Wrocław 2017
Cezary Hołub

Wrocławska Wyższa Szkoła Informatyki Stosowanej

Przedmiot	Programowanie Java
Semestr	Zima 2017/2018

Materiały do ćwiczeń 4

ZADANIE 1	Modelowanie hierarchii dziedziczenia

Cel ćwiczenia:

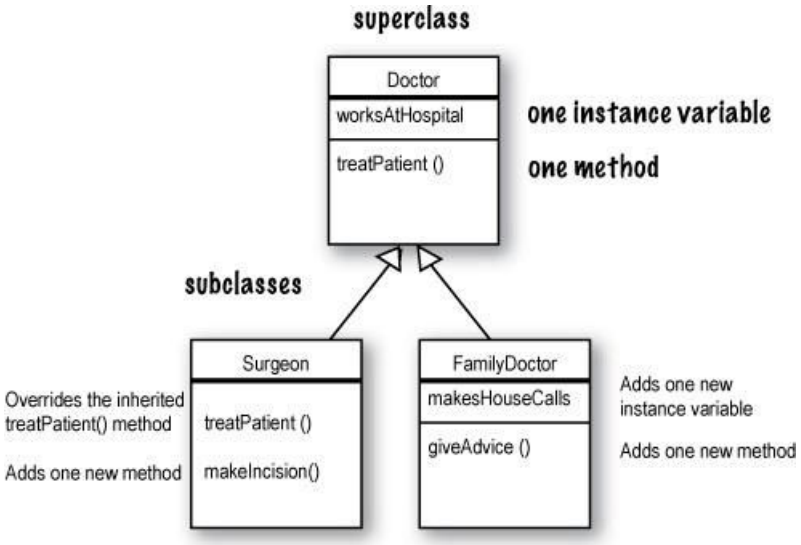
- Utrwalenie praktycznych umiejętności projektowania hierarchii dziedziczenia klas.

Wymagane wiadomości wstępne:

- Teoretyczne podstawy i zasady ‘dziedziczenia’ w projektowaniu obiektowym.

Przebieg ćwiczenia:

Zaprojektować hierarchie dziedziczenia typow dla (dowolnie) wybranej przez siebie dziedziny życia (analogicznie do poniższego przykładu, modelującego fragment przychodni lekarskiej - omawianego w trakcie wykładu/ćwiczeń).



Wymagania:

a. Co najmniej 3 poziomy w hierarchii pionowej (dziedziczące po sobie) .
b. Co najmniej 2 podklasy dla każdej klasy, która podklasy posiada .
c. Co najmniej 1 atrybut i jedna metoda dla każdej klasy .
d. Co najmniej 2 przypadki przesłaniania metody z klasy nadrzędnej .
e. Co najmniej 1 klasa abstrakcyjna [oznaczyć jako ‘abstract’] .

Forma:

Całkowicie dowolna - może być plik z dokumentem programu MS WORD, plik z dowolnego innego programu (StarUML), skan/zdjęcie kartki papieru, na której diagram będzie narysowany odręcznie, lub dowolny inny format

zaproponowany przez Państwa (oby tylko był to poprawny diagram, czytelny i zrozumiały).

ZADANIE 2	Dziedziczenie w języku Java

Cel ćwiczenia:

- Zdobycie umiejętności zapisu hierarchii dziedziczenia między klasami w języku Java.

Wymagane wiadomości wstępne:

- Teoretyczne podstawy ‘dziedziczenia’ w projektowaniu obiektowym.

Przebieg ćwiczenia:

Zaprojektowany przez siebie diagram dziedziczenia i hierarchii typów z ćwiczenia 1 zapisać w języku Java w formie klas. Oprócz relacji dziedziczenia pomiędzy klasami należy również w klasach umieścić treści zaproponowanych atrybutów i metod.

Wymagania:

f. Kod musi być konceptualnie zgodny z własnym projektem z zadania 1
g. Kod musi być poprawny zgodnie z zasadami dziedziczenia w Javie
h. Kod musi się kompilować

Forma:

Całkowicie dowolna - może to być spakowany do archiwum ZIP projekt z Eclipse’a, mogą być pojedyncze pliki z zawartością poszczególnych klas, w formie plików tekstowych.

ZADANIE 3	Dziedziczenie i przesłanianie metod

Cel ćwiczenia:

- Zrozumienie mechanizmu przesłaniania metod przy dziedziczeniu w Javie.

Wymagane wiadomości wstępne:

- Notacja dziedziczenia w języku Java.

Przebieg ćwiczenia:

1. Stwórz następujące klasy w projekcie w Eclipse:

```
package pl.wsis.java;

public class Osoba {

   public String imie;

   public Osoba(String imie) {
        this.imie = imie;
   }

   public void przedstawSie() {
        System.out.println("Witaj, jestem osoba, mam na imie " + this.imie);
   }

}
```

```
package pl.wsis.java;

public class Kobieta extends Osoba {

   public Kobieta(String imie) {
      super(imie);
   }

   // [UZUPELNIJ KOD]
}
```

2. Dodaj następującą klasę narzedziowa:

```
package pl.wsis.java;

public class Main {

   public static void main(String [] args) {
      Osoba kobieta = new Kobieta("Agnieszka");
      kobieta.przedstawSie();
   }

}
```

3. Uruchom program. Na ekranie powinno się wyświetlić:

```
Witaj, jestem osoba, mam na imie Agnieszka
```

4. Zmodyfikuj klasę Kobieta (wstawiając dowolny fragment kodu w miejsce oznaczone przez [UZUPEŁNIJ KOD]) tak aby po ponownym uruchomieniu programu zobaczyć:

```
Witaj, jestem kobieta, mam na imie Agnieszka
```

Rozwiązaniem zadania jest fragment kodu wstawionego w [UZUPELNIJ KOD]

ZADANIE 4	Dziedziczenie i rzutowanie typów

Cel ćwiczenia:

- Zrozumienie mechanizmu rzutowania typów przy dziedziczeniu w Javie.

Wymagane wiadomości wstępne:

- Notacja dziedziczenia w języku Java.

Przebieg ćwiczenia:

1. Stwórz następujące klasy w projekcie w Eclipse:

```
package pl.wsis.java;

public class Osoba {

    public String imie;

    public Osoba(String imie) {
        this.imie = imie;
    }

    public void przedstawSie() {
        System.out.println("Witaj, jestem osoba, mam na imie " + this.imie);
    }

}
```

```
package pl.wsis.java;

public class Kobieta extends Osoba {

    public Kobieta(String imie) {
        super(imie);
    }
}
```

2. Dodaj następującą klasę narzędziową:

```
package pl.wsis.java;

public class Main {

   public static void main(String [] args) {
      Osoba osoba = new Kobieta(“Anna”);
      Kobieta kobieta = osoba;
      kobieta.przedstawSie();
   }

}
```

3. Spróbuj uruchomić program. Pojawi się informacja o błędzie kompilacji:

```
Type mismatch: cannot convert from Osoba to Kobieta
```

4. Zmodyfikuj klasę Main (modyfikując w dowolny sposób linię oznaczona kolorem czerwonym) tak aby po ponownym uruchomieniu programu zobaczyć:

```
Witaj, jestem osoba, mam na imie Anna
```

Rozwiązaniem zadania jest fragment kodu wstawionego zamiast czerwonej linii.

ZADANIE 5	Dziedziczenie i klasy abstrakcyjne

Cel ćwiczenia:

- Zrozumienie mechanizmu klas abstrakcyjnych przy dziedziczeniu w Javie.

Wymagane wiadomości wstępne:

- Notacja dziedziczenia w języku Java.

Przebieg ćwiczenia:

1. Utwórz następującą klasę w projekcie w Eclipse:

```
package pl.wsis.java;

public abstract class Osoba {

    public void przedstawSie() {
         System.out.println("Witaj, jestem osoba");
    }

}
```

2. Dodaj następującą klasę narzedziową:

```
package pl.wsis.java;

public class Main {

   public static void main(String [] args) {
      Osoba osoba = new Osoba();
      osoba.przedstawSie();
   }

}
```

3. Spróbuj uruchomić program. Pojawi się informacja o błędzie kompilacji:

```
Cannot instantiate the type Osoba
```

4. Zmodyfikuj klasę Osoba (modyfikując w dowolny sposób całą klasę) tak aby po ponownym uruchomieniu programu zobaczyć:

```
Witaj, jestem osoba
```

Rozwiązaniem zadania jest kod całej klasy Osoba.