

6. Šesta laboratorijska vježba

6.1. TEMA VJEŽBE

Svrha laboratorijske vježbe je korištenje tekstualnih i binarnih datoteka za čitanje podataka iz njih te zapisivanje podataka u njih, kao i serijaliziranje i deserijaliziranje objekata korištenih u aplikaciji.

6.2. ZADATAK ZA PRIPREMU

Proširiti rješenje pete laboratorijske vježbe na način da se kopira rješenje te preimenuje u naziv koji sadrži indeks „6“, umjesto „5“. 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. Kreirati mapu „dat“ na razini projekta na istoj razini kao što je kreirana mapa „logs“.
2. Unutar mape „dat“ kreirati datoteku „categories.txt“ koja će sadržavati informacije o kategorijama formatiranih na sljedeći način, na primjer:

```
1
Vegetarian
In this category you can find all vegetarian ingredients and meals.
2
Vegan
In this category you can find all vegan ingredients and meals.
3
Meat
In this category you can find all meat ingredients and meals.
```

Svaki zapis o kategoriji mora sadržavati tri varijable: „id“, „name“, „description“. Nakon što se pročitaju podaci iz datoteke, moraju se kreirati odgovarajući objekti klase „Category“ i koristiti u ostatku aplikacije.

Na navedenom primjeru prikazana su tri skupa podataka od po tri retka za tri kategorije označene s tri različite boje.

Na sličan način je potrebno kreirati datoteke koje će sadržavati zapise i za sve ostale entitete korištene u aplikaciji: „ingredients.txt“, „meals.txt“, „chefs.txt“, „waiters.txt“, „deliverers.txt“, „addresses.txt“, „restaurants.txt“, „orders.txt“ i „contracts.txt“.

U slučajevima kad je npr. potrebno povezivati obroka sa sastojcima, to je potrebno učiniti na način da se u jednu liniju postave svi identifikatori sastojaka odvojeni zarezom, npr. „1,2,3“. U slučaju obroka (unutar datoteke „meals.txt“), to može izgledati ovako:

```
1
Pizza Salami
3
1,2,3,4,5
10.00
```

Navedeni primjer bi označavao da jelo „Pizza Salami“ s identifikatorom „1“ ima identifikator kategorije „3“, pet identifikatora sastojaka „1“, „2“, „3“, „4“ i „5“, te cijenu „10.00“.

3. Nakon pokretanja programa moraju se učitati podaci o objektima iz svih datoteka te izvesti ostatak programa (koji se u petoj vježbi izvršavao nakon uspješnog unošenja podataka o objektima).
4. U svim klasama koje predstavljaju entitete u aplikaciji implementirati sučelje koje omogućava serijalizaciju objekata.
5. Na kraju programa serijalizirati i deserijalizirati sve objekte klasa „Restaurant“, „Order“ i „Contract“.
6. Napisati Javadoc dokumentaciju za sve klase u projektu.
7. 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.