

Tutorijal 4

Priprema za tutorijal

Potrebno je kroz Command Prompt napraviti Maven projekat pod nazivom **t4-z1**. U paketu ba.unsa.etf.rpr kreirati sljedeće:

Klasu **Laptop**, koja će imati atribute:
brend (**String**), model (**String**), cijena (**double**), ram (**int**), hdd (**int**), ssd (**int**), procesor (**String**), grafickaKartica (**String**), velicinaEkрана (**double**).
Klasa treba da slijedi *JavaBeans* specifikaciju.

Interface **LaptopDao**, koji u sebi ima metode: *dodajLaptopUListu(Laptop laptop)*, *dodajLaptopUFile(Laptop laptop)*, *getLaptop(String procesor)*, *napuniListu(ArrayList<Laptop> laptopi)*, *vratiPodatkeIzDatoteke()*.

Zadatak 1

Potrebno je napraviti i klase: **LaptopDaoSerializableFile**, **LaptopDaoJSONFile**, **LaptopDaoXMLFile**. Svaka od ovih klasa treba implementirati interfejs **LaptopDao** i imati dva atributa: file (**File**) i laptopi (**ArrayList<Laptop>**).

LaptopDaoSerializableFile klasa treba da bude implementirana na način da se lista laptopa zapiše u .txt file. Koristiti klase za rad sa binarnim datotekama.

Analogno, **LaptopDaoJSONFile** treba da bude implementirana tako što će se vršiti zapis laptopa iz liste u .json file i to ćete realizirati koristeći **ObjectMapper** klasu. **LaptopDaoXMLFile** klasa vrši zapis laptopa iz liste u .xml file, prilikom čega ćete koristiti **XMLMapper** klasu.

Obratite pažnju na to da je potrebno da za korištenje ovih klasa (i općenito za rad sa .json i .fxml datotekama) ubacite odgovarajući dependency u pom.xml datoteku.

Metoda `dodajLaptopUFile(Laptop laptop)` treba da zapiše parametar u datoteku. *Prijašnji zapisani objekti tipa Laptop trebaju ostati u datoteci, a ne vršiti prepisivanje preko tih objekata.* (Misli se na objekte prilikom tog pokretanja programa. Dakle, ako prilikom pokretanja programa hoćete 5 objekata da dodate, *ne smiju* se objekti prepisivati jedan preko drugog. Pri novom pokretanju, objekti se upisuju ispočetka u datoteku.)

Metoda `getLaptop(String procesor)` vraća objekat tipa Laptop (ukoliko takav postoji) sa procesorom koji je poslan kao parametar. *Ako ne postoji takav objekat, metoda baca izuzetak tipa `NeodgovarajuciProcesorException`.*

Metoda `napuniListu(ArrayList<Laptop> laptopi)` dodjeljuje vrijednost atributu *laptopi*.

Metoda `vratiPodatkeIzDatoteke()` treba vratiti listu objekata tipa Laptop koji su upisani u datoteku. *Dakle, potrebno je pročitati sadržaj datoteke i na osnovu toga kreirati listu koja će se vratiti iz metode.*

Za sve tri klase koje rade sa datotekama je potrebno napisati mali testni program.

Zadatak 2

Napisati JUnit 5 testove za prvi zadatak. Potrebno je na barem jednom mjestu upotrijebiti i Mocking.

Napomena:

Voditi računa da commiti budu atomic.

Za Mocking ukratko objasniti u komentaru zašto ste ga baš u tom slučaju upotrijebili.

Vježbu obavezno pushati na GitHub.