Dział Spedycji pobiera towar z magazynu (jeśli stan zapasów jest zbyt mały zamawia dodatkową partię z hurtowni) i realizuje wysyłkę do klienta.
-

Wykrywanie klas

- **Klient** idzie do **Działu Zamówień** i składa **zamówienie** na **towar**. **Dział Zamówień** sprawdza czy towar jest w **magazynie**. Jeśli tak, to przesyła zamówienie do **Działu Spedycji** i **Działu Księgowości**. Dział Księgowości przygotowuje **fakturę** i przeprowadza **transakcję** otrzymując **zapłatę** za pośrednictwem **banku**. Dział Spedycji pobiera towar z magazynu (jeśli stan zapasów jest zbyt mały zamawia dodatkową partię z **hurtowni**) i realizuje wysyłkę do klienta.
-

Klasa w diagramach UML

W diagramach, klasę standardowo przedstawia się jako prostokąt złożony z **min. trzech sekcji**

Nazwa

NazwaKlasy

Atrybuty

+ atrybut1
+ atrybut2

dopuszczalny stan

Operacje

+ operacja1()
+ operacja2()

dopuszczalne zachowanie

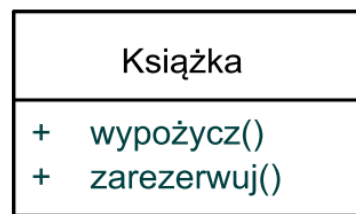
Klasa w diagramach UML



brak atrybutów
i operacji



tylko atrybuty



tylko operacje



atrybuty
i operacje

Związki

- Związek stanowi powiązanie pomiędzy elementami modelu
 - Klasy obiektów – podobnie jak inne elementy języka UML – powiązane są różnego rodzaju związkami. W diagramach klas powszechnie stosuje się cztery rodzaje związków języka UML:
 - Asocjacje
 - Uogólnienia – Generalizacje
 - Zależności
 - Realizacje
-

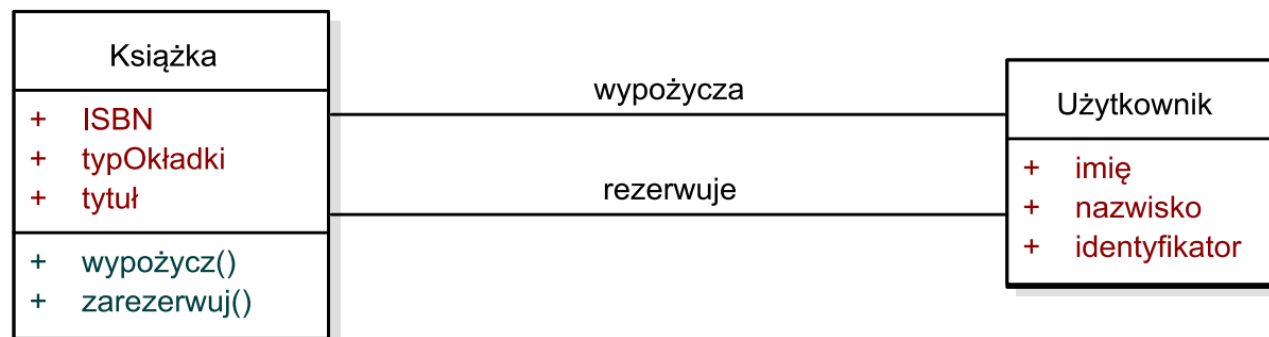
Asocjacja

- Asocjacja jest związkiem pomiędzy dwoma lub więcej klasyfikatorami (reprezentacja klasy), opisującym powiązania pomiędzy ich instancjami
- Związek dwukierunkowy – obie klasy są świadome związku

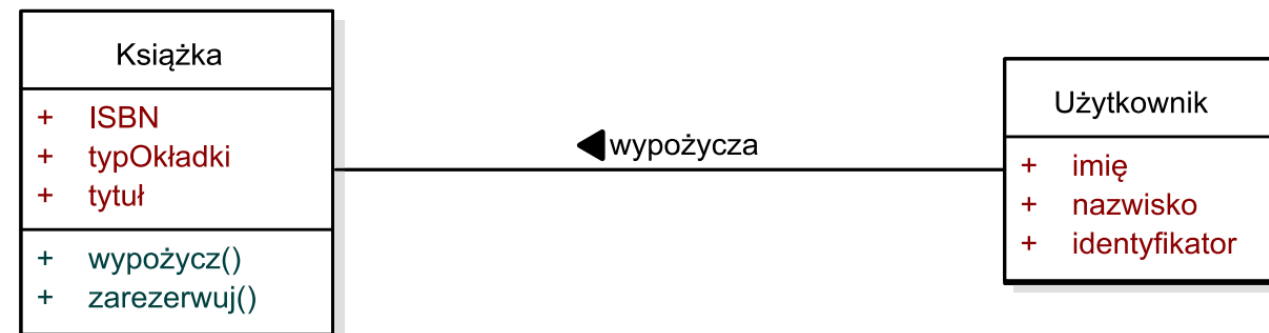


Asocjacja

- Pełna interpretacja związku wymaga wprowadzenia szeregu dodatkowych elementów opisu. Tak więc asocjację można dokładnie sprecyzować poprzez zdefiniowane jej następujących cech:
 - Nazwy
 - Ról powiązanych klas
 - Nawigacji
 - Liczebności
 - Agragacji
-

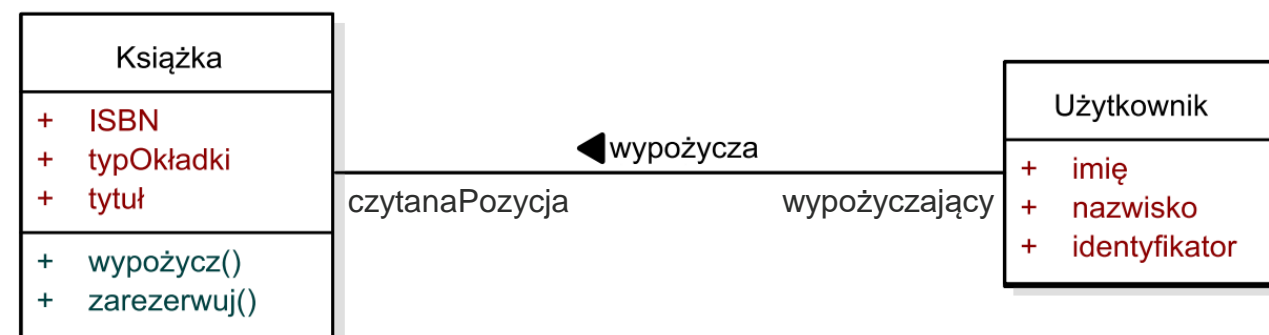


Nazwanie asocjacji:
Wiemy o zależności
wypożyczania i rezerwacji



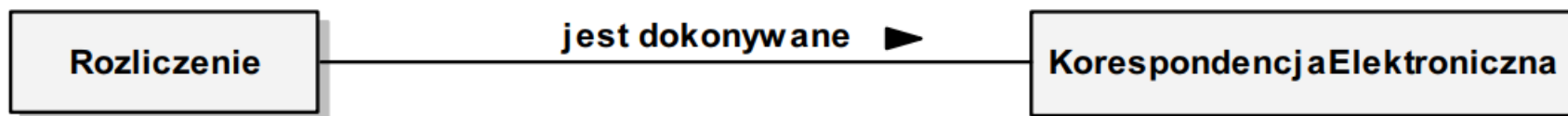
Asocjacja kierunkowa:
Użytkownik wypożycza książkę

Istotne jest przedstawienie
inicjatora interakcji



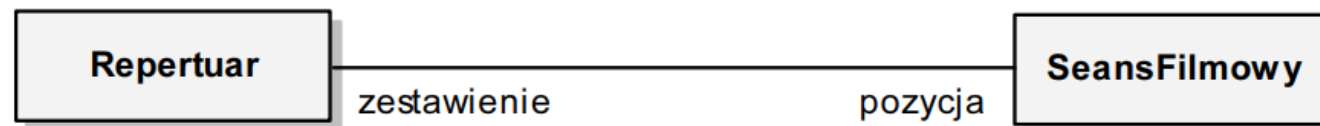
Asocjacja kierunkowa z ze
wskazaniem ról w asocjacji

Więcej przykładów

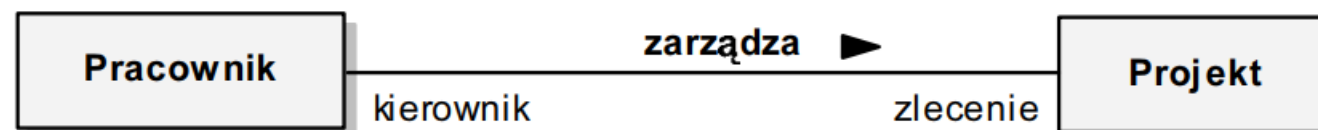


Poszczególne transakcje finansowe reprezentowane przez obiekty klasy Rozliczenie powiązane są z odpowiednimi dokumentami elektronicznymi, które są instancjami klasy KorespondencjaElektroniczna

Rola w asocjacji jest powinnością pełnioną przez jedną klasę obiektu wobec drugiej. Nazwy ról umieszcza się po obu stronach asocjacji.



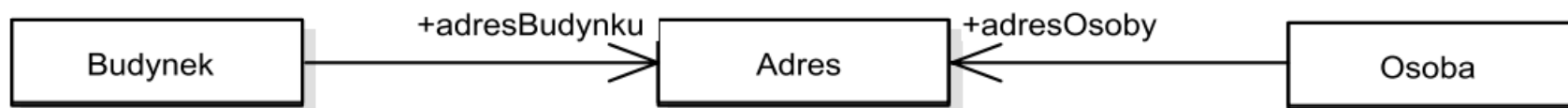
Poszczególne seanse, będące instancjami klasy *SeansFilmowy*, powiązane są z bieżącym *Repertuarem* kinowym. Repertuar jest sumarycznym zestawieniem seansów – a więc dany seans stanowi jedną z pozycji repertuaru.



Kierownik projektu, z którym bezpośrednio powiązany jest projekt, automatycznie klasyfikowany jest jako jego kierownik. Projekt potraktowano w kategoriach zlecenia firmy zarządzanego przez kierownika.

Nawigacja

- Związek jednokierunkowy – czasami komunikacja dwukierunkowa jest zbędna, a wskazanie kierunku nawigacji zwiększa efektywność komunikowania się.
- Tylko jedno strona jest świadoma związku
- „Referencja”

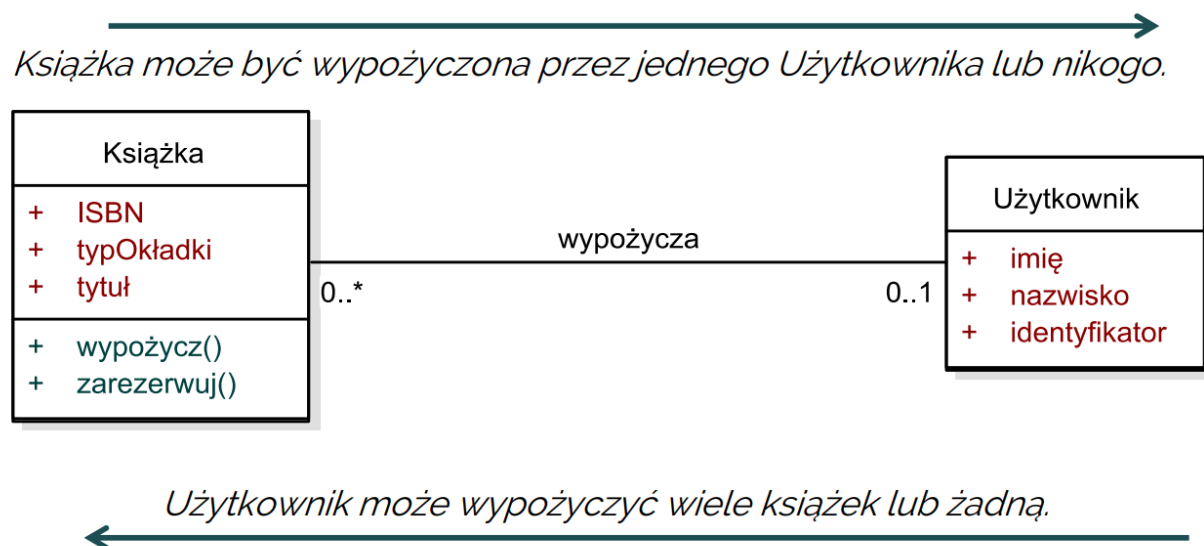


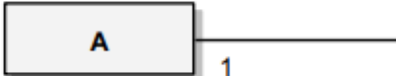










Budynek znajduje się pod adresem

Osoba zamieszkuje adres

Liczebność

- Ograniczenie nakładane na końce asocjacji specyfikujące dopuszczalną liczbę obiektów biorących udział w danym związku.



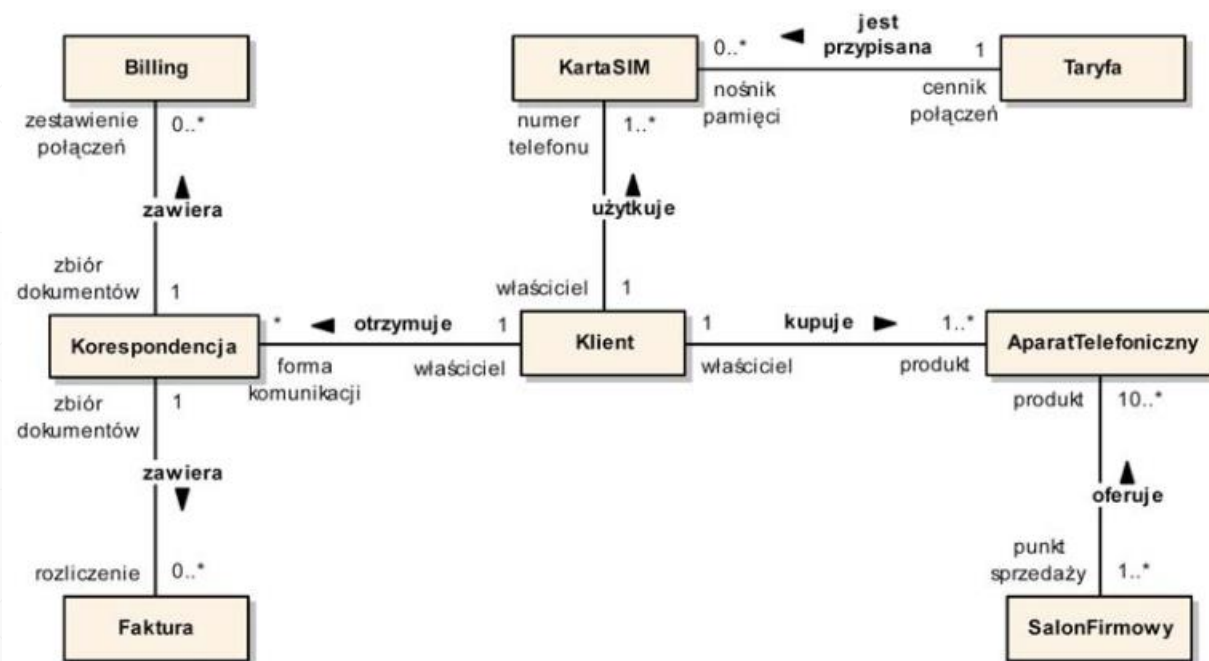
Oznaczenie	Opis	Przykład
1	dokładnie jeden	
1..*	jeden lub wiele	
0..1	zero lub jeden	
*	wiele	
0..*	zero lub wiele	
n	dokładnie n ($n > 1$)	
1..n	od 1 do n	
0..n	od 0 do n	
n..m	od n do m ($n, m > 1$)	
n..*	więcej niż n	
n, m, o..p, q	liczebność złożona	

Poprzez n rozumie się określoną znaną liczbę naturalną

Natomiast w przypadku chęci zasygnalizowania, że w określonej asocjacji bierze udział znaczna, lecz nieznana liczba obiektów, używa się symbolu gwiazdki

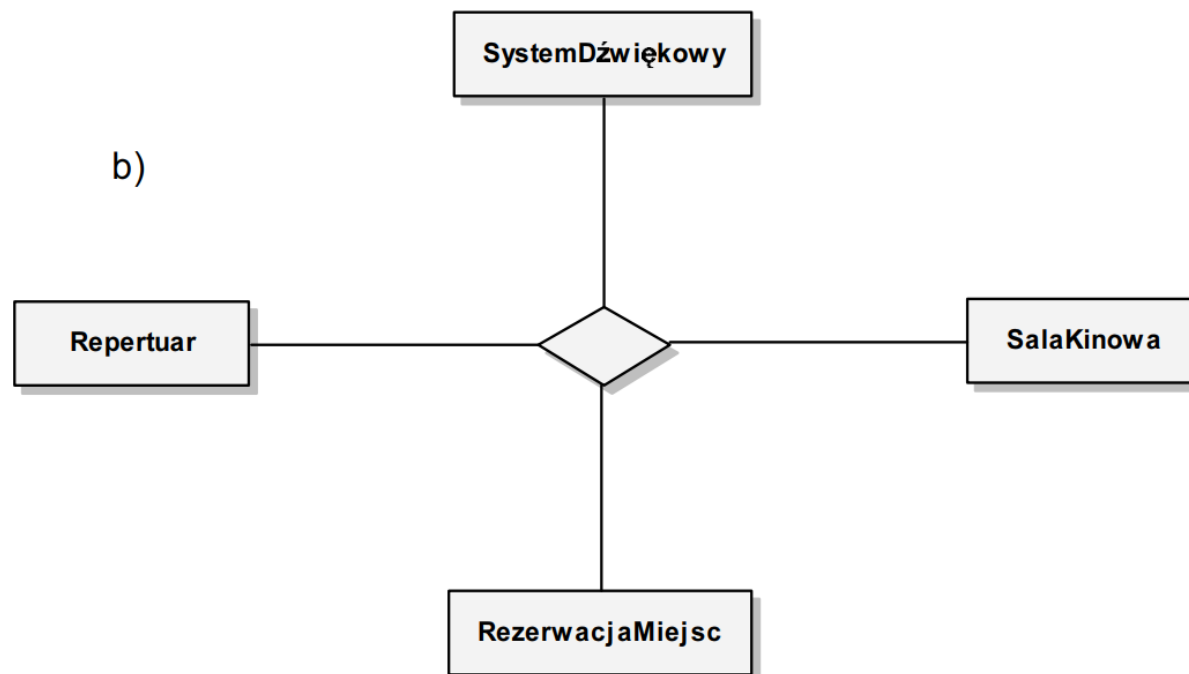
Interpretacja liczebności

- Polega na każdorazowym rozpatrzeniu sytuacji gdy po jednej jej stronie znajduje się jeden obiekt, natomiast liczba obiektów, które odpowiadają mu w drugiej klasie jest ustalana.
- Asocjacja *kupuje* łączy klasy *Klient* i *AparatTelefoniczny*. Jeden obiekt klasy *Klient* może kupić jeden lub wiele obiektów klasy *AparatTelefoniczny*, natomiast jeden *AparatTelefoniczny* może być kupiony tylko przez jednego *klienta*



Ciekawostka – Asocjacje n-arne

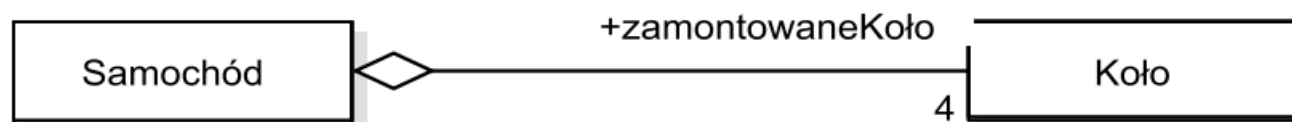
- Ograniczenie nakładane na końcu asocjacji specyfikujące dopuszczalną liczbę obiektów biorących udział w danym związku



Agregacja

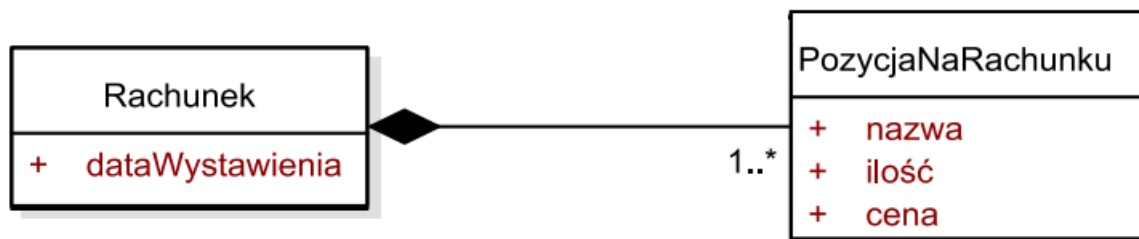
- Związek opisuje relacje typu całość-część pomiędzy klasami
 - Dostępne formy agregacji
 - Agregacja częściowa (agregacja słaba, agregacja współdzielona)
 - Agregacja całościowa (kompozycja, agregacja silna, agregacja składowa)
 - Dwa podstawowe pojęcia
 - Agregat, czyli obiekt stanowiący całość
 - Segment, czyli część
-

- Agregacja częściowa



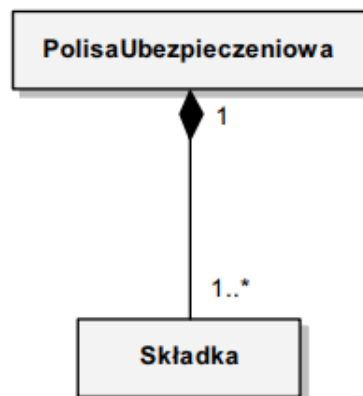
Segment (koło) **może** istnieć „samodzielnie” bez Agregatu (Samochód)

- Agregacja całkowita (kompozycja)

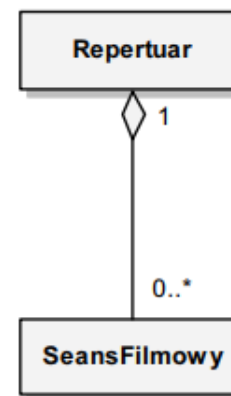


Segment (pozycja na rachunku) **nie może** istnieć „samodzielnie” bez Agregatu (rachunku)

Przykłady agregacji



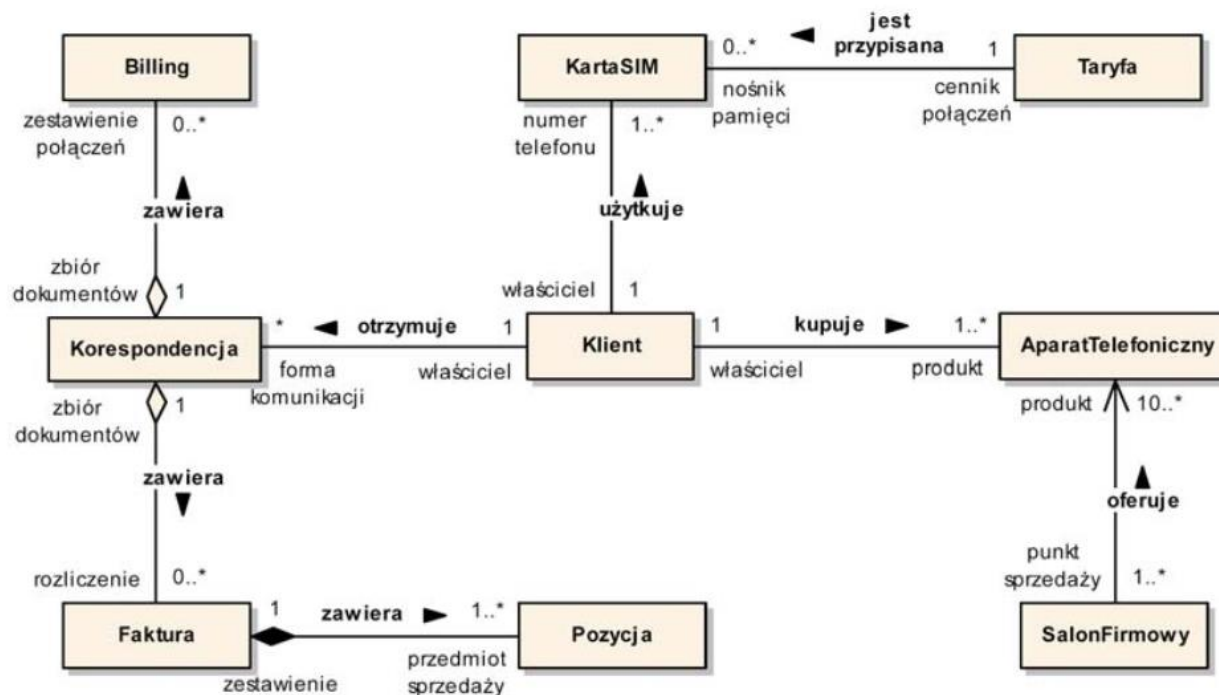
Składka jest integralną częścią polisy ubezpieczeniowej i nie będzie funkcjonować w oderwaniu od niej



Potraktowanie Seansów filmowych w kategorii agregacji częściowej wskazuje, że są częścią składową Repertuaru ale w przypadku zastąpienia Repertuaru innym bytem Seanse nie zostaną utracone

Przykład rozszerzony

Klient otrzymuje wiele zestawów dokumentów traktowanych jako korespondencja. Elementem takiej korespondencji jest Billing będący zestawieniem połączeń oraz Faktura będąca rozliczeniem. Faktura posiada związek kompozycji z Pozycją, określającą temat faktury



Klient użytkuje jedną lub wiele KartSIM. Do każdej kartySim przyporządkowana jest jedna Taryfa ale jedna taryfa może być przypisana do wielu kartSIM

Odpowiedzialnością SalonuFirmowego, jest oferowanie Aparatów Telefonicznych

Atrybuty

- Atrybuty klas mogą przyjąć dodatkowe ograniczenia związane z określeniem **zbioru dozwolonych wartości**
- Określenie **krotności występowania atrybutów**

Osoba
+ imię: string
+ nazwisko: string
+ wiek: integer
+ dataRejestracji: date

Klasa
+ nazwa: string
+ alternatywneNazwy: string [1..*]

Typy atrybutów wywodzą się z typów danych prostych

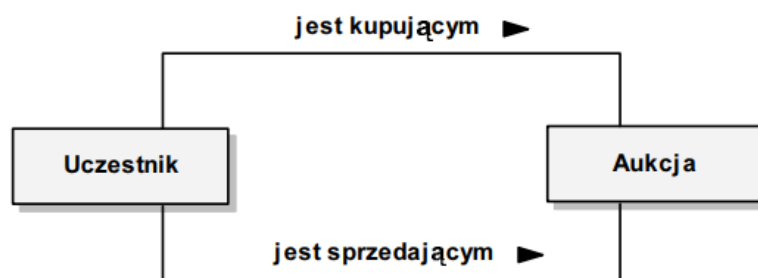
Typy danych	Narzędzie Rational Rose
Logiczne	Boolean
Stałoprzecinkowe	Byte
	Integer
	Long
Zmiennoprzecinkowe	Single
	Double
Znakowe	String
Inne	Date
	Object
	Currency
	Variant

OpcjaFinansowa
premiaOpcyjna: Single = 10 000 wielkośćKontraktu: Double = 1 000 000 tick % = 0,01 wartośćTicku: Byte = 25
liczTick(cenaSprzedaży :Double, cenaNabycia :Double): Single realizuj(ilośćTicków :Single, wartośćTicku :Byte, ilośćKontraktów :Integer): Double

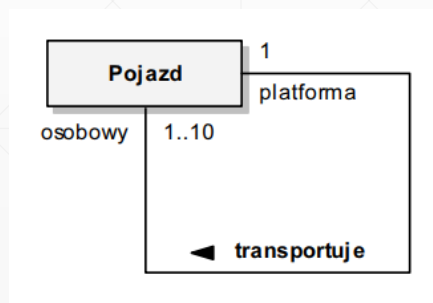
Istnieje dodatkowo opcja stworzenia **tzew. Atrybutu pochodnego**, którego nazwa będzie zaczynała się od „/” np. */wartoscTransakcji*. Wartość takiego atrybutu ustala się na zasadzie przypisania określonej formuły obliczeniowej lub logicznej, która korzysta z innych atrybutów. Formuła jest sformalizowana w postaci reguły biznesowej.

Asocjacje zwrotne i wielokrotne

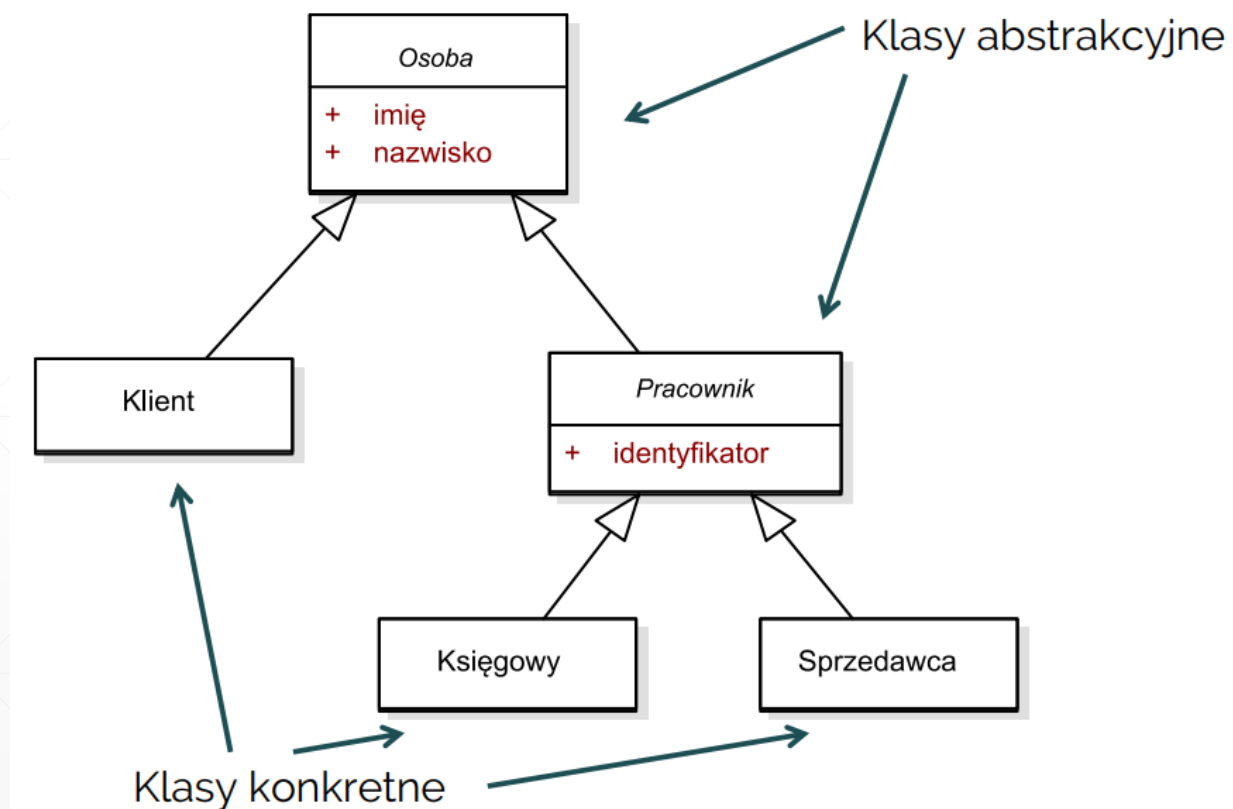
- Asocjacje wielokrotne: powiązane klasy mogą pełnić wobec siebie kilka odmiennych ról



- Asocjacje zwrotne: asocjacja wiążąca daną klasę z samą sobą



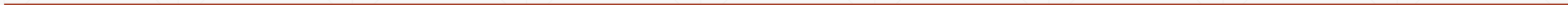
Generalizacja - Uogólnienie



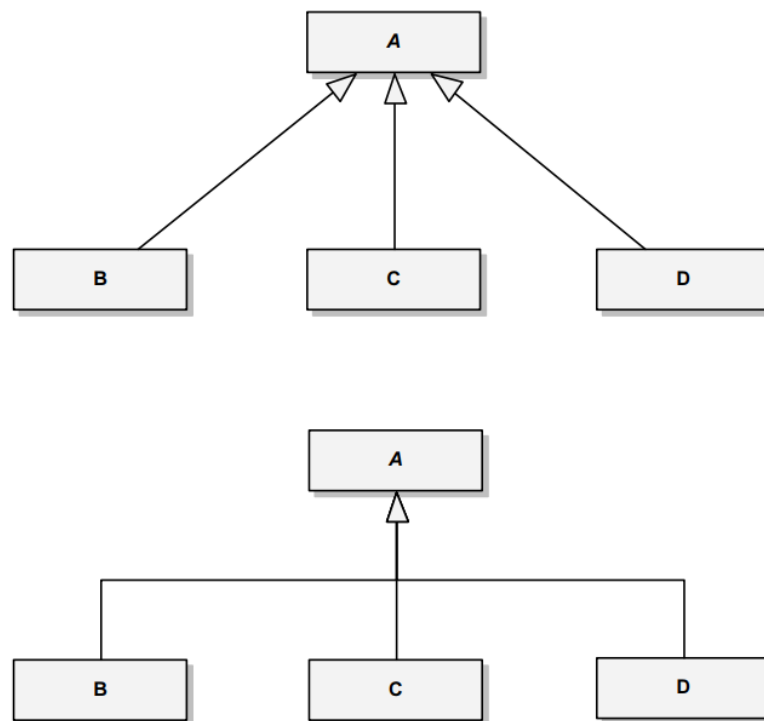
- Klasy abstrakcyjne nie mają instancji !
- Nazwy klas abstrakcyjnych pisane są kursywą
- Atrybuty i operacje klasy abstrakcyjnej podlegają dziedziczeniu zgodnie z hierarchią klas

Dla chętnych

- Ograniczenia complete | incomplete | joint | disjoint

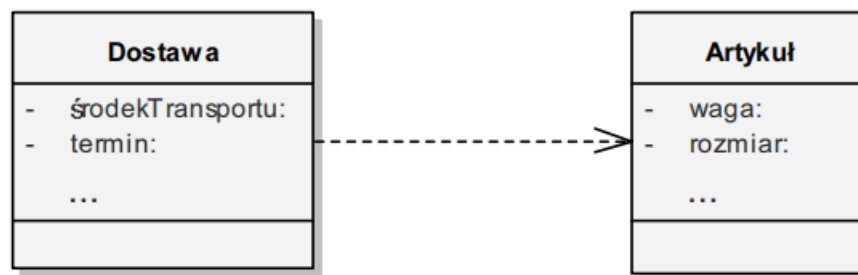


Notacja graficzna generalizacji



Pozostałe związki – Zależności i realizacja

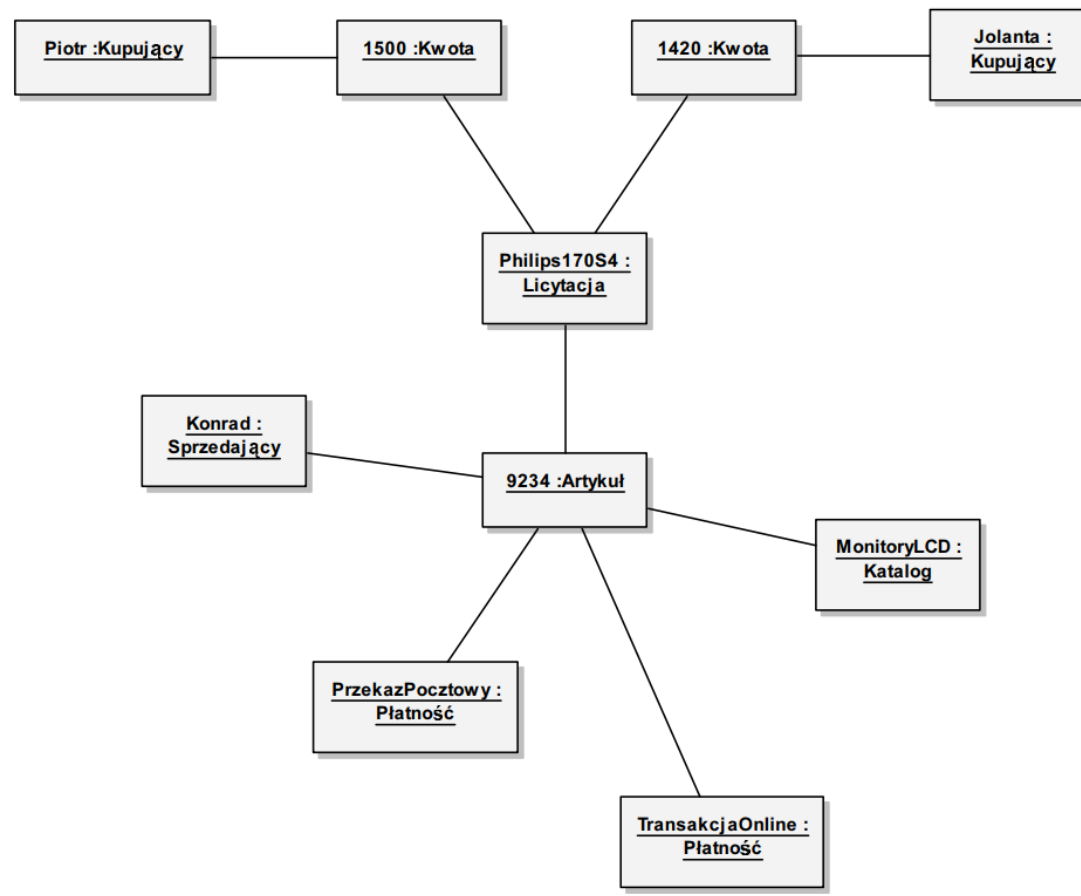
- Niezależna klasa obiektów wykorzystuje klasę zależną do jakiegoś celu

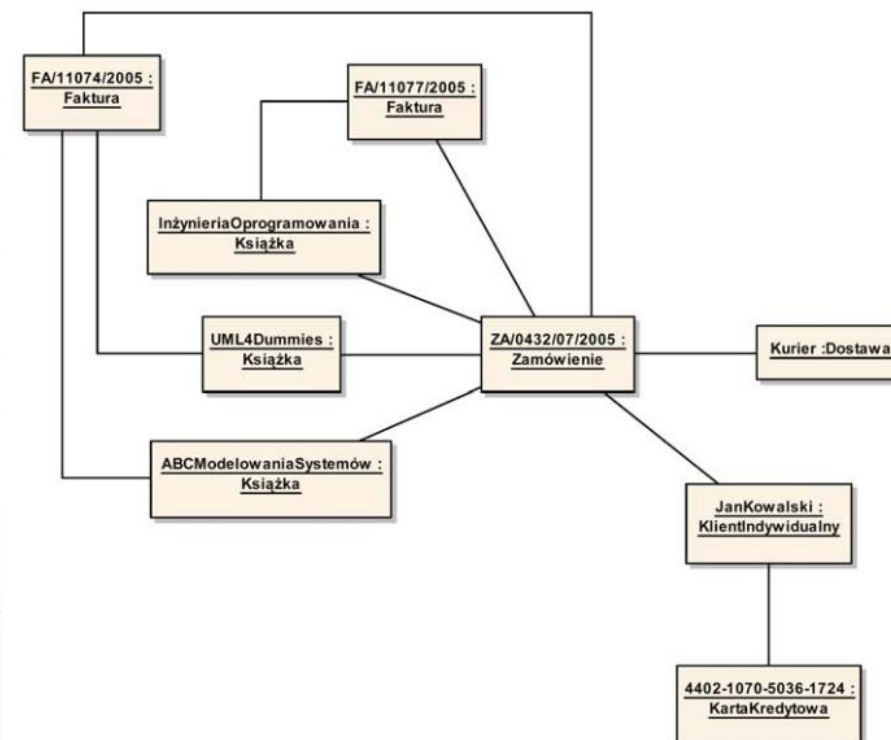
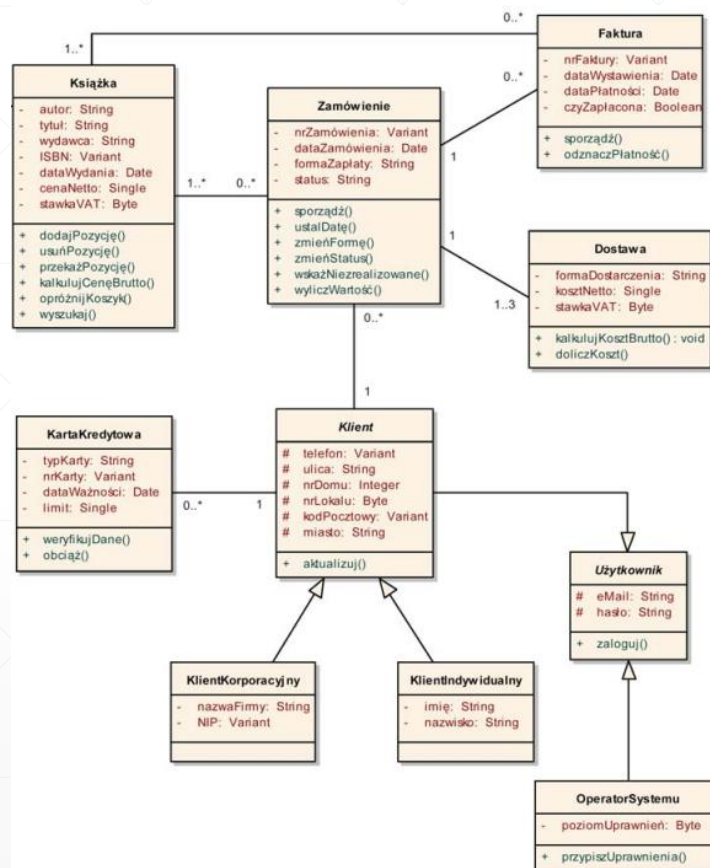


- Szerzej omówimy je w przypadku diagramu przypadków użycia
-

Wiedza uzupełniająca – Diagram obiektów

- Diagram obiektów to wystąpienie („instancja”) diagramu klas, odwzorowuje strukturę systemu w wybranym momencie jego działania





Jak podejść do tworzenia diagramu klas ?

1. Zidentyfikowanie i nazwanie klas
2. Określenie zobowiązań klas
3. Połączenie klas asocjacjami
4. Zidentyfikowanie oraz nazwanie atrybutów i operacji
5. Wspecyfikowanie asocjacji (nazwy, kierunki, role itd.)
6. Opracowanie innych związków (np. generalizacja)

Etapy wykonuj iteracyjnie



Opracowano na podstawie: „Język UML 2.0 w modelowaniu systemów informatycznych”
