Podstawy języka UML

Plan prezentacji

- Wprowadzenie do modelowania
- Wprowadzenie do języka UML
- Diagram klas
- Diagram pakietów
- Diagram przypadków użycia
- Diagram czynności

Terminologia

- **Aplikacja** - program komputerowy, który ma bezpośredni kontakt z użytkownikiem, realizuje zadania dla użytkownika; działanie aplikacji umożliwia system operacyjny.
- **Oprogramowanie** - ogół programów, w które wyposażony jest system komputerowy; rozróżnia się oprogramowanie podstawowe (m.in. system operacyjny, translatory, graficzny interfejs użytkownika) oraz oprogramowanie użytkowe (aplikacyjne), służące do wykonywania określonych, złożonych zadań, np. oprogramowanie statystyczne.

Terminologia

- **Baza danych** - zbiór wzajemnie powiązanych danych, przechowywanych w pamięci komputerów i wykorzystywanych przez programy użytkowe instytucji lub organizacji wraz z oprogramowaniem umożliwiającym definiowanie, wykorzystywanie i modyfikowanie tych danych.
- **Egzemplarz** – jedna sztuka z grupy jednorodnych przedmiotów.

Modelowanie - zagadnienia

- Czym jest model?
- Znaczenie modelowania
- Czym jest modelowanie?
- Do czego służy modelowanie?

Czym jest model?

- Przedstawia interesujący nas fragment rzeczywistości w uproszczony, ale uporządkowany sposób;
- Pozwala lepiej zrozumieć złożoną rzeczywistość.

Czym jest modelowanie?

Modelowanie można określić jako ***tworzenie opisu obiektu/zjawiska*** rzeczywistego lub abstrakcyjnego, wykonywane w założonym celu. Jego ***efektem jest model***, który powinien posiadać określone własności. Z tego względu nie ma modeli złych, są tylko takie, które nie realizują założonego celu.

Znaczenie modelowania

- Wiele czynników ma wpływ na sukces producenta oprogramowania, czy też projektanta baz danych. **Jednym z najważniejszych jest tworzenie modeli**.
- Opracowujemy wszelkiego rodzaju modele, aby przyszli użytkownicy mogli zawczasu wyobrazić sobie gotowe rozwiązanie (np. oprogramowanie)

Do czego służy modelowanie?

Modele tworzone są głównie z dwóch powodów: dla ***lepszego zrozumienia*** i rozwiązania problemu oraz umożliwienia ***wymiany informacji***.
Celem modelowania jest rozpoznanie wszystkich czynników, które mogą wpłynąć na realizację projektu

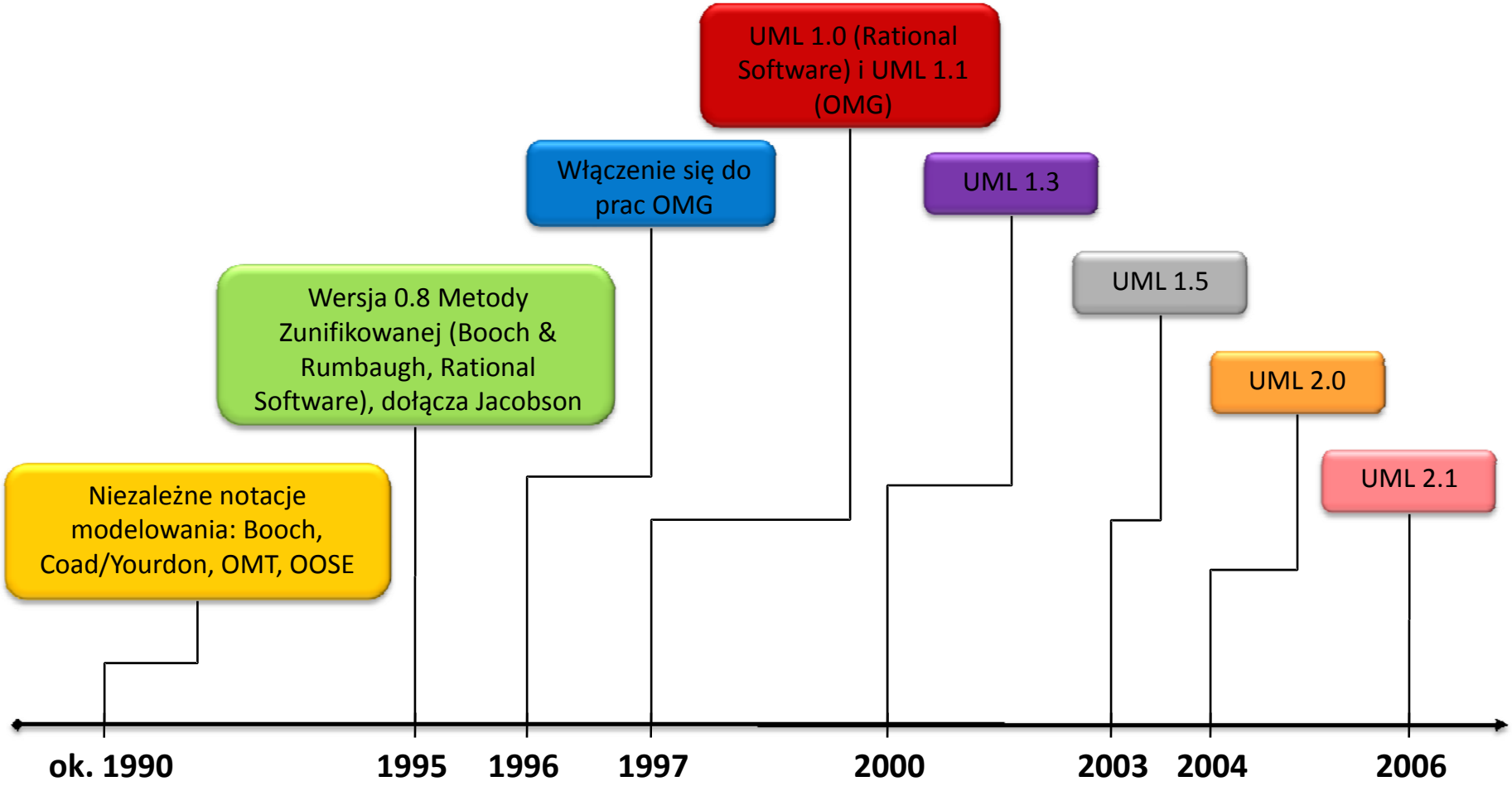
UML - zagadnienia

- Czym jest UML?
- Historia UML
- Zastosowanie języka UML
- Podstawowe elementy UML

Czym jest UML?

- UML (*Unified Modeling Language*) ***jest językiem modelowania***.
- UML ***jest standardem*** (Unified - zunifikowany, **jednolity**).
- UML ***opisuje, co system ma robić***, a nie jak ma to robić.

Historia UML



Zastosowanie języka UML

UML jest językiem przeznaczonym do:

- *obrazowania,*
- *specyfikowania i definiowania,*
- *konstruowania i tworzenia*,
- *dokumentowania.*

Podstawowe elementy UML

W UML można wyróżnić ***kilka grup elementów***, które zostały wykorzystane do budowy modelu:

- *strukturalne,*
- *czynnościowe,*
- *grupujące,*
- *komentujące,*
- *związki,*
- *diagramy*

Systematyka diagramów

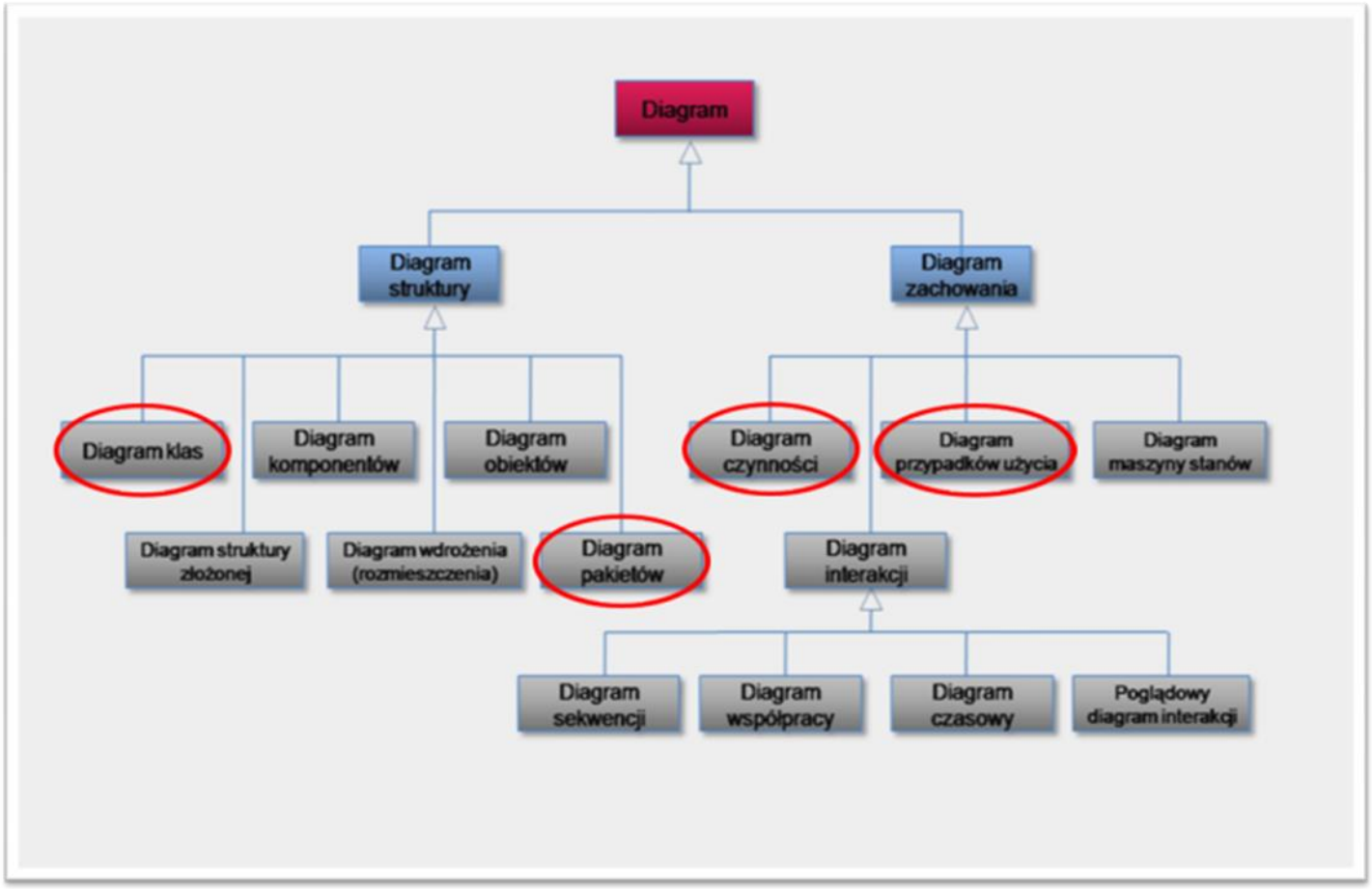
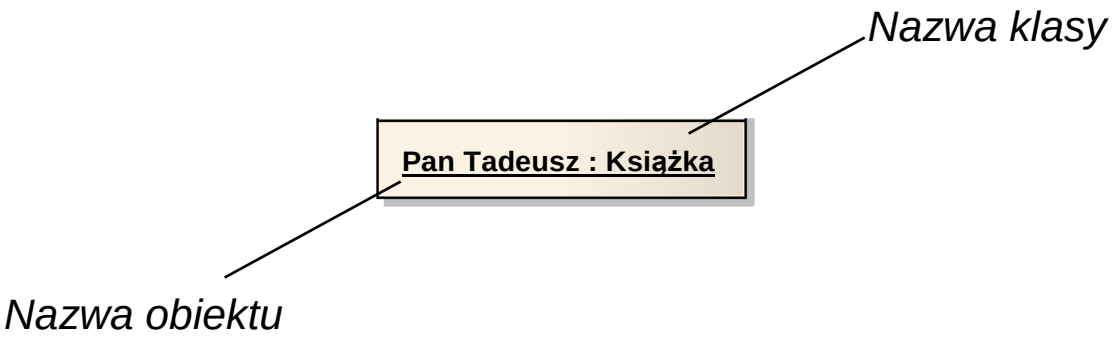


Diagram klas - zagadnienia

- Co to jest i do czego służy obiekt?
- Na czym polega obiektowe podejście?
- Co to jest i do czego służy klasa?
- Podstawowe elementy wchodzące w skład klasy
- Podstawowe związki między klasami
- Liczebność
- Dziedziczenie
- Co to jest klasa abstrakcyjna?
- Co to jest i do czego służy interfejs?
- Co to jest klasa asocjacyjna?

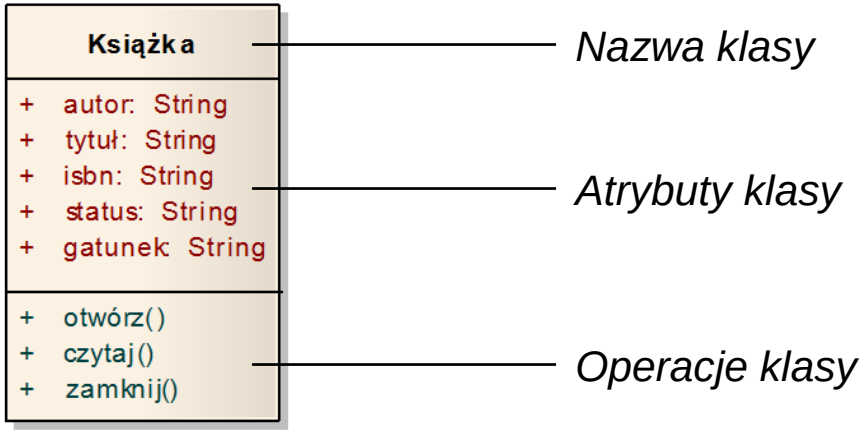
Czym jest obiekt?

- Konkretny lub abstrakcyjny ***byt wyróżnialny w modelowanej rzeczywistości*** posiadający nazwę (identyfikację), określone granice, atrybuty i operacje.
- Obiekt ***ma swoją tożsamość***, która wyróżnia go spośród innych obiektów
- Obiekt jest ***to każdy element mający swój odpowiednik w rzeczywistości***, coś, co ***ma stan i zachowanie***.



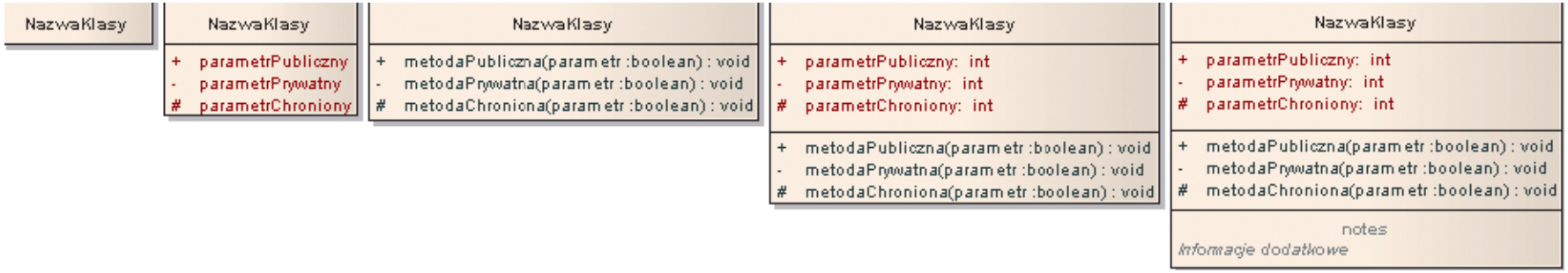
Czym jest klasa?

- Klasa ***jest uogólnieniem zbioru obiektów*** zawierających takie same atrybuty, operacje, związki i znaczenie.
- Klasa ***reprezentuje grupę obiektów*** o wspólnym stanie i zachowaniu.



Różne sposoby zapisu klasy

③ Każda klasa musi mieć przynajmniej nazwę, która wyróżnia ją spośród innych klas.



Dozwolone notacje prezentujące klasy na diagramach UML.

Zapis nazwy klasy

Wyróżniamy nazwy:

- *proste*
- *ścieżkowe*

Co to są atrybuty klasy?

- Atrybuty to ***właściwości klasy***;
- Określają ***zbiór wartości***, jakie można przypisać do poszczególnych egzemplarzy tej klasy;
- Klasa może mieć dowolną liczbę atrybutów lub nie mieć ich wcale;
- Atrybuty mogą być prostymi typami podstawowymi (liczby całkowite, liczby zmiennoprzecinkowe itd.) lub powiązaniami do innych bardziej skomplikowanych obiektów.

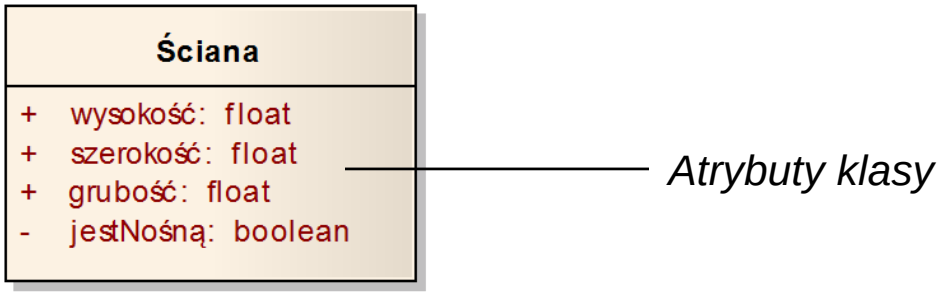
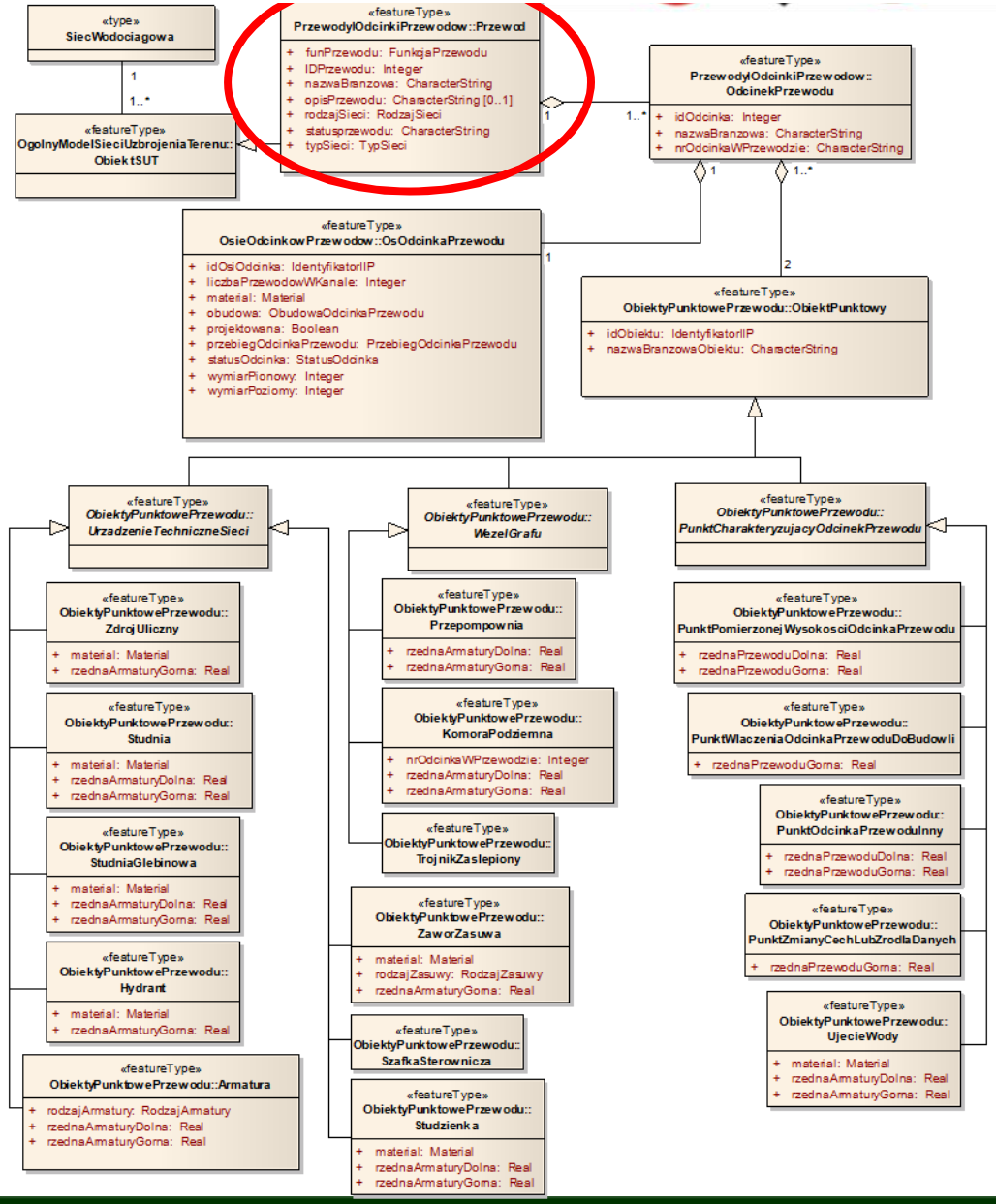


Diagram klas

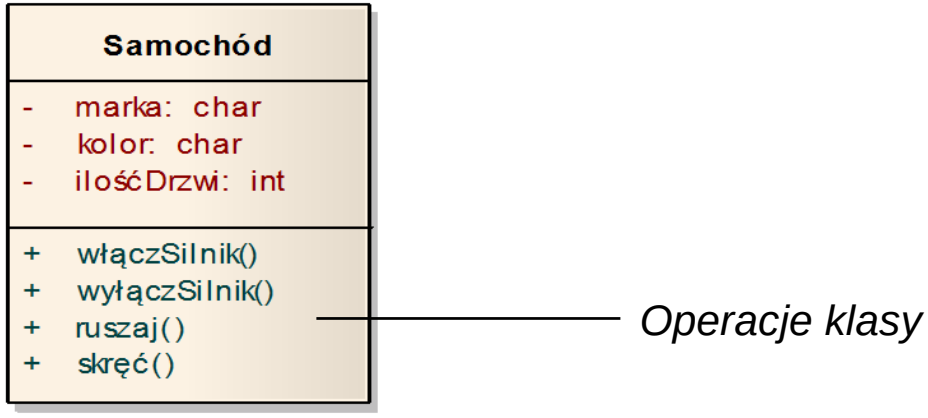


«featureType»
PrzewodyIOdcinkiPrzewodow::Przewod

+ funPrzewodu: FunkcjaPrzewodu
+ IDPrzewodu: Integer
+ nazwaBranzowa: CharacterString
+ opisPrzewodu: CharacterString [0..1]
+ rodzajSieci: RodzajSieci
+ statusprzewodu: CharacterString
+ typSieci: TypSieci

Co to są operacje klasy?

- Operacja to pewna ***usługa***, której wykonania można zażądać od każdego obiektu klasy.
- Jest to ***abstrakcja czegoś, co można zrobić*** z każdym obiektem tej klasy.
- Klasa może mieć dowolną liczbę operacji lub nie mieć ich wcale.

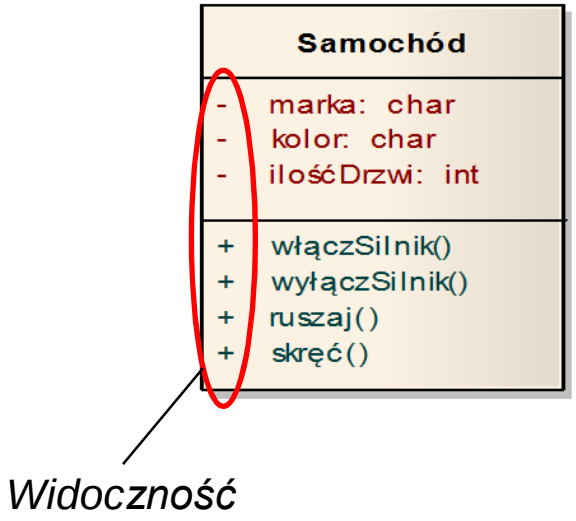


Widoczność

Widoczność to jedna z podstawowych cech, jakie można określić dla atrybutów i operacji klasy.

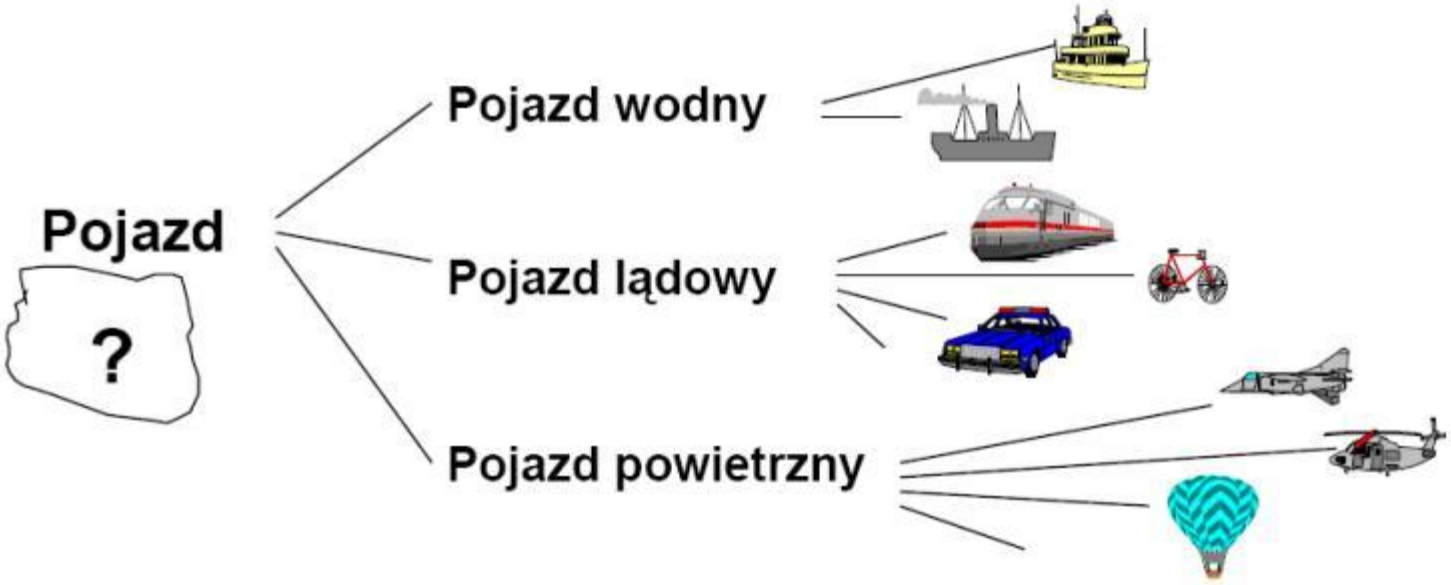
Widoczność określana jest w jeden z następujących sposobów:

- \+ - ***publiczny*** (ang. public)
- \# - ***chroniony*** (ang. protected)
- \- - ***prywatny*** (ang. private)
- ~ - ***pakietowy*** (ang. package)



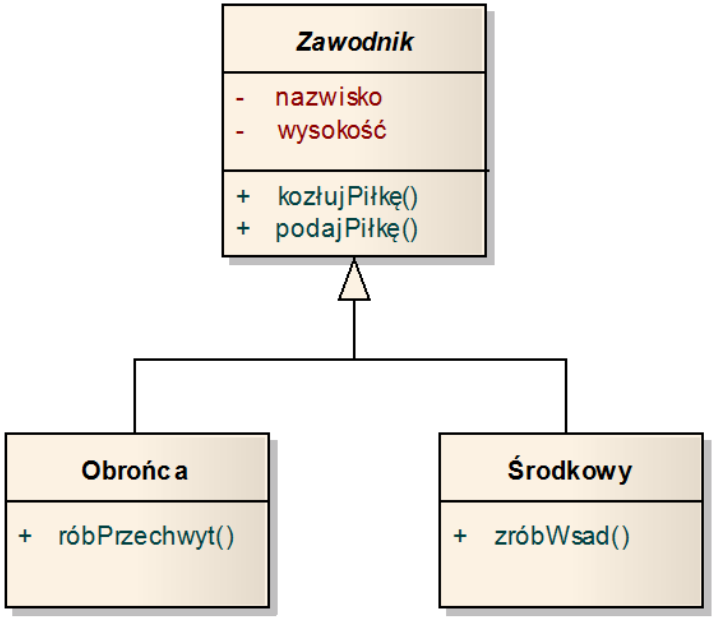
Czym jest dziedziczenie?

Hierarchiczne powiązanie między klasami, w którym ***klasy podrzędne przejmują (dziedziczą) wszystkie właściwości (atrybuty i operacje) klas nadrzędnych***, a ponadto mają właściwości specyficzne dla siebie.



Co to jest klasa abstrakcyjna?

- Klasy abstrakcyjne stanowią uogólnienie obiektów konkretnych znajdujących się na niższych poziomach hierarchii. ***Nie można tworzyć obiektów*** klas abstrakcyjnych.
- ***Nazwa*** klasy abstrakcyjnej musi być ***napisana kursywą*** (pochyłą czcionką).



Związki między klasami

Związek to relacja zachodząca miedzy klasami lub obiektami.

Podstawowe związki:

- ***Asocjacja*** – powiązanie
- ***Agregacja*** – szczegółowy przypadek powiązania
- ***Kompozycja –*** szczegółowy przypadek agregacji
- ***Generalizacja*** – uogólnienie
- ***Realizacja***

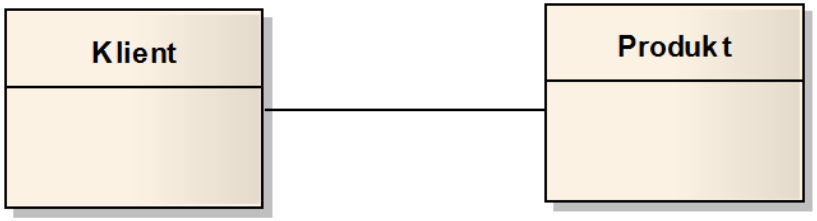
Asocjacja

Asocjacja to ***podstawowy związek między klasami***.

Asocjacja ***oznacza istnienie trwałego powiązania*** pomiędzy klasami.

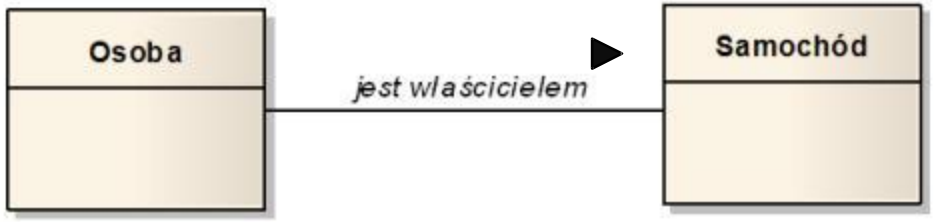
Przykłady asocjacji:

- firma zatrudnia pracowników,
- student studiuje na uczelni,
- klient zamawia produkt.

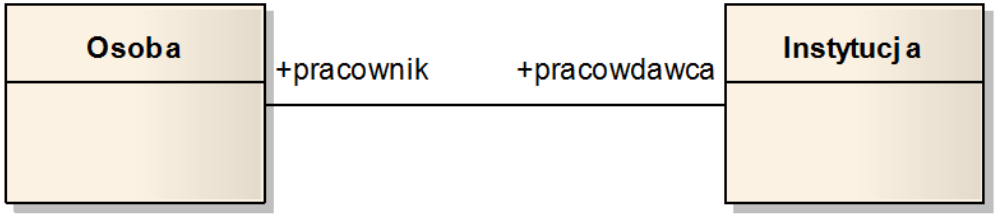


Nazwy i role asocjacji

Sposób powiązania obu klas jest określony poprzez nazwę znajdującą się nad związkiem. Przy nazwie asocjacji czasami można zaobserwować symbol strzałki określający kierunek interpretacji powiązania.



Role określają, jak sama nazwa wskazuje, jaką rolę pełni dana klasa w asocjacji.

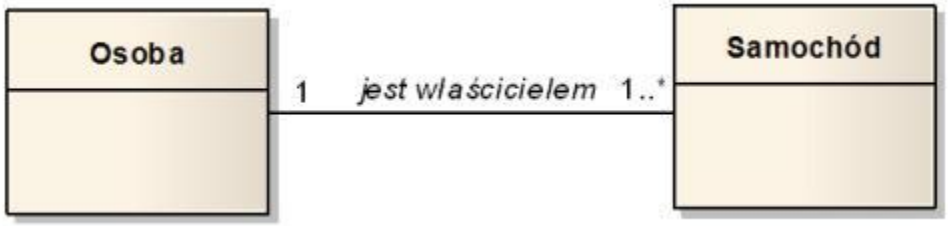


Liczebność

Liczebność ***określa możliwą ilość wystąpienia obiektów*** danej klasy biorących udział w danym związku.

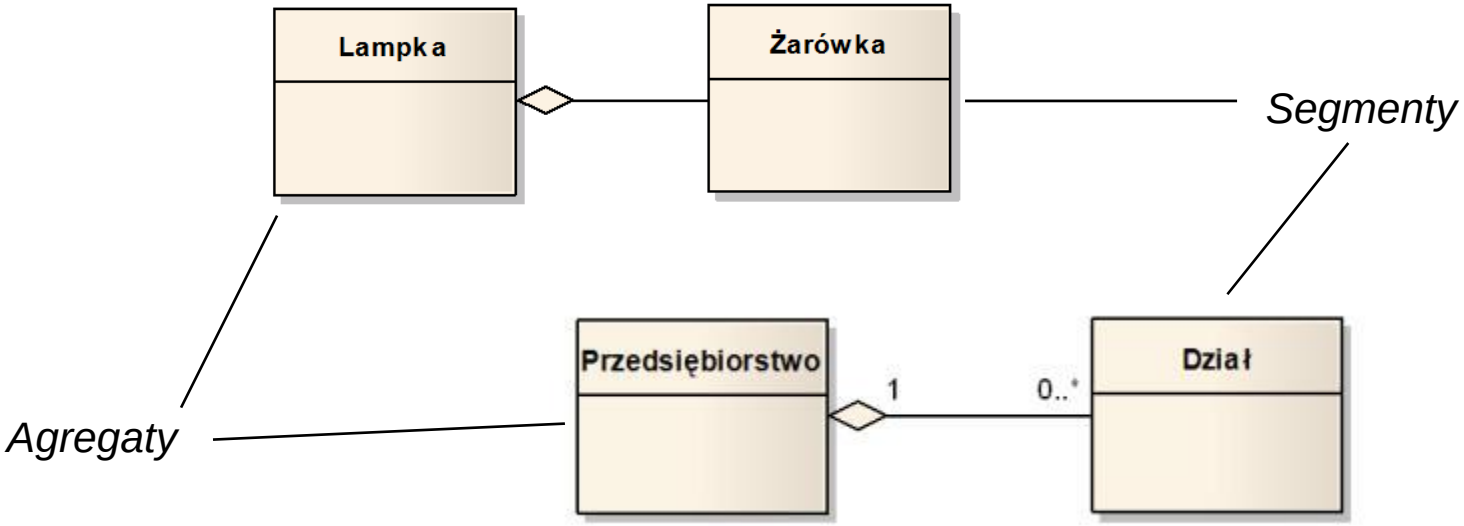
Sposoby przedstawienia liczebności:

- [1], [2], [10]
- [*], [0..*]
- [0..1], [1..5], [10..100], [1..*]
- [0..1, 3..4, 6..*], [1..5, 7, 9]



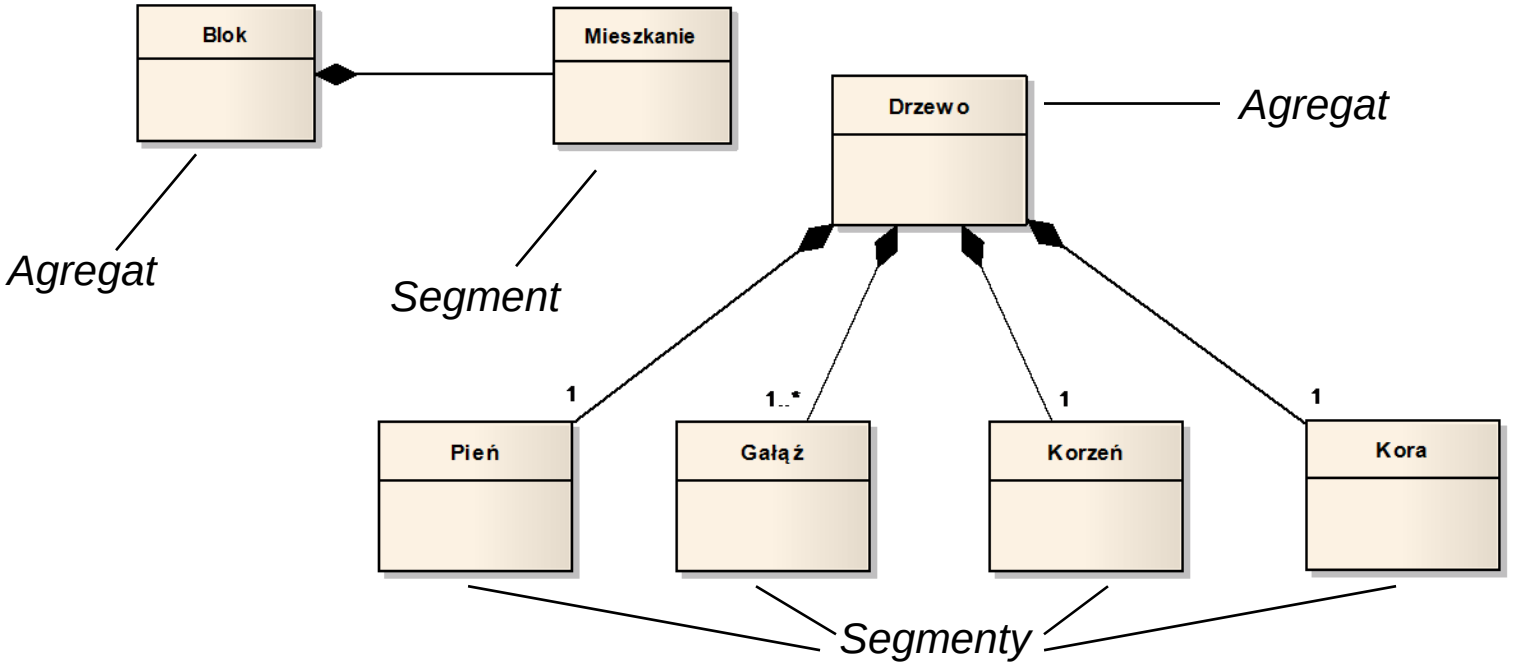
Agregacja

Agregacja jest to ***szczególny rodzaj asocjacji***, która ***określa związek między agregatem (całością), a składnikiem (częścią/segmentem)***. Dana część może należeć do wielu całości i jest od nich niezależna.

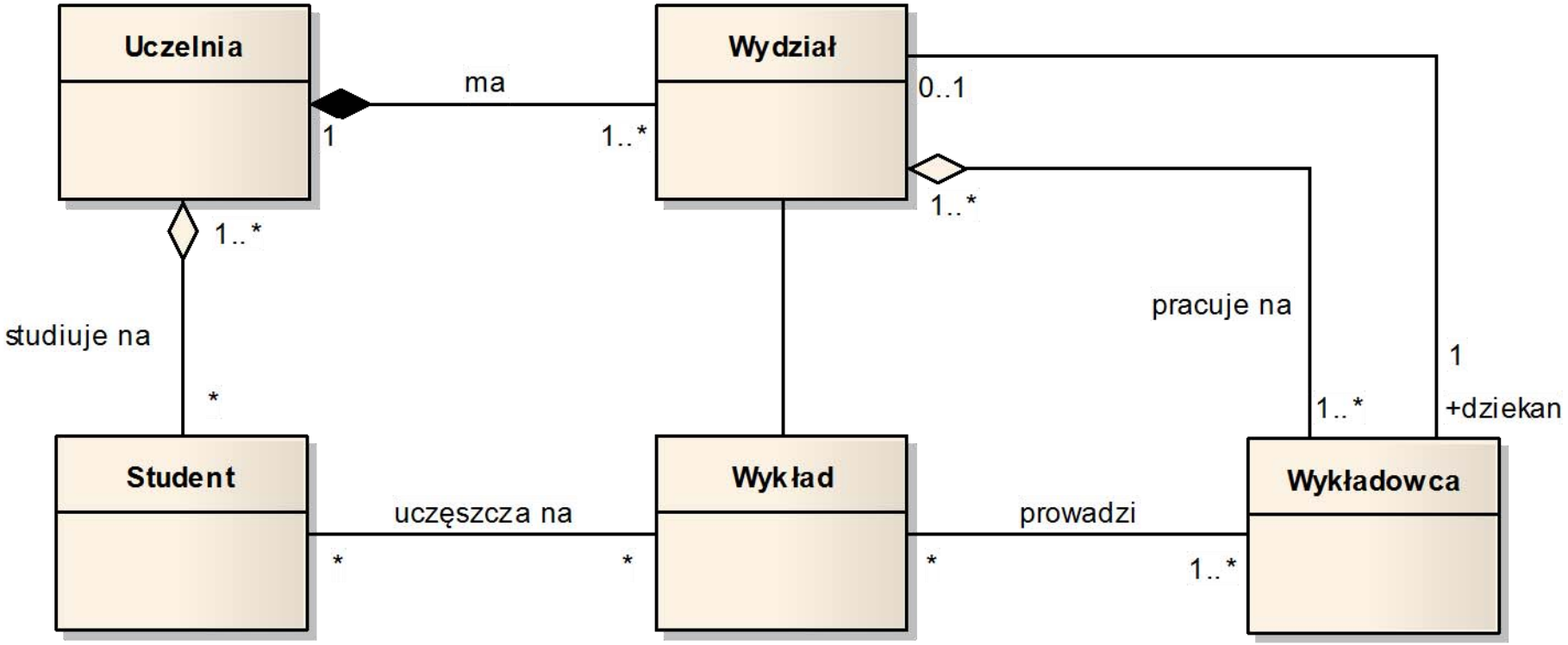


Kompozycja

Szczególnym przypadkiem agregacji jest kompozycja, która oznacza ***składanie się obiektu z obiektów składowych***, które nie mogą istnieć bez obiektu głównego. Kompozycja jest relacją typu "posiada".



Asocjacja, agregacja i kompozycja



Generalizacja

Hierarchiczne powiązania miedzy klasami, dzięki którym klasy podrzędne przejmują własności klas nadrzędnych.

Klasy nadrzędne, znajdujące się na najwyższym poziomie hierarchii określa się mianem korzenia i oznacza się je jako {root}, natomiast te najniższe są to liście {leaf}.

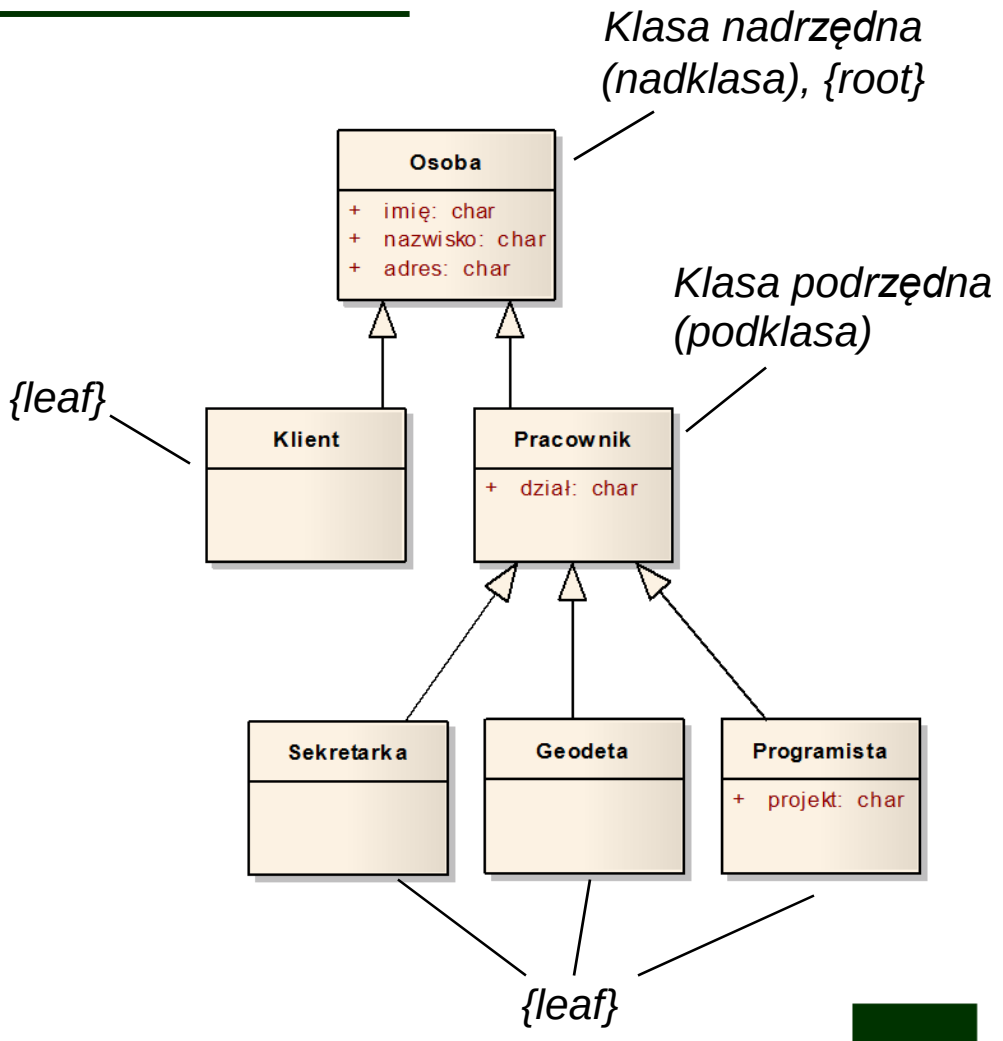
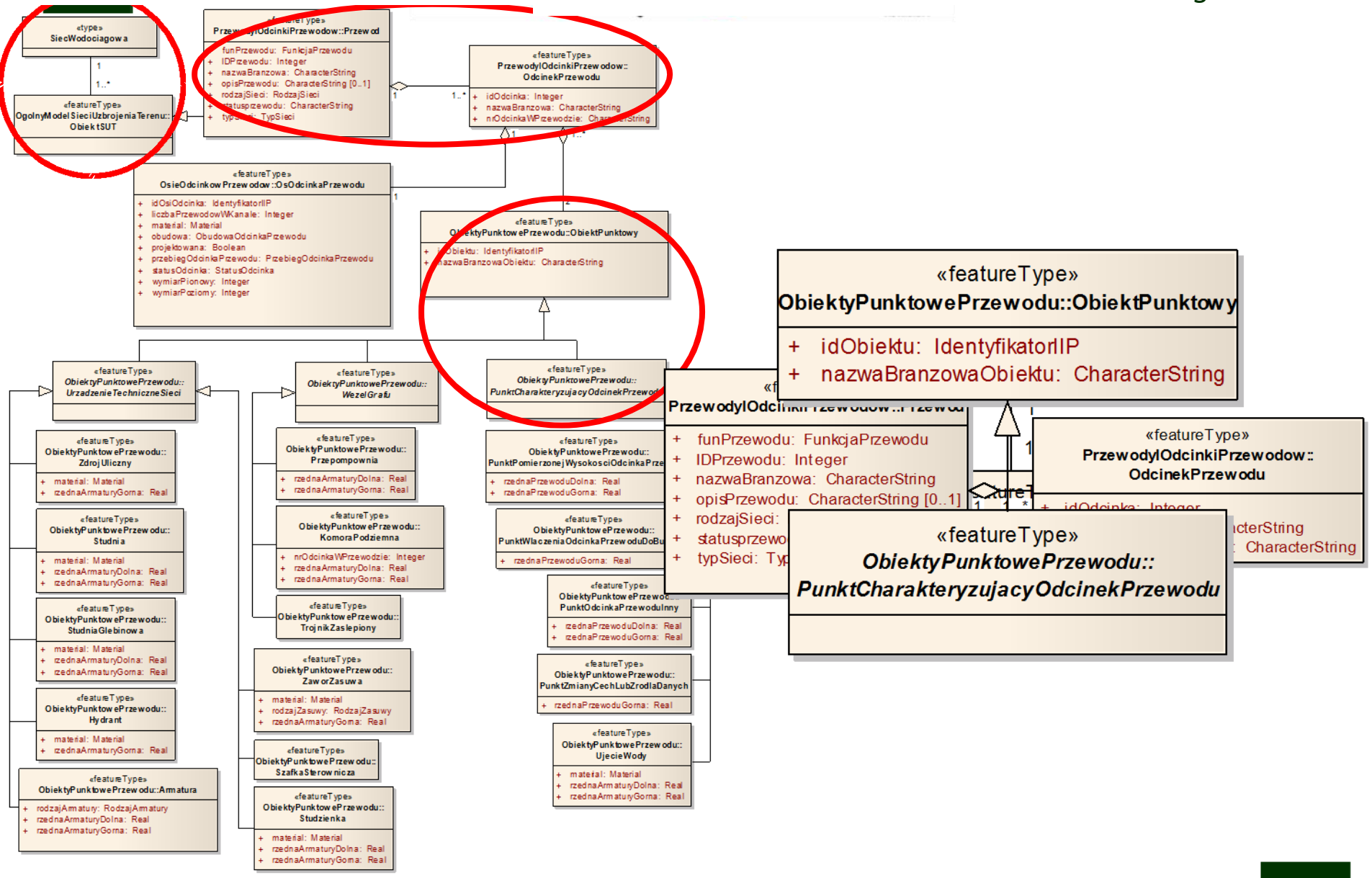
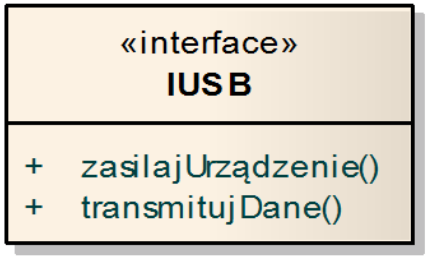


Diagram klas



Interfejsy

- W UML interfejsy ***są zestawem operacji***, które ***wyznaczają usługi oferowane przez klasę*** i sposobem na przejrzystą prezentację projektu.

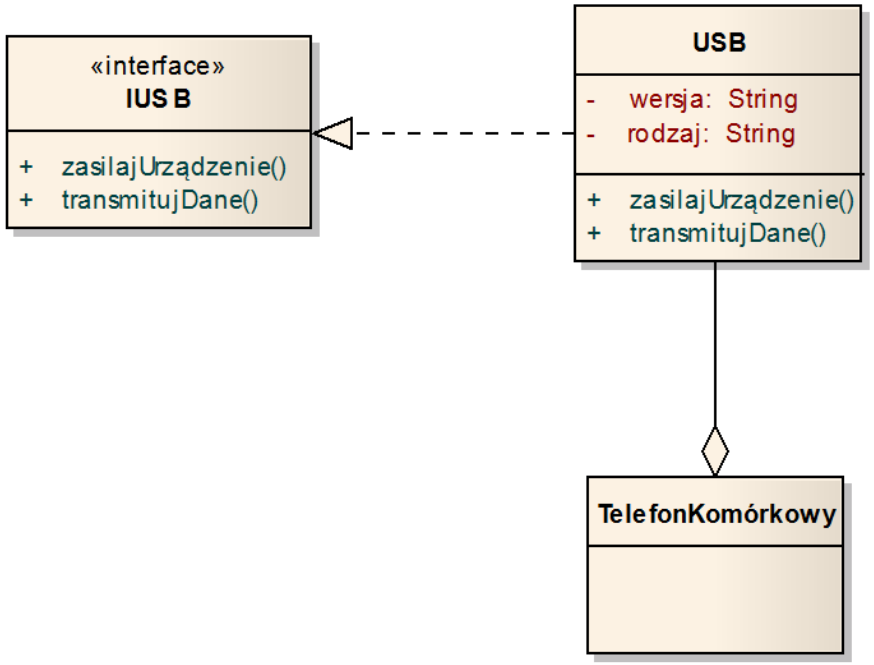


Typowy interfejs

Interfejs – symbol kuli

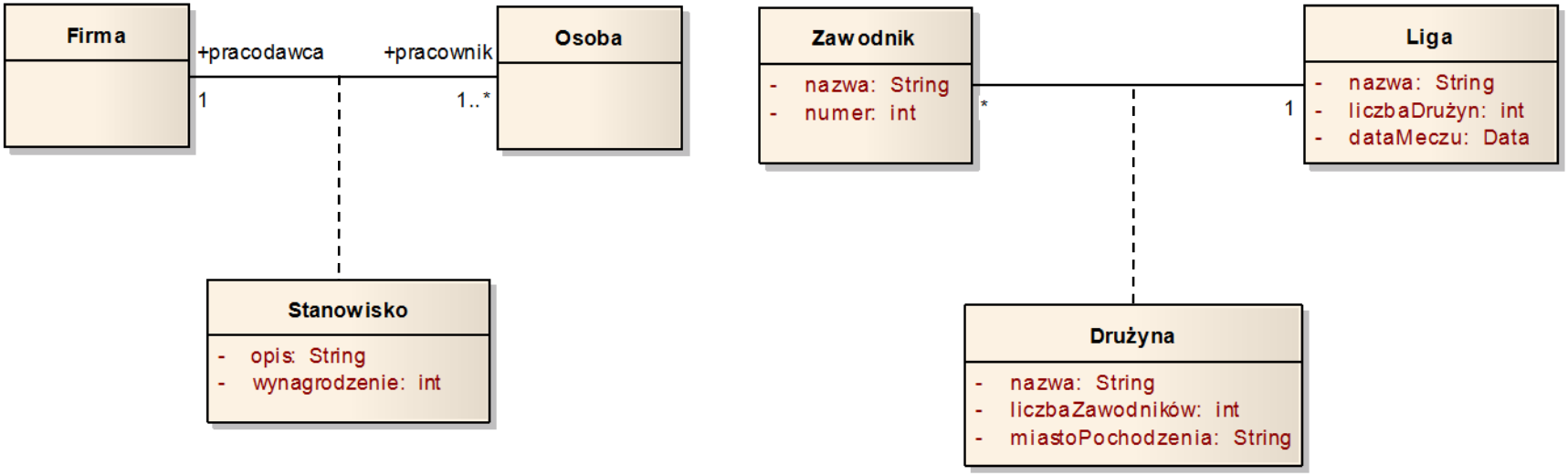
Realizacja

Przedstawia związek znaczeniowy między obiektami diagramu UML (klasami) z których jeden określa kontrakt, a drugi zapewnia wywiązanie się z niego.



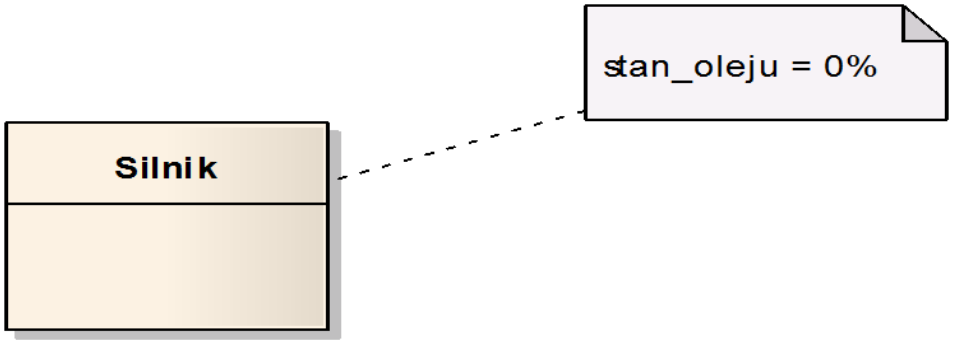
Związek jako klasa – klasa asocjacyjna

- Klasa asocjacyjna jest to byt o właściwościach zarówno klasy, jak i powiązania. Można go postrzegać jako powiązanie mające też właściwości klasy lub jako klasę mającą też właściwości powiązania.



Notatka

Notatka to odpowiednik „żółtej karteczki” przylepionej gdzieś w widocznym miejscu. Notatki pozwalają na dopisywanie komentarzy, ograniczeń i wymagań.



Czym jest diagram klas? - podsumowanie

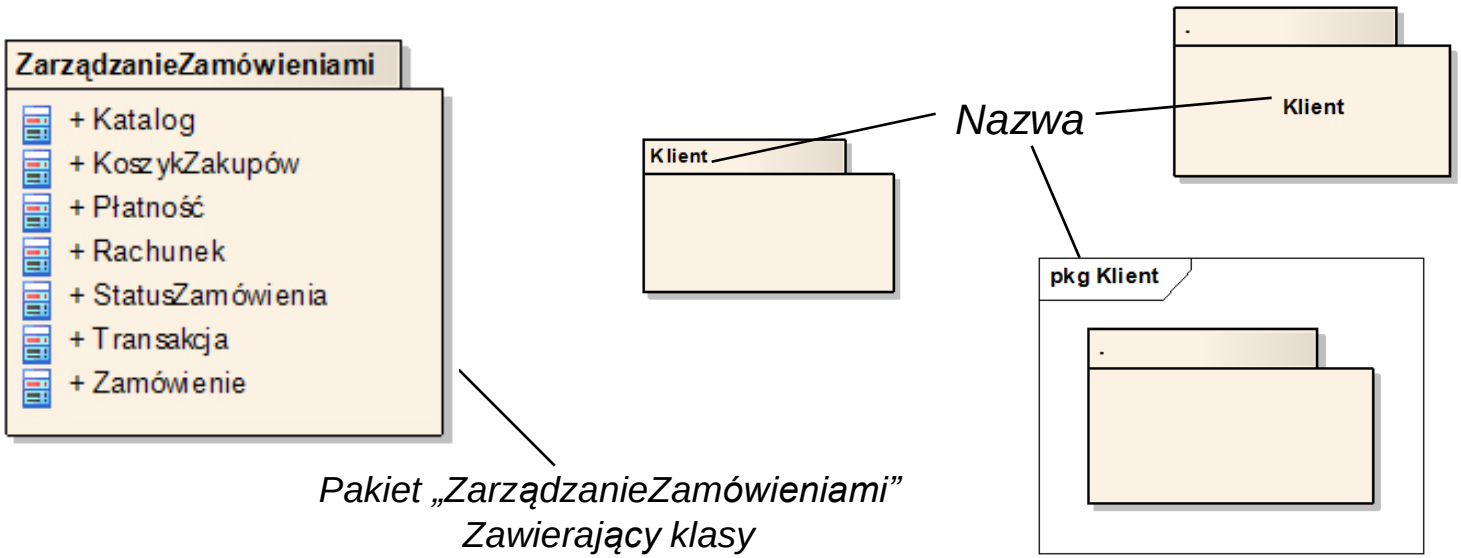
- Podstawowym i najczęściej wykorzystywanym diagramem języka UML.
- Jest wykorzystywany do modelowania statycznych relacji pomiędzy komponentami systemu.
- Ukazuje wzajemne powiązania między klasami tworzącymi dany system, ale nie ukazuje żadnych relacji pomiędzy samymi obiektami.
- Pojedynczy model UML może posiadać wiele diagramów klas pokazujących ten sam system z wielu perspektyw.

Diagram pakietów - zagadnienia

- Czym jest pakiet?
- Czym jest diagram pakietów?
- Elementy diagramu pakietów
- Jakie zastosowanie znajduje diagram pakietów?

Czym jest pakiet?

- Pakiet to mechanizm ogólnego stosowania, służący do organizowania elementów w grupy.
- Dobrze zaprojektowane pakiety składają się z podobnych znaczeniowo i razem zmieniających się bytów.
- Są luźno powiązane ze sobą, ale silnie spójne wewnętrznie



Pakiet „ZarządzanieZamówieniami” Zawierający klasy

Czym jest diagram pakietów?

Diagram pakietów to przedstawienie logicznej struktury systemu w postaci zestawu pakietów połączonych zależnościami i zagnieżdżeniami. Diagram pakietów umożliwia sklasyfikowanie i pogrupowanie elementów takich jak klasy, przypadki użycia itp.

Elementy diagramu pakietów

Podstawowymi elementami diagramu pakietów są:

- Pakiet – zbiór (grupa) elementów modelu.
- Zależności stereotypowe:
  - <<import>>
  - <<access>>
  - <<merge>>

Zależności

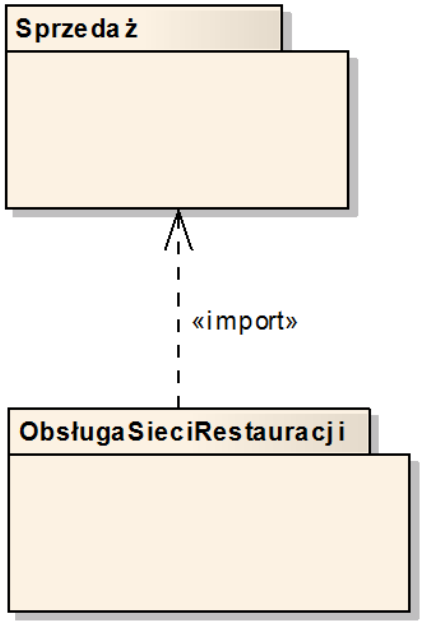
- Zależność ***to związek użycia***.
- Zależności ***mogą być nieokreślone lub należeć do pewnej szczególnej grupy określonej*** przez umieszczony obok linii zależności stereotyp.

Trzy stereotypy do dokładniejszego określenia zależności:

- ***Import*** (import)
- ***Access*** (dostęp)
- ***Merge*** (scalenie)

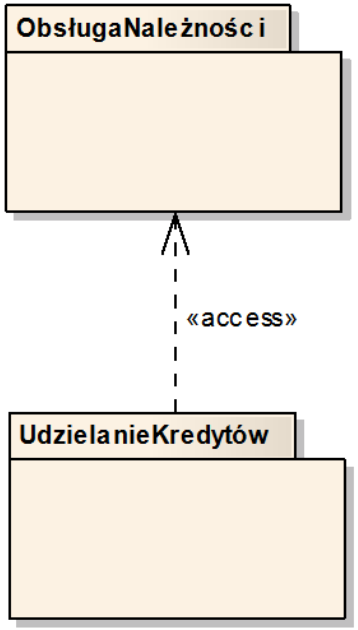
Zależności c.d. - import

③ ***Import*** (import) – określa, że pakiet włącza elementy publiczne innego pakietu do własnej przestrzeni nazw.



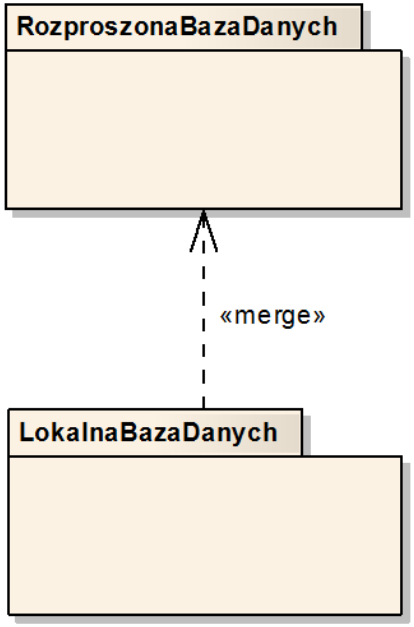
Zależności c.d. - access

- ***Access*** (dostęp) – wskazuje, że pakiet korzysta z publicznych elementów innego pakietu. Każdy element musi być w pełni kwalifikowany poprzez wykorzystanie dłuższej, rozdzielanej dwukropkami nazwy ścieżki.



Zależności c.d. - merge

③ ***Merge*** (scalenie) – wskazuje, że zależny pakiet jest pakietem zaprzyjaźnionym. Pakiet zaprzyjaźniony ma dostęp do wszystkich elementów drugiego pakietu, niezależnie od ich widoczności.



Jakie znajduje zastosowanie?

- Ukrywanie mniej istotnych elementów modelu.
- Ułatwienie podziału prac miedzy członkami zespołu/różnymi zespołami.
- Wizualizacja podstawowych zależności pomiędzy częściami systemu.
- Tworzenie modeli poglądowych dla modeli zawierających wiele elementów.
- Organizacja (uporządkowanie) wielkich modeli.
- Grupowanie elementów.

Diagram przypadków użycia - zagadnienia

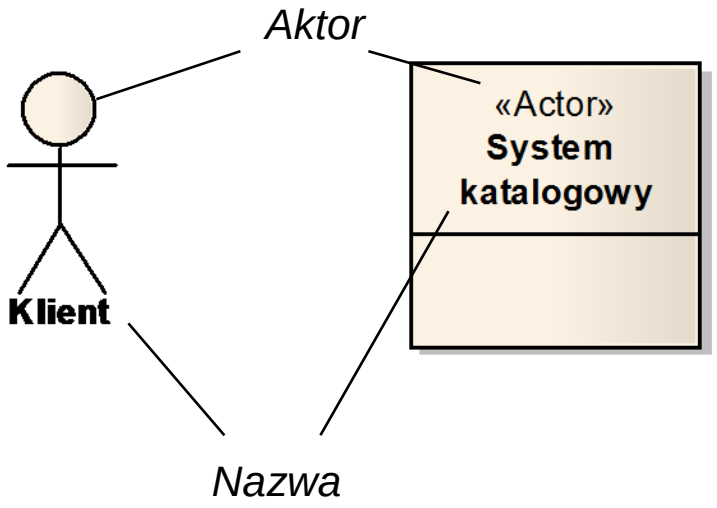
- Czym jest diagram przypadków użycia?
- Aktor
- Identyfikacja aktorów
- Klasyfikacja aktorów
- Co to jest przypadek użycia?
- Identyfikacja przypadków użycia
- Związki w diagramie przypadków użycia
- Asocjacja, zawieranie, rozszerzanie
- Uogólnienie
- Liczebność w diagramie przypadków użycia

Czym jest diagram przypadków użycia?

Diagram przypadków użycia w języku UML ***służy do modelowania funkcjonalności systemu***. Tworzony jest zazwyczaj w początkowych fazach modelowania. Diagram ten ***stanowi tylko przegląd możliwych działań w systemie***.

Aktor

- Aktor (ang. actor) - ***abstrakcyjny użytkownik systemu, reprezentujący grupę rzeczywistych użytkowników*** o podobnych funkcjach i sposobie komunikacji z systemem.
- Najczęściej aktor ***jest sprawcą zdarzenia powodującego uruchomienie przypadku użycia***.



Identyfikacja aktorów

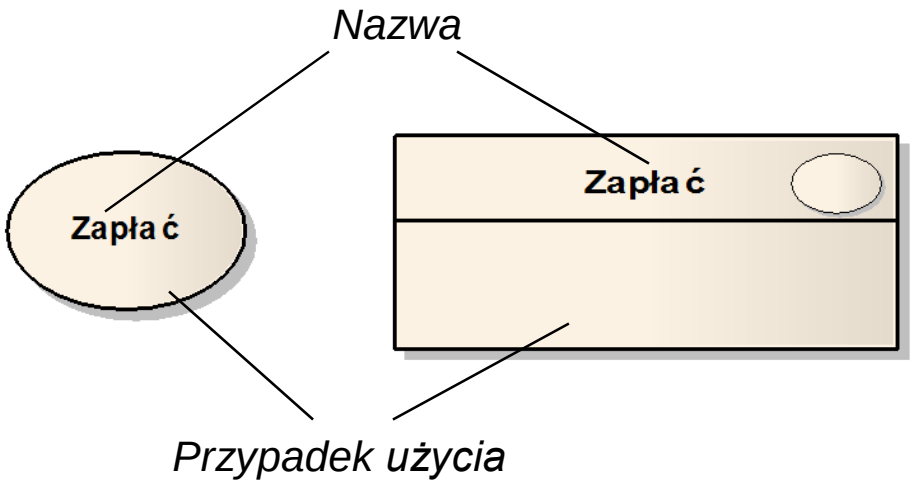
- Kto komunikuje się z systemem?
- Kto będzie korzystał z funkcji systemu?
- Kto będzie system pielęgnował?
- Jakie urządzenia system obsługuje? *(aktorzy nieożywieni)*
- Z jakimi innymi systemami system się komunikuje? *(aktorzy będący innymi systemami)*
- Kto lub co jest zainteresowane wynikami pracy systemu?

Klasyfikacja aktorów

- Aktor ***główny*** – korzysta z podstawowych funkcji systemu
- Aktor ***drugorzędny*** – korzysta głównie z funkcji służących do realizacji zadań pomocniczych *(np. administrowania i pielęgnacji systemu)*
- Aktor ***aktywny*** – inicjuje przypadek użycia
- Aktor ***pasywny*** – nie inicjuje przypadku użycia, lecz tylko w nim uczestniczy
- Aktor ***ożywiony/osobowy*** – reprezentacja ludzi, grupy ludzi
- Aktor ***nieożywiony/bezosobowy*** – reprezentuje system lub urządzenie

Co to jest przypadek użycia?

Przypadek użycia (ang. use case) - ***jednostka funkcjonalności dostarczana przez system***, która jest realizowana jako ciąg interakcji pomiędzy aktorem a systemem.



Identyfikacja przypadków użycia

- Czy aktor musi pamiętać, tworzyć, usuwać, modyfikować informacje w systemie?
- Czy aktor ma być powiadamiany o zdarzeniach w systemie, i na odwrót?

Związki w diagramie przypadków użycia

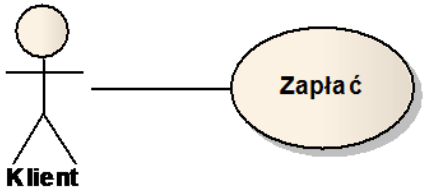
Związek w diagramie przypadków użycia to relacja zachodząca miedzy przypadkami użycia, aktorami lub przypadkami użycia a aktorami.

Podstawowe związki:

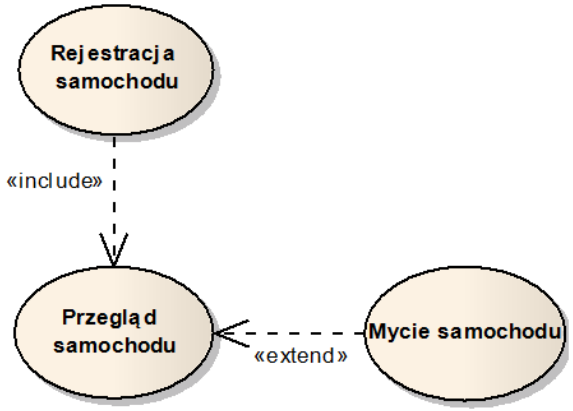
- ***Asocjacja***
- ***Zawieranie***
- ***Rozszerzanie***
- ***Uogólnienie***

Asocjacja, zawieranie, rozszerzenie

- ***Asocjacja*** - dwukierunkowej komunikacji pomiędzy przypadkiem użycia a aktorem.

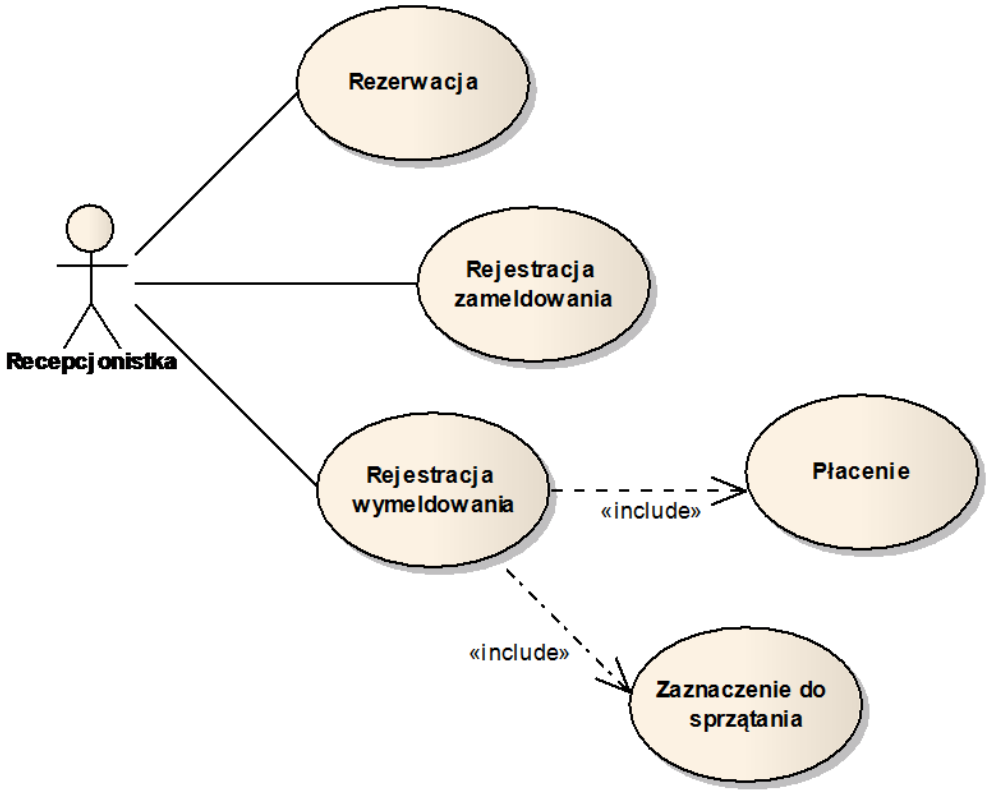


- ***Zawieranie*** <<include>> - służy do modelowania fragmentów przypadku użycia postrzeganych przez użytkownika jako konieczne zachowanie systemu.

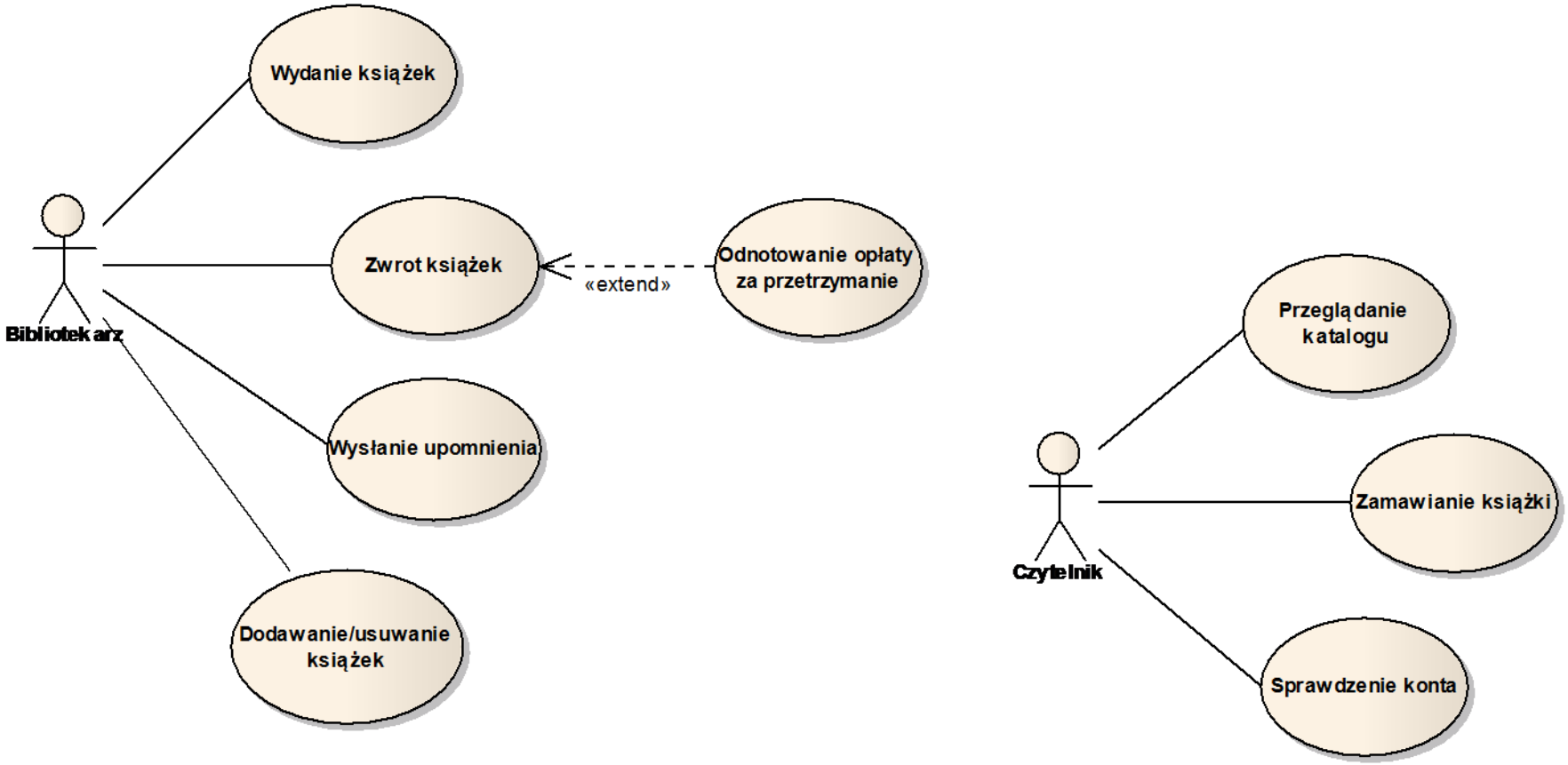


- ***Rozszerzanie*** <<extend>> - służy do modelowania fragmentów przypadku użycia postrzeganych przez użytkownika jako opcjonalne zachowanie systemu.

Asocjacja, zawieranie, rozszerzenie cd.

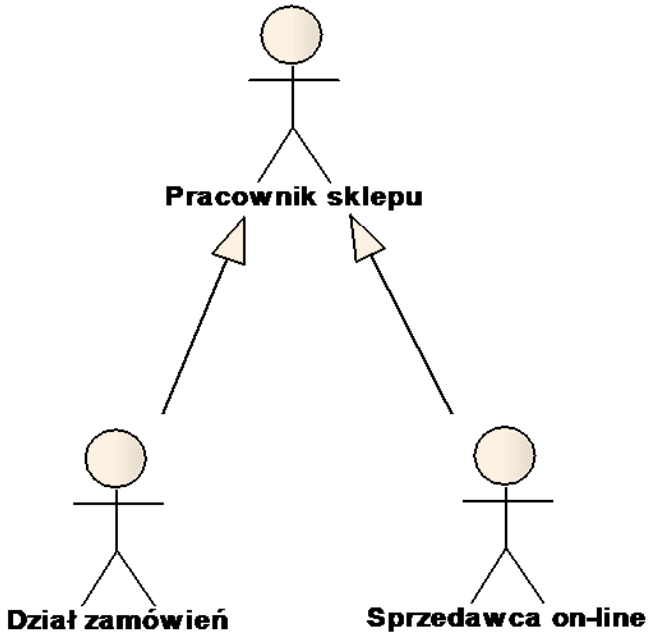


Asocjacja, zawieranie, rozszerzenie cd.



Uogólnienie

Uogólnienie znajduje zastosowanie gdy jeden przypadek użycia może być szczególną odmianą innego, już istniejącego przypadku użycia.



Liczebność

- Typy liczebności są analogiczne do tych występujących w diagramach klas.
- Sposoby przedstawienia liczebności:
  - [1], [10], itd.
  - [*]
  - [0..3], [10..100], [1..*]
  - [1..5, 7, 9]

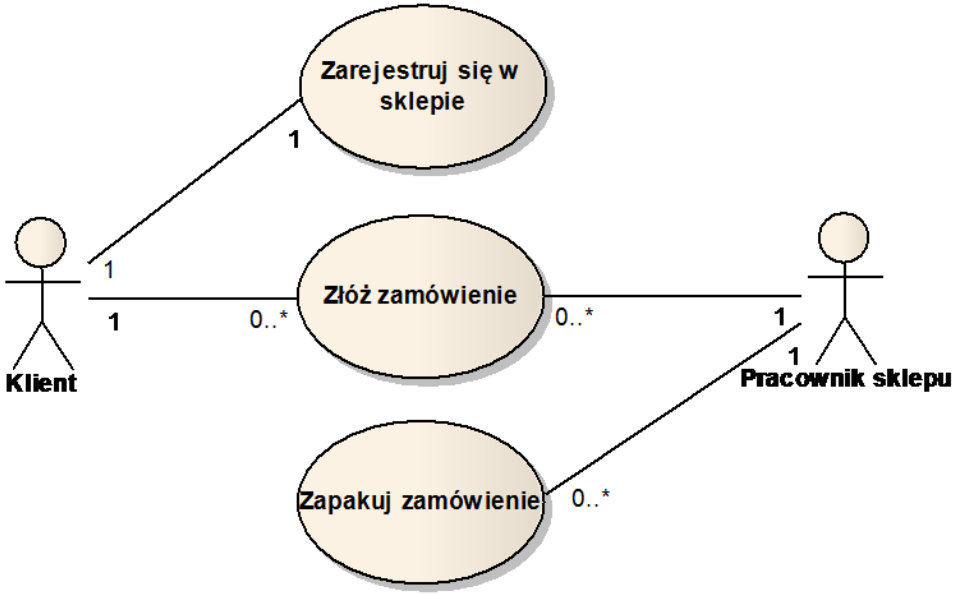


Diagram czynności - zagadnienia

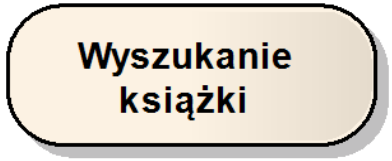
- Czym jest diagram czynności?
- Podstawowe elementy
- Czynność a akcja
- Współbieżność

Czym jest diagram czynności?

Diagram czynności (ang. activity diagram) ***dotyczy jednego lub wielu obiektów***. ***Opisuje czynności i kolejności ich realizowania*** przez obiekty oraz ***reprezentuje interakcję z punktu widzenia wykonywanej pracy***.

Podstawowe elementy

- ***Prostokąt o zaokrąglonych rogach*** przedstawiony poniżej jest graficzną interpretacją czynności lub akcji wykonywanej przez system.
- ***Romby*** reprezentują decyzje.



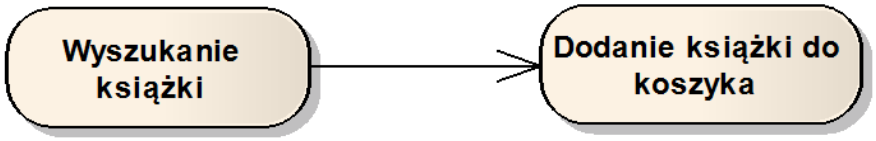
Czynność/akcja

Decyzja

- Początkowy i końcowy stan akcji, przedstawiane są odpowiednio jako ***zamalowane koło*** oraz ***koło z kropką***. Trzeci z poniższych symboli – ***przekreślone koło*** - oznacza zatrzymanie akcji.

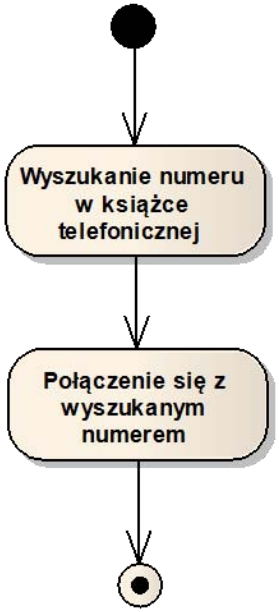
Podstawowe elementy c.d.

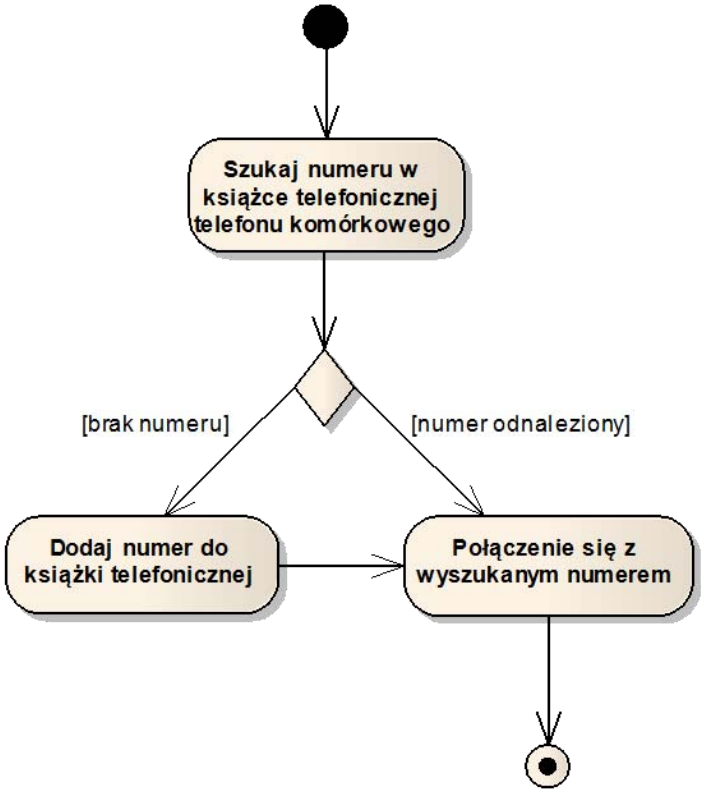
③ ***Strzałki*** określają przepływ sterowania między czynnościami. Zakończenie jednej czynności powoduje rozpoczęcie drugiej.



Przepływ sterowania

Przykłady diagramu czynności





Czynność a akcja

- ***Czynność*** jest bardziej ogólnym pojęciem, w związku z czym ***jest podzielna*** i charakteryzuje się ***dłuższym czasem wykonywania***.

- ***Akcja*** jest pojęciem szczegółowym, a co za tym idzie, ***niepodzielnym***, o ***krótkim czasie realizacji***.

Współbieżność

- To ***jednoczesne wykonywanie kilku czynności***.
- Elementy wykonywane współbieżnie wyróżnia się ***pogrubionymi poziomymi liniami***, która je niejako grupują.