

Metodyki zwinne wytwarzania oprogramowania

Wykład 9

Marcin Młotkowski

6 grudnia 2016

Plan wykładu

- 1 Diagramy
 - Wprowadzenie
 - Przykład
- 2 Typy diagramów
- 3 Jak stosować diagramy

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.

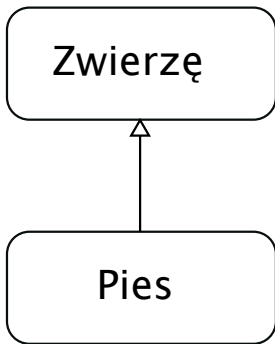
Do czego służą diagramy

Prezentacja koncepcji

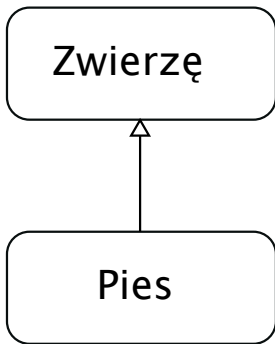
Prezentacja specyfikacji.

Prezentacja implementacji.

Rola UML



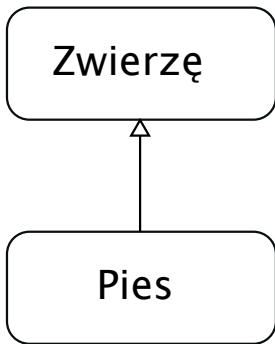
Rola UML



Poziom koncepcyjny

Pies jest zwierzęciem

Rola UML



Poziom koncepcyjny

Pies jest zwierzęciem

Poziom specyfikacji/implementacji

```
public class Zwierzę {}
```

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

Rodzaje diagramów

Diagramy statyczne

Struktura elementów oprogramowania: klasy, obiekty, struktury danych wraz z łączącymi je relacjami.

Rodzaje diagramów

Diagramy statyczne

Struktura elementów oprogramowania: klasy, obiekty, struktury danych wraz z łączącymi je relacjami.

Diagramy dynamiczne

Opisują jak się zmieniają w czasie elementy oprogramowania (zmiana stanu czy przepływ wykonywania)

Rodzaje diagramów

Diagramy statyczne

Struktura elementów oprogramowania: klasy, obiekty, struktury danych wraz z łączącymi je relacjami.

Diagramy dynamiczne

Opisują jak się zmieniają w czasie elementy oprogramowania (zmiana stanu czy przepływ wykonywania)

Diagramy fizyczne

opisują strukturę fizyczną elementów oprogramowania, plików z kodem źródłowym, bibliotek, plików binarnych etc.

Drzewo binarnych poszukiwań

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

Drzewo binarnych poszukiwań

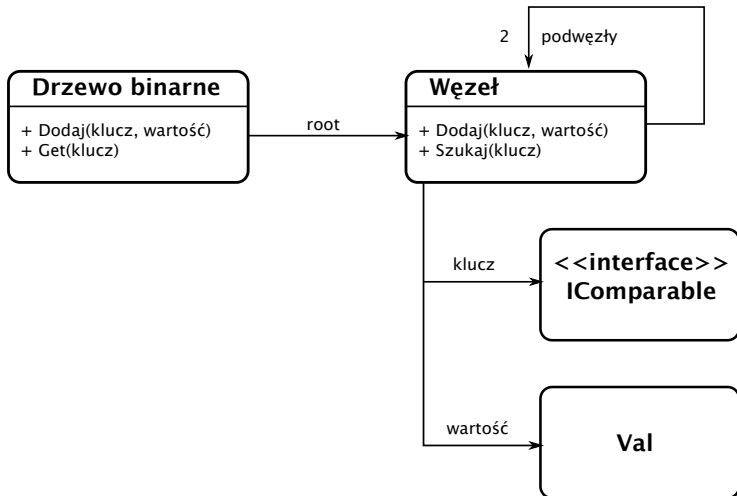
```
public class DrzewoBinarne<Key, Val> {  
    private Węzeł<Key, Val> root = null;  
    public void Dodaj(IComparable<Key> klucz,  
                     Val wartość) { ... }  
}
```

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

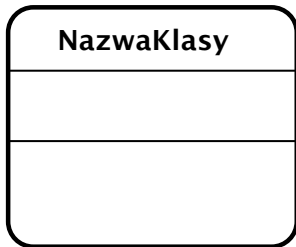
Plan wykładu

- 1 Diagramy
 - Wprowadzenie
 - Przykład
- 2 Typy diagramów
- 3 Jak stosować diagramy

Diagram klas



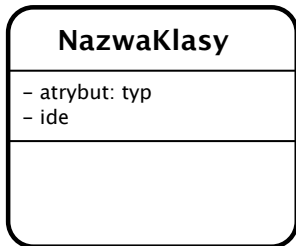
Opis klasy



Opis klasy

- nazwa klasy;

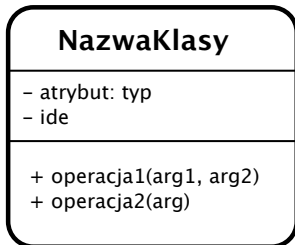
Opis klasy



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.

Opis asocjacji



Opis klasy

- kierunek asocjacji;

Opis asocjacji



Opis klasy

- kierunek asocjacji;
- nazwa zmiennej z referencją;

Opis asocjacji

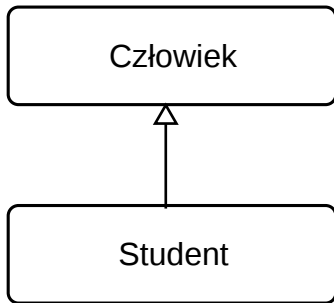


Opis klasy

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

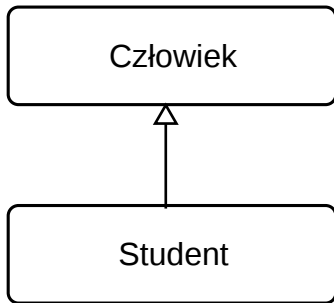
Dziedziczenie i implementacja

Dziedziczenie



Dziedziczenie i implementacja

Dziedziczenie



Implementacja interfejsu

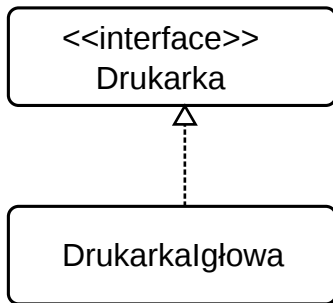
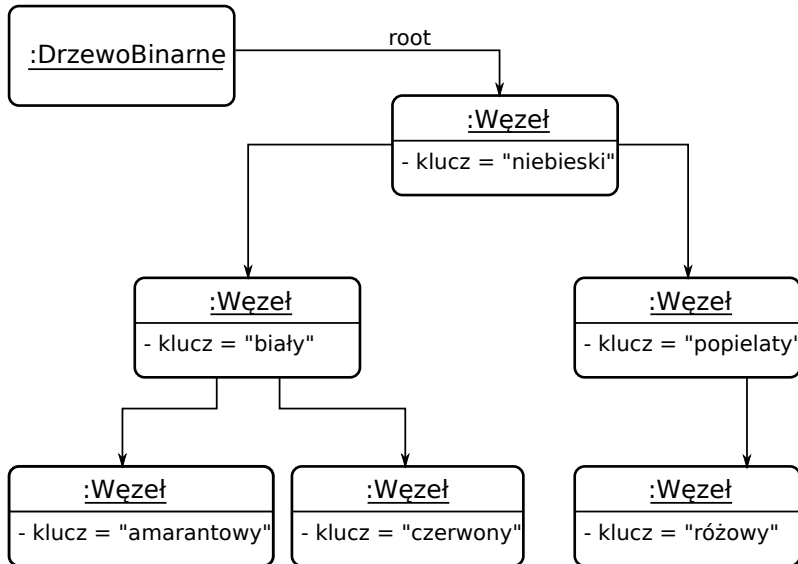


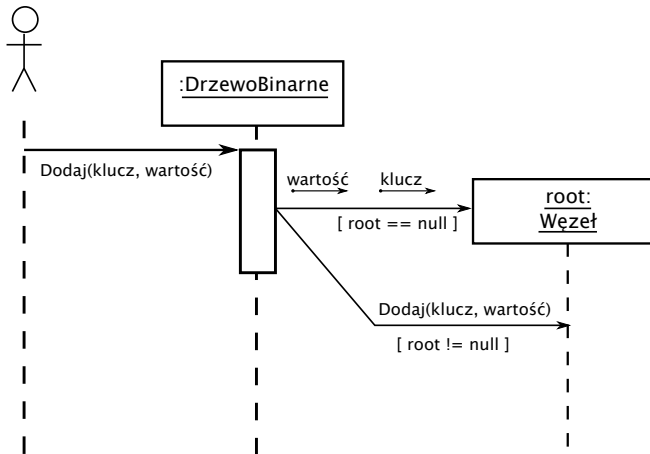
Diagram obiektów



Opis

- nazwy klas są podkreślone;
- po dwukropku : jest klasa obiektu.

Diagram sekwencji



Elementy diagramu sekwencji

Aktor : zewnętrzny "był" inicjujący operację;

[strażnik] : warunek logiczny;

aktywacja (pionowy prostokąt);

tokeny danych

Diagram sekwencji

- diagram opisuje porządek przekazywania komunikatów;
- duże strzałki to łącza;
- małe strzałki to komunikaty.

Diagram współpracy

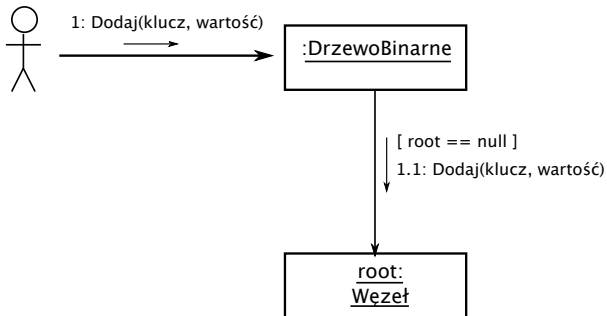
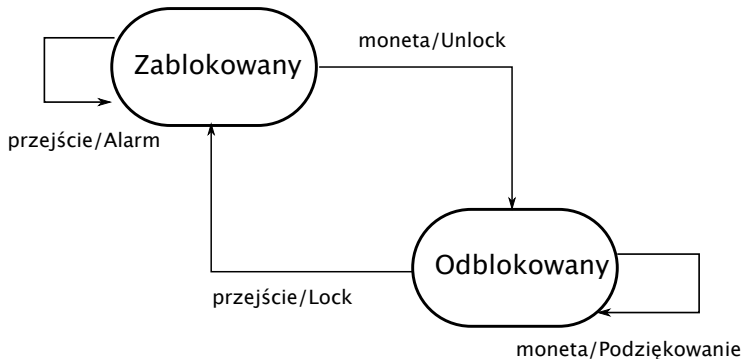


Diagram stanów

Przykład

Płatna bramka (w postaci kołowrotka), wpuszczająca na monety.

Diagram stanów



Plan wykładu

- 1 Diagramy
 - Wprowadzenie
 - Przykład
- 2 Typy diagramów
- 3 Jak stosować diagramy

Zasady stosowania diagramów

Warto stosować do projektowania **przed** zaprogramowaniem klas.

Nie są niezbędne

***Działające oprogramowanie** nad wyczerpującą dokumentacją.*

uwagi

Komunikacja

Dyskusja z innymi programistami nad koncepcjami

Kłopotliwe w wyrażaniu algorytmów.

Co się przydaje

Mapy drogowe

Diagramy klas przedstawiające zależności między klasami:
dziedziczenie, asocjacje etc.

Kolejność

Najpierw diagram współpracy.

Kolejność

Najpierw diagram współpracy.

Potem diagram klas.

Kolejność

Najpierw diagram współpracy.

Potem diagram klas.

Przykład

Oprogramowanie do telefonu komórkowego.

Diagram współpracy

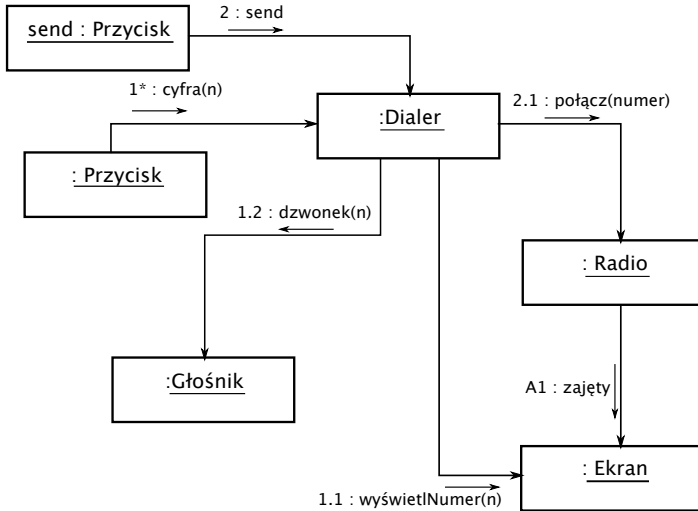


Diagram klas

