

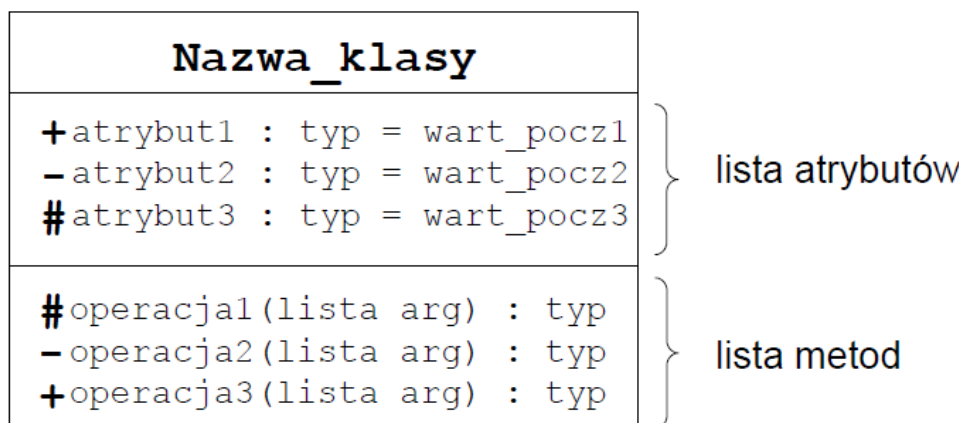
**Inżynieria oprogramowania**  
**Część 5: UML – Diagramy klas**

1. Diagram klas .....	3
Zadanie 1 .....	3
Zadanie 2 .....	4
2. Agregacja .....	4
3. Kompozycja.....	4
4. Generalizacja .....	5
Zadanie 3 .....	5
5. Asocjacje –liczność, nazwy ról .....	6
Zadanie 4 .....	7
Zadanie 5 .....	7

## 1. DIAGRAM KLAS

Diagram klas jest ściśle powiązany z projektowaniem obiektowym systemu informatycznego lub wręcz bezpośrednio z jego implementacją w określonym języku programowania. Elementami tego diagramu są klasy, reprezentowane przez prostokąty, które mogą zawierać informację o polach i metodach klasy.

Klasa obrazowana jest za pomocą podzielonego na części prostokąta –każda część reprezentuje inwarianty o zbliżonym przeznaczeniu.



Rysunek 1: Klasa

Na diagramie klasy dodatkowo określa się widoczność atrybutów i metod przez umieszczanie przed nimi odpowiedniego symbolu:

- + publiczne (public)
- - prywatne (private)
- # chronione (protected)

Dobrze zbudowany diagram klas:

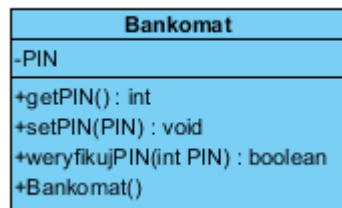
- uwypukla jeden statyczny aspekt perspektywy projektowej systemu
- obejmuje tylko byty niezbędne do zrozumienia tego aspektu
- zawiera szczegóły odpowiednie do przyjętego poziomu abstrakcji, z dodatkami koniecznymi do zrozumienia tego, na czym zależy projektantowi
- nie jest zbyt ogólny

### ZADANIE 1

Utworzyć diagram klas i zapoznać się z podstawowymi własnościami diagramu klas.

## ZADANIE 2

W diagramie klas utwórz klasę przedstawioną poniżej



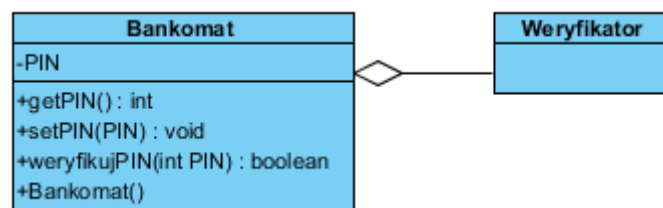
## 2. AGREGACJA

Agregacja najprościej mówiąc oznacza zawieranie.

Na przykład:

- lampka zawiera żarówkę,
- komputer zawiera procesor,
- jabłko zawiera robaka, itd.

Agregację na diagramie UML oznacza się pustym rombem.



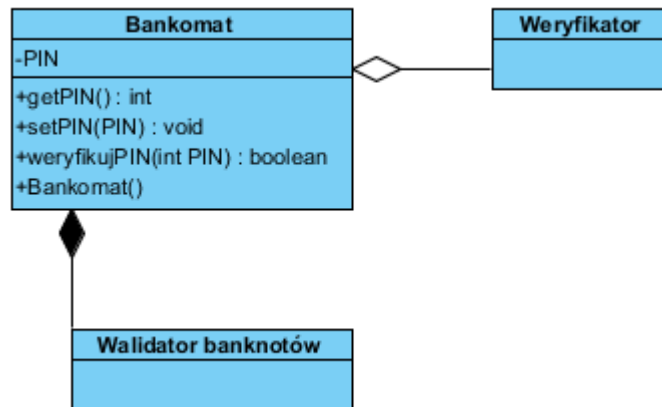
Rysunek 2: Agregacja

## 3. KOMPOZYCJA

Szczególnym przypadkiem agregacji jest kompozycja. Od agregacji różni się tym, że klasa posiada obiekty (składa się z obiektów), które bez tej klasy istnieć by nie mogły. Przykłady takich związków to:

- blok zawiera mieszkania (mieszkania poza blokiem nie istnieją),
- komputer zawiera procesor,
- łazienka zawiera wannę,

Kompozycja na diagramie UML oznaczona jest wypełnionym rombem.



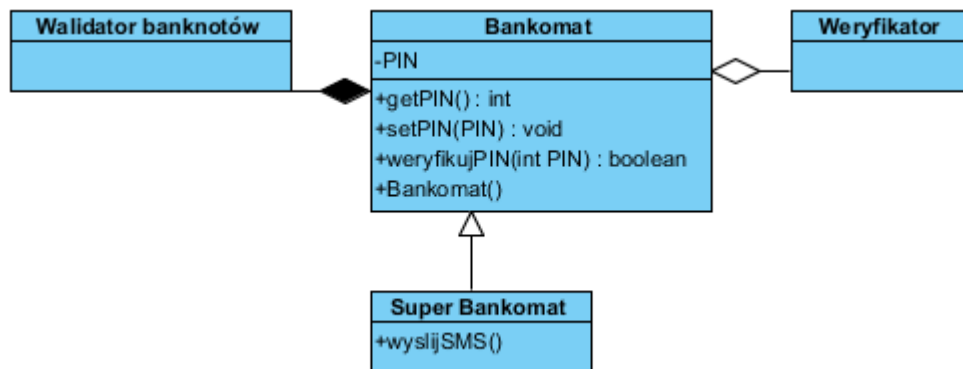
Rysunek 3: Kompozycja

#### 4. GENERALIZACJA

Generalizacja odpowiada dziedziczeniu znanemu z języków programowania. Inaczej mówiąc, jest to związek pomiędzy bardziej ogólną klasą (rodzicem) a klasą bardziej szczegółową (rodzicem).

Oto przykłady generalizacji:

- manager jest pracownikiem,
- sklep jest firmą,
- latarnia jest lampą, podobnie lampą jest lampka nocna,
- kot jest zwierzęciem,



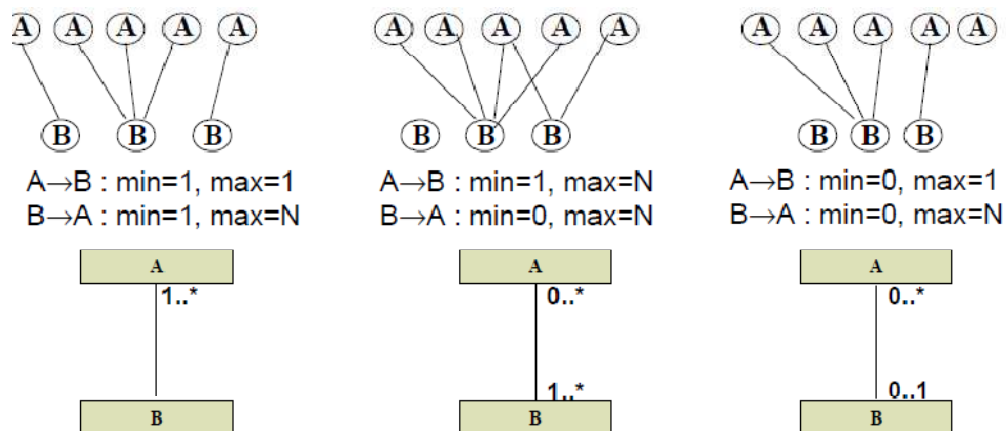
Rysunek 4: Generalizacja

#### ZADANIE 3

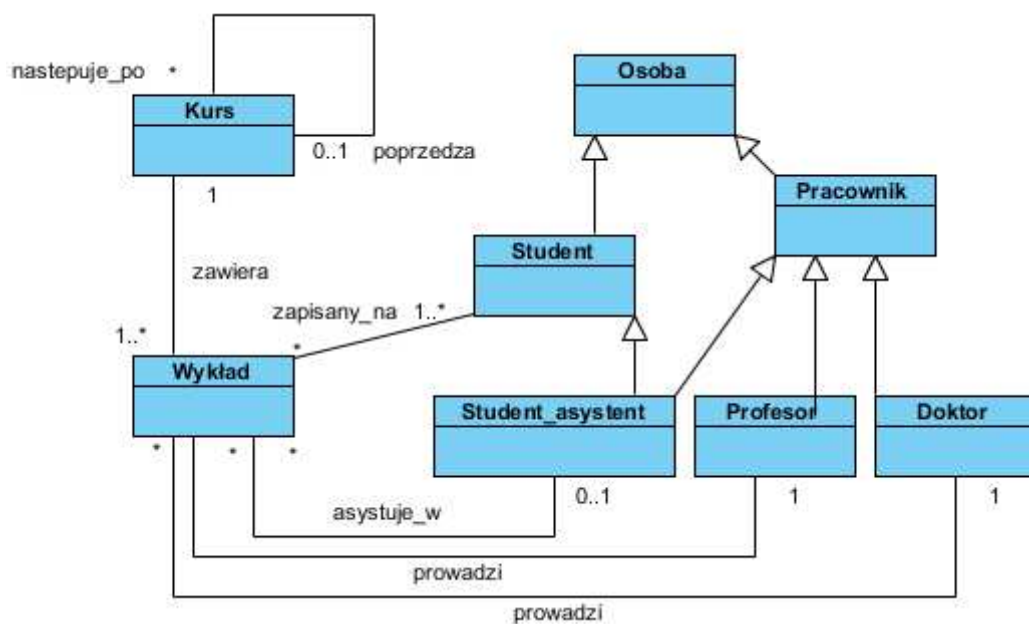
Narysować diagram klas przedstawiony na rysunku 4.

## 5. ASOCJACJE – LICZNOŚĆ, NAZWY RÓL

Asocjacje mogą być wyposażone w oznaczenia liczności

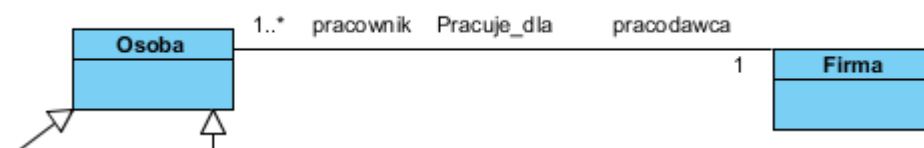


Liczność oznacza, ile obiektów innej klasy może być powiązane z jednym obiektem danej klasy (para liczb oznaczająca ilość minimalną i maksymalną).



Rysunek 5: Asocjacje – liczność

Asocjacje mogą być także wyposażone w dodatkowe nazwy ról (przy odpowiednich końcach).



Rysunek 6: Asocjacje – role

## ZADANIE 4

Narysować diagram klas przedstawiony na rysunku 5.

## ZADANIE 5

Narysować diagram projektowy programu pocztowego przedstawiony na rysunku poniżej.

