

1. Prva laboratorijska vježba

1.1. TEMA VJEŽBE

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 aplikacije za organiziranje proizvodnje različitih dobara. Program je potrebno implementirati prema sljedećim koracima:

1. Preuzeti najnoviju verziju Jave 23 sa stranica <https://www.oracle.com/java/technologies/downloads/#jdk23-windows> i instalirati je na željenu lokaciju.
2. Preuzeti najnoviju inačicu IntelliJ IDEA Ultimate razvojnog okruženja s poveznice: <https://www.jetbrains.com/idea/download/#section=windows>. Na stranici <https://www.jetbrains.com/community/education/#students> je potrebno kreirati JetBrains korisnički račun korištenjem e-maila od TVZ-a. Nakon aktivacije korisničkog računa potrebno je aktivirati IntelliJ IDEA licencu korištenjem ovih uputa: <https://www.jetbrains.com/help/idea/register.html>. U slučaju problema s instalacijom Ultimate verzije IntelliJ alata, dostupna je i Community verzija alata koja je sasvim dostatna za laboratorijske vježbe.
3. Kreirati projekt s nazivom koji odgovara Vašem prezimenu i rednom broju vježbe (npr. „Radovan-1“) korištenjem sljedećih uputa: <https://www.jetbrains.com/help/idea/creating-and-running-your-first-java-application.html>.
4. Unutar projekta iz prošlog zadatka unutar „src“ mape kreirati paket pod nazivom „hr.java.restaurant.model“.
5. U paketu iz četvrtog zadatka kreirati klasu „Category“ koja sadrži dvije „String“ varijable pod nazivom „name“ i „description“. Korištenjem uputa sa stranica <https://www.jetbrains.com/help/idea/generating-code.html#generate-constructors> i <https://www.jetbrains.com/help/idea/generating-code.html#generate-getters-setters>

generirati konstruktor koji prima oba ulazna parametra, te „getter“ i „setter“ metode za te varijable.

6. U paketu iz četvrtog zadatka kreirati klasu „Ingredient“ koja sadrži objekt klase „String“ pod nazivom „name“, objekt „category“ tipa „Category“, objekt „kcal“ tipa „BigDecimal“ te objekt klase „String“ pod nazivom „preparationMethod“. Potrebno je generirati konstruktor koji prima sve ulazne parametre, te „getter“ i „setter“ metode za te varijable.
7. U paketu iz četvrtog zadatka kreirati klasu „Meal“ koja sadrži objekt klase „String“ pod nazivom „name“, objekt „category“ tipa „Category“, polje objekata klase „Ingredient“ te objekt „price“ tipa „BigDecimal“. Na sličan način kao što je opisano u prošlom zadatku je potrebno generirati konstruktor koji prima sve ulazne parametre, te „getter“ i „setter“ metode za te varijable.
8. U paketu iz četvrtog zadatka kreirati Java klasu pod nazivom „Chef“ koja sadrži dva objekta tipa „String“: „firstName“ i „lastName“ te objekt „salary“ tipa „BigDecimal“. Potrebno je generirati konstruktor koji prima sve ulazne parametre, te „getter“ i „setter“ metode za te varijable.
9. U paketu iz četvrtog zadatka kreirati Java klasu pod nazivom „Waiter“ koja sadrži dva objekta tipa „String“: „firstName“ i „lastName“ te objekt „salary“ tipa „BigDecimal“. Potrebno je generirati konstruktor koji prima sve ulazne parametre, te „getter“ i „setter“ metode za te varijable.
10. U paketu iz četvrtog zadatka kreirati Java klasu pod nazivom „Deliverer“ koja sadrži dva objekta tipa „String“: „firstName“ i „lastName“ te objekt „salary“ tipa „BigDecimal“. Potrebno je generirati konstruktor koji prima sve ulazne parametre, te „getter“ i „setter“ metode za te varijable.
11. U paketu iz četvrtog zadatka kreirati Java klasu pod nazivom „Address“ koja sadrži četiri varijable tipa „String“: „street“, „houseNumber“, „city“ i „postalCode“. Na sličan način kao što je opisano u prošlom zadatku je potrebno generirati konstruktor koji prima sve ulazne parametre, te „getter“ i „setter“ metode za te varijable.
12. U paketu iz četvrtog zadatka kreirati Java klasu pod nazivom „Restaurant“ koja sadrži varijable „name“ (tipa String), „address“ (tipa „Address“), „meals“ (tipa polje objekata klase „Meal“), „chefs“ (tipa polje objekata klase „Chef“), „waiters“ (tipa polje objekata klase „Waiter“) i „deliverers“ (tipa polje objekata klase „Deliverer“). Na sličan način kao što je opisano u prošlom zadatku je potrebno generirati konstruktor koji prima sve ulazne parametre, te „getter“ i „setter“ metode za te varijable.

13. U paketu iz četvrtog zadatka kreirati Java klasu pod nazivom „Order“ koja sadrži varijable „restaurant“ tipa „Restaurant“, polje „meals“ koje sadrži objekte klase „Meal“, objekt „deliverer“ klase „Deliverer“ te „deliveryDateAndTime“ klase „LocalDateTime“. Na sličan način kao što je opisano u prošlim zadacima je potrebno generirati konstruktor koji prima sve ulazne parametre, te „getter“ i „setter“ metode za te varijable.
14. Kreirati novi paket pod nazivom „hr.java.production.main“.
15. U paketu iz prošlog zadatka kreirati klasu „Main“ koja sadrži „main“ metodu. U toj metodi je potrebno zatražiti unos podataka za tri kategorije artikala (objekata klase „Category“), pet sastojala (objekata klase „Ingredient“), tri jela (objekata klase „Meal“), tri kuhara (objekata klase „Chef“), tri konobara (objekata klase „Waiter“), tri dostavljača (objekata klase „Deliverer“), tri restorana (objekata klase „Restaurant“), a za svaki od njih i pripadajuću adresu (objekt klase „Address“) te tri narudžbe (objekata klase „Order“). U slučaju neispravnog unosa, korisniku je potrebno prijaviti pogrešku i zatražiti novi unos.
16. Pronaći restoran kojem je poslana najskuplja narudžba. Ako ima više restorana s istom najvišom cijenom narudžbe, potrebno je ispisati sve podatke o svim takvim restoranima.
17. Pronaći dostavljača koji je dostavio najviše dostava. Ako ima više dostavljača s najviše dostava, potrebno je ispisati sve podatke o svim dostavljačima.
18. Logiku unosa svakog od entiteta je potrebno izdvojiti u zasebnu metodu.

NAPOMENE:

1. Osim implementacija vježbe prema uputama, dozvoljeno je uvoditi i promjene ako su opravdane i ne narušavaju koncepte objektno-orijentiranog programiranja.
2. 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>.