

1. Prva laboratorijska vježba

1.1. Klase 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 se ističu principi postavljanja i dohvaćanja vrijednosti polja (varijabli) unutar klasa, te korištenje različitih modifikatora za ograničavanje pristupa tim poljima.

1.2. Zadatak za pripremu

Napisati program koji će se sastojati od nekoliko klasa spremljenih u različitim paketima. Klase simuliraju posuđivanje kreiranje konfiguracije osobnog računala sastavljenog od pojedinih komponenti. Potrebno je kreirati sljedeće klase i pakete prema koracima:

1. Unutar Eclipse razvojnog okruženja potrebno je kreirati Java projekt koji se naziva kao Vaše prezime i sadrži redni broj vježbe (npr. "Horvat-1"). Kreiranje Java projekta u Eclipse razvojnom okruženju opisano je unutar poglavlja 2.2.2. Priručnika za laboratorijske vježbe.
2. Projekt pretvoriti u „Maven Project“ korištenjem opcije „Configure->Convert to Maven Project“ i na dijalogu koji se prikaže nakon toga potvrditi ponuđene parametre.
3. Kreirati paket **hr.java.vjezbe.entitet**. Kreiranje paketa unutar Eclipse razvojnog okruženja opisano je unutar poglavlja 2.2.2. Priručnika za laboratorijske vježbe.
4. Kreirati klasu **MaticnaPloca** unutar paketa **hr.java.vjezbe.entitet** koja se sastoji od dvije varijable: za naziv proizvođača i tip (obje varijable mogu biti tipa "String"). Napisati konstruktor za klasu **MaticnaPloca** tako da prima oba parametra (naziv proizvođača i tip) i sprema ih u privatne varijable klase. 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 2.2.2. Priručnika za laboratorijske vježbe. Automatsko generiranje "get" i "set" metoda unutar razvojnog okruženja Eclipse prikazano je unutar poglavlja 2.7.2. Priručnika za laboratorijske vježbe.
5. Kreirati klasu **Procesor** unutar paketa **hr.java.vjezbe.entitet** koja se sastoji od četiri varijable: za naziv proizvođača, tip, tip sučelja i brzina (prve tri varijable mogu biti tipa "String", a „brzina“ može biti tipa „BigDecimal“). Napisati konstruktor za klasu **Procesor** tako da prima sva četiri parametra i sprema ih u privatne varijable klase. Ograničiti pristup varijablama na način da ih je moguće dohvaćati samo korištenjem javnih "get" metoda i spremanje samo korištenjem javnih „set“ metoda.
6. Kreirati klasu **RadnaMemorija** unutar paketa **hr.java.vjezbe.entitet** koja se sastoji od tri varijable: za naziv proizvođača, tip i kapacitet (prve dvije varijable mogu biti tipa "String", a „kapacitet“ može biti tipa „Integer“). Napisati konstruktor za klasu **RadnaMemorija** tako da prima sva tri parametra i sprema ih u privatne varijable klase. 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 **TvrđiDisk** unutar paketa **hr.java.vjezbe.entitet** koja se sastoji od tri varijable: za naziv proizvođača, tip i kapacitet (prve dvije varijable mogu biti tipa "String", a „kapacitet“ može biti tipa „BigDecimal“). Napisati konstruktor za klasu **TvrđiDisk** tako da prima sva tri parametra i sprema ih u privatne varijable klase. 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 **Racunalo** unutar paketa **hr.java.vjezbe.entitet** koja se sastoji od četiri objekta: za matičnu ploču (objekt klase **MaticnaPloca**), procesor (objekt klase **Procesor**), radnu memoriju (objekt klase **RadnaMemorija**) i tvrdi disk (objekt klase **TvrđiDisk**). Napisati konstruktor za klasu **Racunalo** tako da prima sva četiri parametra i sprema ih u privatne varijable klase. Ograničiti pristup varijablama na način da ih je moguće dohvaćati samo korištenjem javnih "get" metoda i spremanje samo korištenjem javnih „set“ metoda.
9. Kreirati paket **hr.java.vjezbe.glavna**.
10. 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 jedan objekt klase **MaticnaPloca**, **Procesor**, **RadnaMemorija**, **TvrđiDisk** i **Racunalo**. Za svaki objekt je potrebno od korisnika zatražiti unos svih parametara pojedine komponente i spremiti ih u pripadajuće objekte korištenjem „set“ metode.
11. Unutar metode **main** klase **Glavna** potrebno je ispisati sve informacije o kreiranoj konfiguraciji računala. Kod ispisa je informacije potrebno dohvaćati korištenjem „get“ metoda iz objekta klase **Racunalo**.

Primjer izvođenja programa:

```
Unesite proizvođača matične ploče računala: Asrock
Unesite tip matične ploče računala: Desktop board P4I65G
Unesite proizvođača procesora računala: Intel
Unesite tip procesora računala: Celeron G1840
Unesite tip sučelja procesora računala: Socket 1150
Unesite brzinu procesora računala (u GHz): 2,8
Unesite proizvođača radne memorije računala: Corsair
Unesite tip radne memorije računala: DDR3 1600 MHz
Unesite kapacitet radne memorije računala (u GB): 4
Unesite proizvođača tvrdog diska računala: Seagate
Unesite tip tvrdog diska računala: Barracuda
Unesite kapacitet tvrdog diska računala (u TB): 4
Unesena konfiguracija računala je sljedeća:
Proizvođač matične ploče: Asrock
Tip matične ploče: Desktop board P4I65G
Proizvođač procesora: Intel
Tip procesora: Celeron G1840
Tip sučelja procesora: Socket 1150
Brzina procesora: 2.8 GHz
Proizvođač radne memorije: Corsair
Tip radne memorije: DDR3 1600 MHz
Kapacitet radne memorije: 4 GB
Proizvođač tvrdog diska: Seagate
Tip tvrdog diska: Barracuda
```

Kapacitet tvrdog diska: 4 TB

NAPOMENE:

1. Osim implementacija vježbe prema uputama, dozvoljeno je uvoditi i promjene ako su opravdane i ne narušavaju koncepte objektno-orientiranog 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. Prema potrebi moguće je koristiti rješenje prve laboratorijske vježbe iz prošle akademske godine. Način „eksportanja“ i „importanja“ novog projekta u Eclipse radnu okolinu opisan je poglavljju 2.3. unutar Priručnika za laboratorijske vježbe.

MOGUĆNOSTI UNAPREĐENJA ZADATKA:

1. Kreirati metode za unos podataka i ispis podataka za konzolu pri čemu je dozvoljeno koristiti samo jednu instancu objekta klase „Scanner“.
2. Implementirati metodu „toString“ koja služi za ispis konfiguracije računala prema prikazanom primjeru.
3. Ograničiti unos samo kompatibilnih komponenti u računalu korištenjem dodatnim atributa klase (kao što su sučelja za procesor i matične ploče, broj modula radne memorije i sl.).
4. Ugraditi provjere unesenih podataka i dojavljivati poruku o pogreškama korisniku koji unosi podatke te traženje unošenja ispravnih podataka.