

2. Druga laboratorijska vježba

2.1. TEMA VJEŽBE

Svrha laboratorijske vježbe je korištenje koncepata objektno orijentiranog programiranja u programskom jeziku Java, uz korištenje podklasa i nadklasa kod nasljeđivanja, korištenje modifikatora „protected“ i „super“, konstruktora u podklasama, nadjačavanje metoda, polimorfizma, apstraktnog metoda i sučelja te novih funkcionalnosti u programskom jeziku Java kao što su zapisi (engl. *records*) te zapečaćenih (engl. *sealed*) sučelja i klasa.

2.2. ZADATAK ZA PRIPREMU

Proširiti rješenje prve laboratorijske vježbe na način da se kopira rješenje prethodne laboratorijske vježbe te preimenuje u naziv koji sadrži indeks „2“, umjesto „1“. Osim same mape s projektom, potrebno je promijeniti i naziv projekta unutar IntelliJ-a korištenjem opcije „Refactor->Rename“. Program je potrebno proširiti na sljedeći način:

1. U paket „model“ dodati novu apstraktnu klasu „NamedEntity“ koji sadrži „name“ tipa „String“, konstruktor koji postavlja taj objekt te „getter“ i „setter“ metode za taj objekt.
2. Sve klase koje imaju objekt „name“ promijeniti na način da nasljeđuju klasu „NamedEntity“, a samim time i objekt „name“, koriste njen konstruktor te „getter“ i „setter“ metode.
3. U paketu „model“ kreirati sučelje „Edible“ koje ima metodu „calculateKilocalories“ koja vraća broj kilokalorija (cijeli broj) koji ima navedena namirnica koja je jestiva. Također u isto sučelje dodati metodu „calculatePrice“ koja vraća cijenu za namirnicu koja će se izračunavati prema težini.
4. U paketu „model“ kreirati dvije klase (imenovati ih po želji) koje nasljeđuju klasu „Item“, implementiraju sučelje „Edible“ te sadržava cjelobrojnu konstantu koja definira broj kalorija po kilogramu jestive namirnice. Te klase je potrebno proširiti varijablom „weight“ tipa „BigDecimal“ koja definira težinu namirnice (u kilogramima). Kod

unosa artikla od korisnika zatražiti unos informacije želi li odabrati artikl koji predstavlja hranu ili ostale artikle te ako je odabrao da želi odabrati hranu, ponuditi mu te dvije namirnice koje su kreirane. Nakon odabira namirnice je potrebno izračunati broj kilokalorija namirnice te ih ispisati na ekran te cijenu s obzirom na težinu (vrijednost u varijabli „price“ u klasi „Item“ uzimati kao cijenu za jedan kilogram namirnice). Također je potrebno preračunati cijenu namirnice na način da se uzima u obzir težina po kilogramu te cijena po kilogramu.

5. Na kraju programa ispisati podatke o namirnicama koja ima najveći broj kilokalorija te najveću ukupnu cijenu s obzirom na težinu.
6. U paket „model“ dodati zapis (engl. *record*) pod nazivom „Discount“ te ga dodati kao novi objekt unutar klase „Item“. Taj zapis mora sadržavati podatak „discountAmount“ koji označava iznos popusta (u postocima). Kod unosa podataka za artikle je od korisnika potrebno odabrati i iznos popusta te kod preračunavanja cijene uzeti i popust u obzir.
7. U paket „model“ dodati zapečaćeno sučelje (engl. *sealed interface*) koje će se zvati „Technical“ i predstavlja tehničku robu kao artikl. To sučelje mora moći implementirati samo klasa „Laptop“ koja će se kreirati u osmom koraku. Sučelje „Technical“ mora sadržavati metodu koja vraća trajanje garantnog roka tehničke robe kao artikla (cjelobrojnu vrijednost u mjesecima).
8. Kreirati klasu „Laptop“ koja će nasljeđivati klasu „Item“ i implementirati sučelje „Technical“. Kod unosa novog artikla od korisnika zatražiti informaciju želi li odabrati laptop (pored već postojećih vrsta artikala) te ako želi, kreirati objekt klase „Laptop“ te od korisnika zatražiti dodatnu informaciju o trajanju garantnog roka laptopa.
9. Na kraju programa ispisati podatke o laptopu koji sadrži najkraći garantni rok od svih unesenih laptopa.
10. Implementirati „Builder pattern“ za klasu „Address“ te umjesto poziva konstruktora za klasu „Address“ unutar „main“ metode koristiti „Builder pattern“.

NAPOMENE:

1. Osim implementacija vježbe prema uputama, dozvoljeno je uvoditi i promjene ako su opravdane i ne narušavaju koncepte objektno-orijentiranog programiranja.