

## 3. Treća laboratorijska vježba

### 3.1. TEMA VJEŽBE

Svrha laboratorijske vježbe je obrađivanje različitih vrsta iznimaka te kreiranje i bacanje iznimaka kojima se signalizira nastanak specifičnih situacija prilikom izvođenja Java aplikacije. Osim toga je opisan način korištenje vanjske biblioteke za upravljanje log datotekama – Logback. Također, potrebno je aplikaciju dokumentirati Javadoc dokumentacijom.

### 3.2. ZADATAK ZA PRIPREMU

Proširiti rješenje druge laboratorijske vježbe na način da se kopira rješenje te preimenuje u naziv koji sadrži indeks „3“, umjesto „2“. 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. Na svim mjestima u aplikaciji gdje se unose brojčane vrijednosti implementirati obradu iznimke „InputMismatchException“ na način da se korisniku ispiše poruka o pogrešci te se od njega traži ponovni unos vrijednosti.
2. Kreirati novi paket „hr.java.restaurant.exception“.
3. U paket „hr.java.restaurant.exception“ dodati novu označenu iznimku proizvoljnog imena koja će se bacati u slučaju kad se unesu nazivi entiteta koji već prethodno uneseni (imena i prezimena osoba, restorana, jela itd.). Iznimku je u tom slučaju potrebno obraditi, ispisati poruku od pogrešci i od korisnika zatražiti unos podataka koji se ne poklapa s već unesenim podacima. Klasa koja označava tu označenu iznimku mora sadržavati sve konstruktore koje je moguće generirati pomoću IntelliJ okruženja.
4. U paket „hr.java.restaurant.exception“ dodati novu neoznačenu iznimku proizvoljnog imena koja će se bacati u slučaju kad se unesu cijene koje su negativne ili nerealne ili plaće manje od minimalno zadane plaće. Iznimku je u tom slučaju potrebno obraditi, ispisati poruku od pogrešci i od korisnika zatražiti unos podataka koji je u skladu s poslovnim pravilima koje je potrebno proizvoljno definirati. Klasa koja označava tu označenu iznimku mora sadržavati sve konstruktore koje je moguće generirati pomoću IntelliJ okruženja.
5. Proširiti rješenje vježbe na način da se u „pom.xml“ datoteku doda ovisnost o posljednjoj verziji „logback-classic“ biblioteke. U mapu „resources“ IntelliJ projekta dodati datoteku „logback.xml“. Nakon toga odabrati opciju „Maven-Reload project“.

6. Kreirati detaljne logove svih razina po cijelom kodu te testirati zapisivanje u „log“ datoteku. „Logiranje“ je potrebno obaviti za sve korisničke unose podataka, sve pozive metoda, sve ulazne i izlazne vrijednosti parametara metoda te svih pogrešaka u kodu. Logovi se ne smiju ispisivati u konzolu, već samo u „log“ datoteku.
7. Na kraju vježbe je potrebno korištenjem Javadoc dokumentacije dokumentirati sve klase, konstruktore i metode (osim „getter“ i „setter“ metoda) te generirati Javadoc dokumentaciju korištenjem uputa sa sljedećeg linka:  
<https://www.jetbrains.com/help/idea/working-with-code-documentation.html#generate-javadoc>.
8. Ako je potrebno, doraditi klase na način da nijedna nema 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.
2. Nije dozvoljeno korištenje elemenata gradiva koje još nije obrađeno, kao što su liste i slično.