

Programowanie obiektowe

Wykład 11

Marcin Młotkowski

4 maja 2017

Plan wykładu

- 1 Software crisis
- 2 Analiza obiektowa
- 3 Unified Modelling Language
 - Diagramy klas
 - Przykład
- 4 Przykład

Statystyki

Statystyka skuteczności wydawania pieniędzy federalnych USA w 1979 roku:

- 2% budżetu: uzyskano dokładnie taki efekt jak planowano;
- 3% oprogramowanie, które było przydatne po niewielkich modyfikacjach.

Polski wkład do statystyk

- Komputeryzacja urzędów skarbowych (połowa lat 90):
100 mln \$;
- Komputeryzacja ZUS'u: 714 mln zł.

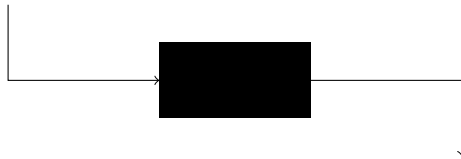
Przyczyny kłopotów z tworzeniem oprogramowania

- Rozbudowa oprogramowania: zwiększanie zarówno dziedziny jak i funkcjonalności;
- błędne założenia o czasie użytkowania;
- zmienność potrzeb, w tym zmienność prawa;
- problemy zapanowania nad dużym projektem.

Metody projektowania: prosta dziedzina

Przykłady: numeryczna analiza danych, przetwarzanie danych finansowych.

Dane wejściowe



Wyniki

Metody projektowania

Dekompozycja funkcjonalna

podział modułów ze względu na funkcje

Graf przepływu danych

Ustalenie kolejności wykonywania operacji na danych

Rozbudowana dziedzina

- Modelowanie danych;
- modelowanie związków między danymi.

Analiza problemu

- Proces poznawania dziedziny;
- zrozumienie szczegółów;
- ustalenie granic dziedziny.

Cechy projektu

- Abstrakcja danych i abstrakcja operacji
- Enkapsulacja: ukrycie detali implementacyjnej
- Modularność: podział programu na wyraźne fragmenty
- Hierarchia pojęć

Waga dobrego projektu

- Dziedzina zmienia się wolno
- Funkcjonalność może się szybko zmieniać

Plan wykładu

- 1 Software crisis
- 2 Analiza obiektowa**
- 3 Unified Modelling Language
 - Diagramy klas
 - Przykład
- 4 Przykład

Podział projektu na warstwy

- 1 Warstwa klas i obiektów
- 2 Warstwa związków
- 3 Warstwa tematów
- 4 Warstwa atrybutów
- 5 Warstwa usług

Klasy i obiekty

- Osoby, przedmioty, pojęcia
- Zdarzenia
- Role
- Jednostki organizacyjne

Warstwa związków

- generalizacja
- specjalizacja
- agregacja
- kompozycja
- asocjacja

Warstwa tematów

Grupowanie klas w tematyczne klastry

Warstwa tematów

Grupowanie klas w tematyczne klastry

Wielkość tematu: wskazówka

7 ± 2

Warstwa atrybutów

Atrybuty proste

Np imię, nazwisko

Warstwa atrybutów

Atrybuty proste

Np imię, nazwisko

Atrybuty złożone

Np. adres

Warstwa atrybutów

Atrybuty proste

Np imię, nazwisko

Atrybuty złożone

Np. adres

Atrybuty obliczane

Np. wiek,

Warstwa atrybutów

Atrybuty proste

Np imię, nazwisko

Atrybuty złożone

Np. adres

Atrybuty obliczane

Np. wiek,

Unikatowe identyfikatory

Własny 'numer seryjny' obiektu.

Warstwa atrybutów

Atrybuty proste

Np imię, nazwisko

Atrybuty złożone

Np. adres

Atrybuty obliczane

Np. wiek,

Unikatowe identyfikatory

Własny 'numer seryjny' obiektu.

Referencje do innych obiektów

Warstwa usług

- Inicjowanie
- Dostęp do atrybutów
- Zmiana stanu

Plan wykładu

- 1 Software crisis
- 2 Analiza obiektowa
- 3 Unified Modelling Language**
 - Diagramy klas
 - Przykład
- 4 Przykład

Unified Modelling Language

Notacja graficzna, umożliwiająca ilustrację zagadnień związanych z oprogramowaniem.

Do czego służą diagramy

Prezentacja koncepcji

Do czego służą diagramy

Prezentacja koncepcji

Prezentacja specyfikacji.

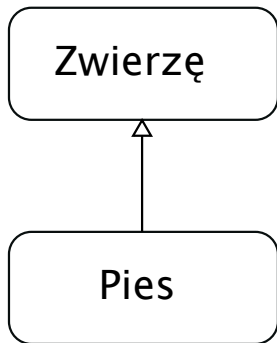
Do czego służą diagramy

Prezentacja koncepcji

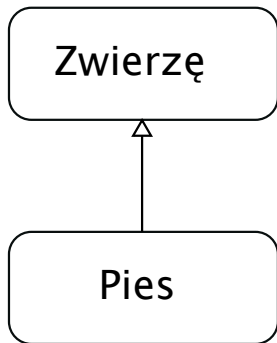
Prezentacja specyfikacji.

Prezentacja implementacji.

Rola UML



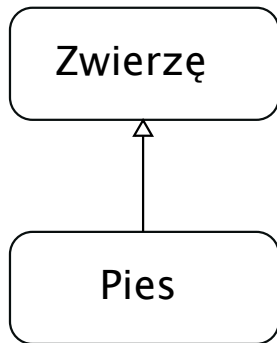
Rola UML



Poziom koncepcyjny

Pies jest zwierzęciem

Rola UML



Poziom koncepcyjny

Pies jest zwierzęciem

Poziom specyfikacji/implementacji

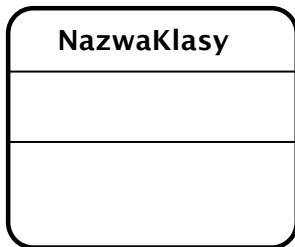
```
public class Zwierzę {}
```

```
public class Pies : Zwierzę {}
```

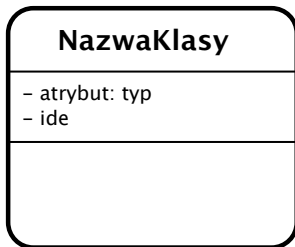
Opis klasy

Opis klasy

- nazwa klasy;



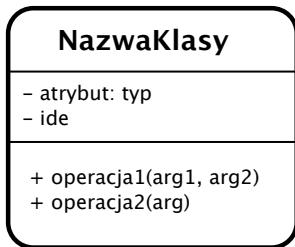
Opis klasy



Opis klasy

- nazwa klasy;
- atrybuty klasy, opcjonalnie z typem atrybutu i widzialnością (+: publiczny, #: zabezpieczony, -: prywatny);

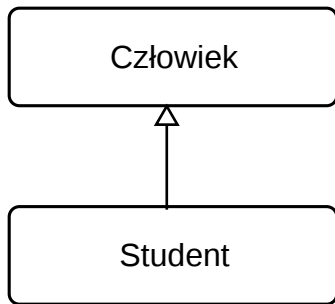
Opis klasy



Opis klasy

- nazwa klasy;
- atrybuty klasy, opcjonalnie z typem atrybutu i widzialnością (+: publiczny, #: zabezpieczony, -: prywatny);
- operacje na klasie, opcjonalnie z typami i widzialnością.

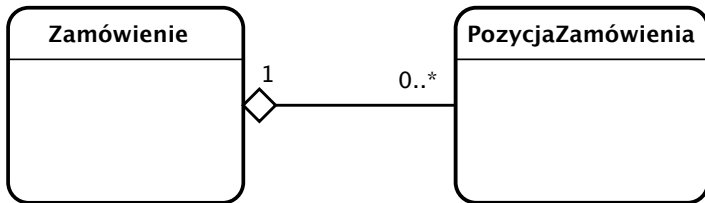
Generalizacja/Specjalizacja



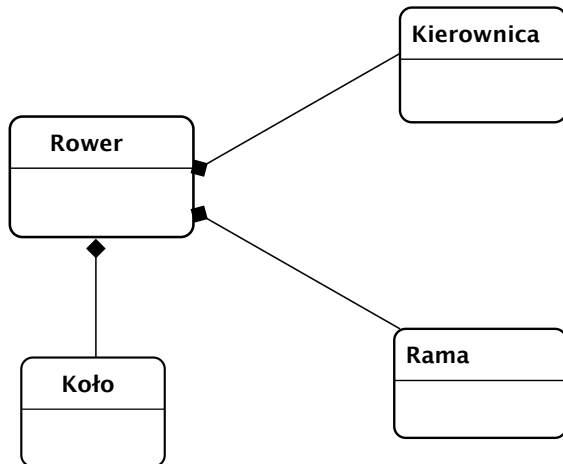
Generalizacja

Specjalizacja

Agregacja



Kompozycja



Opis asocjacji



Opis klasy

- kierunek asocjacji;

Opis asocjacji

→
zmienna

Opis klasy

- kierunek asocjacji;
- nazwa zmiennej z referencją;

Opis asocjacji



Opis klasy

- kierunek asocjacji;
- nazwa zmiennej z referencją;
- "liczność asocjacji".

Drzewo binarnych poszukiwań

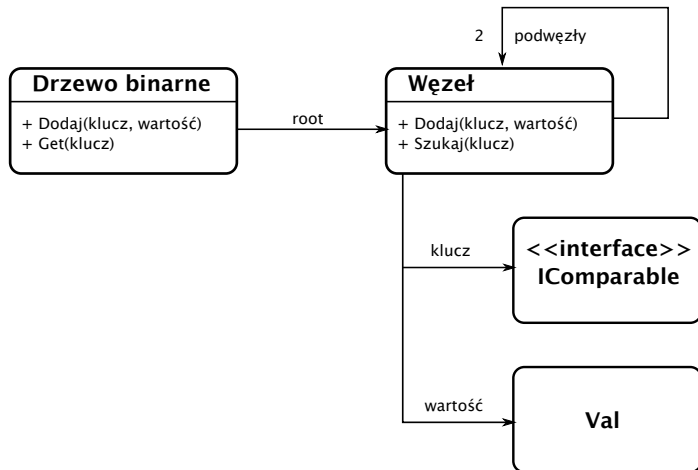
```
public class DrzewoBinarne<Key, Val> {  
    private Węzeł<Key, Val> root = null;  
    public void Dodaj(Comparable<Key> klucz,  
                     Val wartość) { ... }  
}
```

Drzewo binarnych poszukiwań

```
public class DrzewoBinarne<Key, Val> {  
    private Węzeł<Key, Val> root = null;  
    public void Dodaj(IComparable<Key> klucz,  
                     Val wartość) { ... }  
}
```

```
class Węzeł<Key, Val> {  
    Val wartość;  
    IComparable<Key> klucz;  
  
    Węzeł<Key, Val>[] podwężły = new Węzeł<Key, Val>[2];  
}
```

Diagram klas



Plan wykładu

- 1 Software crisis
- 2 Analiza obiektowa
- 3 Unified Modelling Language
 - Diagramy klas
 - Przykład
- 4 Przykład

Biblioteka osiedlowa

- Książki i czasopisma
- Czytelnicy
- Karty biblioteczne
- Tymczasowi czytelnicy