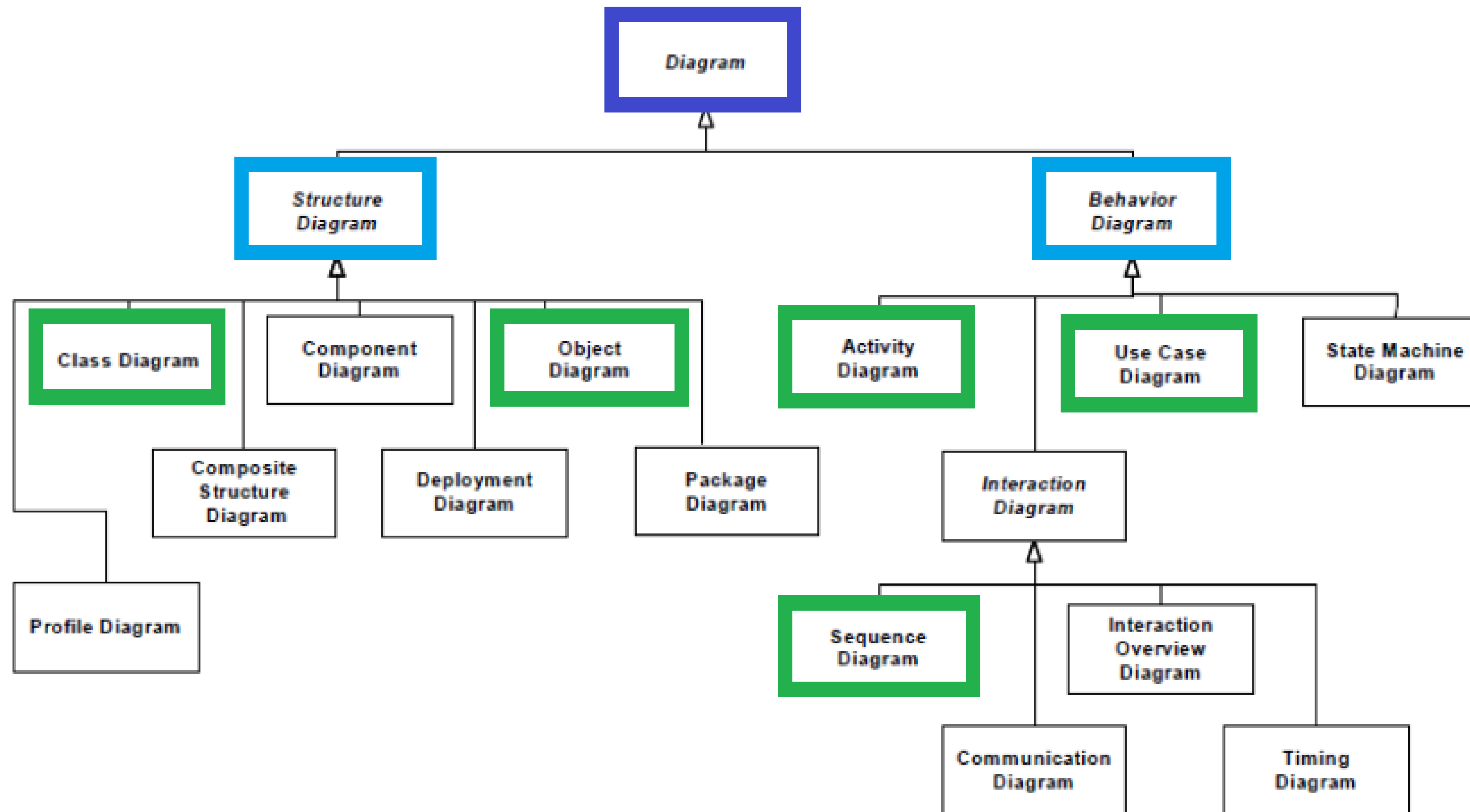


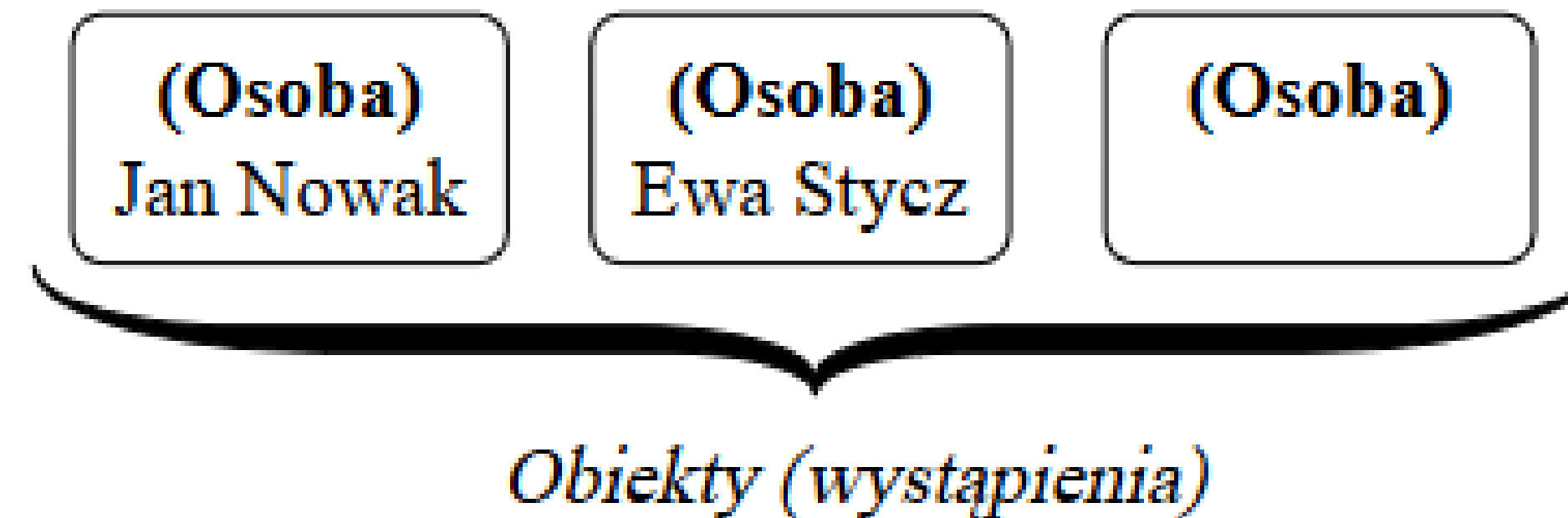
Diagram Klas

Diagramy języka UML



Obiekt

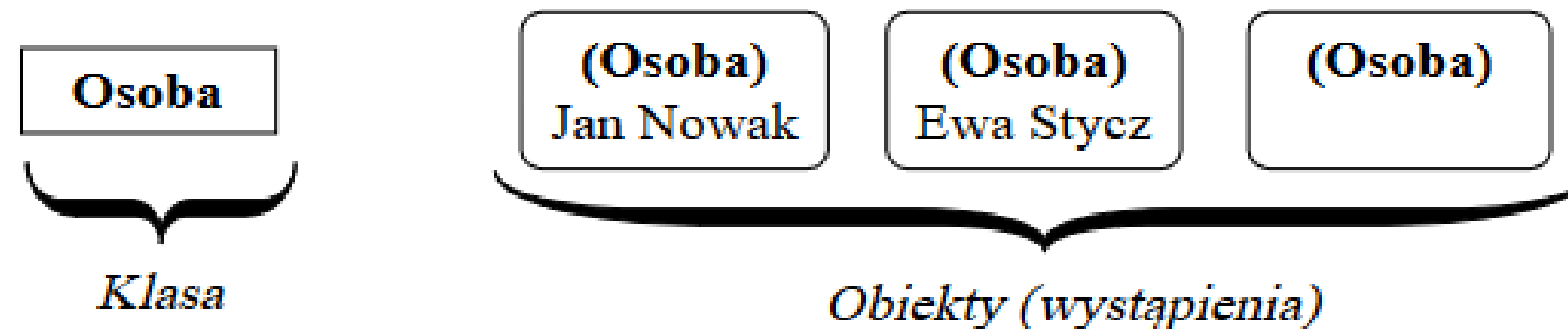
Obiekt - pojęcie, byt lub rzecz istniejąca w świecie rzeczywistym, posiadająca określone granice oraz tożsamość i stan, sprzyjająca zrozumieniu modelowanego świata rzeczywistego.



np.: *Jan Nowak, Ewa Strycz, laptop Ewy, Pałac Kultury, Pani Irena ze sklepu na rogu, Pan Waldemar.*

Obiekt a klasa (obiektów)

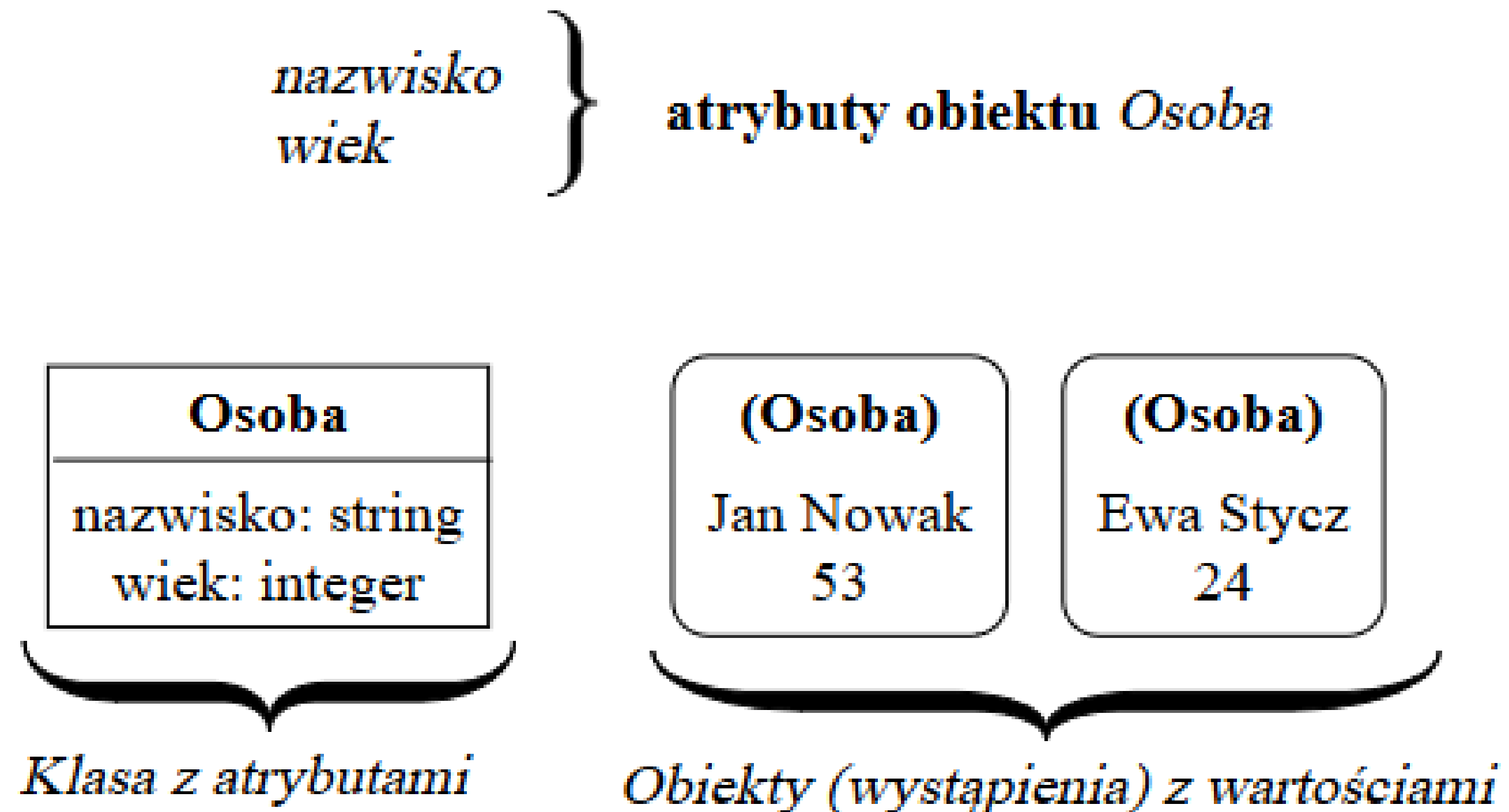
Klasa - grupa obiektów posiadająca:
podobne **własności (atrybuty)**;
podobne **zachowanie (operacje)**;
podobne **związki z innymi obiektami**;



np.: Jan, Ewa, Pani Irena i Pan Waldemar są osobami, zatem możemy je powiązać i stworzyć klasę *Osoba*

Atrybuty (obiektów)

Atrybut jest wartością przechowywaną przez obiekty w ramach klasy. Nie posiada tożsamości.



Operacje i metody

Operacja jest funkcją lub transformacją, która może być stosowana do obiektu lub przez obiekt.

Operacja własnością klasy.

Wszystkie obiekty klasy podlegają tym samym operacjom.

zatrudnij
zwolnij
wypłać_dewidendę } operacje na obiektach klasy *Firma*

Ta sama operacja może być zastosowana do obiektów **wielu** klas (**polimorfizm**).

Natomiast **metoda** jest implementacją operacji dla **jednej klasy**.

Diagram klas

- jest podstawowym diagramem obrazującym logiczną **strukturę systemu**,
- przedstawia występujące w systemie **klasy** i zachodzące pomiędzy nimi statyczne **relacje**,
- klasy na diagramie reprezentowane przez prostokąty mogą oprócz **nazwy klasy** zawierać informacje o jej **atrybutach** i **operacjach** dostarczanych przez klasę (metodach),
- jest najbardziej „**pojemnym**” diagramem, dlatego jest najczęściej stosowany do generowania kodu na podstawie modelu.

Tworzenie diagramu klas

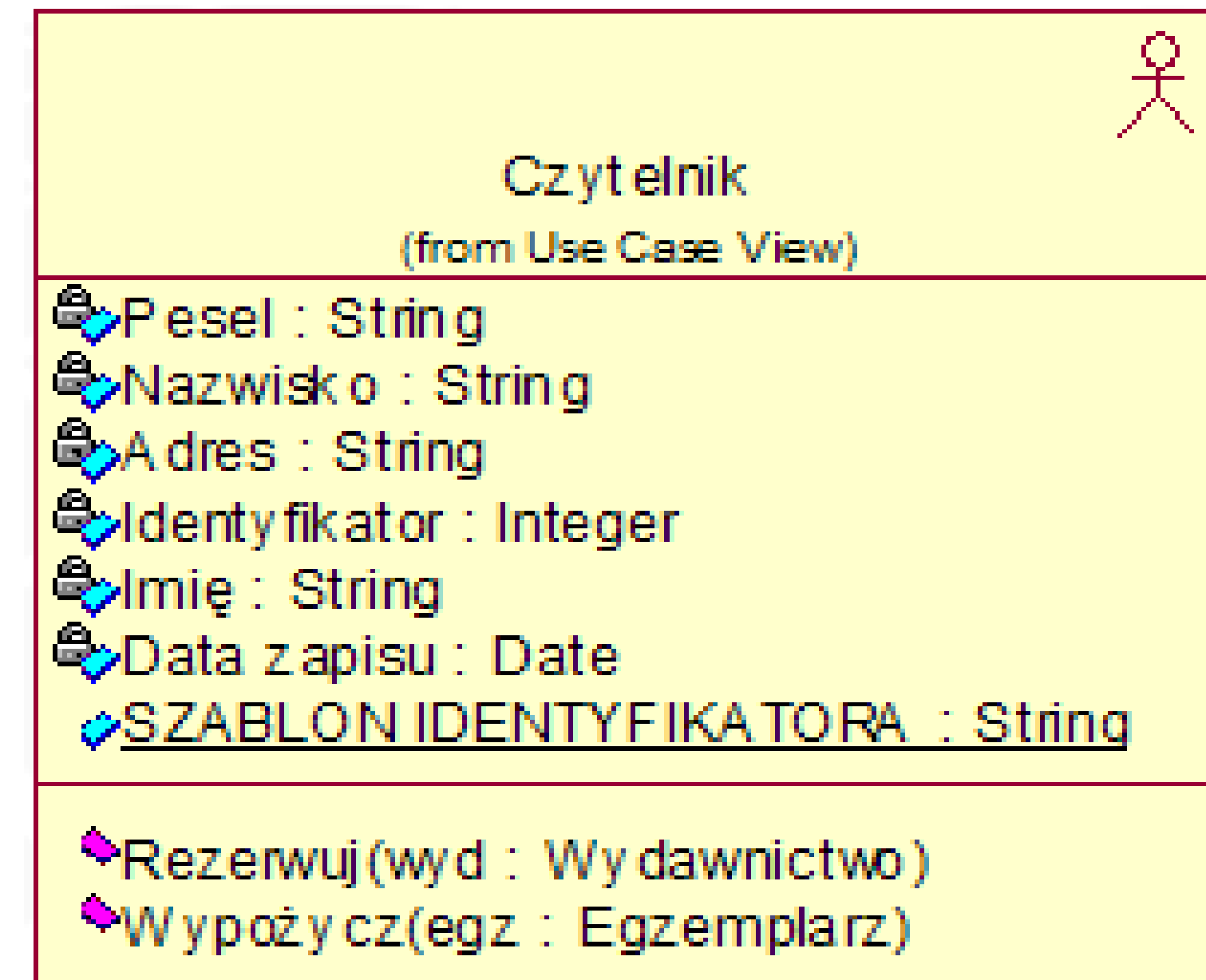
obejmuje:

identyfikację klas;
identyfikację związków pomiędzy klasami;
identyfikację i definiowanie atrybutów;
identyfikację i definiowanie operacji (metod) i komunikatów.

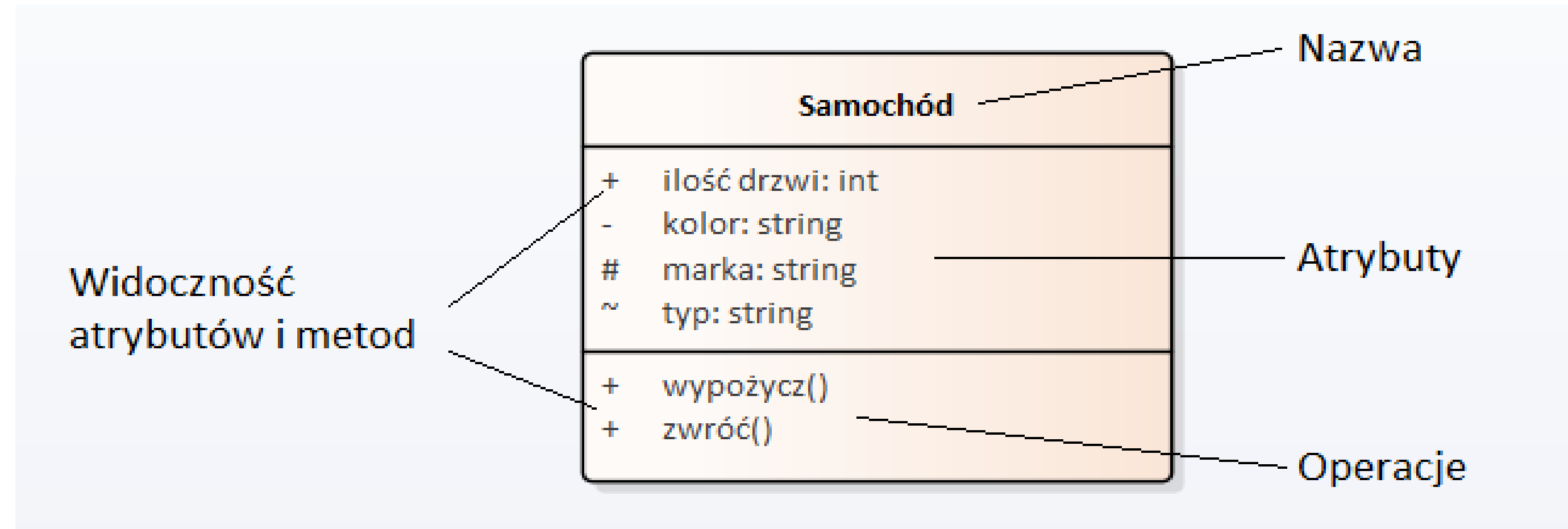
Notacja klasy

Klasa jest reprezentowana na diagramie przez prostokąt z wydzielonymi przedziałami:

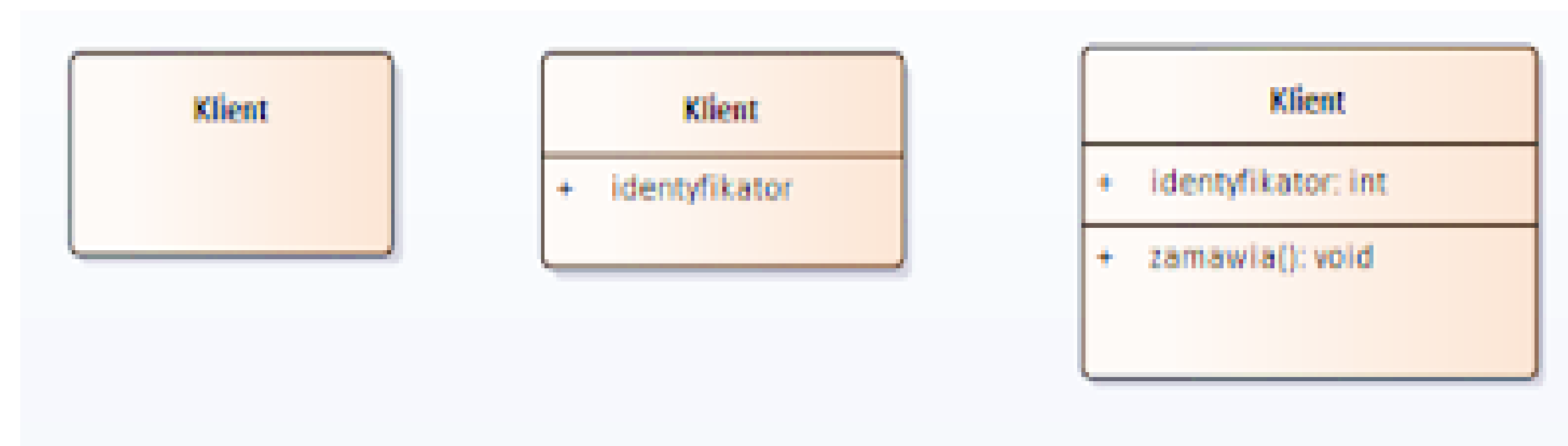
nazwą;
atrybutami;
operacjami.



nazwa
atrybuty
operacje



Klasę można zaprezentować w sposób mniej lub bardziej szczegółowy, ukazując lub ukrywając pola opisujące własności klasy.

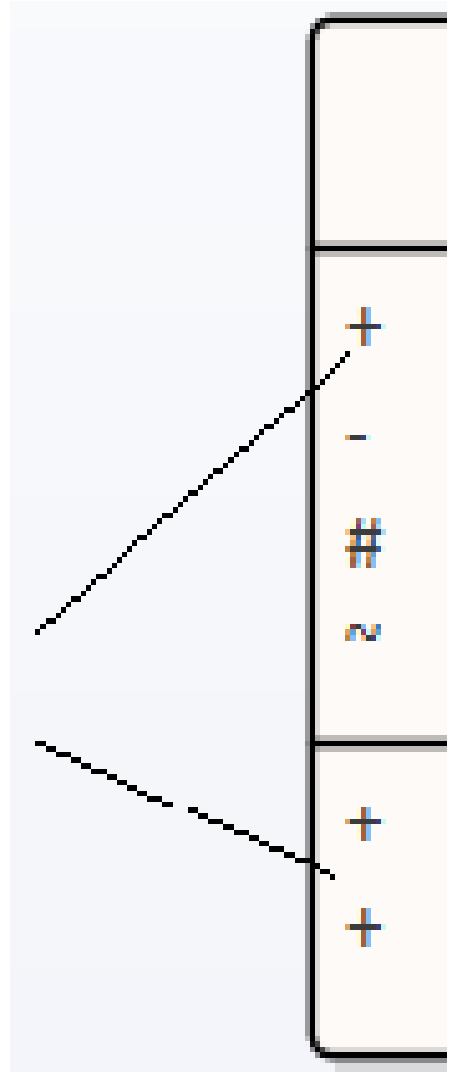


Notacja widoczności

Widoczność (scope) - nazywana również **dostępnością**. Określa czy i jak widoczna jest cecha lub operacja dla innych elementów systemu.

Rozróżniamy 4 rodzaje widoczności:

- +** **publiczny** – widoczny z każdego miejsca systemu,
- **prywatny** - widoczny tylko wewnątrz swojej klasy,
- #** **chroniony** - widoczny wewnątrz klasy i jej podklas,
- ~** **publiczny w zakresie pakietu** - widoczny wewnątrz własnego pakietu, czyli w obrębie pewnego zestawu klas.



Związki klas

zależność (ang. *dependency*),
asocjacja (ang. *association*),
agregacja (ang. *aggregation*),
kompozycja (ang. *composition*),
generalizacja-specjalizacja (ang. *generalization-specialization*).

Zależność (ang. *dependency*)

to najłagodniejsza relacja łącząca klasy. Informuje nas, że zmiana w jednej klasie powoduje zmianę w drugiej (chwilową). Relacja ta ma wiele typów, np. use, create i inne.



np.: Klasa *Samochód* używa klasy *Droga*
(często opisuje się frazami: „korzysta z”, „oddziałuje na”, „ma wpływ na”, „tworzy”).

Zależność c.d.

«call» - operacje w klasie A wywołują operacje w klasie B

«create» - klasa A tworzy instancje klasy B

«instantiate» - obiekt A jest instancją klasy B

«use» - do zaimplementowania klasy A wymagana jest klasa B



Asocjacja (ang. *association*)

informuje o tym, że obiekt jednej klasy, jest czasowo powiązany z obiektem klasy drugiej (często do relacji asocjacji dodaje się nazwy asocjacji, krotności i role).

Krotność informuje, jaka ilość obiektów jest zaangażowana w asocjację.

Rola jak sama nazwa wskazuje, informuje klasę o roli jaką posiada druga klasa.



np.: relacja asocjacji *ma konto* pomiędzy klasą *Osoba* a klasą *Bank*.

Krotność	Znaczenie
0..1	Brak obiektu lub jeden obiekt.
0..*	Bez ograniczenia liczby obiektów (łącznie z ich brakiem).
1	Dokładnie jeden obiekt.
1..*	Przynajmniej jeden obiekt.
n	Dokładnie n obiektów.
m .. n	Od m do n obiektów.

Krotności są pojedynczymi liczbami albo zakresami liczb.

Nawigowalność (asocjacje skierowane)

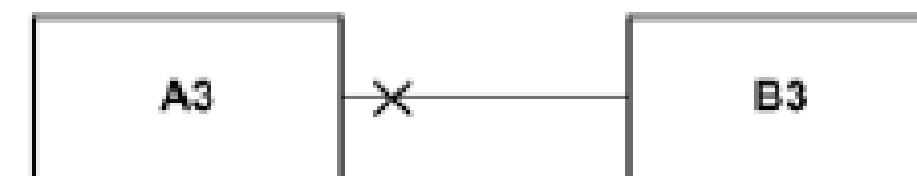
- posiadanie odwołania do klas
- przejście z klasy A do klasy B
- przejście z klasy A4 do klasy B4 z brakiem możliwości powrotu



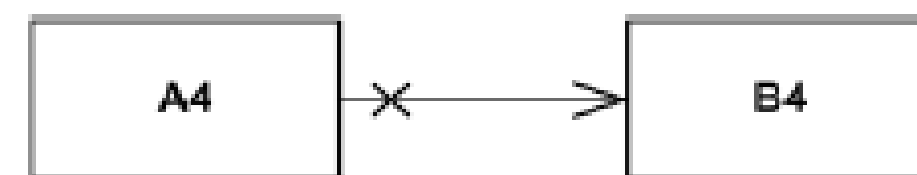
nieokreślona nawigowalność między A1 a B1 oraz B1 a A1



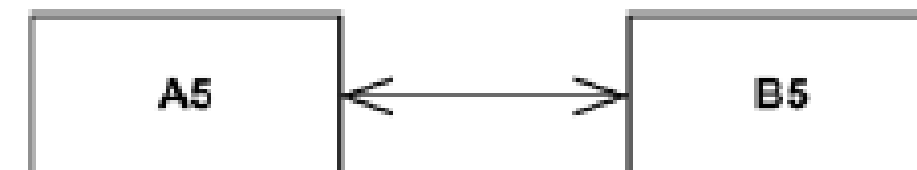
nawigowalność między A2 a B2 oraz nieokreślona nawigowalność między B2 a A2



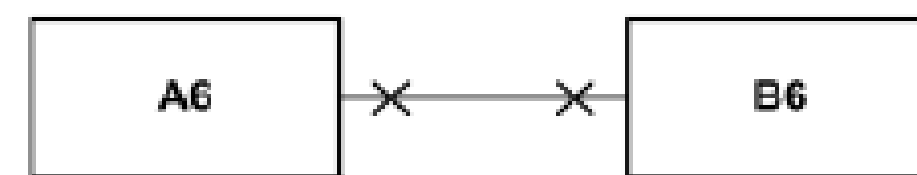
brak nawigowalności między B3 a A3 oraz nieokreślona nawigowalność między A3 a B3



nawigowalność między A4 a B4 oraz brak nawigowalności między B4 a A4



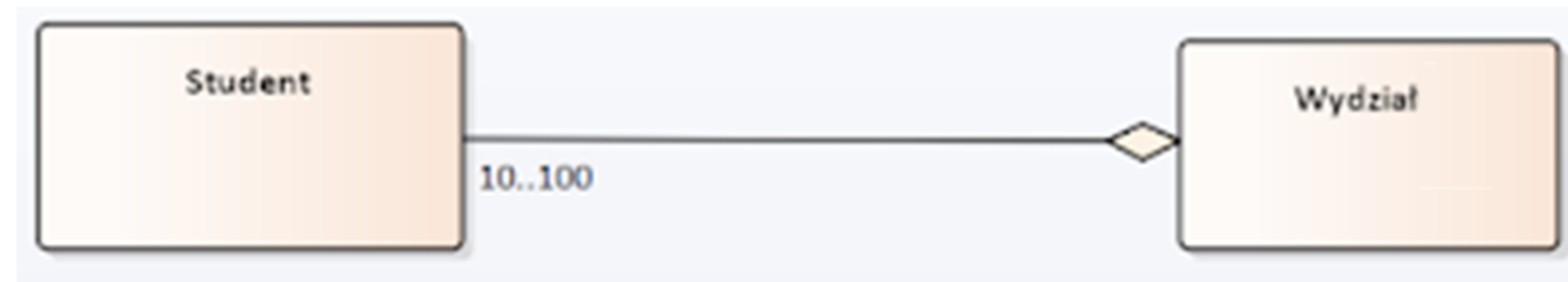
obustronna nawigowalność między A5 a B5



obustronny brak nawigowalności między A6 a B6

Agregacja (ang. *aggregation*)

nazywana również **agregacją częściową** jest relacją typu część - całość. Ważne jest, że część może należeć do kilku całości, a jej istnienie jest od nich niezależne.



np.: klasa *Wydział* zawiera od 10 do 100 *Studentów*. Zlikwidowanie klasy *Wydział* nie spowoduje skasowania klasy *Student*, ponieważ może ona być częścią innej klasy.

notacja: romb po stronie klasy reprezentującej właściciela

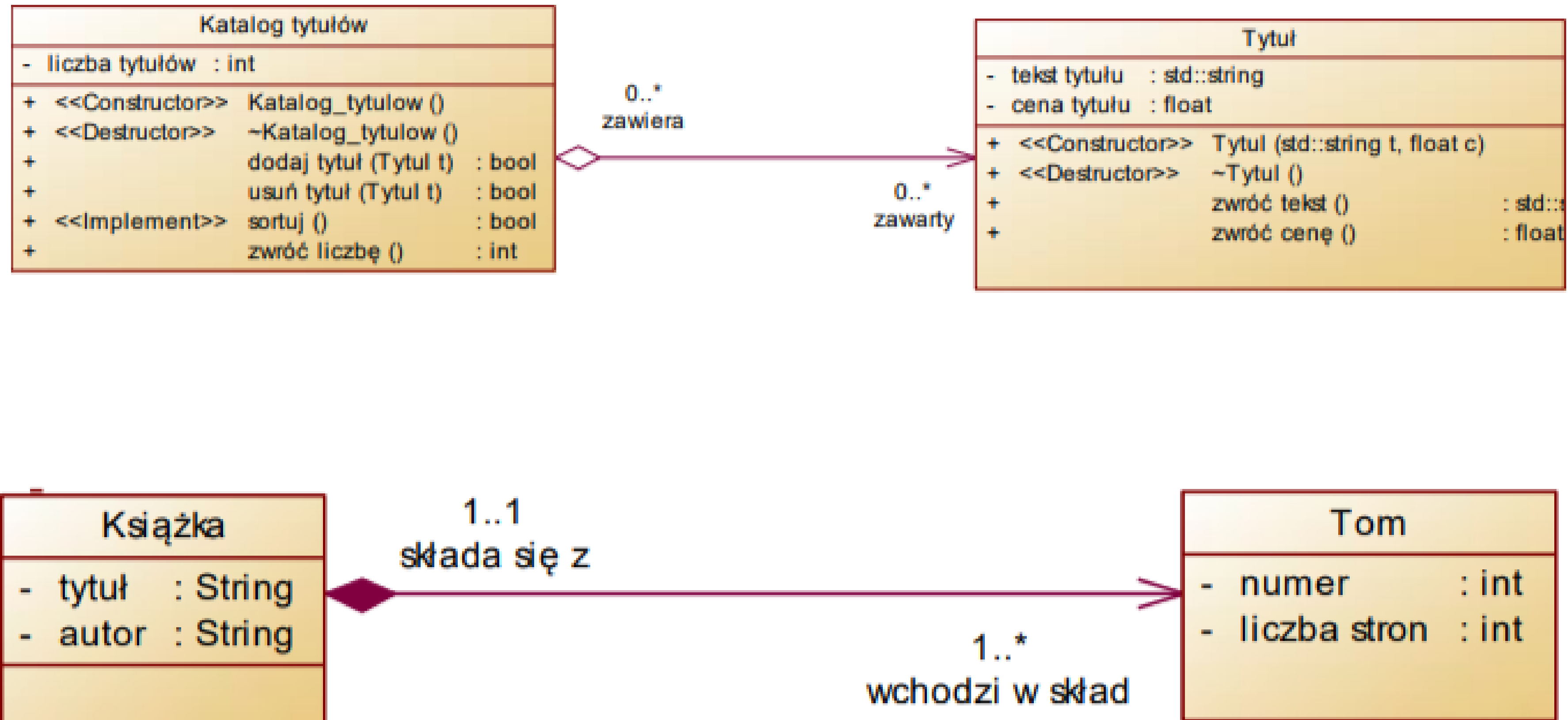
Kompozycja (ang. *composition*)

inaczej zawieranie, lub agregacja całkowita. Podobnie jak agregacja, informuje nas o tym, że jedna klasa zawiera obiekty innej (całość-część). Różnica polega na tym, że klasa która stanowi całość jest wyłącznym właścicielem części. Skasowanie całości powoduje skasowanie części.



np.: klasa Blok zawiera 15 Mieszkań. Likwidacja klasy Blok spowoduje skasowanie klasy Mieszkania.

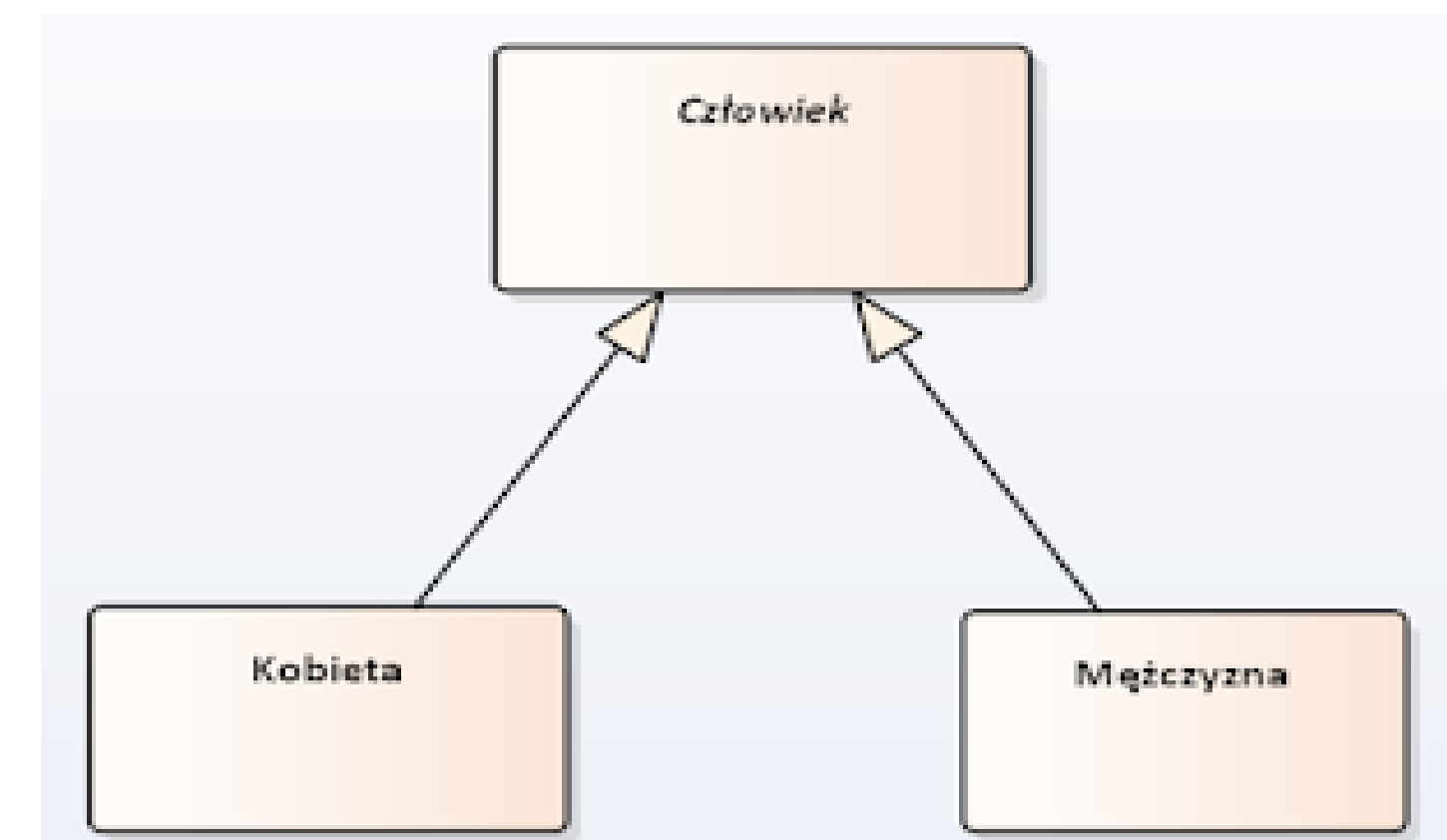
notacja: zaczerntoniony romb po stronie klasy reprezentującej całość

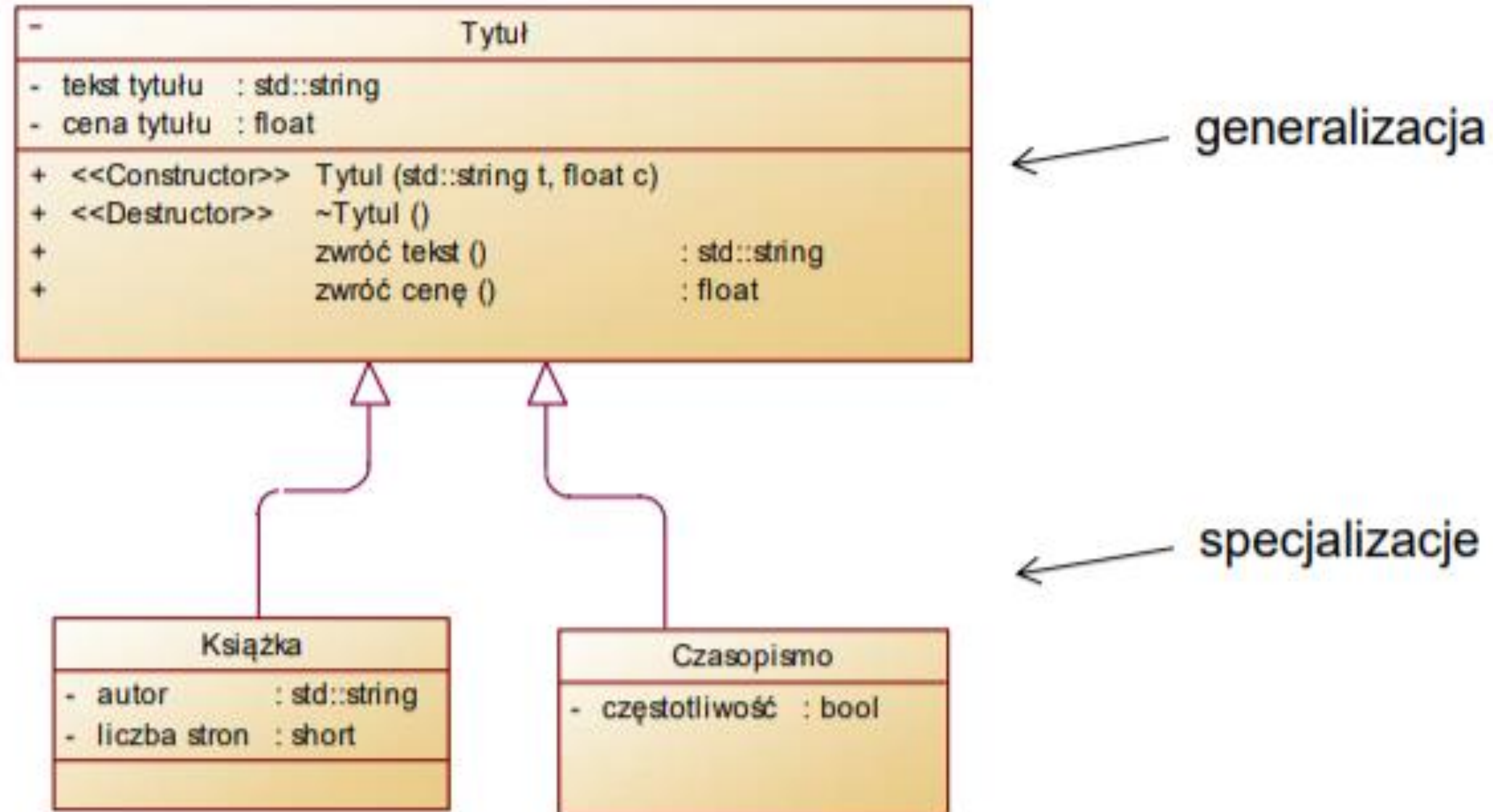


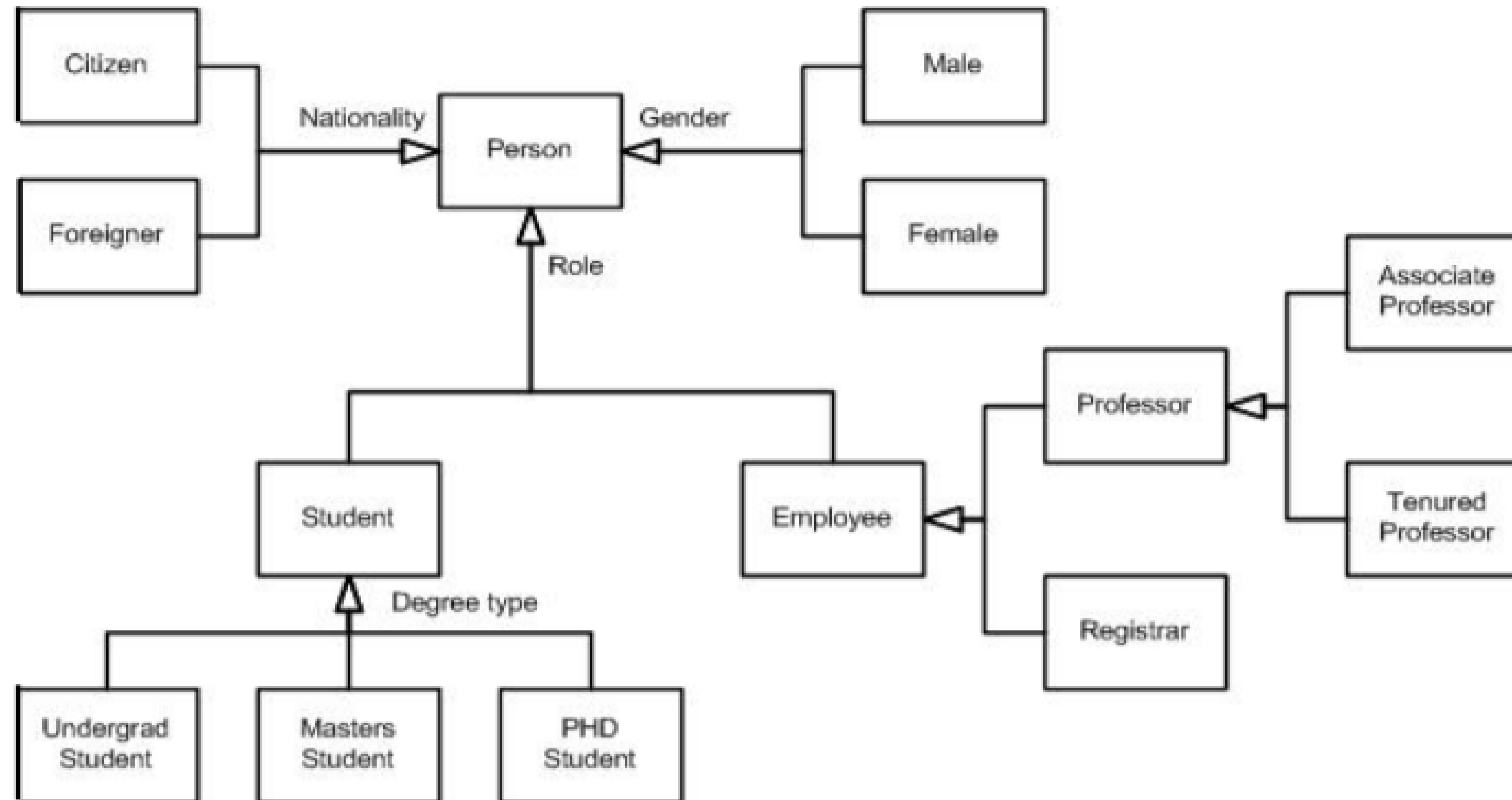
Generalizacja (ang. generalization)

jest relacją między klasami - ogólną i szczególnymi. Klasa ogólna posiada swoje atrybuty, które jej podklasy dziedziczą. Zdarza się również, że klasa ogólna jest klasą abstrakcyjną, czyli taką, która nie zawiera w sobie żadnych obiektów.

np.: ogólna klasa abstrakcyjna
Człowiek generalizuje klasę Kobieta i
klasę Mężczyzna.







Podsumowanie relacji między klasami



Diagram klas - przykład

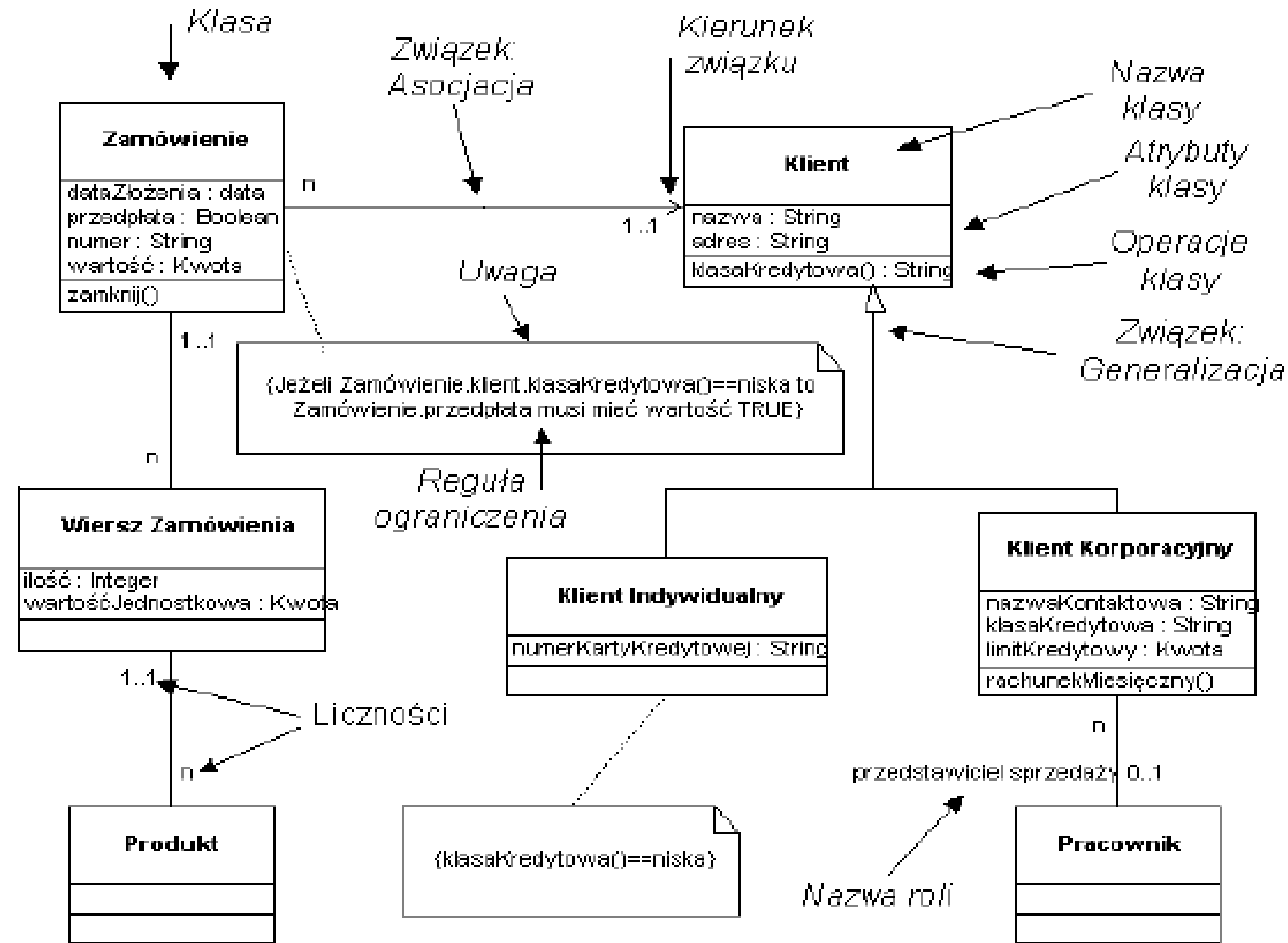
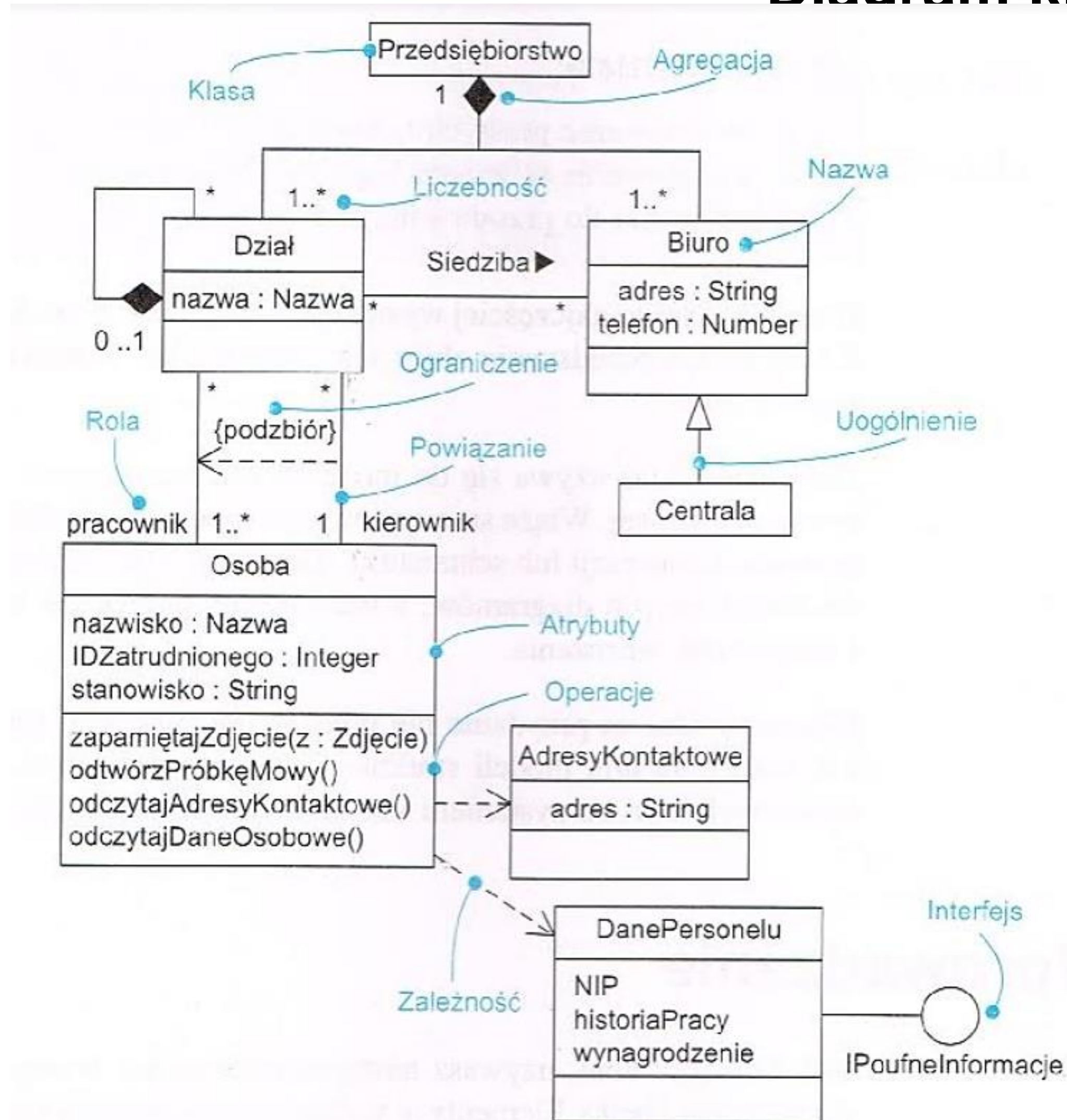


Diagram klas - przykład



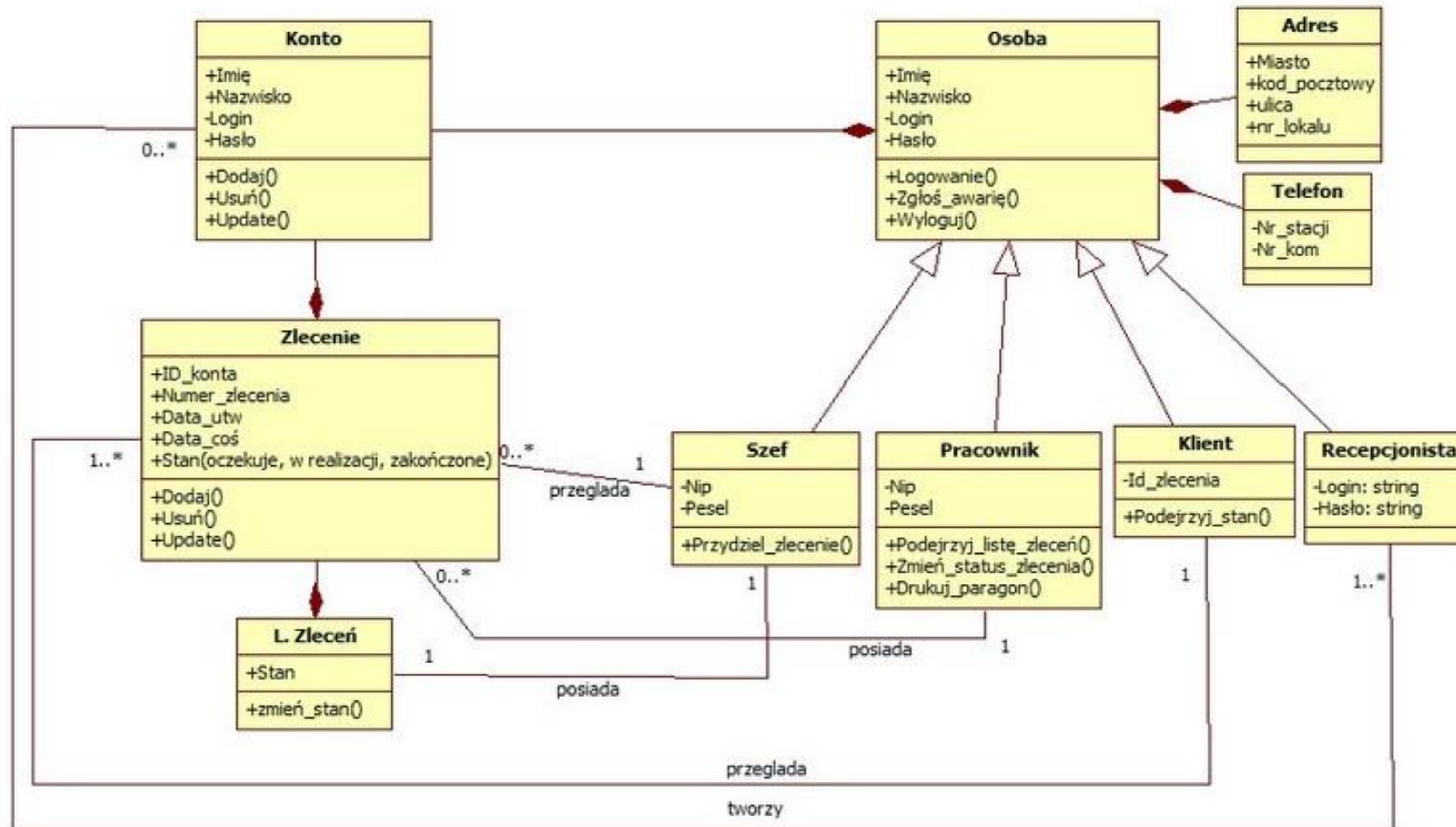


Diagram klas

UWAGA 1: diagram klas nie może być jednak „kalką” diagramu ERD – powinien uwzględniać, oprócz odmiennej notacji, także specyficzne aspekty „obiektowości”: operacje, sekcje klasy, związki dziedziczenia, ewentualne klasy abstrakcyjne oraz interfejsy, związki agregacji, kompozycji, asocjacje, itp.

UWAGA 2: na diagramie klas uwzględniamy jedynie klasy wchodzące w skład warstwy danych projektowanego systemu; nie jest wymagany diagram klas realizujących np. warstwę interfejsu użytkownika.

Na dzisiejszych zajęciach

Opracować diagram klas. Zadbać o zgodność z poprzednimi etapami projektu.

Ostatnie 15 minut zajęć poświęcimy na pochwalenie się opracowanymi materiałami.

W razie pytań czy wątpliwości jestem dostępny pod moim adresem mailowym
m.krok@po.edu.pl

(przypominam, że regularne wysyłanie postępów jest jednym z warunków zaliczenia)