

Programski jezici 2 - 2246

06.09.2018.

1. **(30)** Napisati aplikaciju simulacije sipanja goriva na benzinskoj pumpi. Gorivo mogu da sipaju automobili, kamioni, motocikli i traktori. Pod benzinskom pumpom se smatra objekat sa četiri pristupna mjesta od kojih se na svakom može sipati benzin, dizel i lož ulje. Sva vozila koja dolaze na benzinsku pumpu sipaju određeni tip goriva. Svako vozilo ima svoj naziv, kapacitet rezervoara i trenutno stanje goriva (manje od kapaciteta rezervoara) kao i tip goriva koje koristi (benzin ili dizel). Sipanje lož ulja u vozilo je zabranjeno ali se može desiti (u 10% slučajeva) kao i sipanje pogrešnog tipa goriva (u 10% slučajeva). Regularno je sipanje lož ulja u vozila sa dizel pogonom dok u vozila sa benzinskim pogonom nije i u tom slučaju se smatra da je nasut pogrešan tip goriva. U svako vozilo se sipa gorivo do punog kapaciteta rezervoara. Prilikom sipanja goriva posebna klasa policajac svakih 5 sekundi provjerava koji tip goriva se sipa na svakom od pristupnih mjesta i ukoliko se u neko od vozila sipa lož ulje i vozilo je tipa dizel detektuje se i evidentira takav slučaj. Klase u vezi vozila nalaze se u jednom modulu, klasa policajac u drugom modulu a preostale klase u trećem modulu. Simulacija rada benzinske pumpe sastoji se od sljedećeg:
 - Kreira se 20 vozila koja trebaju nasuti gorivo (kreiranje podrazumijeva automatsko generisanje svih atributa vozila, kapacitet rezervoara je od 30 do 100 litara).
 - Nakon kreiranja, vozila se postavljaju u red pred benzinsku pumpu.
 - Korisničkim unosom „START“ sa tastature benzinska pumpa počinje sa radom. Jedna odvojena nit predstavlja „pregled“ stanja na pristupnim mjestima za sipanje goriva gdje se smještaju vozila iz reda. Sipanje goriva se vrši po pravilu 10 litara po sekundi. Obrađeni se tom prilikom uklanjaju iz reda. Prilikom obrade vozila na konzolu se ispisuju podaci o vozilu te koliko i kojeg tipa je nasuto gorivo.
 - Na kraju simulacije ispisuju se i serijalizuju u fajl vozila.ser vozila u koje je sipano pogrešno gorivo kao i vozila koja su primjećena da sipaju lož ulje.
2. **(15)** Napisati program za rad sa fajl sistemom. Program se može pokrenuti isključivo iz komandne linije, a dostupne opcije koje se mogu iskoristiti su:
 - pregled svih fajlova u folderu koji počinju određenim stringom (*--starts-with vrijednost*),
 - pregled svih fajlova u folderu koji završavaju određenim stringom (*--ends-with vrijednost*),
 - sortiranje fajlova u ispisu po veličini (*--sort [asc|desc]*),
 - ispis podataka o fajlovima u tekstualni fajl (*>> ime_fajla*), pri čemu se u fajl upisuju puni nazivi fajlova i njihova veličina.Sve opcije se mogu kombinovati, a redoslijed unosa naredbi ne smije biti ograničen. Primjer: *java App --starts-with ispit --sort*
3. **(25)** Napisati program koji upotrebom *Java NIO* čita tekst iz fajla koji je dostupan na *Moodle* stranici predmeta. Nakon što se fajl pročita počinje se sa obradom koja treba da se uradi pomoću *Java Stream API*-ja, *Predicate* i *Consumer* objekata, na sljedeći način:
 - na osnovu podataka iz *txt* fajla kreiraju se objekti koji imaju atribut ključ i vrijednost i svi kreirani objekti se smještaju u proizvoljno odabranu kolekciju,
 - poziva se funkcija kojoj se proslijeđuje lambda izraz za ispis svih ključeva na konzolu,
 - kolekcija se filtrira tako da se ispiše 10 ključeva sa najvećim vrijednostima,
 - kolekcija se filtrira tako da se 10 ključeva sa najvećim vrijednostima upiše u tekstualni fajl,
 - kolekcija se filtrira tako da se ispišu svi ključevi čije su vrijednosti veće od 10,
 - ispisuje se zbir svih vrijednosti i ukupan broj ključeva.

Vrijeme rada: 180 minuta

Programski jezici 2 - A401

06.09.2018.

1. **(30)** Napisati pojednostavljenu aplikaciju za upravljanje proizvodnim procesom fabrike. Fabrika ima više proizvodnih traka, a svaka traka ima po četiri mašine: mašina za pripremu materijala, mašina za grubu obradu materijala, mašina za finu obradu materijala i mašina za pakovanje. Svaka mašina ima svoj jedinstveni identifikator, brzinu rada (u milisekundama) i maksimalan kapacitet (broj u intervalu od 10 do 100). Materijali koji se koriste su: plastika, staklo i metal, a svi imaju naziv i osobine. Plastika se može savijati i topiti, staklo se može sjeći i topiti, a metal se može sjeći i savijati. Tokom proizvodnog procesa kreira se po 10 materijala koji se šalju na sve mašine u jednoj traci. Mašina za grubu obradu materijala može da obradi materijale koji se mogu topiti i sjeći, a mašina za finu obradu materijala može da obradi materijale koji se mogu savijati. Ukoliko na mašinu dođe materijal koji ta mašine ne može obraditi baca se izuzetak. Mašine rade konstantno, a njihova brzina je vrijeme koje mašina potroši za obradu jednog materijala. Obrada materijala treba da se implementira pozivom metode koja ispisuje odgovarajuću poruku. Pokretanje rada proizvodne trake obavlja se ručno. Mašina za pakovanje sastavlja sve obrađene materijale u konačan proizvod. Potrebno je ispisivati vrijeme izvršavanja svakog proizvodnog ciklusa na proizvodnoj traci.
2. **(20)** Napisati RMI aplikaciju za rad sa fajl sistemom. Korisnici u klijentskom dijelu unose naredbe koje se izvršavaju na serverskom dijelu sistema:
 - pregled svih fajlova u folderu koji počinju određenim stringom (*--starts-with vrijednost*),
 - pregled svih fajlova u folderu koji završavaju određenim stringom (*--ends-with vrijednost*),
 - sortiranje fajlova u ispisu po veličini (*--sort [asc|desc]*),
 - ispis podataka o fajlovima u tekstualni fajl (*>> ime_fajla*), pri čemu se u fajl upisuju puni nazivi fajlova i njihova veličina.
3. **(20)** *JavaSensor* je pojednostavljena serverska aplikacija koja omogućava očitavanje većeg broja uređaja na određenoj lokaciji. Očitavanje uređaja se simulira metodama koje generišu slučajne vrijednosti temperature, pritiska i vlažnosti vazduha. Serverska aplikacija podržava istovremen rad više korisnika. Potrebno je omogućiti rad većeg broja ovih aplikacija.

JavaSensorControl je aplikacija koja ima sljedeće opcije:

- pregled svih aktivnih *JavaSensor* servera
- pregled detalja svakog *JavaSensor* servera (broj aktivnih korisnika, port na kojem server sluša, IP adresa)
- gašenje određenog *JavaSensor* servera

JavaSensorControl aplikacija u svom podešavanju ne može imati informacije o serverima, ali serveri mogu znati za *JavaSensorControl* aplikaciju.

Vrijeme rada: 180 minuta