

4. Četvrta laboratorijska vježba

4.1. TEMA VJEŽBE

Svrha laboratorijske vježbe je korištenje zbirke iz skupina listi, setova i mapa te zamjena svih polja u aplikaciji sa zbirkama. U sklopu vježbe je također sve skupine konstanti potrebno zamijeniti enumeracijama. Također se u vježbi implementira i algoritam prema kojim se sortiraju objekti unutar zbirki.

4.2. ZADATAK ZA PRIPREMU

Proširiti rješenje druge laboratorijske vježbe na način da se kopira rješenje te preimenuje u naziv koji sadrži indeks „4“, umjesto „3“. 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. Unutar klase „Meal“ umjesto polja klase „Ingredient“ koristiti set te ispraviti sve preostale detalje u kodu koji se pojave nakon te promjene. U klasi „Ingredient“ potrebno je nadjačati metode „hashCode“ i „equals“. Za generiranje tih metoda koristiti opciju „Generate“ u IntelliJ okruženju.
2. Unutar klase „Restaurant“ umjesto polja objekata „meals“, „chefs“, „waiters“ i „deliverers“ također koristiti setove te ispraviti sve preostale detalje u kodu koji se pojave nakon te promjene. U klasama „Meal“, „Chef“, „Waiter“ i „Deliverer“ također nadjačati metode „hashCode“ i „equals“. Za generiranje tih metoda koristiti opciju „Generate“ u IntelliJ okruženju.
3. Unutar klase „Order“ umjesto polja „Meals“ koristiti listu te ispraviti sve preostale detalje u kodu koji se pojave nakon te promjene.
4. Na svim ostalim mjestima u programu polja zamijeniti s listama.
5. Kreirati novi paket „hr.java.restaurant.enum“.
6. U paketu iz petog koraka kreirati enumeraciju „ContractType“ koji će zamijeniti konstante kreirane u klasu „Contract“ koje predstavljaju vrijednosti „FULL_TIME“ i „PART_TIME“ te doraditi sve ostale dijelove koda koji su koristiti konstante da sad koriste vrijednosti iz enumeracije.

7. U „main“ metodi koristiti mapu koja će sadržavati informacije o tome u kojim restoranima se priprema određeno jelo. Ključ u mapi mora biti objekt klase „Meal“, a vrijednost lista restorana u kojima je moguće naručiti to jelo. Na kraju programa omogućiti korisniku odabir nekog od postojećih jela, a program mora ispisati podatke o restoranima u kojima je moguće naručiti to jelo.
8. Kreirati novi paket „hr.java.restaurant.sort“.
9. U paketu iz prethodnog koraka napisati klasu proizvoljnog imena koja će implementirati sučelje „Comparator“ te implementirati metodu „compare“ koja sortira zaposlenike po plaći, od najveće do najmanje. Iskoristiti tu klasu za sortiranje zaposlenika u svakom restoranu te ispisati koji zaposlenik u svakom od restorana ima najveću plaću.
10. U paketu iz osmog koraka napisati klasu proizvoljnog imena koja će implementirati sučelje „Comparator“ te implementirati metodu „compare“ koja sortira zaposlenike po trajanju ugovora, od najkraćeg do najdužeg. Iskoristiti tu klasu za sortiranje zaposlenika u svakom restoranu te ispisati sortirana trajanja ugovora zaposlenika u restoranima.
11. U paketu iz osmog koraka napisati klasu proizvoljnog imena koja će implementirati sučelje „Comparator“ te implementirati metodu „compare“ koja sortira jela po broju restorana u kojima se jelo može naručiti, od jela s najviše restorana do jela s najmanje restorana. Ispisati sortirana jela po tom kriteriju. Potrebno je koristiti mapu iz sedmog koraka.
12. U paketu iz osmog koraka napisati klasu proizvoljnog imena koja će implementirati sučelje „Comparator“ te implementirati metodu „compare“ koja sortira namirnice po abecedi. To sortiranje potrebno je iskoristiti kod ispisa namirnica koje su korištene kod pripreme jela.
13. Za sve nove klase i metode napisati Javadoc dokumentaciju.
14. Nijedna klasa ne smije imati više od 200 linija koda.

NAPOMENE:

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