

UML - zarys

2007/2008

Modelowanie

- Jest ważne przy tworzeniu wysokiej jakości oprogramowania
- Jest przydatne przy tworzeniu i analizie działania organizacji
- Modelujemy aby:
 - Zrozumieć system
 - Określić pożądaną strukturę i zachowania
 - Określić architekturę i móc ją zmieniać
 - W celu zarządzania ryzykiem

UML - historia

- Prace nad UML rozpoczęły się w 1994
- Pierwsze w miarę kompletne wersje w 1996
- Wersja 1.0 w 1997. Wersja ta została później przekazana do Object Management Group (OMG).
- Wersja 1.5 stała się oficjalna w 2003
- Wersja 2.0 została wprowadzona w 2003. Jest to pierwsza istotna zmiana standardu, wprowadzająca liczne nowe diagramy i kategorie modelowania.

Diagramy UML

- Model w UML jest graficzną reprezentacją systemu
- Reprezentacja składa się z logicznie połączonych diagramów
- Wersja 2.0 zawiera 13 typów diagramów
- Istotnym pojęciami są pojęcia **klasyfikatora** (classifier) – abstrakcyjnej kategorii która uogólnia kolekcję wystąpień mających te same cechy, oraz **wystąpienia** (instance) – egzemplifikacji klasyfikatora

Diagramy przypadków użycia

- Umożliwiają:
 - Identyfikację i dokumentację wymogów
 - Analizę zakresu aplikacji
 - Komunikację pomiędzy twórcami, właścicielami, klientami itd.
 - Opracowanie projektu przyszłego systemu, organizacji
 - Określenie procedur testowych dla systemu

Diagramy przypadków użycia

- Zawierają:
 - Przypadki użycia
 - Aktorów
 - Związki (relacje)

Przypadki użycia

- Specyfikacja sekwencji akcji (i ich wariantów) które system może wykonać poprzez interakcję z aktorami tego systemu
- Przypadek użycia jest spójnym fragmentem funkcjonalności systemu
- Nazwą jest rozkaz wypełnienia określonej funkcji. Nazwa jest umieszczona w owalu



Zweryfikuj
użytkownika

Sprawdź
hasło

Aktor

- Aktor jest spójnym zbiorem ról odgrywanych przez użytkowników podczas interakcji z przypadkami użycia
- Aktorami mogą być:
 - Osoby (pojedyncza osoba, grupa, organizacja itp.)
 - Zewnętrzne systemy (programowe bądź sprzętowe)
 - Czas
- Nazwa to rzeczownik odzwierciedlający rolę odgrywaną w systemie
- Aktor może używać wielu przypadków użycia, przypadek użycia może być używany przez wielu aktorów

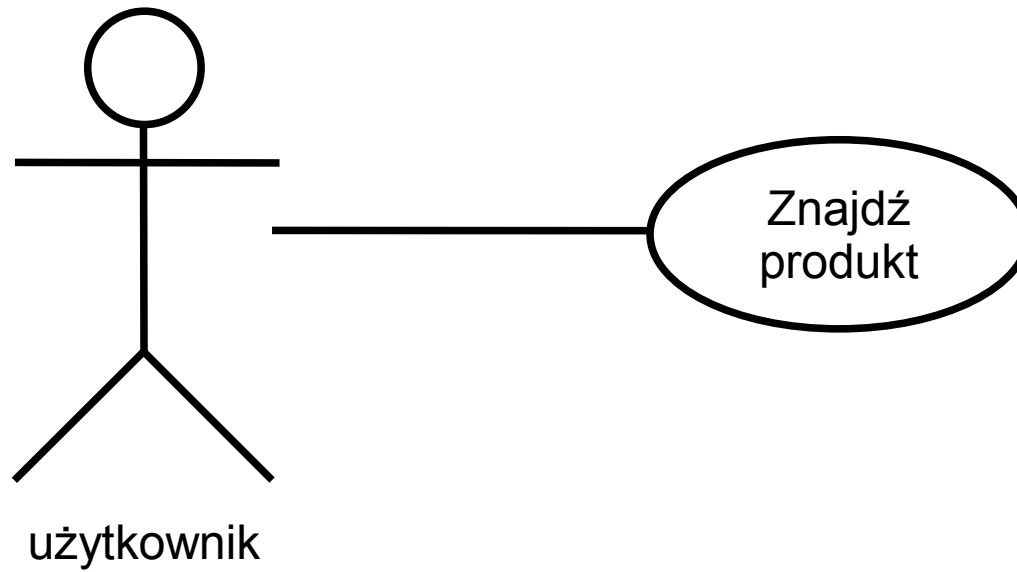
Związek (relationship)

- Wiąże ze sobą elementy diagramu (np. aktorów i przypadki użycia)
- Są 4 rodzaje związków:
 - Asocjacja (association)
 - Uogólnienie (generalisation)
 - Zależność (dependence)
 - Realizacja (realisation)

Asocjacja

- Asocjacja opisuje związek pomiędzy (dwoma lub więcej) wystąpieniami klasyfikatorów
- W diagramie przypadków użycia reprezentuje dwukierunkową komunikację pomiędzy aktorem i przypadkiem użycia
- Jej reprezentacją graficzną jest ciągła linia
- Zazwyczaj nie są nazywane

Asocjacja



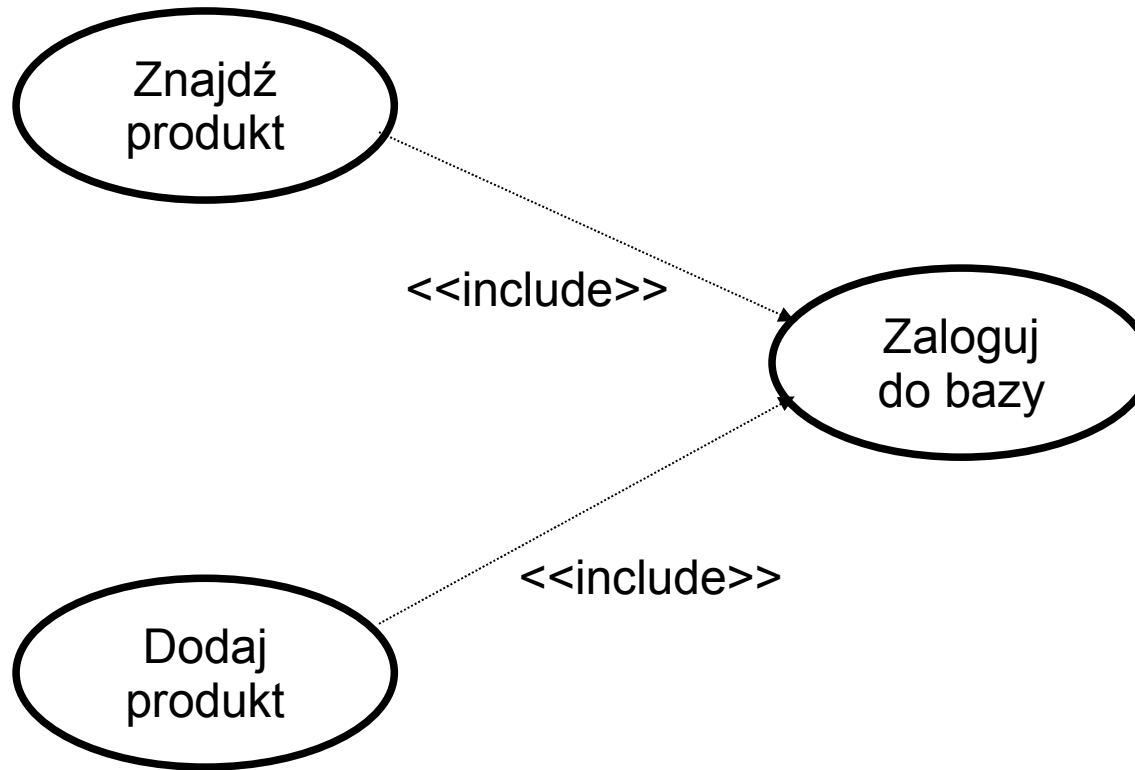
Zależność (dependency)

- Zależność jest związkiem pomiędzy dwoma elementami modelu gdzie zmiana w jednym elemencie (niezależnym one) ma wpływ na drugi element (zależny)
- Jest obrazowana jako linia przerywana zakończona otwartą strzałką
- W diagramach przypadków użycia zależność jest stereotypowana w:
 - Zależność <<include>>
 - Zależność <<extend>>

Zależność <<include>>

- Związek między przypadkiem zawierającym i zawieranym
- Przypadek zawierany jest wykonywany zawsze gdy wykonywany jest przypadek zawierający – i tylko wtedy
- Jest przydatna gdy kilka przypadków użycia zawiera tę samą wspólną część
- Strzałka skierowana jest od przypadku zawierającego do zawieranego

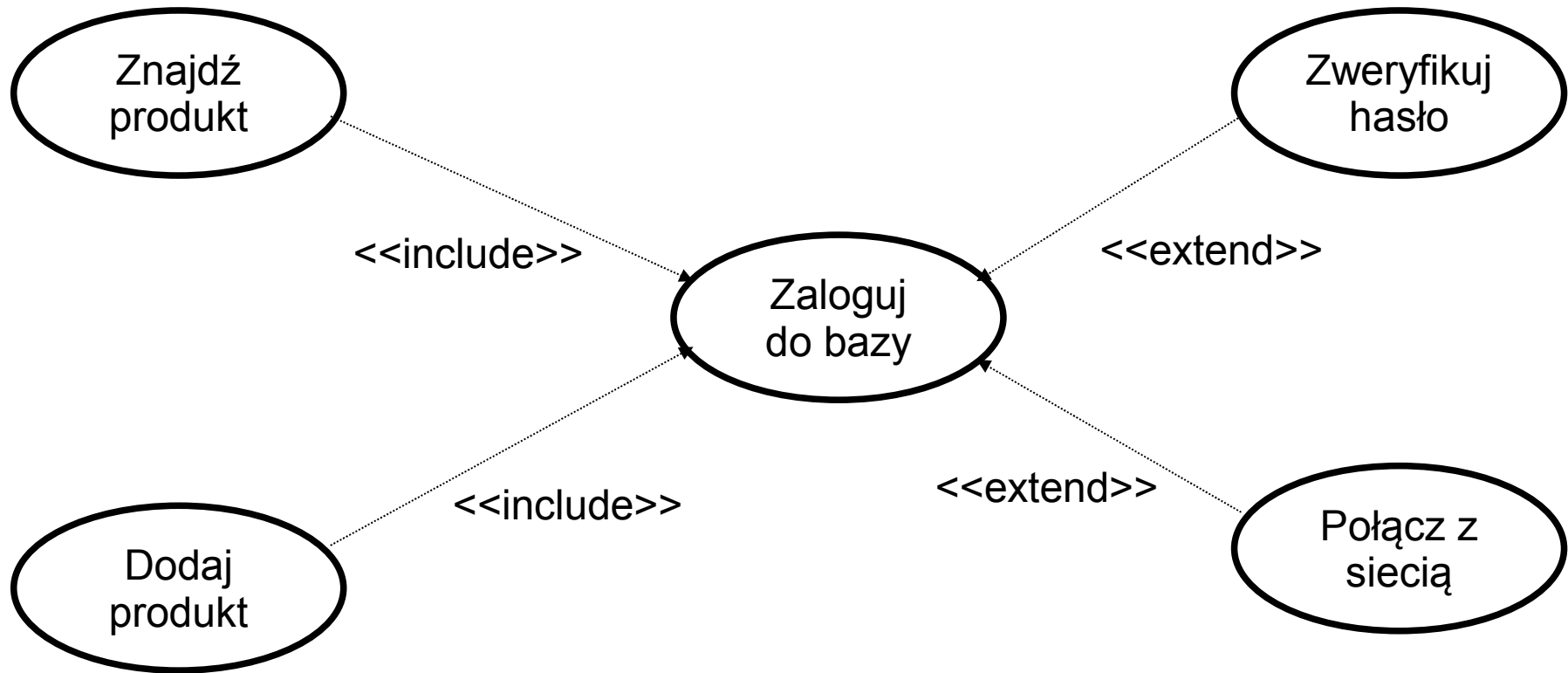
Zależność <<include>>



Zależność <<extend>>

- Zależność między przypadkiem podstawowym i przypadkiem który opcjonalnie może wprowadzić dodatkową funkcjonalność do przypadku podstawowego
- Jest przydatna gdy przypadek może, w pewnych warunkach, być uzależniony od innych przypadków
- Strzałka wskazuje od rozszerzenia do przypadku podstawowego

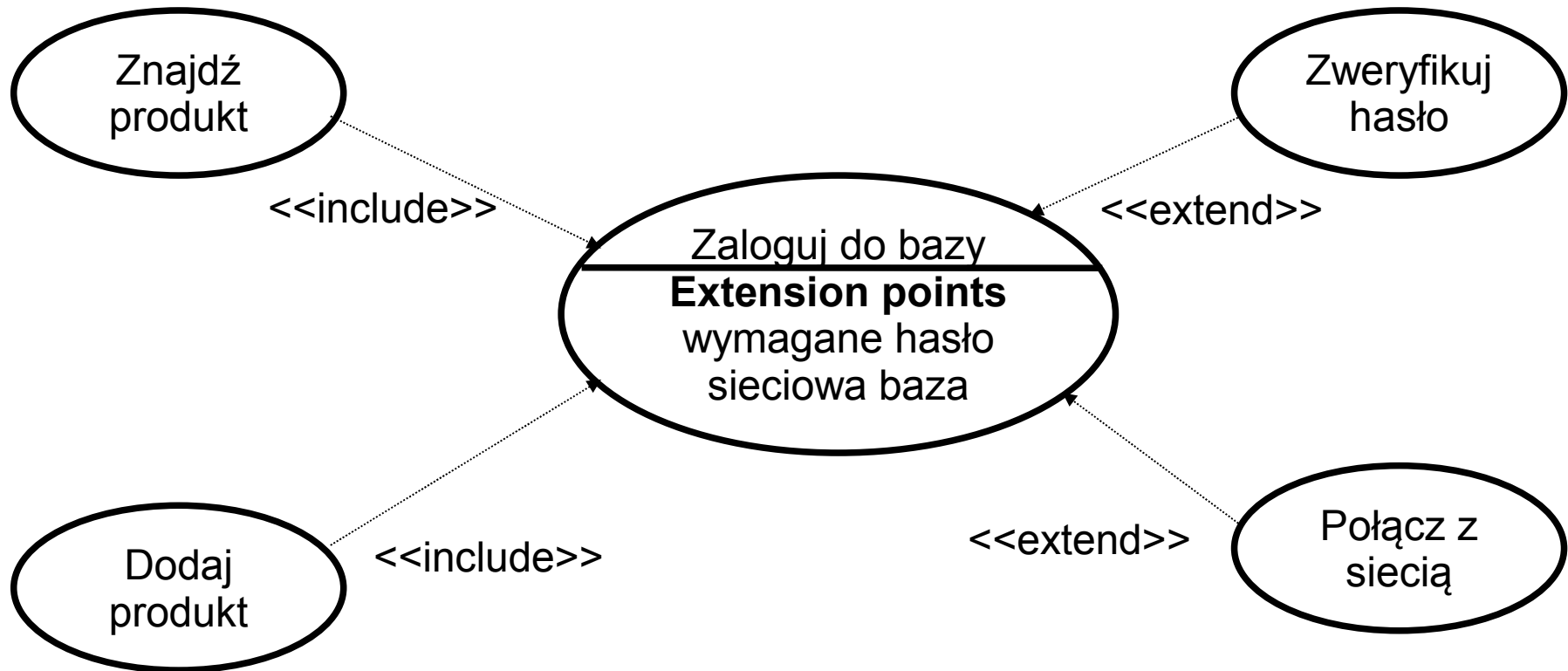
Zależność <<extend>>



Punkty rozszerzenia

- Jest możliwe określenie sytuacji (warunków) w których musi nastąpić dołączenie przypadków rozszerzających
- Są one wyszczególnione w rozszerzanym przypadku, pod poziomą linią

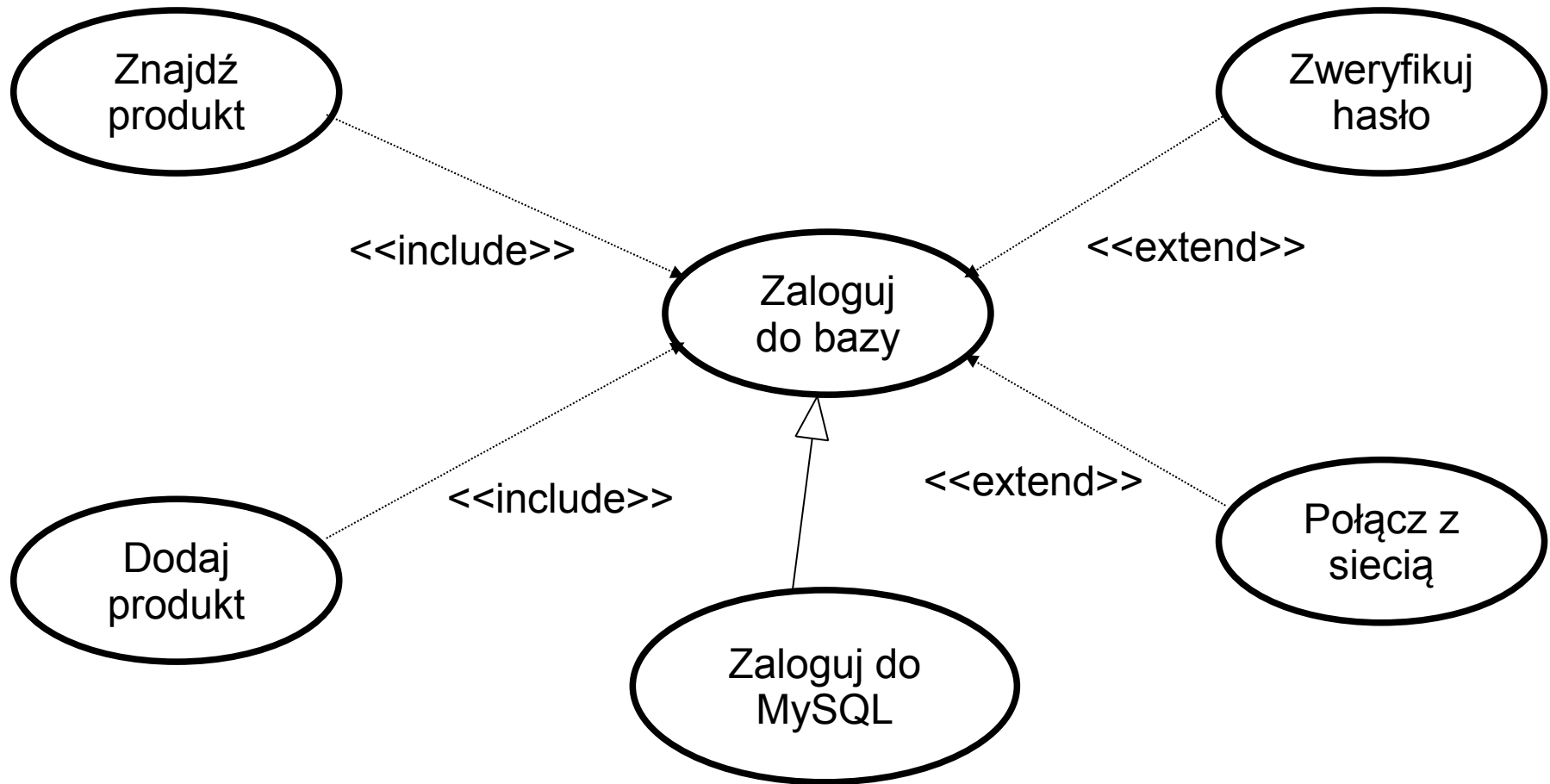
Punkty rozszerzenia



Uogólnienie

- Uogólnienie jest taksonomiczną relacją pomiędzy ogólnym i specjalizowanym klasyfikatorem
- Specjalizowany klasyfikator dziedziczy wszystkie cechy klasyfikatora ogólnego
- Jest obrazowana linią zakończoną zamkniętą strzałką, wskazującą w stronę klasyfikatora ogólnego

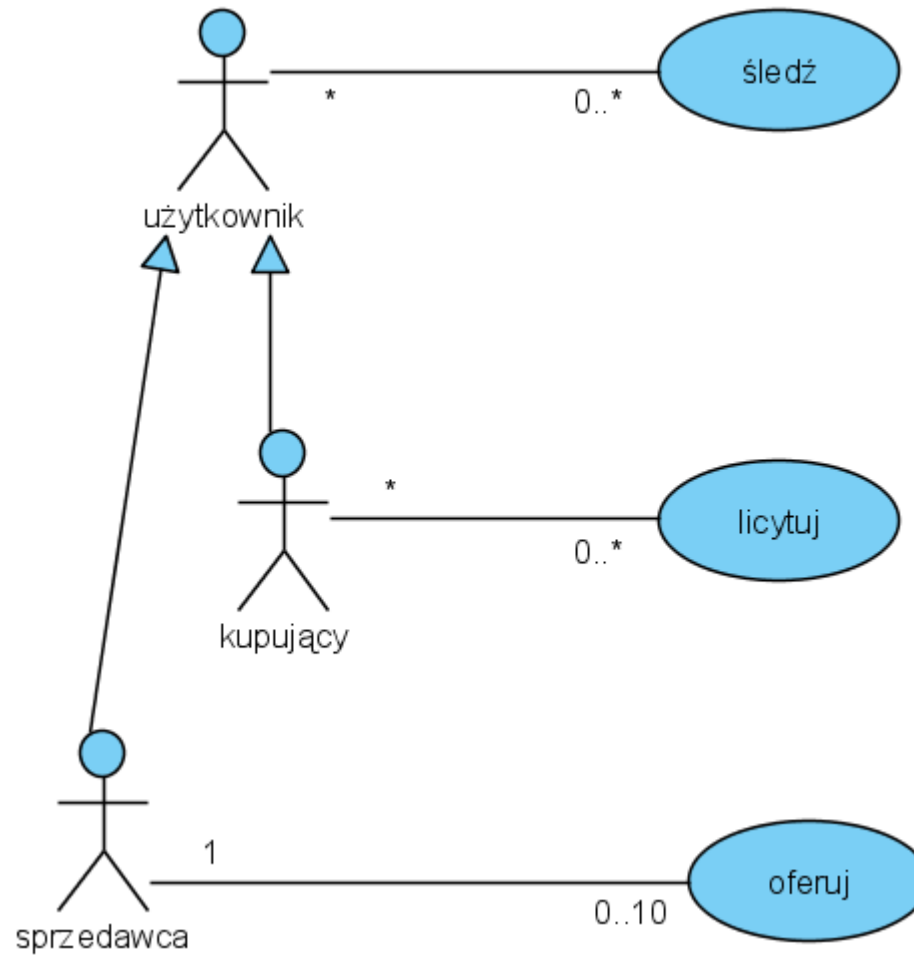
Uogólnienie



Liczebność

- Umożliwia określenie liczby elementów biorących udział w tej asocjacji, na każdym jej końcu
- Możliwe przypadki:
 - n ($n > 0$) dokładnie n
 - $n..*$ ($n \geq 0$) n lub więcej
 - $n..m$ ($m > n \geq 0$) od n do m
 - $*$ wiele
(nieznana liczba)
 - $n, m, o..p, q$ ($q > p...$) lista wartości

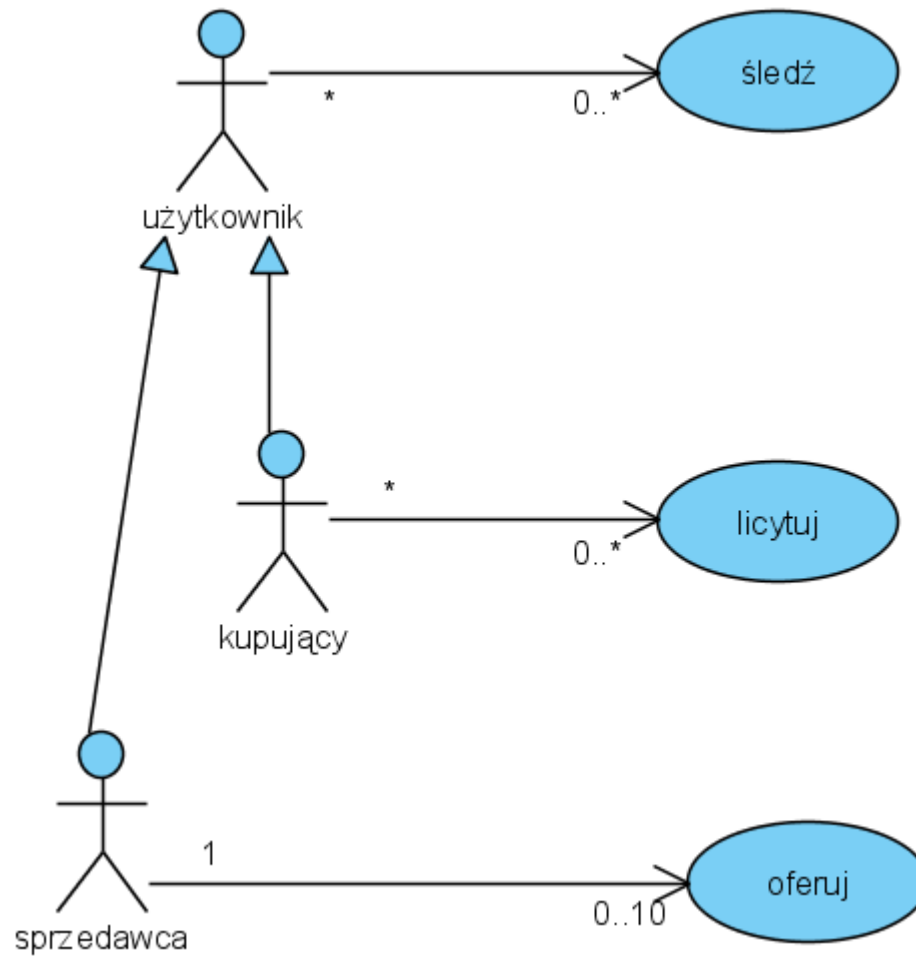
Liczebność



Skierowanie (navigability)

- Dwukierunkowa asocjacja może być wzbogacona o informację określającą stronę inicjującą komunikację
- Jest to zobrazowane przy pomocy strzałki
- Strzałka nie wskazuje kierunku przepływu danych

Skierowanie



Dokumentowanie przypadków użycia

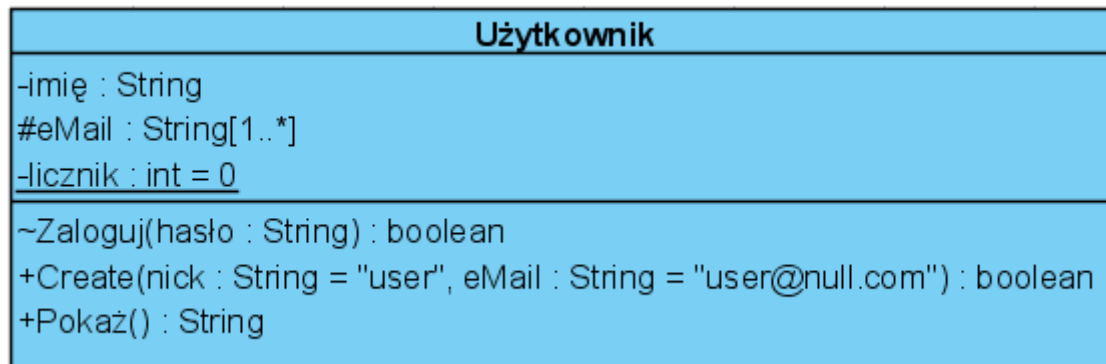
- Diagram przypadków użycia sam w sobie jest bardzo ogólnikowy
- Aby precyzyjnie określić pożądane zachowanie systemu, każdy przypadkiem użycia powinien posiadać dodatkową informację, tzw. scenariusz
- Scenariusz jest sekwencją akcji, określającą zachowanie
- Dla złożonych przypadków można zdefiniować główny oraz alternatywne scenariusze
- Scenariusz może zostać zapisany w języku naturalnym, pseudo-kodzie, tabeli itp.

Diagramy klas

- Zawiera informacje o statycznych związkach między elementami (klasami)
- Są ściśle powiązane z technikami programowania zorientowanego obiektowo
- Są jednymi z istotniejszych diagramów w UML

Symbol klasy

- Symbolem klasy jest prostokąt, zwykle podzielony poziomymi liniami na trzy sekcje:
 - nazwy
 - atrybutów
 - operacji
- W razie potrzeby może zostać uzupełniony dodatkowymi sekcjami (np. wyjątków)



Kontrola dostępu

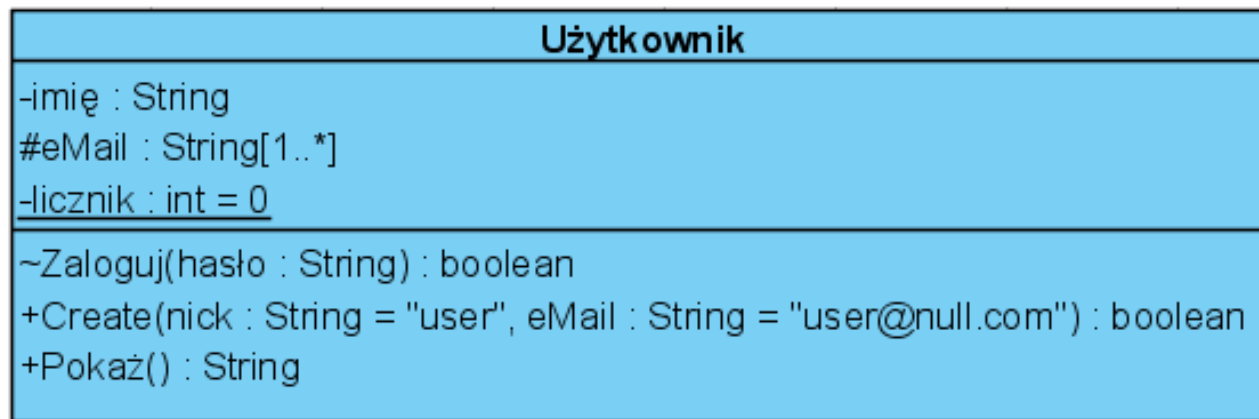
- Można określić modyfikatory dostępu dla składowych
- Są one ściśle powiązane z koncepcjami programowania zorientowanego obiektowo
- Możliwe rodzaje dostępu:

- + publiczny
- - prywatny
- # chroniony
- ~ pakietu

Użytkownik
-imię : String #eMail : String[1..*] <u>-licznik : int = 0</u>
~Zaloguj(hasło : String) : boolean +Create(nick : String = "user", eMail : String = "user@null.com") : boolean +Pokaż() : String

Składniki statyczne

- Składniki można zadeklarować jako statyczne – działające na rzecz klasy, nie obiektu
- Koncepcja identyczna jak w językach zorientowanych obiektowo
- Reprezentacją graficzną jest podkreślenie



Specyfikacja składników

- Atrybuty mogą mieć określone:
 - Typ. Typ jest umieszczany po nazwie, oddzielony dwukropkiem
 - Liczebność
 - Wartość początkową
- Operacje mogą mieć określone:
 - Typ zwracany. Typ jest umieszczany po nazwie, oddzielony dwukropkiem
 - Argumenty. Każdy argument może być określony tak jak atrybut, z dodatkowym oznaczeniem kierunku przekazywania wartości (domyślnie “in”)

Specyfikacja składników

Użytkownik
-imię : String #eMail : String[1..*] <u>-licznik : int = 0</u>
~Zaloguj(hasło : String) : boolean +Create(nick : String = "user", eMail : String = "user@null.com") : boolean +Pokaż() : String

Związki

- Wszystkie 4 typy związków są używane
- Głównym typem jest asocjacja
- Może mieć następujące cechy (pogrubiono nowe w stosunku do diagramów przypadków użycia):
 - **nazwa**
 - **role**
 - kierunek nawigacji
 - liczebność
 - **agregacja**

Nazwa

- Można nazwać asocjację aby doprecyzować jej znaczenie
- Nazwa może zawierać kierunek



Role

- Inny sposób doprecyzowania asocjacji
- Rola klasy jest określona przez tekst umieszczony w pobliżu tej klasy
- Można określić jednocześnie nazwę i rolę



Kierunek nawigacji

- Domyślnie asocjacja jest dwukierunkowa
- Aby była jednokierunkowa, dodaje się strzałkę
- Oznacza to że **komunikacja jest jednokierunkowa**



Liczebność

- Znaczenie identyczne jak w diagramach przypadków użycia



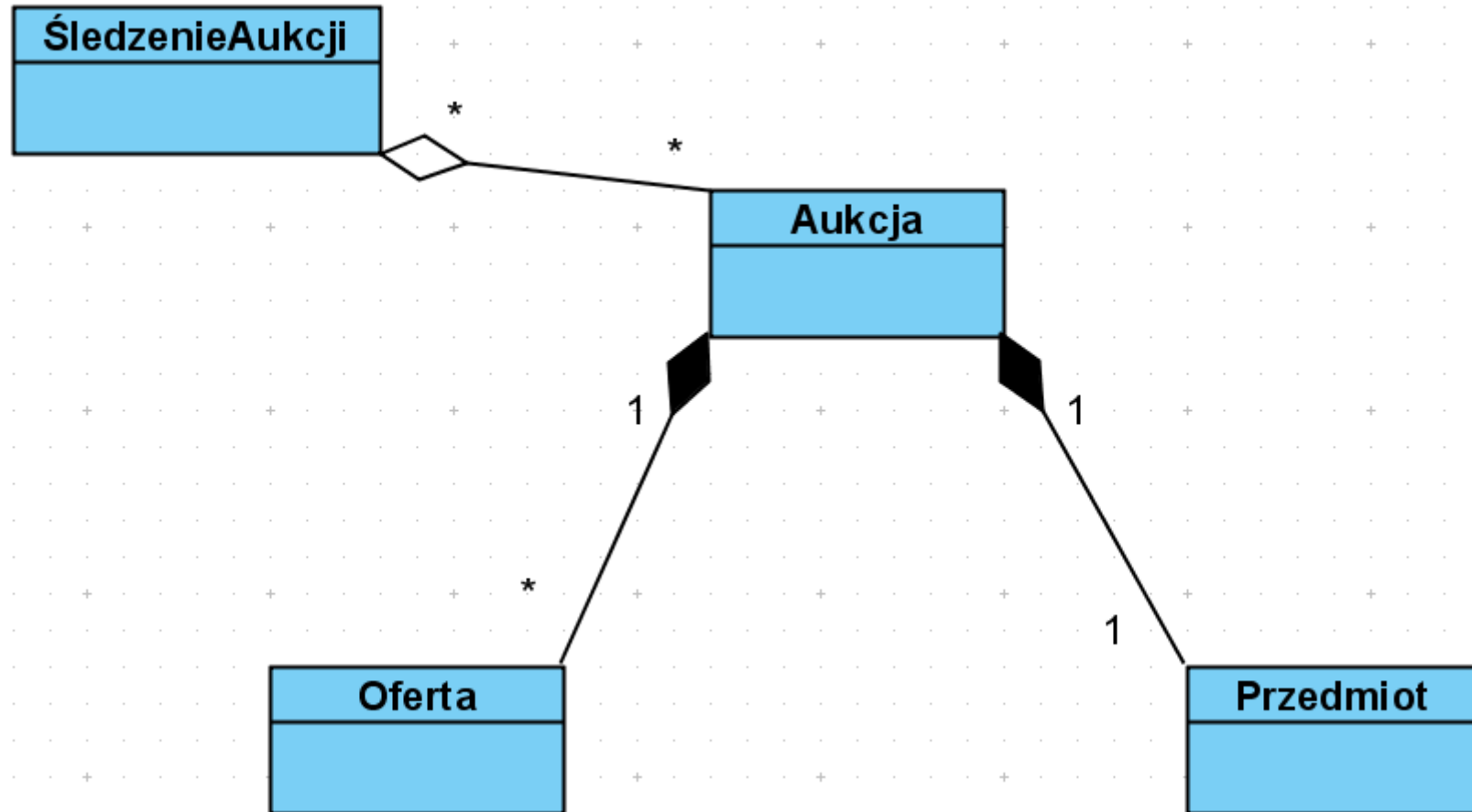
Agregacja

- Określa związek między całością i częścią
- Są dwa typy:
 - całkowita (kompozycja, silna agregacja)
 - częściowa (agregacja, słaba agregacja)
- Jest obrazowana przez romb umieszczony przy symbolu określającym całość
- Silna agregacja jest zobrazowana przez pełen romb, słaba – przez pusty

Silna i słaba agregacja

- W przypadku silnej agregacji części składowe nie mogą istnieć jeśli symbol określający całość jest usunięty
- W przypadku słabej agregacji jest to możliwe. Jeden obiekt może być też zawierany przez wiele innych.

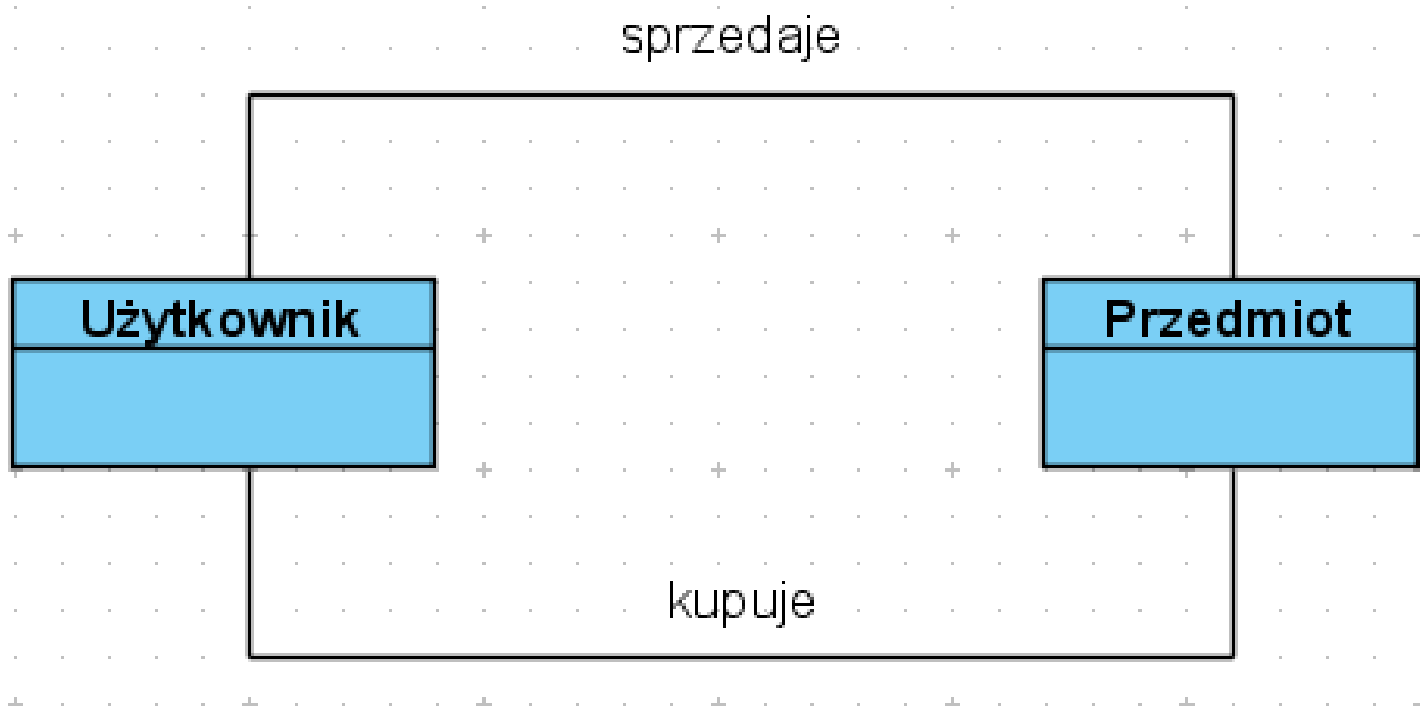
Silna i słaba agregacja



Asocjacja wielokrotna

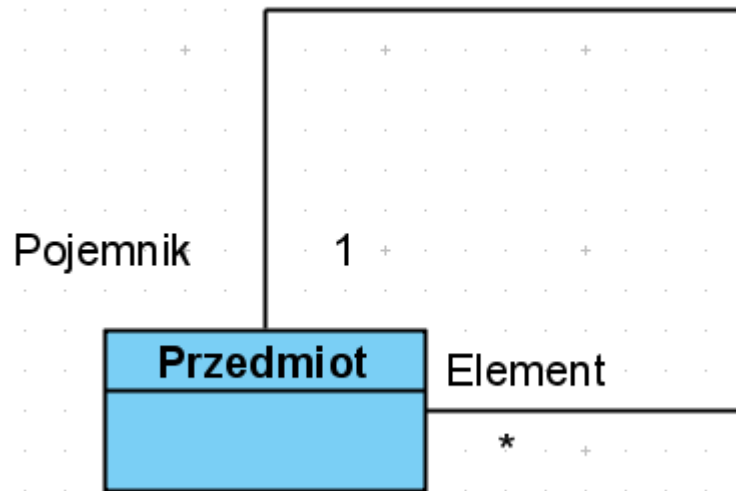
- Dwie klasy mogą być w odmienny sposób związane ze sobą w różnych kontekstach
- W efekcie może być więcej niż jedna asocjacja między klasami
- W takim wypadku każda powinna być nazwana

Asocjacja wielokrotna



Asocjacja zwrotna

- Asocjacja może być zwrotna

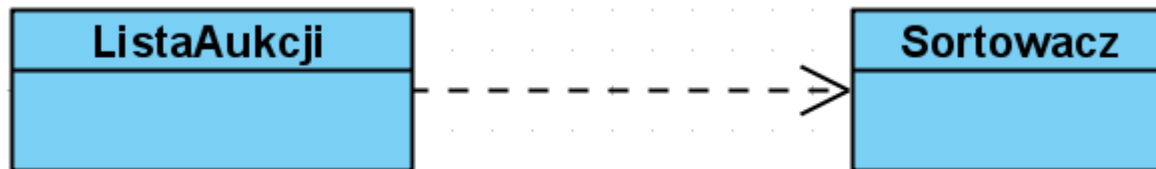


Podsumowanie asocjacji

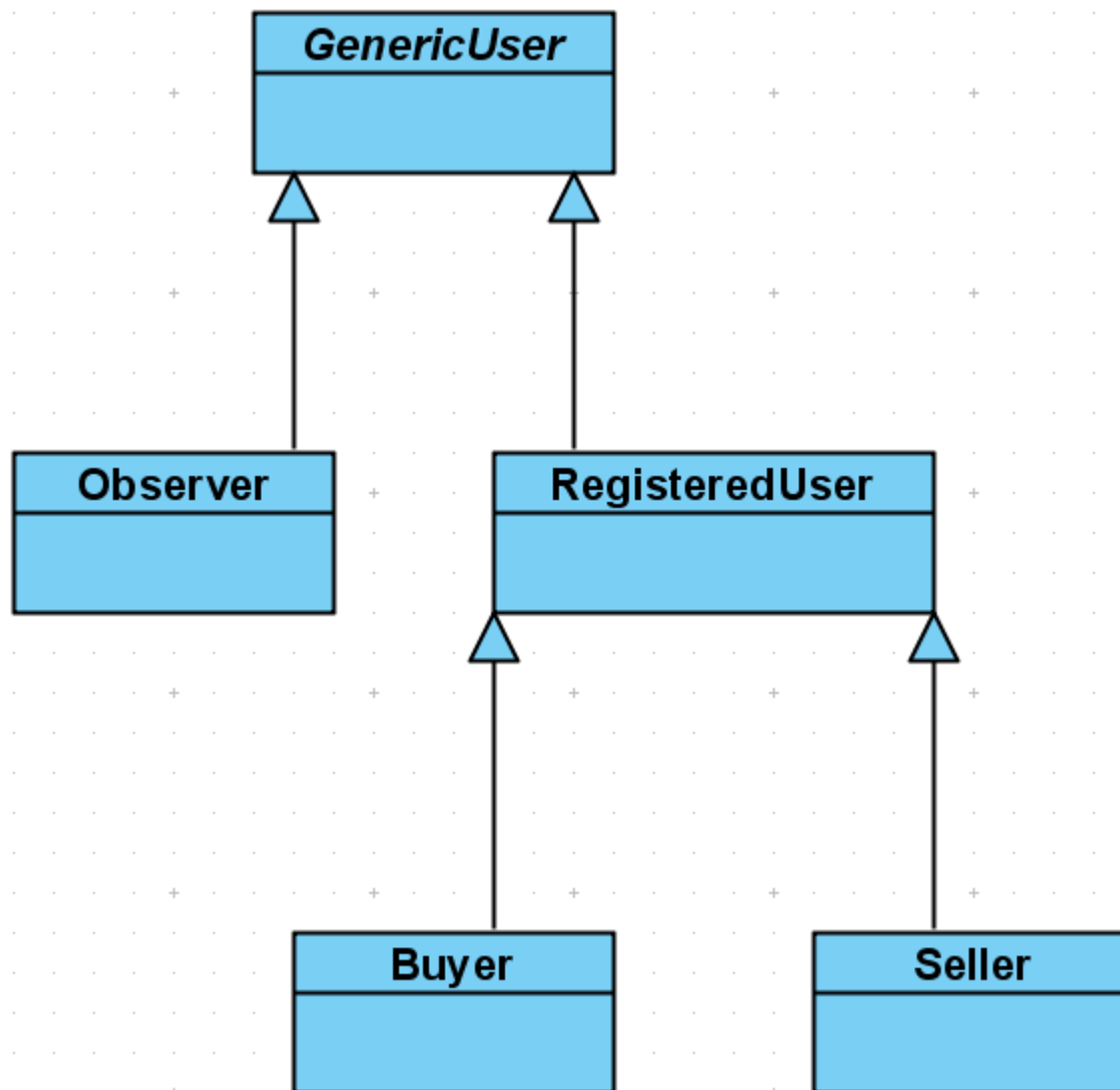
- Asocjacja – związek między dwiema klasami (student – nauczyciel, sprzedający – kupujący)
- Słaba agregacja – obiekty jednej klasy należą do obiektów drugiej, ale fragmenty mogą istnieć bez całości (zamówienie – produkty, biblioteka – książki)
- Silna agregacja (kompozycja) – j.w., ale część nie może istnieć bez całości (wielokąt – jego wierzchołki, zamówienie – adres dostarczenia)

Zależność

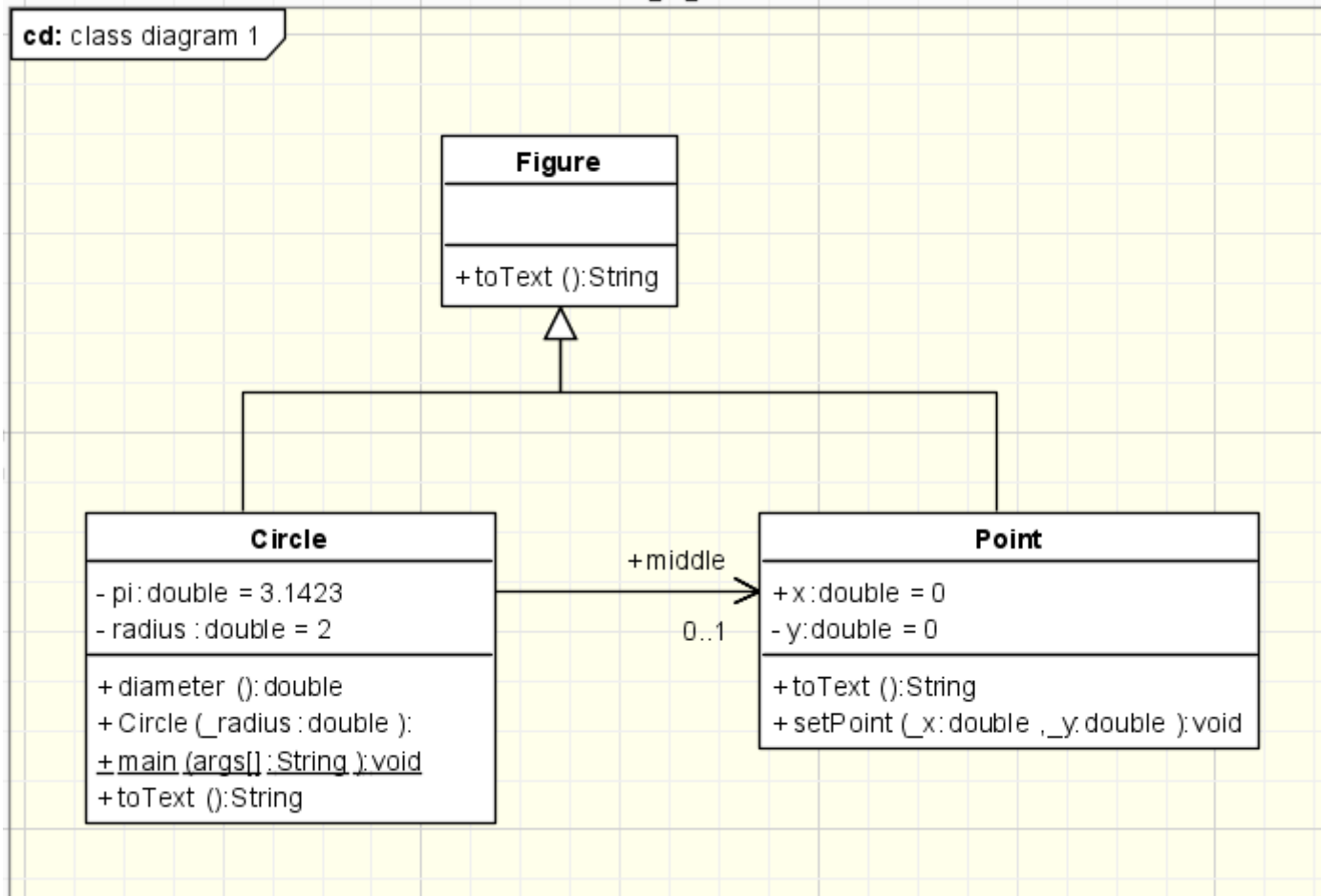
- Oznacza że jedna klasa (klient) w jakiś sposób używa innej klasy (dostawca)
- Jest obrazowana linią przerywaną zakończoną strzałką wskazującą na dostawcę



Uogólnienie

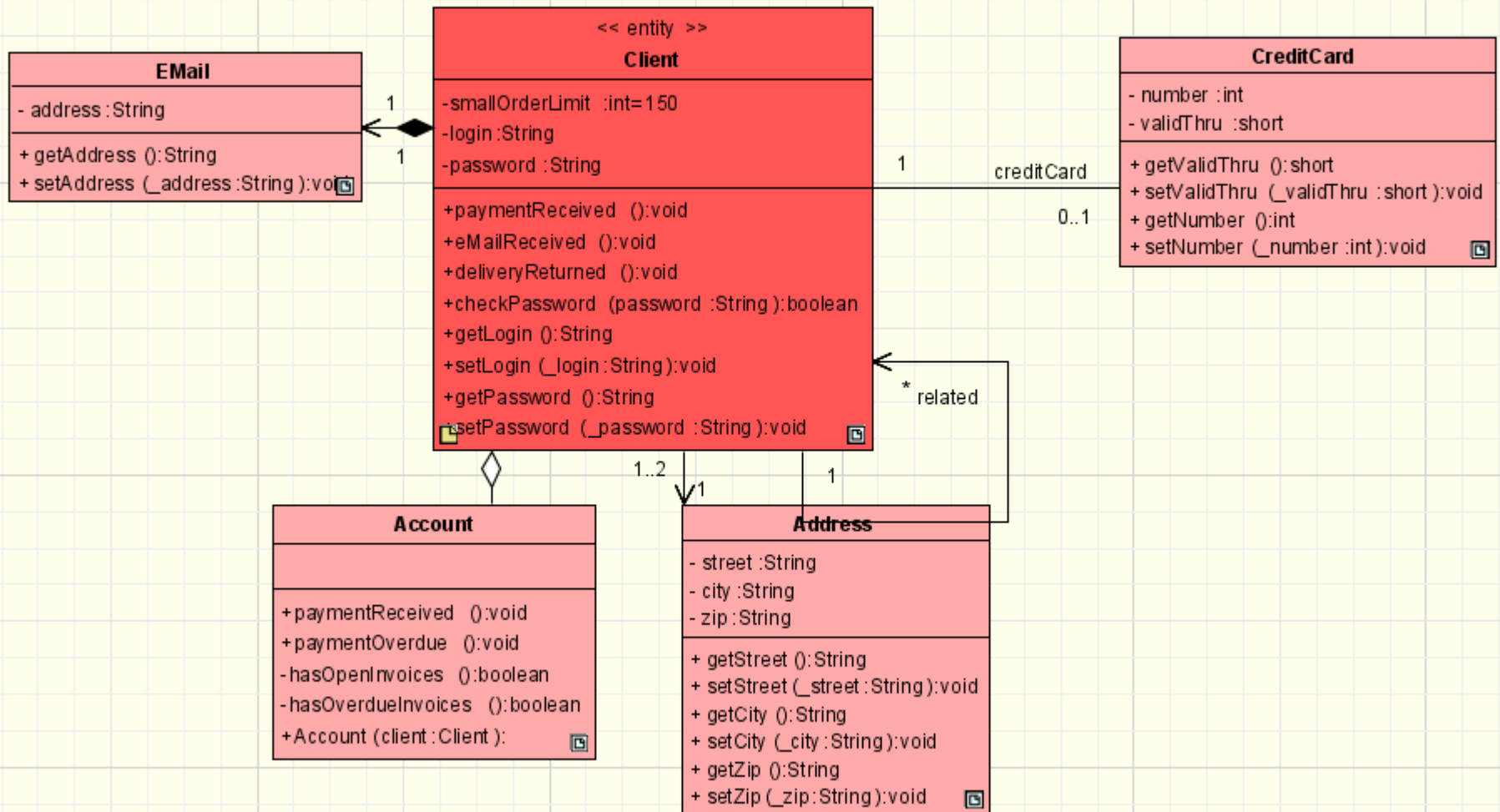


Przykładowe diagramy

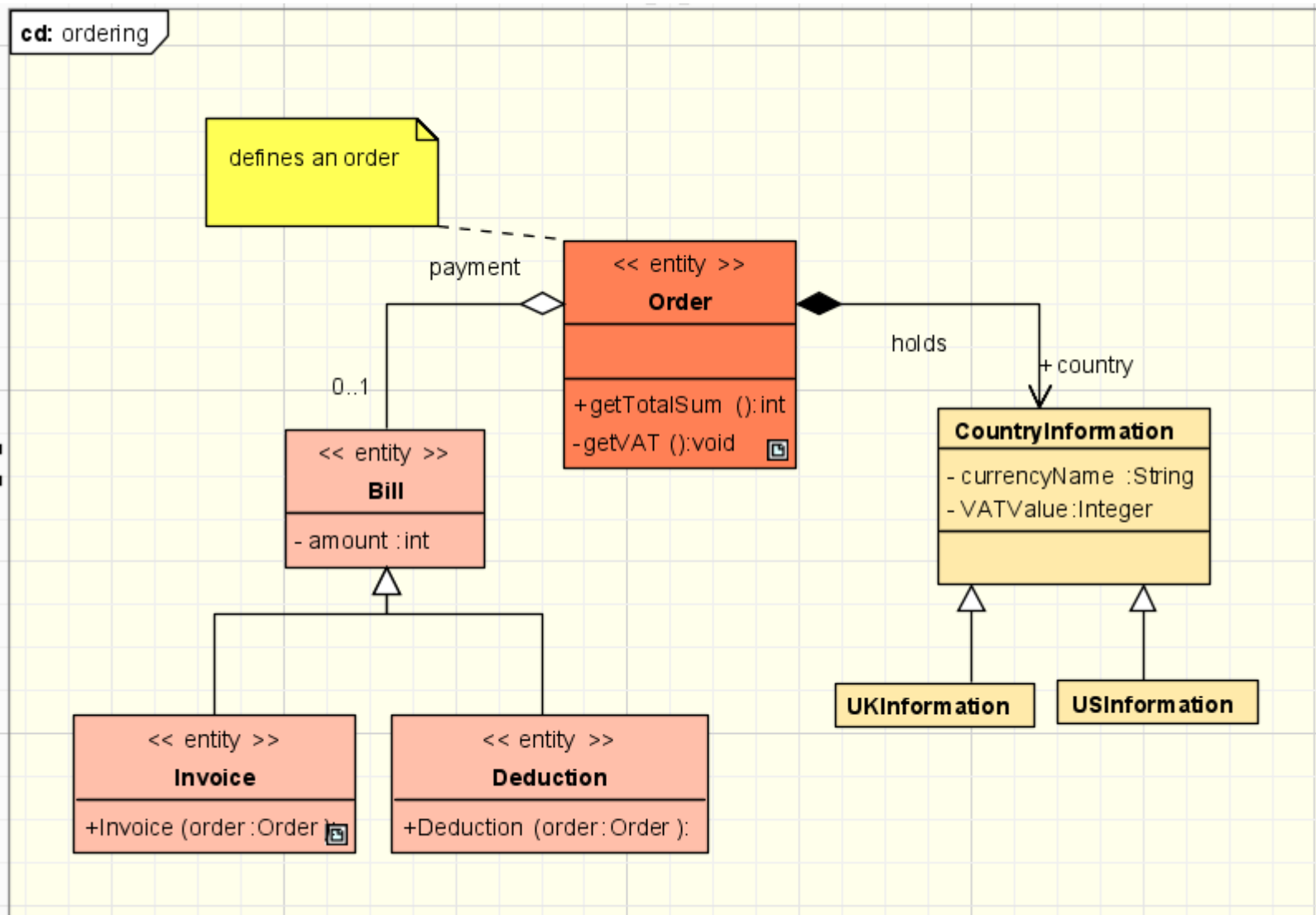


Przykładowe diagramy

cd: clients

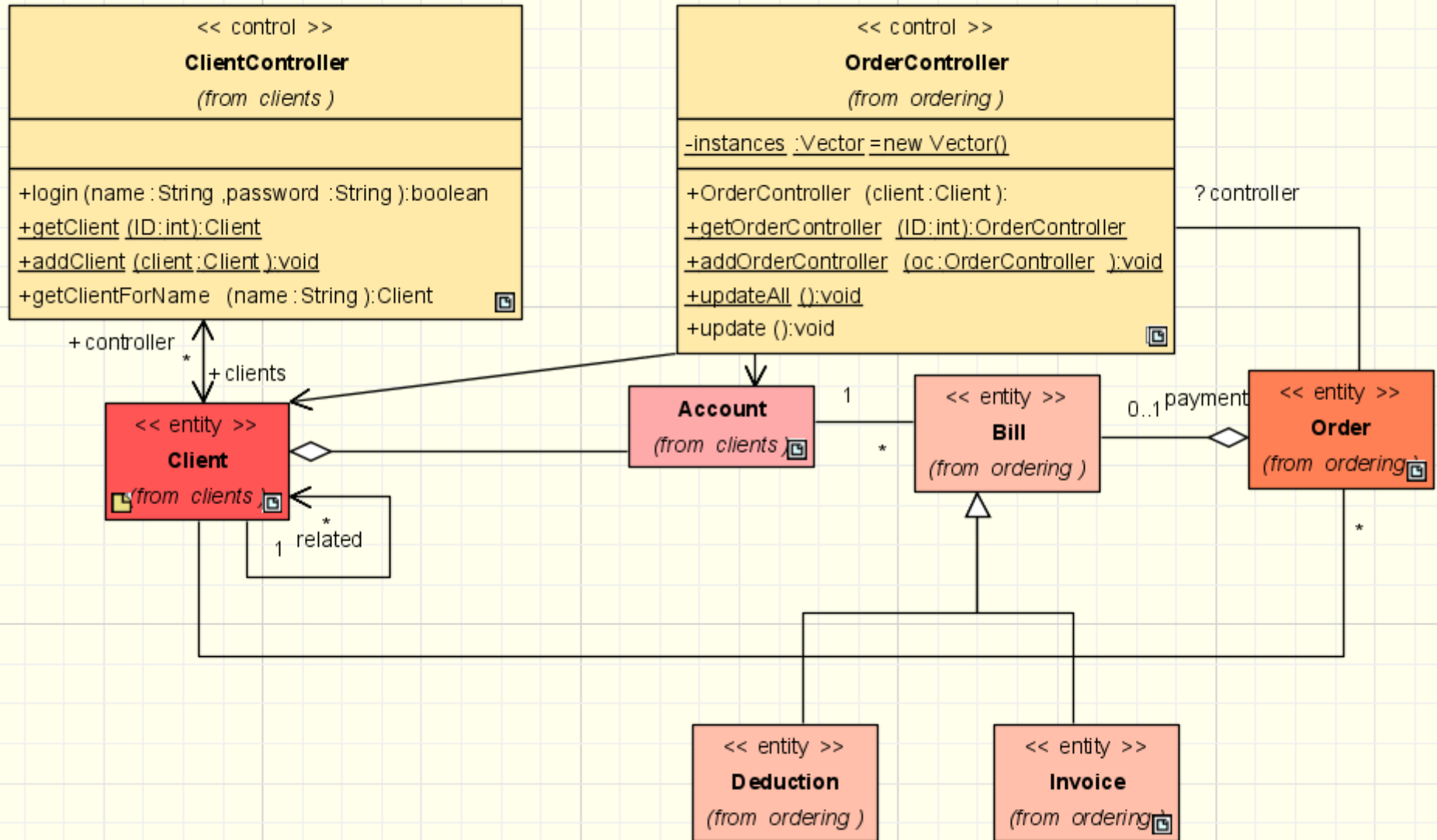


Przykładowe diagramy

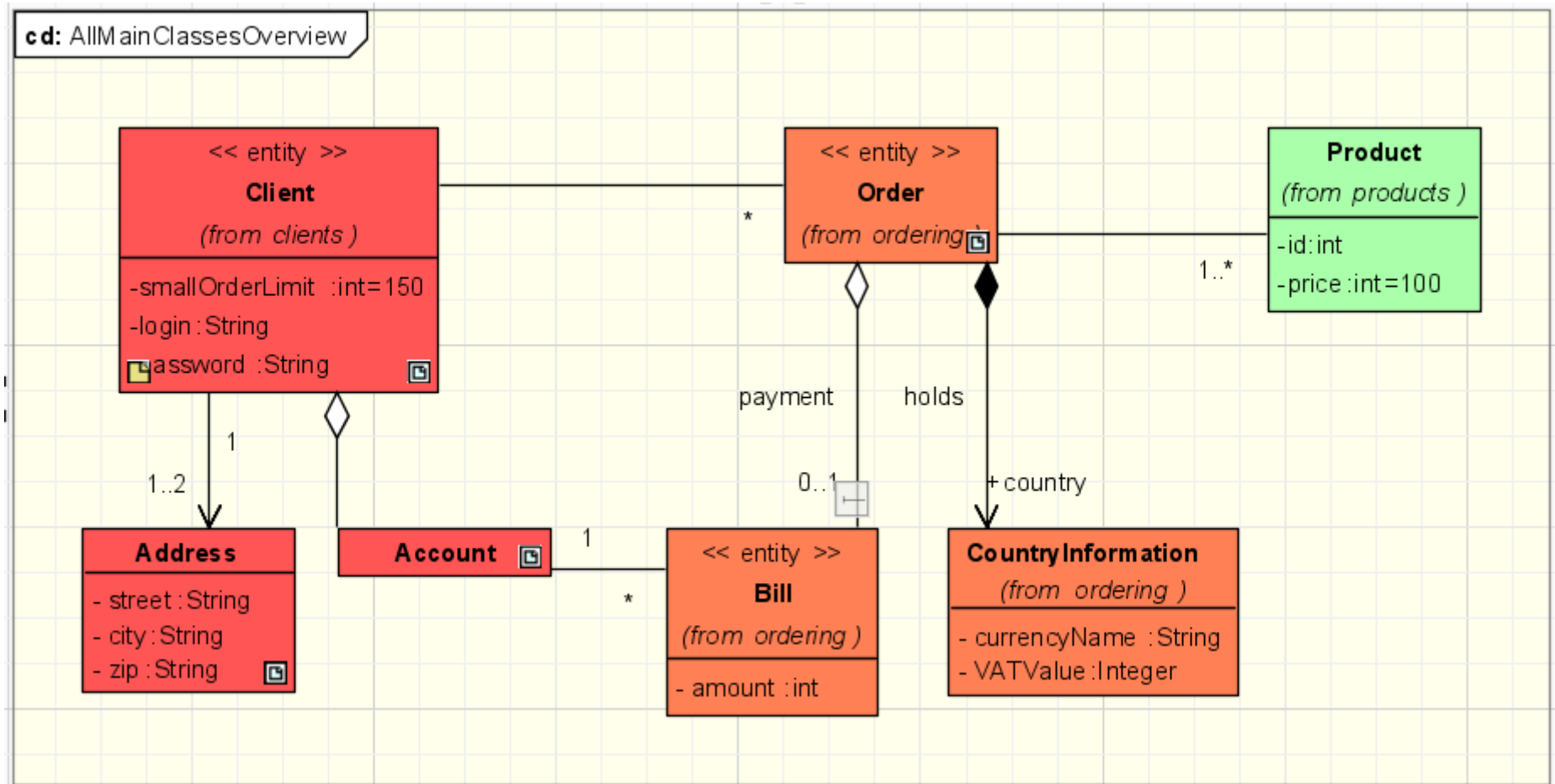


Przykładowe diagramy

cd: BCE Schema



Przykładowe diagramy



Przykładowe diagramy

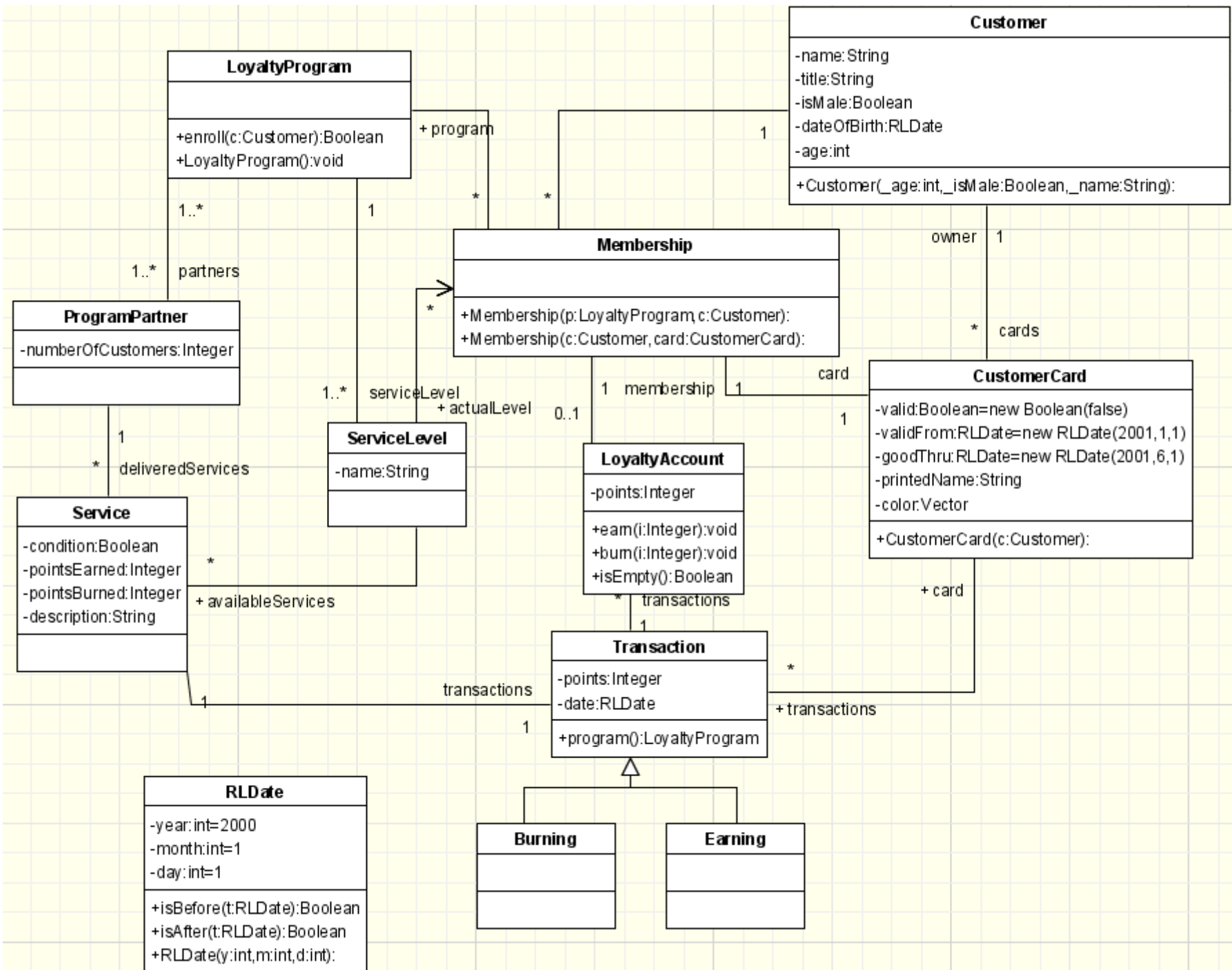


Diagram sekwencji (sequence)

- Opisuje interakcje między obiektami przy pomocy sekwencji wiadomości
- Dobrze nadaje się do dokumentowania przypadków użycia
- Diagram jest zorientowany w dwu wymiarach:
 - Oś pozioma związana jest z kolejnymi obiektami biorącymi udział w wymianie wiadomości
 - Oś pionowa związana jest z upływem czasu

Główne elementy

- Obiekt
- Linia życia
- Wiadomość

Obiekt

- Obiekty klas są podstawowymi elementami w diagramie sekwencji
- Diagram może również zawierać instancję innych klasyfikatorów: przypadków użycia, aktorów, sygnałów itp.
- Obiekt jest przedstawiany jako prostokąt z nazwą (niekiedy podkreśloną)
- W prostych diagramach wszystkie obiekty umieszczone są przy górnej krawędzi diagramu

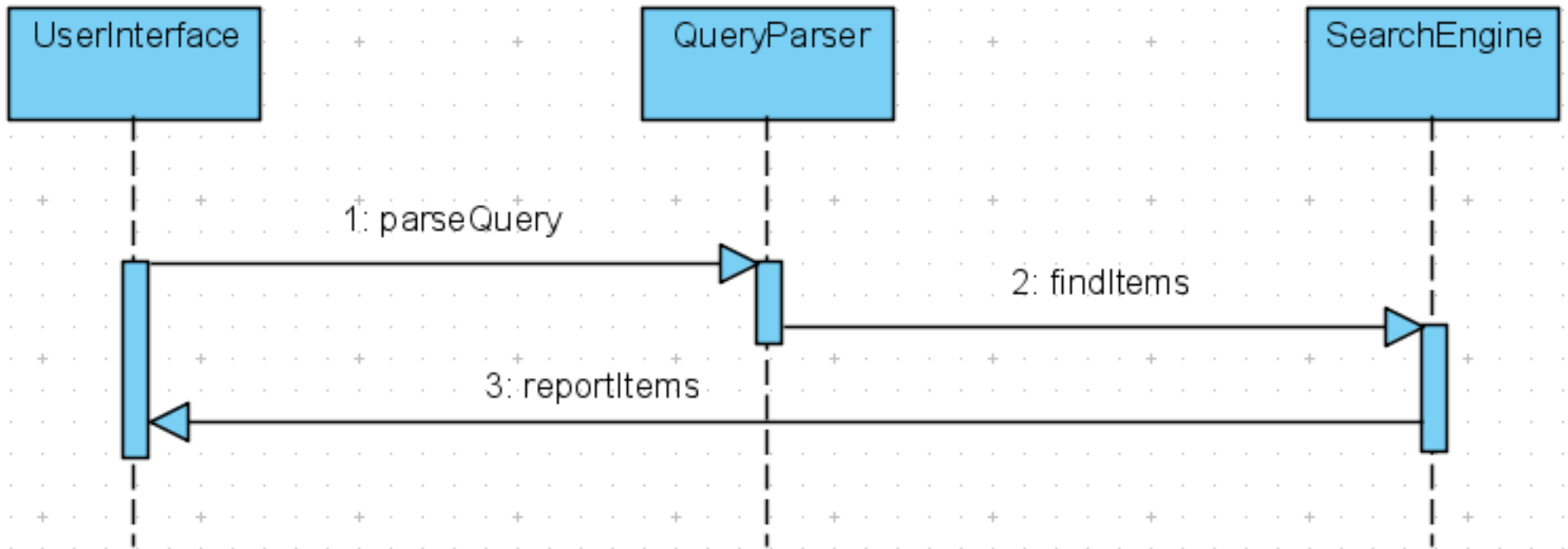
Linia życia

- Reprezentuje przedział czasu w którym obiekt istnieje
- Jest zobrazowana przez przerywana pionową linię, zaczynającą się na obiekcie i idącą w dół

Wiadomości

- Wiadomości reprezentują przepływ informacji między obiektami. Wiadomość jest poleceniem dla obiektu aby wykonać pewne operacje
- Najważniejszym (i koniecznym) składnikiem opisu wiadomości jest sygnatura, składająca się co najmniej z nazwy

Przykładowy diagram



Rodzaje wiadomości

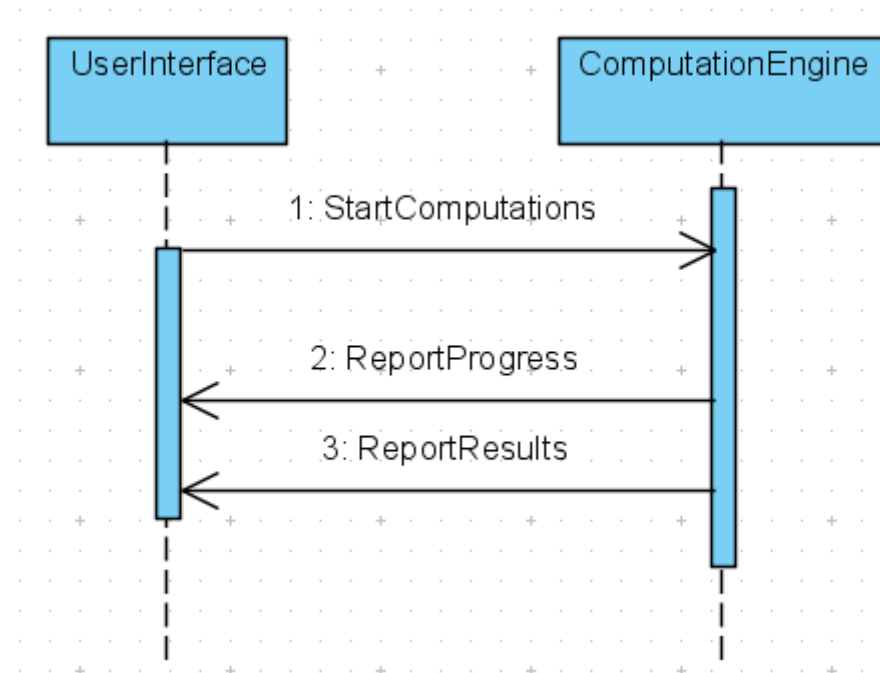
- Synchroniczna
- Asynchroniczna
- Zwrotna

Wiadomość synchroniczna

- Sterowanie jest przekazane do obiektu wywoływanego
- Przetwarzanie w obiekcie wywołującym jest wstrzymywane do momentu zakończenia wywołanej czynności
- Jest obrazowane pełną strzałką
- Jest odpowiednikiem wywołania funkcji

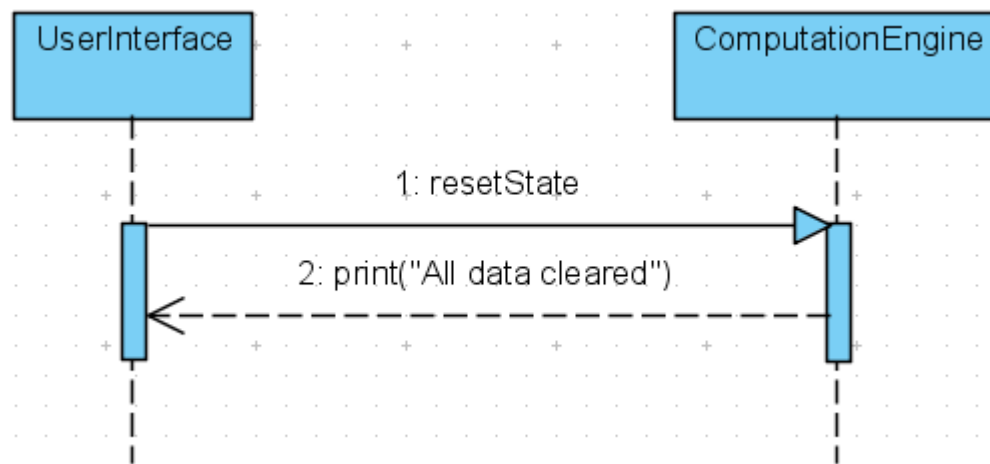
Wiadomość asynchroniczna

- Przetwarzanie w obiekcie wywołującym nie jest przerwane
- Jest obrazowane otwartą strzałką



Wiadomość zwrotna

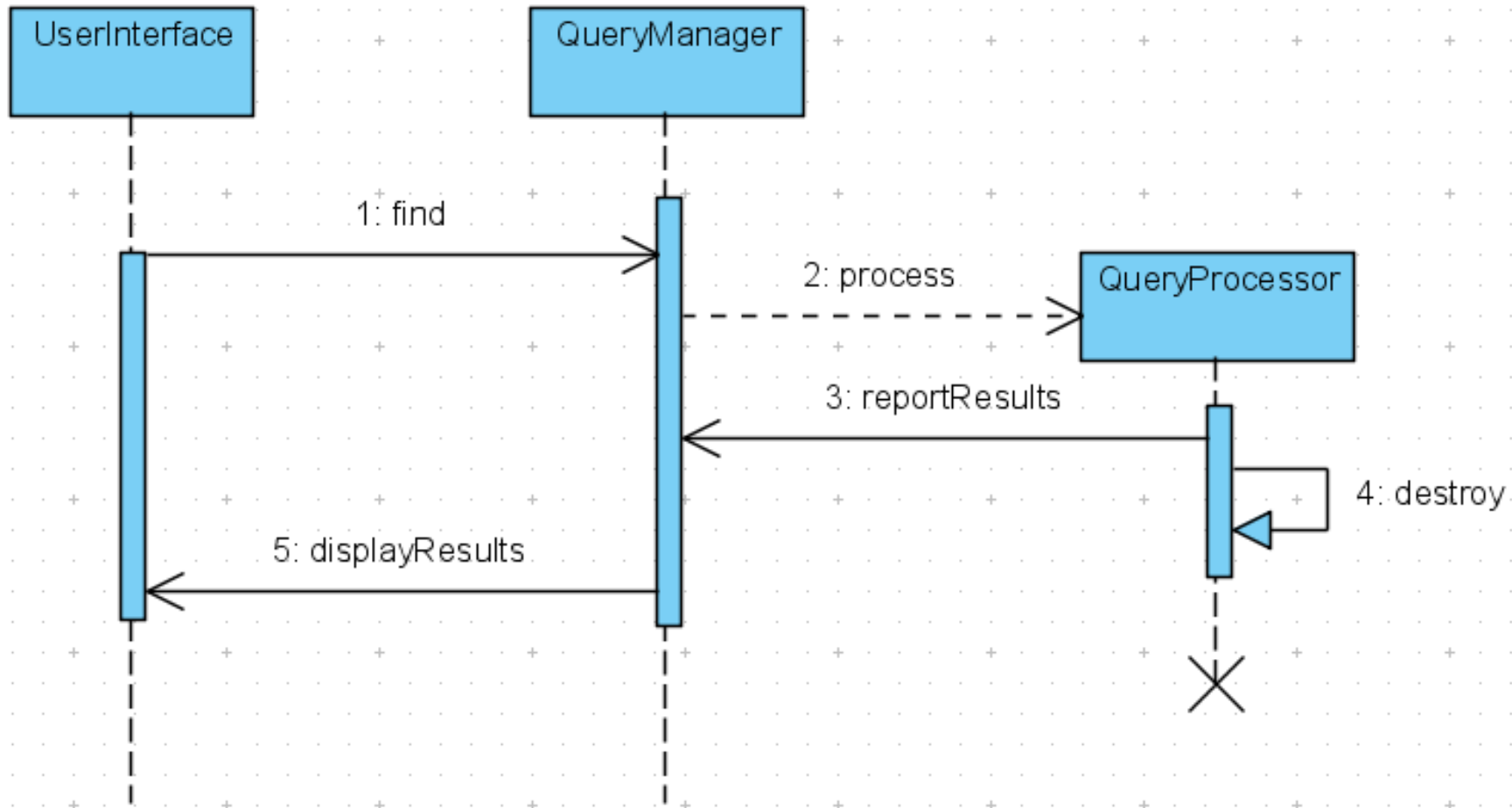
- Obrazuje oddanie sterowania do obiektu wywołującego
- Jest opcjonalna
- Jest przedstawiona jako strzałka z linią przerywaną



Tworzenie i niszczenie obiektów

- Tworzenie i niszczenie są powodowane przez odpowiednie wiadomości
- Wiadomości te mają stereotypy, odpowiedni “create” i “destroy”
- Na końcu wiadomości “create” umieszcza się tworzony obiekt (co powoduje że znajduje się on poniżej innych obiektów)
- Po otrzymaniu wiadomości “destroy” linia życia obiektu kończy się. Jest to dodatkowo wyróżnione umieszczeniem znaku X na końcu linii.

Przykładowe tworzenie i niszczenie



Fragmenty złożone

- Są to wybrane fragmenty diagramu sekwencji, do których odnosi się odpowiedni operator interakcji
- Są zobrazowane ramą otaczającą wybrany region. Rama ma nagłówek w lewym górnym rogu zawierający operator interakcji
- Niektóre operatory wymagają wyodrębnienia podfragmentów regionu. Są one wyodrębniane linią kropkowo-kreskową

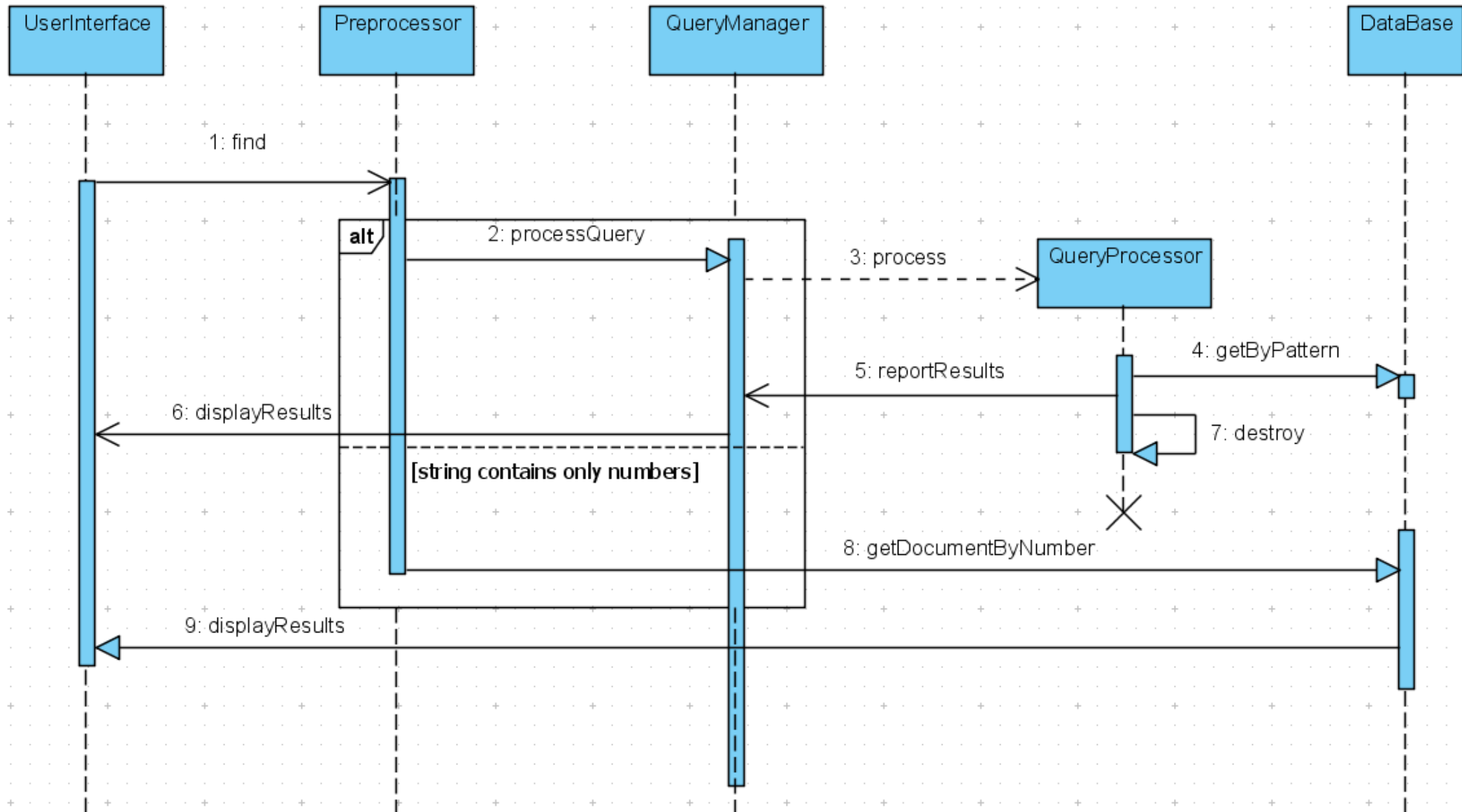
Wybrane operatory

- Alt
- Opt
- Loop
- Par

Operator alt - alternatywa

- Oznacza, że tylko jeden z podobszarów (operandów) obszaru objętego ramą może być wybrany
- Wybór ten zależy od warunków umieszczonych w podfragmentach
- Podobszar bez warunku jest wyborem domyślnym

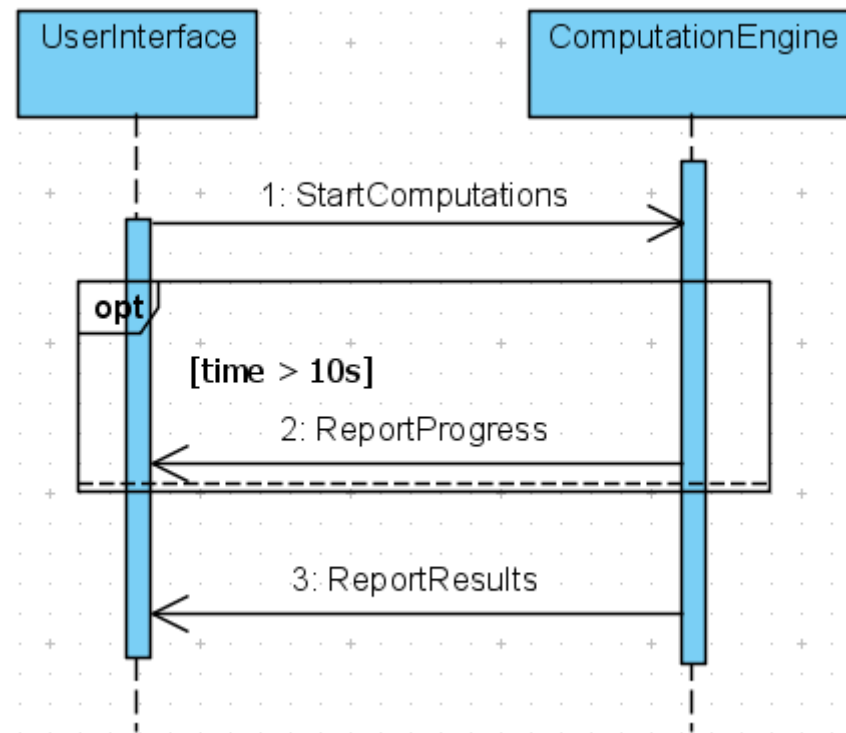
Przykład operatora alt



Operator opt – fragment opcjonalny

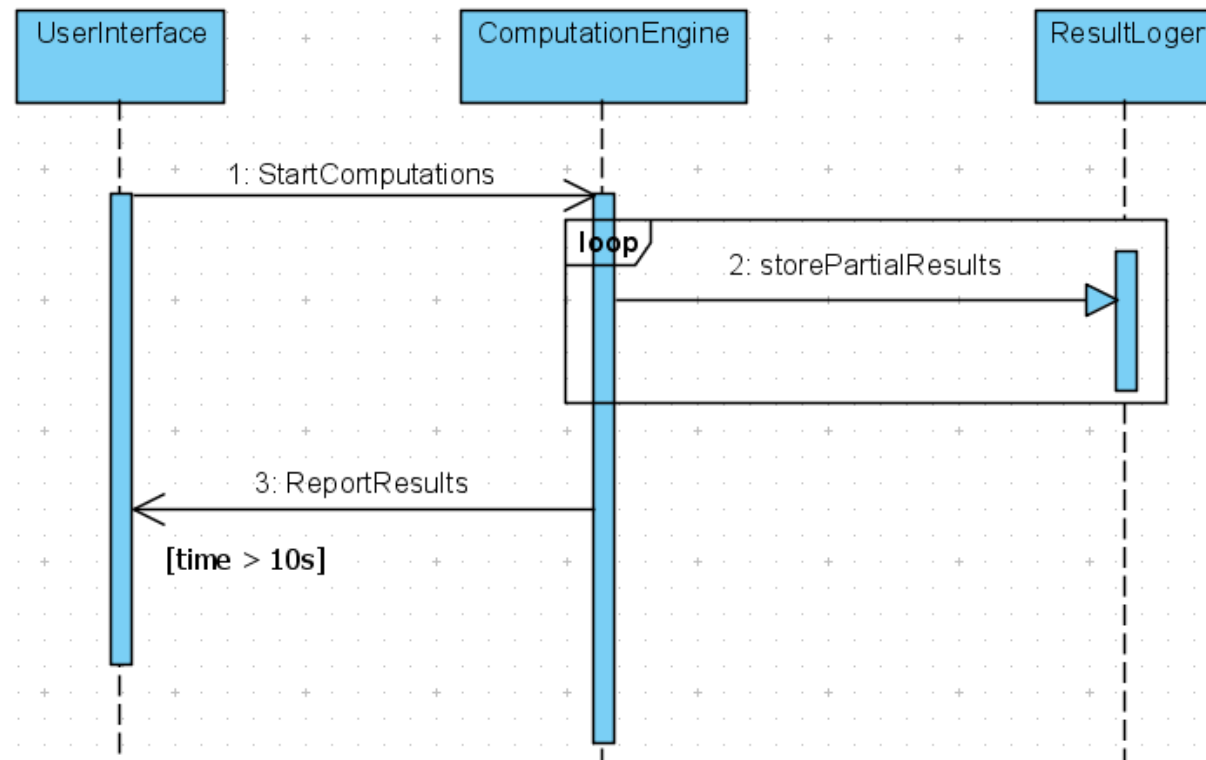
- Część diagramu zostanie wykonana tylko jeśli spełniony będzie warunek
- Odpowiada operatorowi alt z pustym domyslnym operandem

Przykład operatora opt



Operator loop - iteracja

- Umożliwia wielokrotne powtórzenie wybranego fragmentu
- Liczba interakcji może zostać określona



Operator par – wykonanie równoległe

- Oznacza, że podfragmenty fragmentu objętego ramą są wykonywane równoległe

