

## Programski jezici 2 – 20.04.2022. –

**1. (30)** U JavaSvijetu organizuje se uvoz vozila. Svako vozilo ima vozača, broj putnika, cijenu, identifikator, motor koji takođe ima identifikator i tip motora (motor može biti dizelski, benzinski, električni ili hibrid). Vozila mogu biti automobili (broj putnika od 1 do 8), kamioni (broj putnika od 1 do 3) i autobusi (broj putnika od 8 do 52). Automobili se mogu uvoziti pojedinačno ili na kamionu. Automobili se uvoze regularno, kamioni i autobusi se uvoze uz posebnu proceduru. Kamioni koji prevoze automobile ne trebaju da se uvoze. Vozači za svako prevozno sredstvo imaju ime i prezime. Svako vozilo treba da se kreće po stazama koje su predstavljene nizom polja. Do graničnog prelaza vodi jedna staza na kojoj je 15 uvodnih polja ali se prije graničnog prelaza (na po tri uzastopna dodatna polja) odvajaju u posebnu traku automobili, a u posebnu traku autobusi i kamioni (uključujući i kamione koji prevoze automobile). Na kraju odvojenih traka nalaze se terminali. Procedura uvoza na terminalima za automobile traje do 1 sekunde, a procedura uvoza za autobuse i kamione do 2 sekunde (sekunda po vozilu koje se uvozi na kamionu). Prilikom uvoza za svako vozilo se računa odgovarajuća taksa i to po sljedećem modelu: Električna vozila ne plaćaju taksu, hibridim se naplaćuje taksa u iznosu od 50 KM, u svim ostalim tipovima automobila taksa iznosi 100KM, za kamione 500KM, a za autobuse po formuli broj\_putnikax20. Simulacija započinje kreiranjem reda vozila (po pet automobila, kamiona i autobusa), koji su raspoređeni na prvih 15 polja. Vozilo može preći na naredno polje samo ako je ono slobodno. Svi podaci za vozilo su slučajno generisani. U sakom trenutku se na konzoli ispisuju trenutni događaji na terminalima sa svim podacima (koji terminal, koje vozilo sa svim podacima vozila, kolika je taksa...). U posebnim odvojenim fajlovima se u relnom vremenu upisuje u tekstualne fajlove ukupna određena taksa za regularnu proceduru i posebnu proceduru, respektivno. U trenutku kada se simulacija završi potrebno je ispisati na konzolu ukupan iznos obračunate takse i broj putnika koji su prošli terminale.

Obavezno napraviti hijerarhiju klasa koja omogućava da ne dođe do dupliranja koda. Ne koristiti enumeracije. Takođe je potrebno voditi računa o performansama izvršavanja.

**2. (20)** Napisati interfejs koji ima metode `Double getValue()`, `String getName()` i `Status getStatus()`. Status je enumeracija sa vrijednostima: `NEW`, `PROCESSING`, `DONE`. Napraviti 3 klase koje implementiraju ovaj interfejs. Nazive klasa odrediti proizvoljno. Napraviti tri niza objekata ovih klasa sa po 10 objekata. Statuse postaviti naizmjenično. Napisati generičku metodu koja u proslijeđenih  $n$  nizova grupiše podatke na način da ostanu samo jedinstveni name atributi, sa sumarnim value vrijednostima. Metoda vraća jedan niz, a prima argument Status po kojem filtrira podatke iz ulaznih nizova. Iskoristiti ovu metodu da sumirate tri niza u jedan. Primjer:

```
A = [{name:'pj2', value: 100, status: NEW}, {name:'java', value: 20, status: PROCESSING},...]
```

```
B = [{name:'ispit', value: 100, status: DONE}, {name: 'pj2', value: 200, status: NEW},...]
```

```
REZ = [{name: 'pj2', value: 300}, {name: 'java', value: 20}, {name: 'ispit', value: 100}, ...]
```

3. (20) Na Moodle-u je dostupan tekstualni fajl *songs.txt* u kom su vrijednosti razdvojene sa ###. Kreirati odgovarajuću klasu *Song* koja sadrži sve potrebne attribute. Korištenjem *stream*-ova obaviti sljedeće operacije i rezultate prikazati na konzoli:

- a. kreirati listu pjesama grupisanih po godini objavljivanja,
- b. kreirati listu pjesama grupisanih po zadatim izvođačima (minimalno dva izvođača),
- c. kreirati listu pjesama čija je dužina trajanja veća od 5 minuta (dužina trajanja je izražena u sekundama),
- d. kreirati listu pjesama koji su objavljeni u drugoj deceniji 2000-tih i
- e. izračunati prosječnu dužinu pjesama objavljenih 80-ih godina.

**Vrijeme za rad: 180 minuta**