

## Programski jezici 2

– 07.02.2024. –

1. **(30)** U fikcionoj laboratoriji RoboTech, inženjeri rade na razvoju robota specijalizovanih za izvršavanje različitih zadataka. Roboti imaju svoje serijske brojeve (cjelobrojni podatak), tipove (montažni, čistački, istraživački) i tehničke karakteristike (snaga, autonomija i visina). Roboti sadrže tri različite komponente: senzor, motor i bateriju. Svaka komponenta ima svoj identifikator, proizvođača i godinu proizvodnje. Motore karakteriše snaga u kilovatima, senzore odlikuje vrsta (optički, ultrazvučni) i raspon detekcije, dok baterije imaju kapacitet. Tokom rada, kod robota može doći do oštećenja sa određenom vjerovatnoćom (funkcija koja u svake sekunde provjerava da li je slučajno generisan broj od 1 do 100 manji od 5). Sve akcije robota se predstavljaju putem ispisivanja odgovarajućih poruka na konzolu svake sekunde. Na primer, tokom rada čistački robot ispisuje da je u statusu čišćenja prostora sa svim podacima o istom (koji je serijski broj, koji tip, tehničke karakteristike, podaci o komponentama, koji senzor, kapacitet baterije itd) i nakon završetka rada javlja da je uspešno očišćen prostor. Vrijeme rada robota je slučajna vrijednost između 10 i 30 sekundi. Laboratorija ima poseban sistem za uzbunu koji se može aktivirati unosom vrijednosti 1 u tekstualnu datoteku pod nazivom *alarm.txt*. Tada svi roboti prestaju sa radom (vrijeme rada se isto zamrzne), dok se u datoteku ne unese vrijednost 0. Na početku simulacije, kreira se po 10 robota svakog tipa sa pripadajućim komponentama (po 5 robota ima jedan tip senzora). Nakon toga, roboti obavljaju zadatke prema svojoj specijalizaciji. Tokom simulacije, detaljno ispisivati poruke o svim značajnim događajima na konzolu. Svi događaji u vezi oštećenja robota upisuju se u odvojenu tekstualnu datoteku. Nije dozvoljena upotreba enumeracija.
2. **(20)** Napisati program koji pretvara objekte iz XML i JSON formata u Java objekte. XML i JSON fajlovi se nalaze na Moodle stranici predmeta. Metode za učitavanje fajla i dobijanje atributa i njihovih vrijednosti trebaju biti: *HashMap<String, String> load(File f)*, a razlikuju se implementacije za parsiranje XML i JSON formata. Učitani atributi i vrijednosti se proslijeđuju odgovarajućoj metodi za kreiranje objekata: *T convert(HashMap<String, String> attr, T obj)*, koje na osnovu imena atributa setuju odgovarajuće vrijednosti (dozvoljeno je ručno mapiranje). U *main* metodi pozvati odgovarajuće metode i kreirati objekte, a zatim ih ispisati na konzolu pozivom metoda *toString* (ispisati vrijednosti svih atributa).
3. **(20)** Kreirati klasu Automobil. Svaki automobil ima godinu proizvodnje, marku, model, broj vrata, snagu motora, i atribut tip (enum tip, koji ima minimalno sljedeće tri vrijednosti: limuzina, *hatchback* i SUV). Dva automobila su jednaka ukoliko su istog tipa i sva ostala polja su im jednaka. Grupa automobila predstavlja *LinkedList* automobila. Korištenjem Java Stream API-ja i lambda izraza implementirati sljedeće funkcionalnosti:
  - a. Spajanje grupa automobila koji su SUV-ovi - spojiti dvije grupe automobila, tako što se spajaju u novu listu i pri tom se spajaju samo SUV-ovi čija je snaga motora veća od 150KS.
  - b. Sortiranje grupe automobila po snazi motora od veće ka manjoj - sortirati ih i ispisati na konzoli korištenjem *stream-a*.
  - c. Sumiranje kapaciteta vrata svih automobila iz grupe koji su tipa *hatchback* i imaju više od dvoje vrata korištenjem *Function* interfejsa.
  - d. Prikazivanje automobila koji je najbliži prosječnoj snazi motora korištenjem *stream-a*.

**Vrijeme za rad: 180 minuta**