

1. Prva laboratorijska vježba

1.1. KLASSE I OBJEKTI U JAVI

Svrha laboratorijske vježbe je usvajanje osnovnih principa objektno orijentiranog programiranja, kao što su kreiranje vlastitih klasa te kreiranje objekata tih klasa. Osim toga, ističu se i principi postavljanja i dohvaćanja vrijednosti polja (varijabli) unutar klasa, te korištenje različitih modifikatora za ograničavanje pristupa tim poljima.

1.2. ZADATAK ZA PRIPREMU

Napisati Java program koji će se sastojati od nekoliko klasa pohranjenih u različitim paketima. Klase simuliraju strukturu za implementaciju meteorološkog sustava.

1. Preuzeti najnoviju verziju Eclipse Oxygen razvojnog okruženja. To je potrebno napraviti iz razloga što starije inačice nisu u potpunosti kompatibilne s Javom 9.
2. Unutar Eclipse razvojnog okruženja potrebno je kreirati Java projekt koji se naziva kao Vaše prezime i sadrži redni broj vježbe (npr. "Horvat-1", ako se prezivate „Horvat“). Kreiranje Java projekta opisano je u online tutorialu [1] u poglavlju 8.1.
3. Projekt pretvoriti u „Maven Project“ korištenjem opcije „Configure->Convert to Maven Project“ i na dijalogu koji se prikaže nakon toga potvrditi ponuđene parametre.
4. Kreirati paket **hr.java.vjezbe.entitet**. Kreiranje paketa unutar Eclipse razvojnog okruženja opisano je unutar poglavlja 8.2. Eclipse online tutoriala [1].
5. Kreirati klasu **GeografskaTocka** unutar paketa **hr.java.vjezbe.entitet** koja se sastoji od dvije varijable: „x“ i „y“. Obje varijable moraju biti tipa „BigDecimal“. Napisati konstruktor klase **GeografskaTocka** tako da prima obje varijable i pohranjuje ih u privatne varijable. Ograničiti pristup varijablama na način da ih je moguće dohvaćati samo korištenjem javnih "get" metoda i spremanje samo korištenjem javnih „set“ metoda. Kreiranje klasa unutar Eclipse razvojnog okruženja opisano je unutar poglavlja 8.3. Eclipse online tutoriala [1], automatsko generiranje konstruktora te "get" i "set" metoda unutar razvojnog okruženja Eclipse prikazano je unutar poglavlja 15.
6. Kreirati klasu **Drzava** unutar paketa **hr.java.vjezbe.entitet** koja se sastoji od dvije varijable: „naziv“ i „povrsina“. Varijabla „naziv“ mora biti tipa „String“, a „povrsina“ tipa „BigDecimal“. Napisati konstruktor klase **Drzava** tako da prima obje varijable i pohranjuje ih u privatne varijable. Ograničiti pristup varijablama na način da ih je moguće dohvaćati samo korištenjem javnih "get" metoda i spremanje samo korištenjem javnih „set“ metoda.
7. Kreirati klasu **Zupanija** unutar paketa **hr.java.vjezbe.entitet** koja se sastoji od dvije varijable: „naziv“ i „drzava“. Varijabla „naziv“ mora biti tipa „String“, a „drzava“ tipa „Drzava“. Napisati konstruktor klase **Zupanija** tako da prima obje varijable i pohranjuje ih u privatne varijable. Ograničiti pristup varijablama na način da ih je moguće dohvaćati samo korištenjem javnih "get" metoda i spremanje samo korištenjem javnih „set“ metoda.

8. Kreirati klasu **Mjesto** unutar paketa **hr.java.vjezbe.entitet** koja se sastoji od dvije varijable: „naziv“ i „zupanija“. Varijabla „naziv“ mora biti tipa „String“, a „zupanija“ tipa „Zupanija“. Napisati konstruktor klase **Mjesto** tako da prima obje varijable i pohranjuje ih u privatne varijable. Ograničiti pristup varijablama na način da ih je moguće dohvaćati samo korištenjem javnih "get" metoda i spremanje samo korištenjem javnih „set“ metoda.
9. Kreirati klasu **MjernaPostaja** unutar paketa **hr.java.vjezbe.entitet** koja se sastoji od tri varijable: „naziv“, „mjesto“ i „geografskaTocka“. Varijabla „naziv“ mora biti tipa „String“, „mjesto“ tipa „Mjesto“ i „geografskaTocka“ tipa „GeografskaTocka“. Napisati konstruktor klase **MjernaPostaja** tako da prima sve tri varijable i pohranjuje ih u privatne varijable. Ograničiti pristup varijablama na način da ih je moguće dohvaćati samo korištenjem javnih "get" metoda i spremanje samo korištenjem javnih „set“ metoda.
10. Kreirati paket **hr.java.vjezbe.glavna**.
11. Unutar paketa **hr.java.vjezbe.glavna** kreirati klasu **Glavna** koja će se sastojati od **main** metode. Unutar nje potrebno je napisati programski kod za unošenje podataka za po tri objekta klase **MjernaPostaja**. Objekte je potrebno kreirati tek kad se unesu svi podaci za svaki od objekata. Ni jedan objekt ne smije sadržavati *NULL* vrijednosti. Na primjer, kada je potrebno kreirati mjernu postaju, zbog referenci i kako bi programski kod bio ispravan, potrebno je unijeti geografsku točku, mjesto, županiju i državu. Za svaki objekt potrebno je od korisnika zatražiti unos svih parametara pojedine komponente i korištenjem konstruktora kreirati pripadajuće konstruktore. Unos podataka za pojedine objekte potrebno je izdvojiti u zasebne metode koje će se pozivati iz **main** metode. **Nije dozvoljeno sve imati unutar main metode.** Svaka od tih metoda mora vraćati objekt kojeg popunjava, a primiti objekt klase **Scanner**. Dozvoljeno je koristiti samo jedan objekt klase **Scanner**.
12. Unutar metode **main** klase **Glavna** potrebno je ispisati sve informacije o kreiranim objektima. Kod ispisa je informacije potrebno dohvaćati korištenjem „get“ metoda iz objekta klase **MjernaPostaja**.

Primjer izvođenja programa:

```
Unesite prvu mjernu postaju:
Unesite naziv mjerne postaje:
Maximir
Unesite naziv mjesta:
Zagreb
Unesite naziv županije:
Grad Zagreb
Unesite naziv države:
Hrvatska
Unesite površinu države:
56594
Unesite Geo koordinatu X:
```

45,8288

Unesite Geo koordinatu Y:

16,0211

Unesite drugu mjernu postaju:

Unesite naziv mjerne postaje:

Zemunik

Unesite naziv mjesta:

Zadar

Unesite naziv županije:

Zadarska županija

Unesite naziv države:

Hrvatska

Unesite površinu države:

56594

Unesite Geo koordinatu X:

44,1058

Unesite Geo koordinatu Y:

15,3793

Unesite treću mjernu postaju:

Unesite naziv mjerne postaje:

Zavižan

Unesite naziv mjesta:

Zavižan

Unesite naziv županije:

Ličko-senjska županija

Unesite naziv države:

Hrvatska

Unesite površinu države:

56594

Unesite Geo koordinatu X:

44,49

Unesite Geo koordinatu Y:

14,59

Naziv mjerne postaje: Maximir

Postaja se nalazi u mjestu Zagreb, županiji Grad Zagreb, državi Hrvatska

Točne koordinate postaje su x:45.8288 y:16.0211

Naziv mjerne postaje: Zemunik

Postaja se nalazi u mjestu Zadar, županiji Zadarska županija, državi Hrvatska

Točne koordinate postaje su x:44.1058 y:15.3793

Naziv mjerne postaje: Zavižan

Postaja se nalazi u mjestu Zavižan, županiji Ličko-senjska županija, državi Hrvatska

Točne koordinate postaje su x:44.49 y:14.59

NAPOMENE:

1. Osim implementacija vježbe prema uputama, dozvoljeno je uvoditi i promjene ako su opravdane i ne narušavaju koncepte objektno-orijentiranog programiranja.
2. Prilikom unosa podataka tipa **BigDecimal** koristiti decimalni zarez, a ne točku.
3. Nakon naredbi za unos numeričkih podataka (BigDecimal ili Integer) izvršiti metodu „nextLine“ nad objektom klase Scanner koji se koristi. Proučiti odgovore na sljedećem linku: <http://stackoverflow.com/questions/13102045/skipping-nextline-after-using-next-nextint-or-other-nextfoo-methods>.
4. U slučaju potrebe isključivanja nekih prečaca u Eclipseu (kao što je kombinacija tipki „Alt Gr + B“ ili „Alt Gr + F“ koje služe za ispisivanje zagrada) koristiti upute sa sljedećem linku: <http://stackoverflow.com/questions/12477738/how-to-disable-ctrl-alt-m-in-eclipse>.
5. Za pretvorbu objekta tipa „String“ i objekt tipa „LocalDateTime“ koristiti upute na sljedećem linku: <http://stackoverflow.com/questions/22463062/how-to-parse-format-dates-with-localdatetime-java-8>.

LITERATURA

1. Vogella, GmbH: Eclipse Oxygen Tutorial:
<http://www.vogella.com/tutorials/Eclipse/article.html>