

Diagramy klas

Dr hab. inż. Ilona Bluemke

Plan wykładu

- Klasa
 - Relacje między klasami (asocjacje, generalizacje, agregacje)
 - Ograniczenia relacji
 - Klasy interfejsowe
-

Diagramy klas

- Przedstawiają typy obiektów występujących w aplikacji i powiązania między nimi.
- Jest to model statyczny dziedziny problemu.
- Określają **CO JEST** w systemie potrzebne do realizacji wymaganych funkcji (model use case).

Klasa

Osoba
Wiek Adres Data_urodzenia
Podaj_wiek Daj_adres Ustaw_adres

nazwa

atrybuty

operacje

Graficzna reprezentacja klasy

Osoba

Osoba
Podaj_wiek
Podaj_adres
Ustaw_adres

Osoba
wiek
adres
data_urodzenia

Klasa

- Opisuje grupę obiektów o podobnych własnościach (atrybutach), zachowaniach (operacjach), wspólnych relacjach z innymi obiektami, identycznym znaczeniu.
- Jest uogólnieniem zbioru obiektów. Obiekt jest instancją klasy.
- Klasy pomagają w abstrakcji, generalizacji problemu.

Atrybuty

- wartości danych przechowywane w obiekcie określonej klasy, są unikalne w klasie, różne klasy mogą mieć takie same nazwy atrybutów.
- są określonego typu
- mogą mieć wartość domniemaną.
- może mieć określony **zakres dostępności** :
 - prywatny (private) -name : tylko klasa
 - publiczny (public) +name : wszystkie klasy
 - chroniony (protected) #name : klasa, podklasy

Atrybut -2

może być:

- implementacyjny (implementation) ?name
- wyprowadzalny (derived) /name
daje się wyprowadzić z innych atrybutów
- kluczowy (key) *name
jednoznacznie identyfikuje instancję klasy lub połączenie

Właściwości atrybutu definiują dodatkowe cechy

- *changeable* modyfikowalny bez ograniczeń
- *addOnly* brak możliwości usuwania
- *frozen* stały

Operacja

Operacja jest funkcją lub transformacją, którą można zastosować **do** lub może być stosowana **przez** obiekty tej klasy.

Wszystkie obiekty danej klasy posiadają te same operacje.

Metoda - implementacja operacji dla klasy.

Operacja -2

może mieć **argumenty parametryzujące** (ale nie wpływają one na wybór metody).

- może **zwracać wynik** (określonego typu).
- może mieć określony **zakres widoczności**, określający dostępność operacji, jakie inne klasy mogą z niej korzystać.
 - prywatna (private) **-name** : tylko klasa
 - publiczna (public) **+name** : wszystkie klasy
 - chroniony (protected) **#name** : klasa, podklasy
 - implementacyjny (implementation) **?name**
zasięg będzie określony w implementacji

Specyfikacja operacji klasy w języku UML:

widoczność nazwa (lista-argumentów) : typ { właściwości }

lista-argumentów

określa argumenty operacji:

sposób-przekazywania nazwa : typ = wartość-domyślna

gdzie:

sposób-przekazywania

- *in* przekazanie przez wartość
- *out* przekazanie przez referencję
- *inout* przekazanie przez referencję

właściwości definiują dodatkowe cechy operacji:

- *leaf* operacja nie jest polimorficzna
- *isQuery* operacja nie zmienia atrybutów
- *sequential* wymaga sekwencyjnego działania obiektu
- *guarded* wykonywana rozłącznie z innymi
- *concurrent* wykonywana współbieżnie z innymi

Relacje

- Połączenie - wiąże ze sobą obiekty.
- Fizyczne lub conceptualne połączenie między obiektami.
- Obiekt współpracuje z innymi obiektami z którymi jest połączony.
- Poprzez link obiekt klient prosi o usługę inny obiekt lub poprzez link może sterować innym obiektem, wywoływać operacje, otrzymywać rezultaty operacji.

Powiązanie - (association)

- Modeluje relacje takie jak: dotyczy, komunikuje się z, obsługuje
- Asocjacja dwukierunkowa:

nazwa



- Asocjacja jednokierunkowa

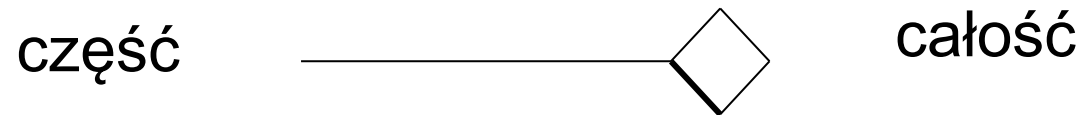
nazwa



Agregacja

modeluje relacje takie jak:

- składa się z,
- jest zbudowany z,
- zawiera



Generalizacja -dziedziczenie

modeluje dziedziczenie klas

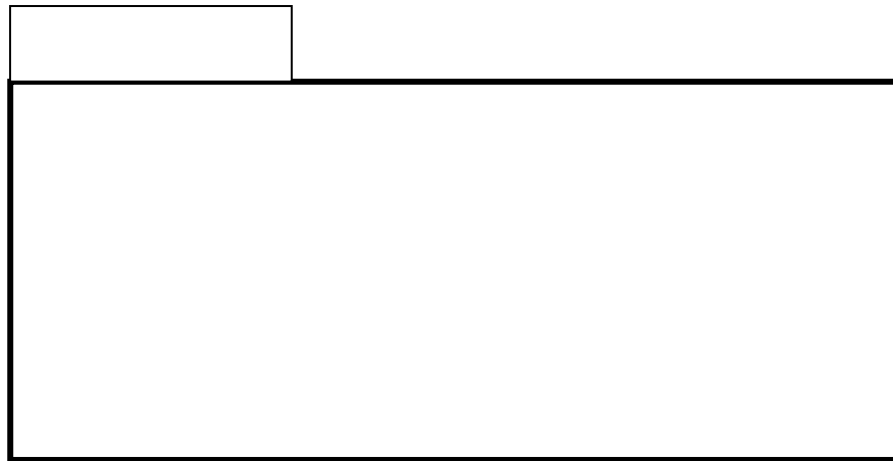
Klasa ogólna

Klasa specjalizowana



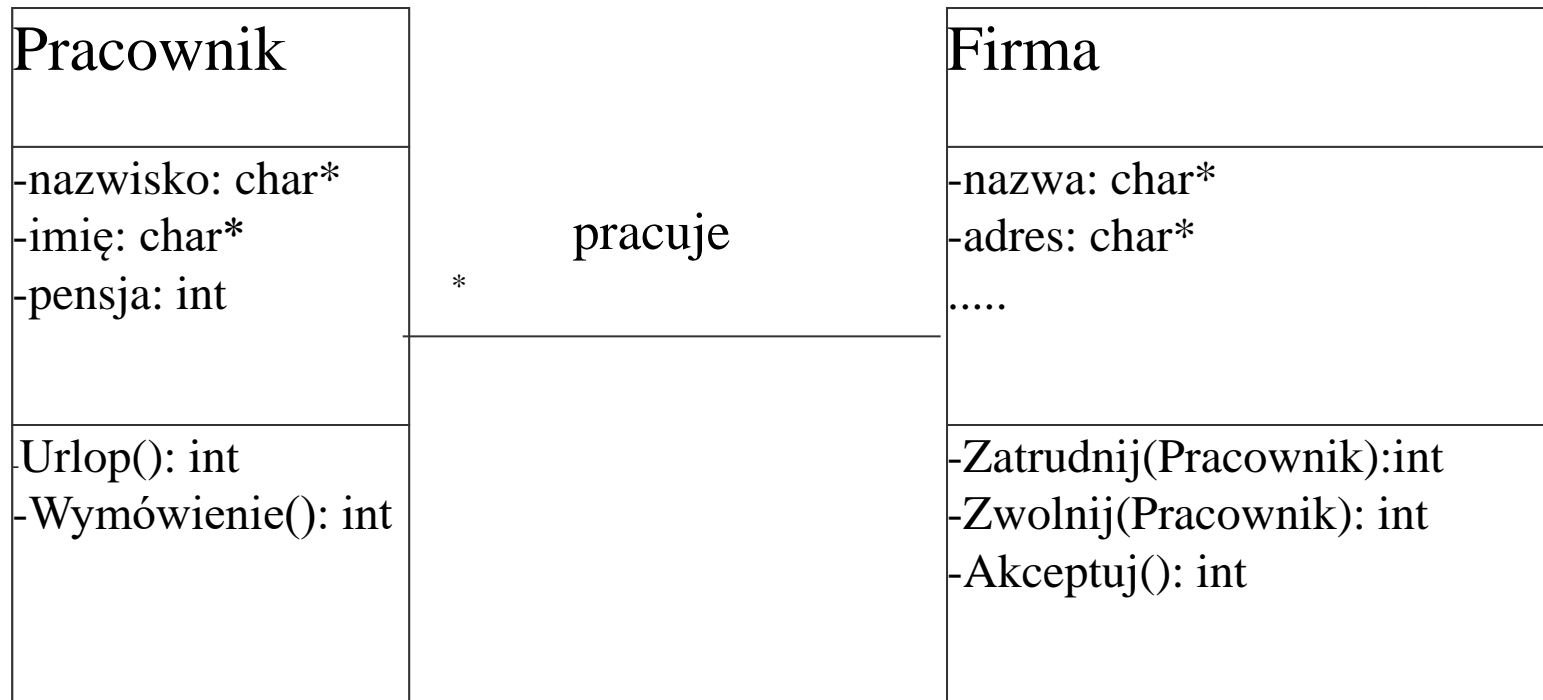
Pakiet

- Grupuje części diagramu klas
- Służy do porządkowania i hierarchizacji diagramów



przykład

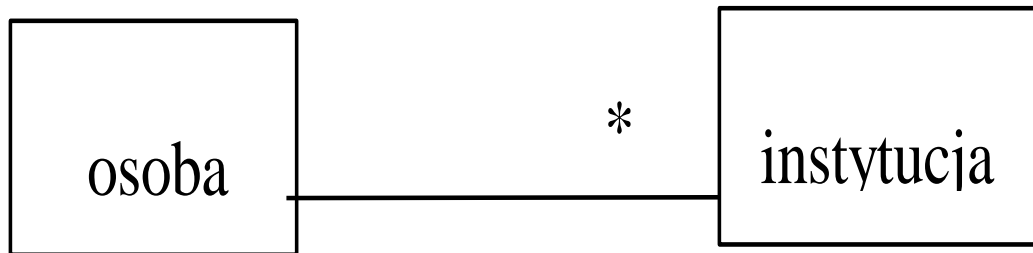
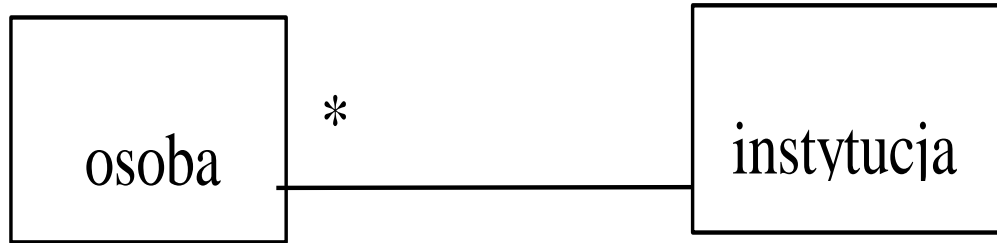
pracownik „pracuje ” dla firmy



powiązania

- czasowniki w opisie problemu.
- Nazwa dwukierunkowego powiązania jest zwykle czytana w określonym kierunku np. pracownik pracuje dla firmy, firma zatrudnia pracownika.
- Powiązania mogą być **wielowartościowe**, zależnie od założeń modelu. **Krotność** określa ile obiektów danej klasy może być w relacji z obiektem innej klasy np.:
 - 1 dokładnie jeden
 - 0 ..1 zero lub jeden
 - M ..N od M do N (liczby naturalne)
 - * od zera do dowolnej liczby całkowitej
 - 1 .. * od jednego do dowolnej liczby całkowitej

Przykład – różnice znaczeniowe



role

- Można określać role obiektów pełnione w powiązaniu np.

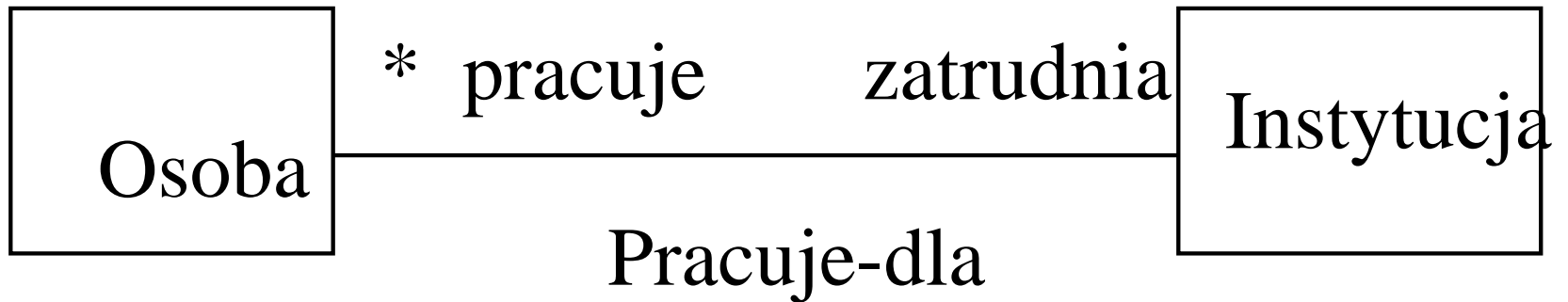
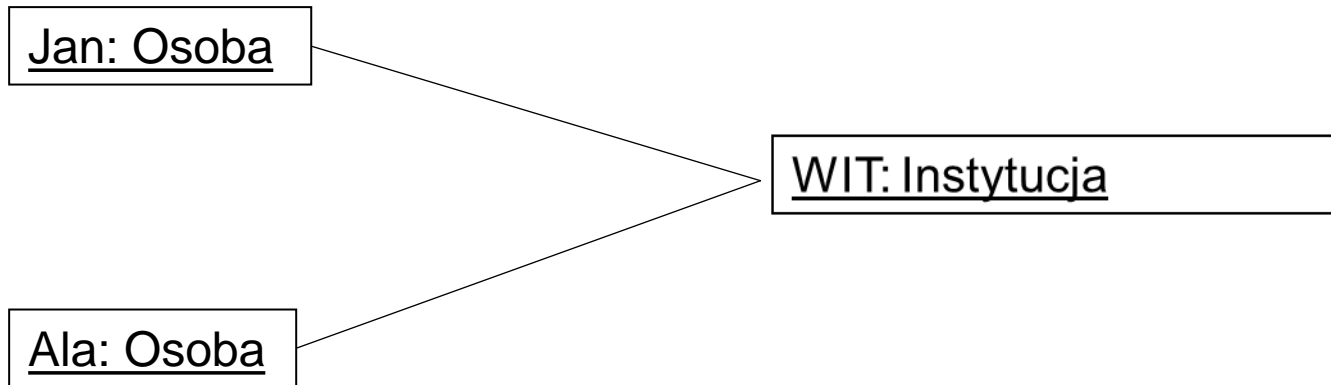


Diagram obiektów



Asocjacja zwrotna

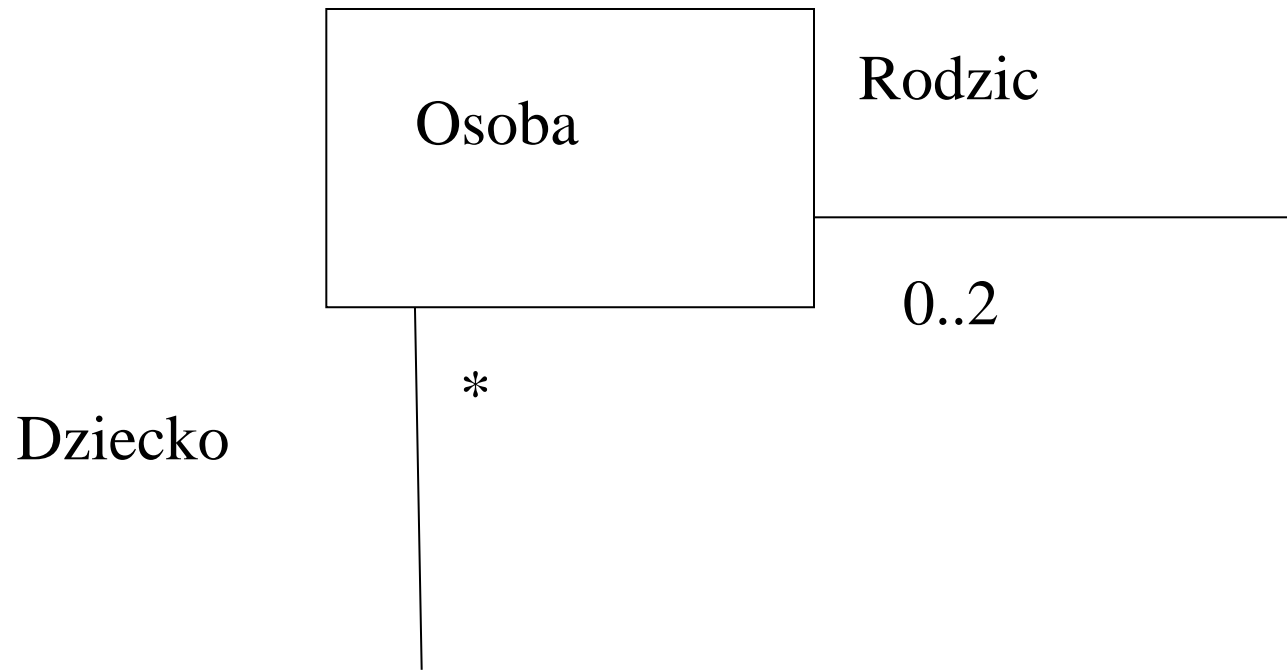
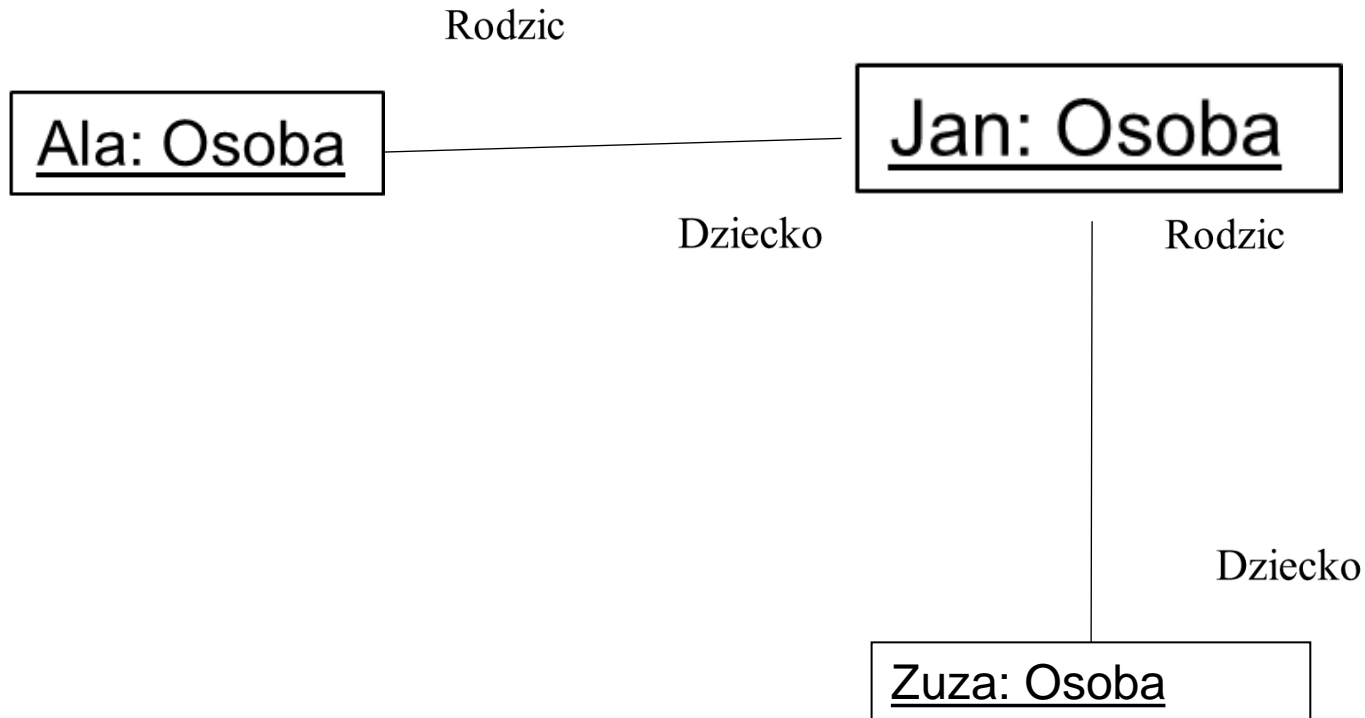
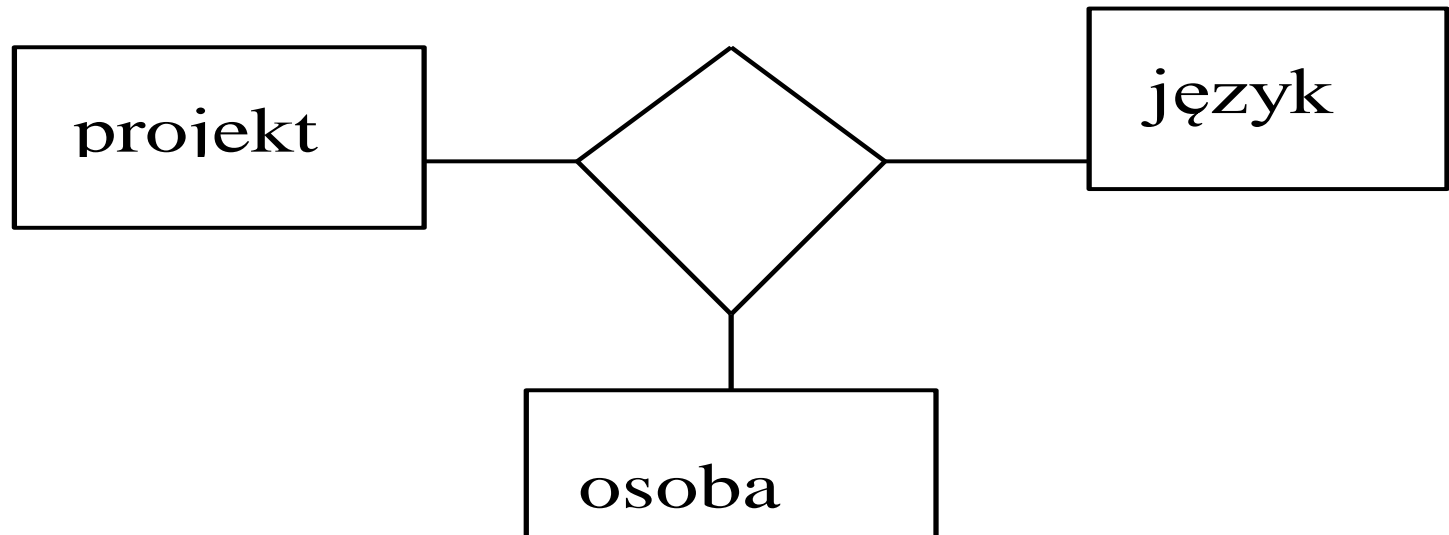


Diagram obiektów - Asocjacja zwrotna



Powiązania ternarne (ang. ternary association)

Pomiędzy 3 lub więcej klasami, nie może być podzielone bez utraty informacji.



Bez relacji ternarnej

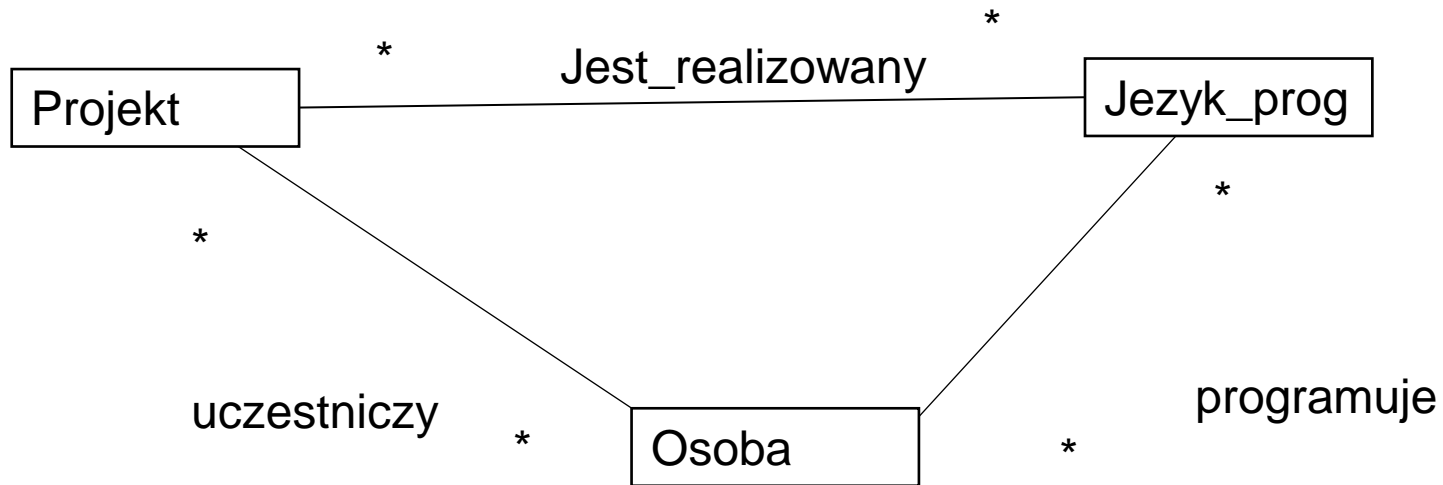


Diagram obiektów

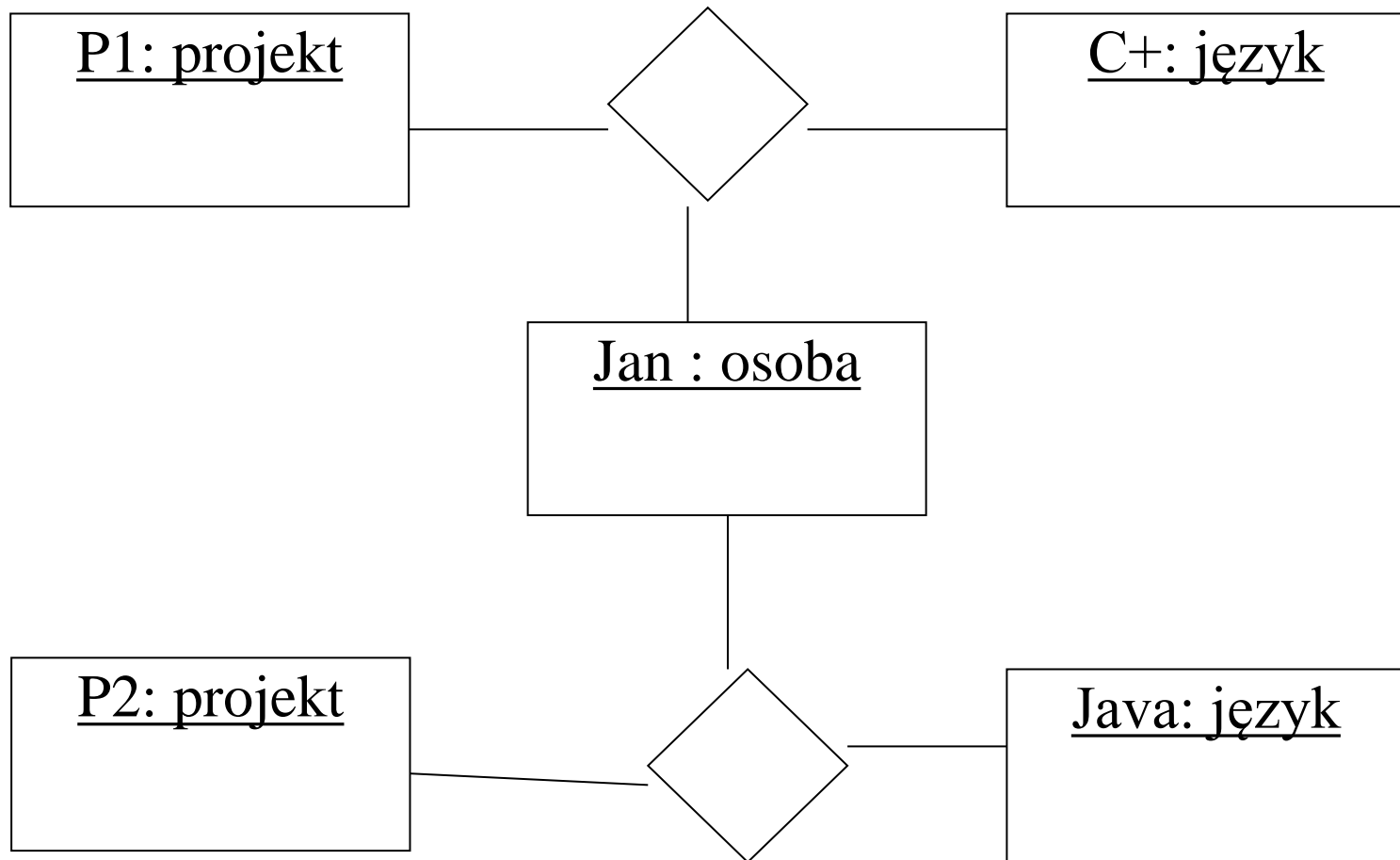
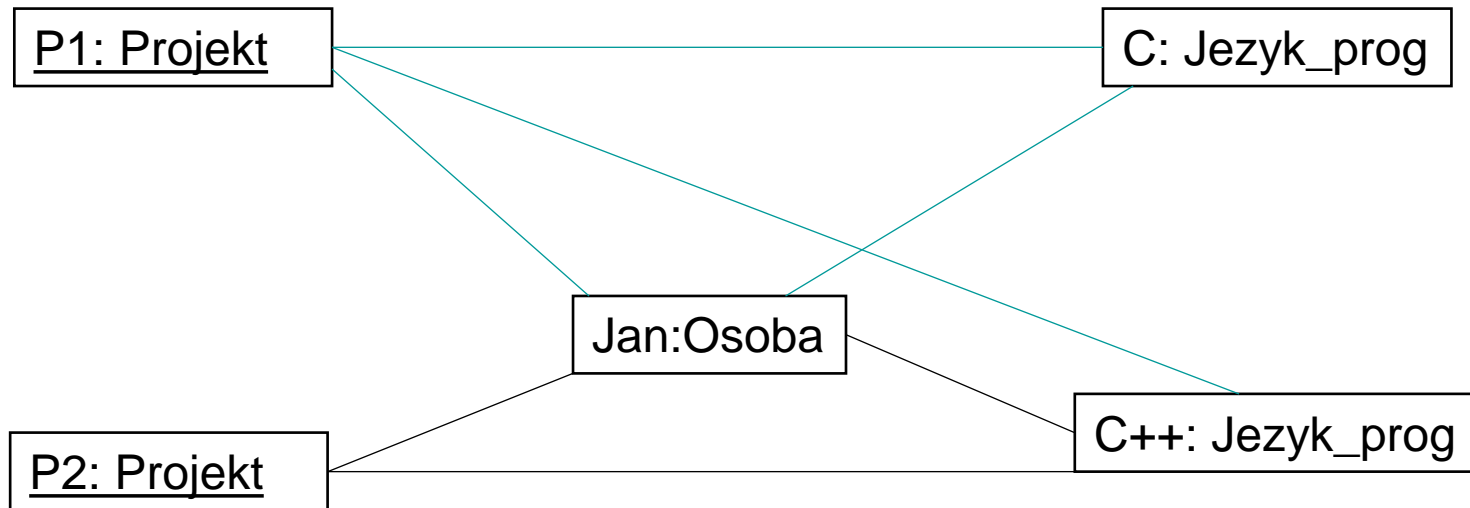


Diagram obiektów



Klasy Plik- User

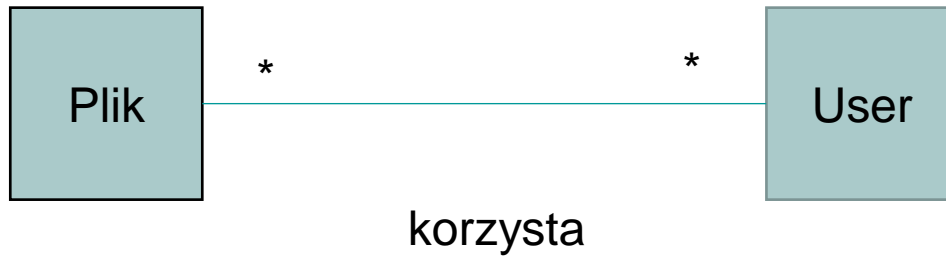
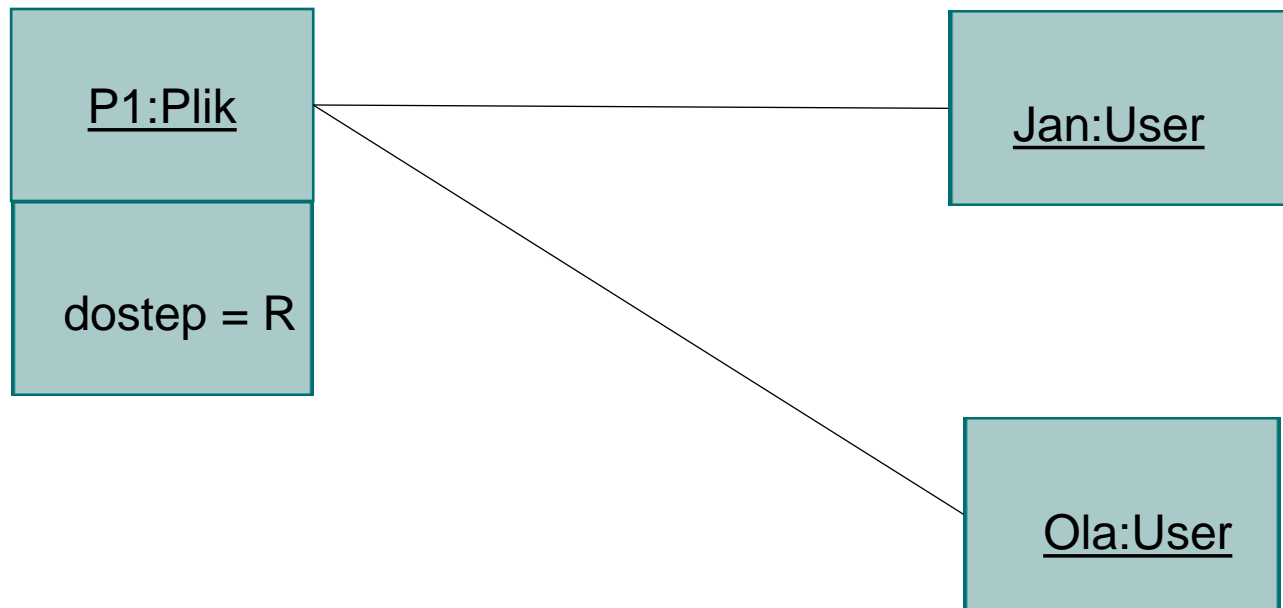


Diagram obiektów



Co jeśli prawa dostępu „włożymy” do klasy User ?

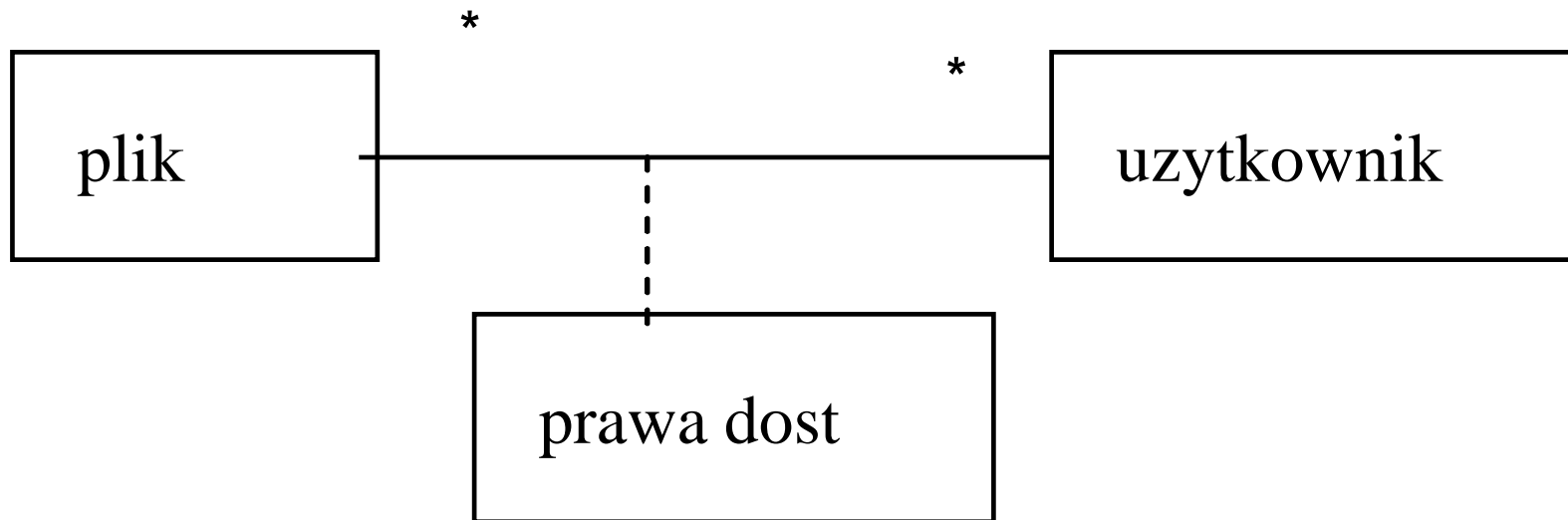
Wariant ze zwykłą klasą

Przy wartościowościach wiele - wiele atrybut ten nie może być dołączony do obiektu np.

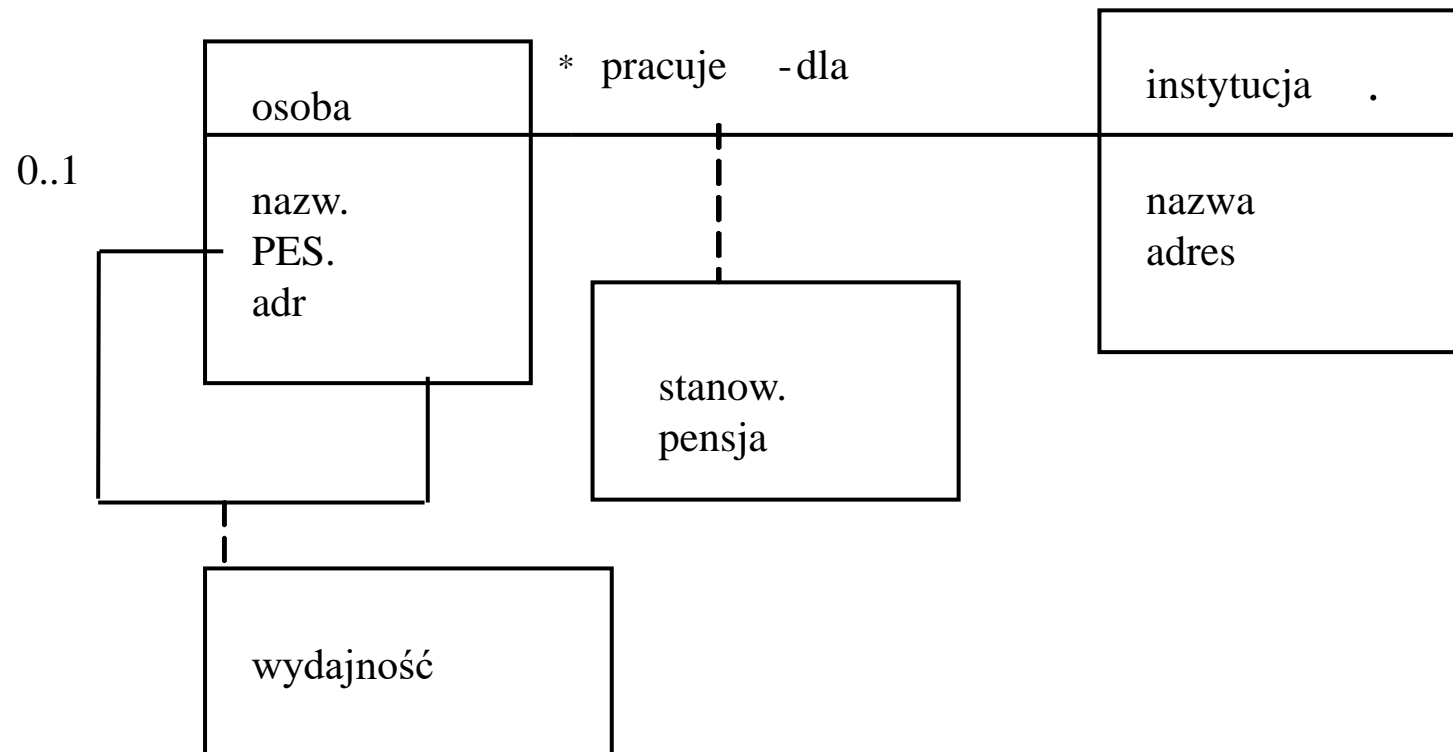


Atrybuty powiązań

Określają pewną własność powiązania
klasa asocjacyjna

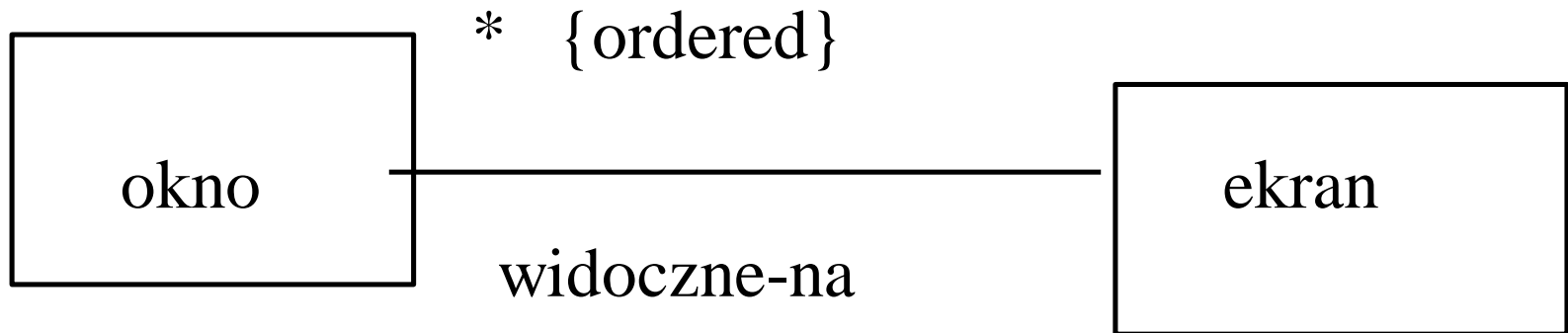


Przykład z klasami asocjacyjnymi



Ograniczenie {ordered}

Powiązania mogą także posiadać własność "uporządkowania"



Różne asocjacje pomiędzy tymi samymi klasami

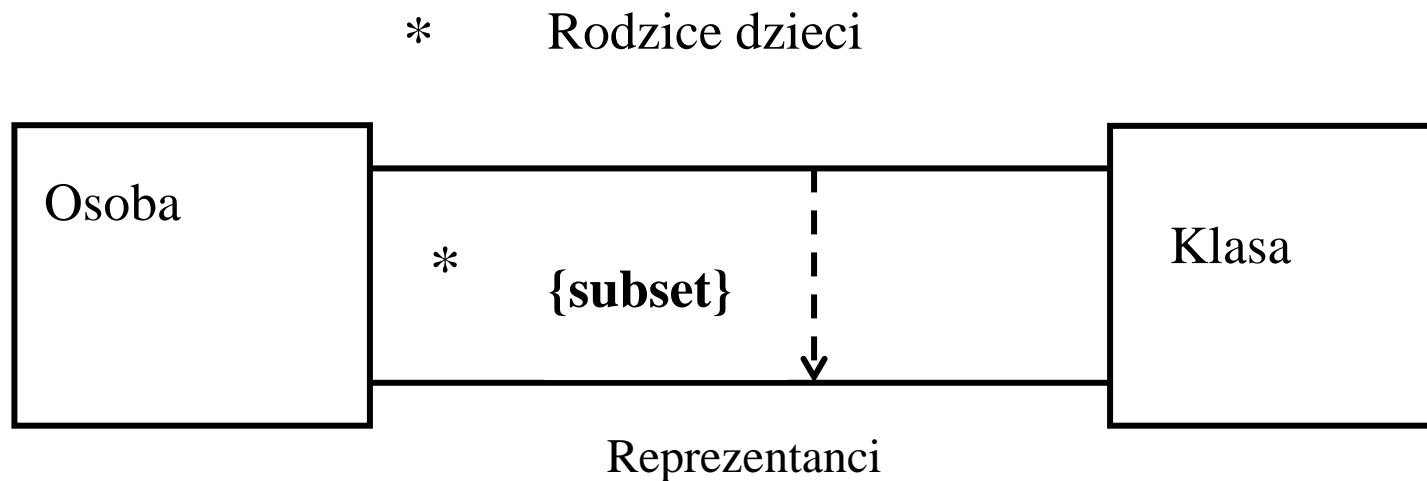
Obiekt Osoba może być w relacji „Reprezentanci” , „Rodzice dzieci” z różnymi obiektami Klasa.



Ograniczenie {subset}

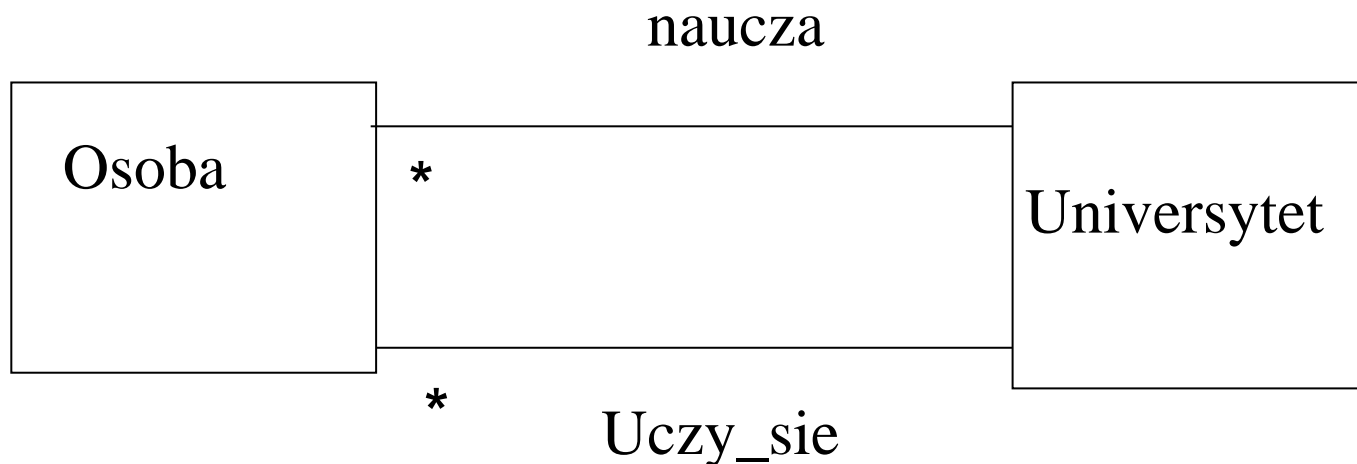
wskazuje, że pewien zbiór (kolekcja) jest włączony w inny zbiór np.

Reprezentanci rodziców dzieci są także rodzicami dzieci.



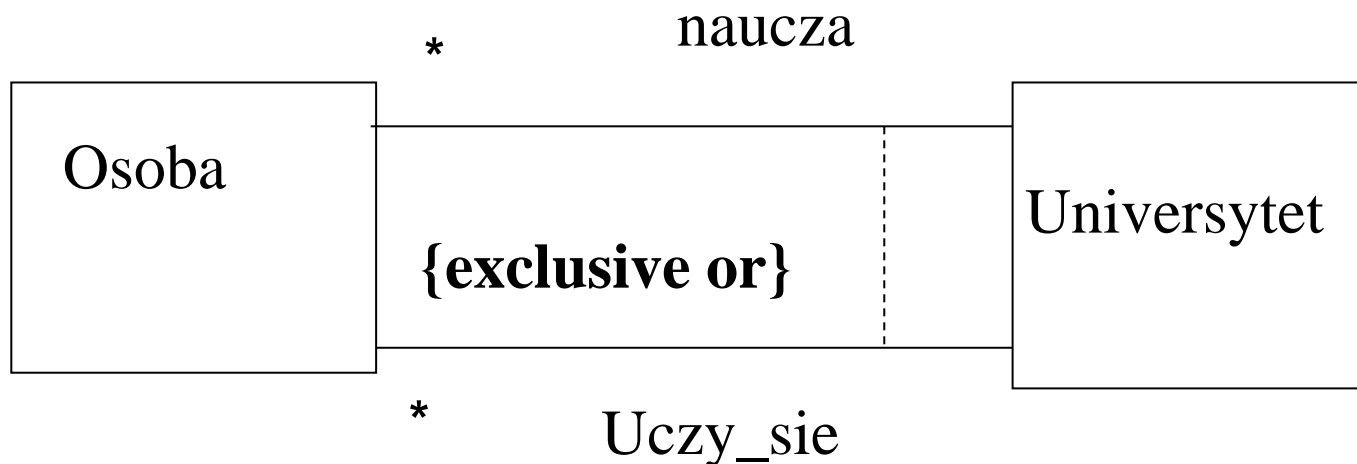
Różne asocjacje pomiędzy tymi samymi klasami

Dla danego obiektu obie relacje asocjacji są możliwe.



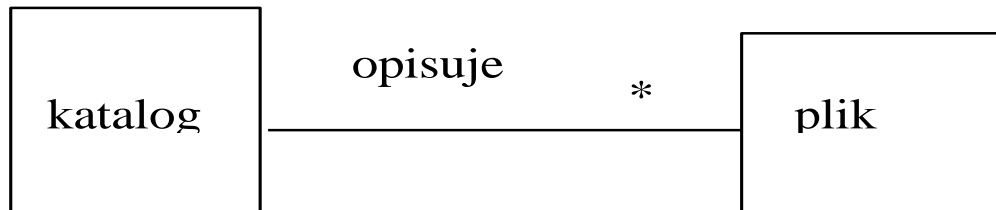
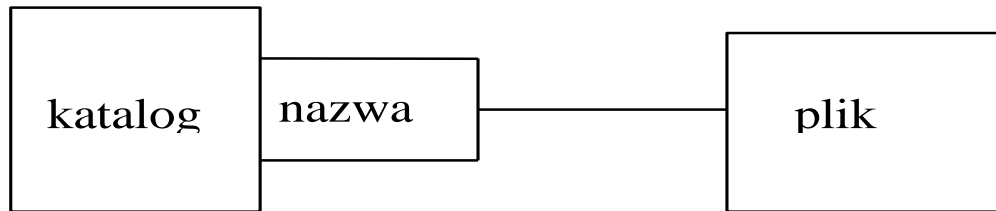
Ograniczenie {exclusive or}

wskazuje, że dla danego obiektu jedynie jedna relacja asocjacji spośród grupy relacji jest właściwa.

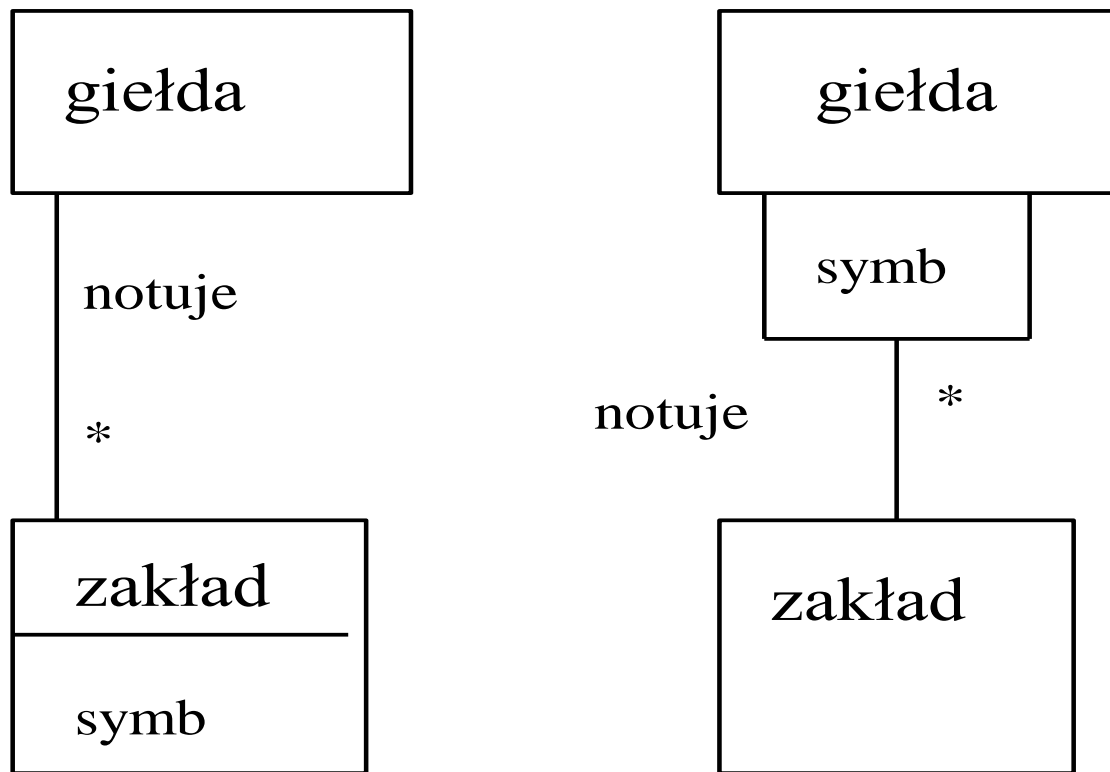


Powiązania "kwalifikowane"

Kwalifikator wyróżnia między wieloma obiektami, sprowadza punkt wiele do 1.



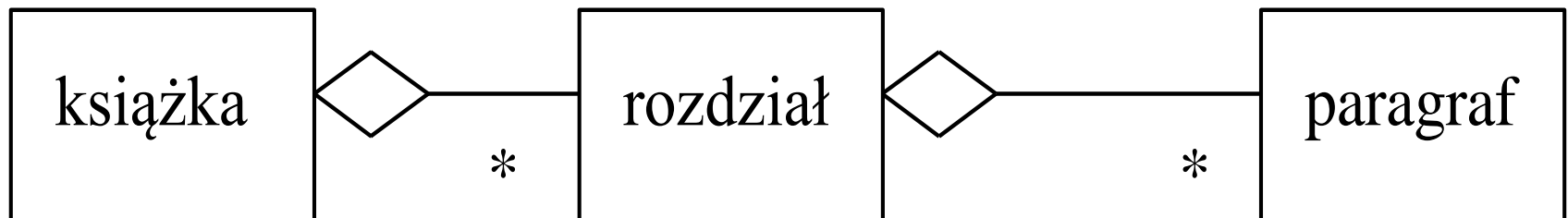
Przykład z kwalifikatorami powiązań



Agregcja -kompozycja

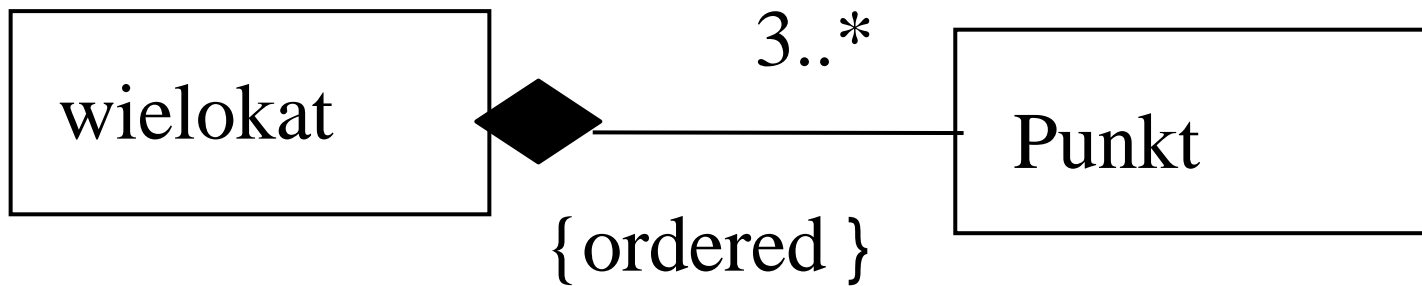
Określa relację :

- "składa się z"
- "jest zbudowany z"
- "jest częścią"



Kompozycja – agregacja całościowa

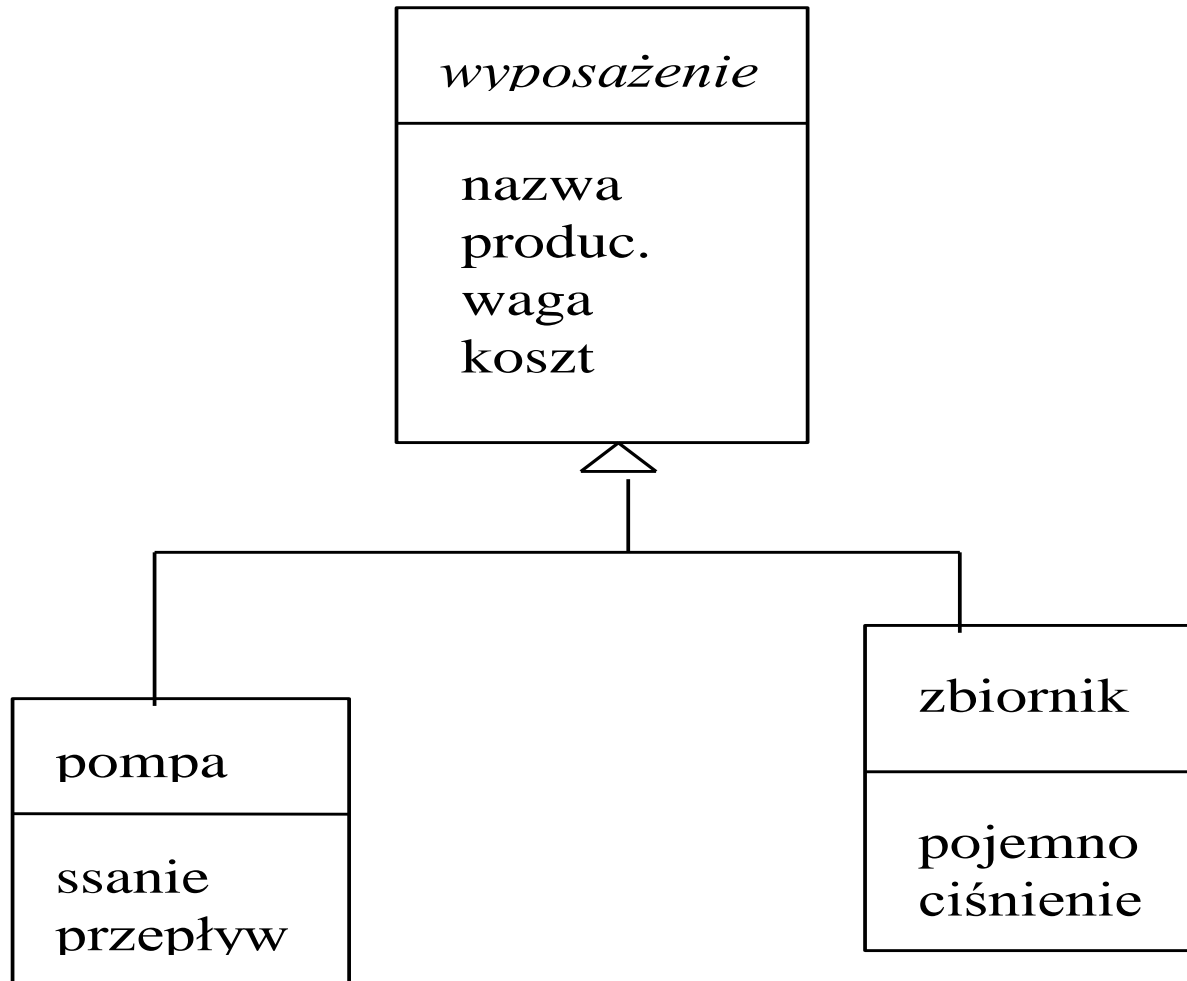
oznacza „fizyczną” agregację np. wielokat składa się z uporządkowanych punktów.



Dziedziczenie i generalizacja

- **Generalizacja** jest związkiem między klasą a jej "ulepszeniami" -podklasami.
- Atrybuty i operacje wspólne dla grupy podklas są umieszczane w superklasie.
- "podklasa" dziedziczy atrybuty, operacje.
- Dziedziczenie jest przechodnie.

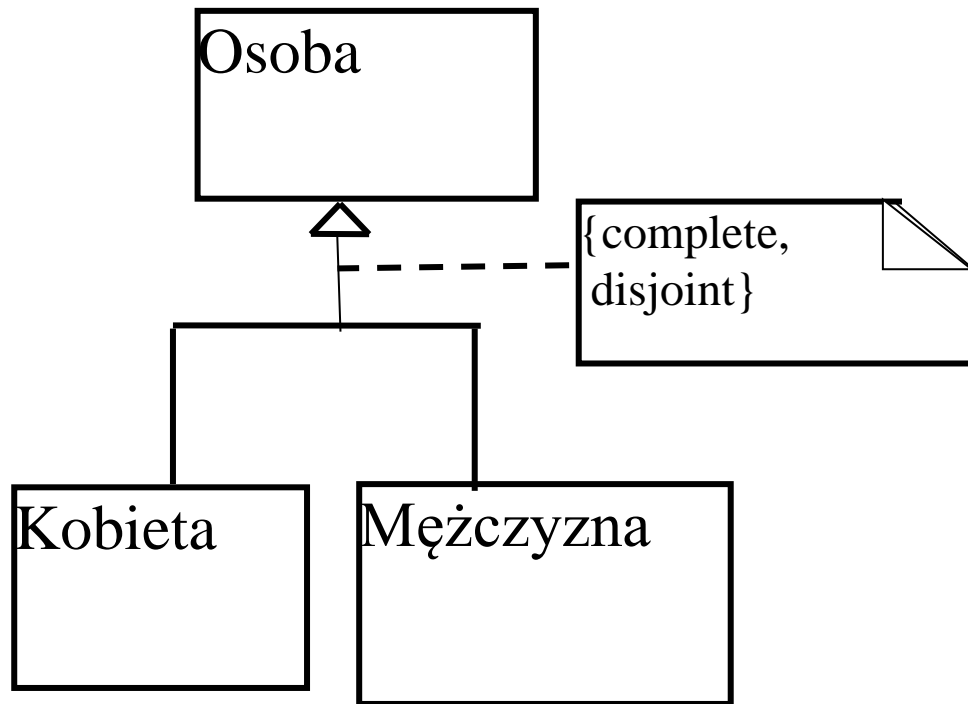
Przykład dziedziczenia



Ograniczenia relacji dziedziczenia

- **{exclusive}** domniemana, obiekt jest instancją tylko jednej podklasy
- **{disjoint}** rozłączna: klasa pochodna od A jest podklasą tylko jednej podklasy klasy A
- **{overlapping}** nakładająca się : klasa pochodna od A należy do produktu kartezyjskiego podklas klasy A np. (urządzenia, urządzenia do rejestracji obrazu, urządzenia do rejestracji dźwięku, magnetowid)
- **{complete}**
- **{incomplete}**

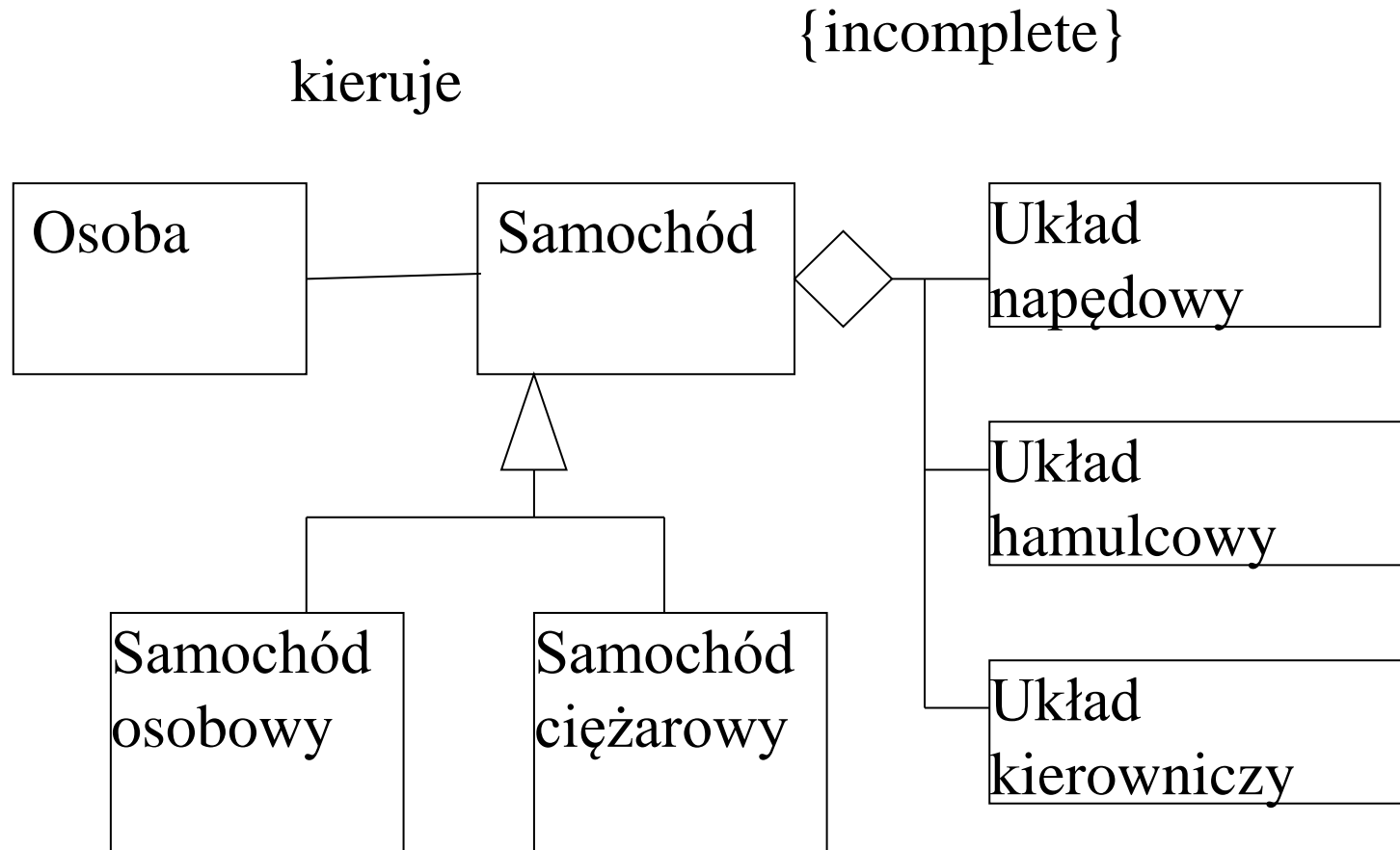
Przykład ograniczeń generalizacji



Podsumowanie

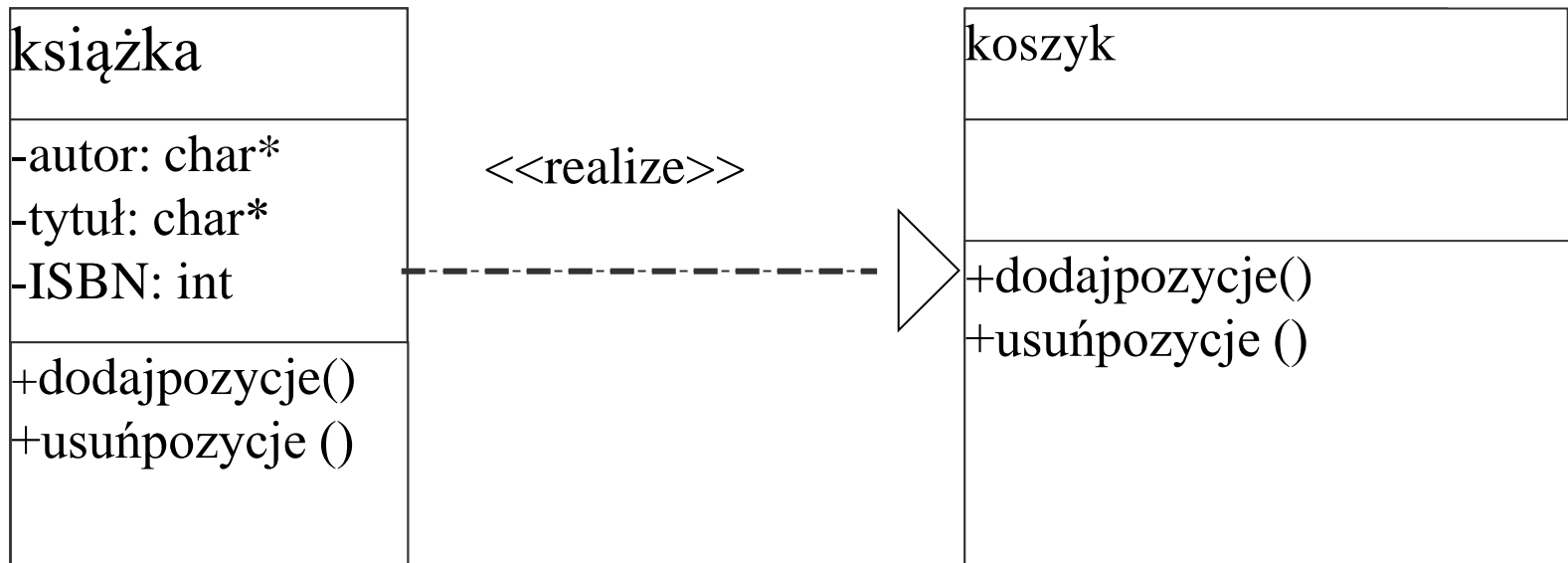
- ze specyfikacji wyodrębnij obiekty (ważne rzeczowniki z dziedziny problemu ale nie dotyczące implementacji),
 - utwórz powiązania,
 - zastosuj generalizację, kompozycję,
 - dodaj atrybuty, operacje,
 - podziel klasy na moduły spójne pod względem świadczonych usług, umieść je w pakietach.
-

Przykład – różne typy relacji



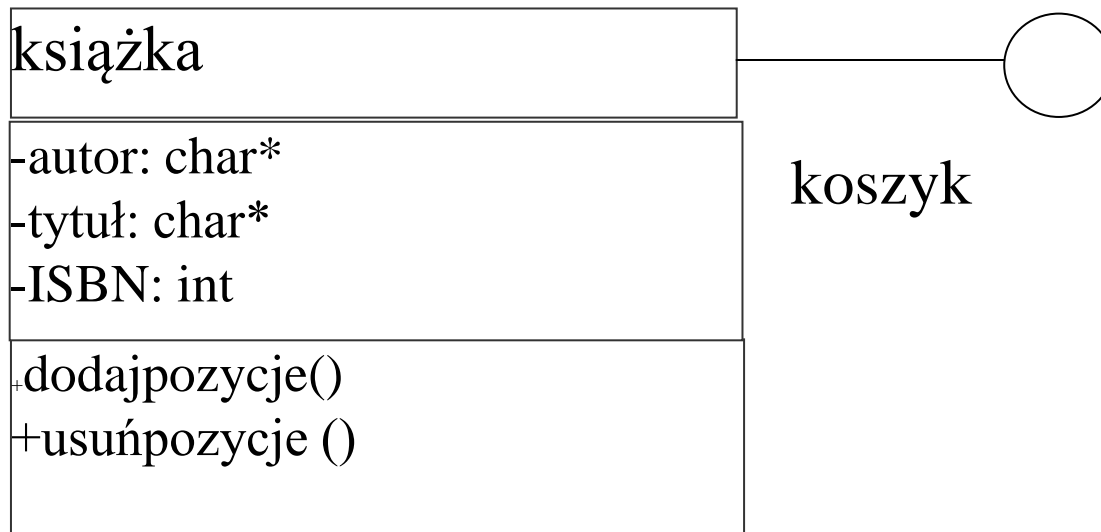
Klasa interfejsowa

Klasa definiuje operacje udostępniane innym obiektom



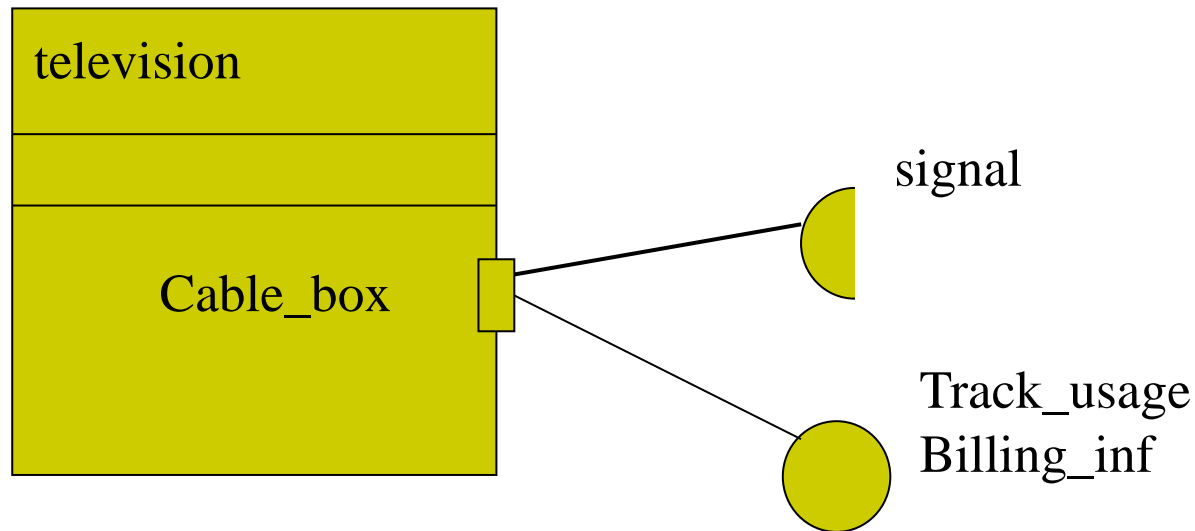
Klasa interfejsowa -2

Inna notacja



Bramy (ports)

Klasa *television* wymaga sygnału z zewnętrznego źródła, firma telewizja kablowa chce mieć możliwość śledzenia używania telewizji. Brama *Cable_box* dostarcza tego interfejsu (*Track_usage*), także dostarczony jest interfejs umożliwiający obciążanie klienta (*Biling_inf*).



Zadanie 1

Określ i narysuj w notacji UML typy relacji pomiędzy obiektami w poniższych zdaniach. Odpowiedź należy uzasadnić

1. Klient ma miejscówkę na określony pociąg
2. Listonosz dostarcza przesyłki
3. Bileter sprzedaje bilety
4. W plecaku znajdują się książki, zeszyty
5. Magnetofon, magnetowid są urządzeniami do rejestracji dźwięku

Zadanie 2

Na podstawie podanego poniżej zbioru słów opracuj spójny diagram klas w UML, pokazujący relacje między obiektami klas.

Należy podać typ relacji (powinny występować wszystkie typy relacji), jej nazwę ewentualnie krotność. **Odpowiedź należy uzasadnić.**

**{Zamek, most zwodzony, wieża, schody,
korytarz, pokój, okno, podłoga, duch,
strażnik, kucharz, hrabia, hrabina, lokaj}**
