

2. Druga laboratorijska vježba

2.1. Klase i objekti u Javi

Svrha laboratorijske vježbe je korištenje objektno orijentiranih principa kao što su nasljeđivanje i polimorfizam, te korištenje apstraktnih klasa, sučelja, korištenja varijabli i metoda iz nadklasa i sučelja. Time se omogućava višestruko iskorištavanje postojećeg programskog koda, dinamičko povezivanje tipova podataka kod izvođenja programa i korištenje *cast* operacije kad je potrebno odrediti konkretan tip objekta u programu.

2.2. Zadatak za pripremu

Nastaviti razvoj aplikacije iz prve laboratorijske vježbe i uvesti značajke objektno-orijentiranog programiranja kroz sljedeće korake, uz obvezno poštivanje smjernica iz napomena na kraju pripreme:

1. Iskopirati projekt iz prve laboratorijske vježbe i kopiju preimenovati u naziv koji sadrži redni broj vježbe „2“.
2. Unutar paketa „hr.java.vjezbe.entitet“ kreirati apstraktnu klasu „Komponenta“ koja sadrži varijablu za spremanje podatka o nazivu proizvođača, konstruktor koji prima tu varijablu i sprema je u privatnu varijablu, kao i „getter“ i „setter“ metode za tu varijablu.
3. Iz klasa „MaticnaPloca“, „Procesor“, „RadnaMemorija“ i „TvrdiDisk“ obrisati varijablu „nazivProizvodjaca“ i pripadajuće „getter“ i „setter“ metode. U tim klasama potrebno je naznačiti da nasljeđuju klasu „Komponenta“ i unutar njihovih konstruktora ubaciti naredbu koja poziva konstruktor nadklase i pomoću njega postavlja varijablu „nazivProizvodjaca“.
4. Unutar paketa „hr.java.vjezbe.entitet“ kreirati sučelje „Memorijska“ s metodom „uvecajKapacitet“ koja prima „Integer“ parametar i ne vraća ništa. Tu metodu biti će potrebno implementirati u klasama koje će predstavljati komponente računala i implementirati to sučelje na način da se njihova memorija uveća za zadani faktor.
5. U klasama „RadnaMemorija“ i „TvdiDisk“ implementirati sučelje „Memorijska“ i implementirati metodu „uvecajKapacitet“ koja će za zadani faktor definiran „Integer“ parametrom uvećati kapacitet radne memorije, odnosno tvrdog diska te dotične komponente.

6. U sučelje „Memorijska“ dodati statičku implementaciju metode pod nazivom „pretvoriUTB“ koja prima iznos kapaciteta u gigabajtima, a vraća taj isti iznos izražen u terabajtima.
7. Unutar paketa „hr.java.vjezbe.entitet“ kreirati sučelje „Frekvencijska“ koja će imati jednu „default“ metodu pod nazivom „overclock“. Ta metoda mora primiti brzinu komponente, a vraćati uvećanu brzinu za 50%. Naznačiti da to sučelje implementira klasa „Procesor“.
8. U klasu „Procesor“ uvesti četiri konstante koje će označavati četiri moguća sučelja procesora za priključivanje na matičnu ploču: socket AM2, socket AM3, LGA 1151 i socket G3.
9. Glavnu klasu doraditi na način da korisnik unosi dvije konfiguracije računala, prema ih u polje (nije dozvoljeno koristiti dvije odvojene varijable) i ispisuje njihove karakteristike, pri čemu kapacitet radne memorije ispisuje u terabajtima (za pretvorbu koristiti metodu „pretvoriUTB“ iz sučelja „Memorijska“). Kod unosa sučelja za procesor moguće je unijeti samo jednu od ponuđenih vrijednosti (definiranih konstantama iz klase „Procesor“). Nakon ispisa računala s manjim iznosom radne memorije udvostručiti iznos korištenjem metode iz sučelja „Memorijska“ (ako su kapaciteti isti nije potrebno ništa promijeniti), a računalo sa sporijim procesorom „overclokirati“ brzinu za 50% korištenjem metode iz sučelja „Frekvencijska“ (ako su brzine iste, nije potrebno ništa promijeniti) te na kraju ponovno ispisati sve karakteristike od oba računala.

Primjer izvođenja programskog koda:

```
Unos podataka prvog računala:
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 (odaberite broj ispred željenog sučelja):
1) SOCKET_AM2
2) SOCKET_AM3
3) SOCKET_LGA_1151
4) SOCKET_G3
10
Pogreška kod odabira, molimo pokušajte ponovno!
Unesite tip sučelja procesora računala (odaberite broj ispred željenog sučelja):
1) SOCKET_AM2
2) SOCKET_AM3
3) SOCKET_LGA_1151
4) SOCKET_G3
3
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
Unos podataka drugog računala:
Unesite proizvođača matične ploče računala: Asrock
Unesite tip matične ploče računala: Z97 KILLER
Unesite proizvođača procesora računala: Intel
Unesite tip procesora računala: i5-4690K
Unesite tip sučelja procesora računala (odaberite broj ispred željenog
sučelja):
1) SOCKET_AM2
2) SOCKET_AM3
3) SOCKET_LGA_1151
4) SOCKET_G3
4
Unesite brzinu procesora računala (u GHz): 3,9
Unesite proizvođača radne memorije računala: Kingston
Unesite tip radne memorije računala: HyperX Savage DDR3 2400MHz
Unesite kapacitet radne memorije računala (u GB): 8
Unesite proizvođača tvrdog diska računala: Western Digital
Unesite tip tvrdog diska računala: Caviar Green WD20EZRX
Unesite kapacitet tvrdog diska računala (u TB): 2
Unesena konfiguracije računala su sljedeće:
Prvo računalo:
Proizvođač matične ploče: Asrock
Tip matične ploče: Desktop board P4I65G
Proizvođač procesora: Intel
Tip procesora: Celeron G1840
Tip sučelja procesora: SOCKET_LGA_1151
Brzina procesora: 2.8 GHz
Proizvođač radne memorije: Corsair
Tip radne memorije: DDR3 1600 MHz
Kapacitet radne memorije: 0.00390625 TB
Proizvođač tvrdog diska: Seagate
Tip tvrdog diska: Barracuda
Kapacitet tvrdog diska: 4 TB
Drugo računalo:
Proizvođač matične ploče: Asrock
Tip matične ploče: Z97 KILLER
Proizvođač procesora: Intel
Tip procesora: i5-4690K
Tip sučelja procesora: SOCKET_G3
Brzina procesora: 3.9 GHz
Proizvođač radne memorije: Kingston
Tip radne memorije: HyperX Savage DDR3 2400MHz
Kapacitet radne memorije: 0.0078125 TB
Proizvođač tvrdog diska: Western Digital
Tip tvrdog diska: Caviar Green WD20EZRX
Kapacitet tvrdog diska: 2 TB
Nakon promjena, konfiguracije računala su sljedeće:
Prvo računalo:
Proizvođač matične ploče: Asrock
Tip matične ploče: Desktop board P4I65G
Proizvođač procesora: Intel
Tip procesora: Celeron G1840
Tip sučelja procesora: SOCKET_LGA_1151
Brzina procesora: 4.20 GHz
```

```
Proizvođač radne memorije: Corsair
Tip radne memorije: DDR3 1600 MHz
Kapacitet radne memorije: 0.0078125 TB
Proizvođač tvrdog diska: Seagate
Tip tvrdog diska: Barracuda
Kapacitet tvrdog diska: 4 TB
Drugo računalo:
Proizvođač matične ploče: Asrock
Tip matične ploče: Z97 KILLER
Proizvođač procesora: Intel
Tip procesora: i5-4690K
Tip sučelja procesora: SOCKET_G3
Brzina procesora: 3.9 GHz
Proizvođač radne memorije: Kingston
Tip radne memorije: HyperX Savage DDR3 2400MHz
Kapacitet radne memorije: 0.0078125 TB
Proizvođač tvrdog diska: Western Digital
Tip tvrdog diska: Caviar Green WD20EZR
Kapacitet tvrdog diska: 2 TB
```

NAPOMENE:

1. Programski kod za unos karakteristika računala potrebno je odvojiti u zasebnu metodu koja se poziva dvaput (za svaki unos konfiguracije računala). Ta metoda mora primiti objekt klase „Scanner“, a vratiti objekt klase „Racunalo“.
2. Programski kod za ispis karakteristika računala također odvojiti u zasebnu metodu koja prima objekt klase „Racunalo“ i ne vraća ništa te se poziva kod ispis karakteristika svakog od računala.
3. Kod unosa sučelja procesora koristiti beskonačnu petlju koja se prekida tek nakon što korisnik odabere jednu od ponuđenih vrijednosti.

MOGUĆNOSTI UNAPREĐENJA ZADATKA:

1. Ograničiti maksimalne brzine i količine memorije komponenti računala i u slučaju prekoračenja tih vrijednosti kod povećanja memorije ili „overclockiranja“, ispisati poruku o pogrešci.
2. Uvesti apstraktnu klasu proizvoljnog imena koju nasljeđuje računalo i koja ima komponente spremljene u polje tipa „Komponenta“ te ih dohvaća i „prepoznaje“ korištenjem operatora „instanceof“. Tu klasu mora koristiti klasa „Racunalo“ za spremanje i dohvaćanje komponenti iz polja.
3. Uvesti novo sučelje „Prikljucna“ koja ima dvije metode: „prikljuci“ koja se poziva prilikom dodavanja komponenti na matičnu ploču i „brojPriključenihKomponenti“ koja vraća broj priključenih komponenti na matičnu ploču. To sučelje potrebno je implementirati u klasi „MaticnaPloca“ i implementirati navedene metode na odgovarajući način.