

Programski jezici 2

- 24. 04. 2024. –

1. **(30 bodova)** Potrebno je implementirati Java aplikaciju koja simulira sistem za biblioteku. Svaka knjiga u biblioteci predstavlja objekat odgovarajuće klase knjiga. Svaka knjiga ima sljedeće attribute: naslov, autor, broj stranica i godina izdavanja. Postoje tri vrste knjiga koje se mogu naći u biblioteci: beletristika, knjige za djecu i stručna literatura. Beletristika ima dodatni atribut žanr knjige, knjige za djecu imaju preporučenu godinu starosti djeteta, dok stručna literature ima oblast. Knjige za djecu mogu da se preuzmu samo u biblioteci, stručna literature samo *online*, dok se beletristika može preuzeti i *online* i u biblioteci. Aplikacija treba da omogući rad sa korisnicima biblioteke. Svaki korisnik ima svoje ime, prezime, godine starosti, oblast interesovanja i broj članske karte. Korisnici mogu podizati knjige iz biblioteke.

Simulacija podizanja knjiga obavlja se na sljedeći način:

- Kreira se matrica 20x10 (kolona x redova) na koju se na slučajne pozicije postavljaju knjige. Pri postavljanju knjige voditi računa o tome da se ne prepisu jedna preko druge. Broj knjiga koji se postavljaju unosi se kao argument komandne linije prilikom pokretanja simulacije, a minimalno ih mora biti 30. Potrebno je kreirati podjednak broj knjiga različite vrste, npr. ako je broj knjiga proslijeđen kao argument komandne linije 30, potrebno je kreirati 10 knjiga beletristike, 10 knjiga stručne literature i 10 knjiga za djecu. Ukoliko zadati broj nije djeljiv sa 3, kreirati cio broj knjiga, a ostatak prikazati kao nekreirane knjige. Npr. ako korisnik unese 32, kreiraće se 3x10 knjiga i nekreiranih knjiga će biti 2.
- Kreiraju se 10 korisnika i postavljaju na početak reda redoslijedom kojim su kreirani, tako da se u jednom redu kreće jedna osoba i započinju kretanje. Korisnik za jednu sekundu prelazi jedno polje.
- Ukoliko korisnik naiđe na knjigu, prikazuju se podaci o knjizi i korisniku, a nakon toga provjerava se tip knjige. Korisnik može da podigne knjigu ukoliko se njegova oblast interesovanja poklapa sa žanrom ili oblašću, dok se u slučaju nailaska na knjigu za djecu provjerava broj godina starosti. Slučaj da je korisnik mlađi ili stariji od preporučene starosti kod knjiga za djecu obraditi korisnički definisanim izuzetkom. Ukoliko je korisnik podigao knjigu, u tekstualni fajl imenovan kao *BIBLIOTEKA-trenutno_vrijeme_u_milisekundama.txt* upisuju se podaci o korisniku – broj članske karte i knjizi – naslov (pod navodnicima), u jednom redu, razdvojeni znakom "#", npr. 45678#"Alisa u Zemlji čuda" i na mjesto u matrici na kom se nalazila knjiga postavlja se *null*.
- Simulacija završava kada svi korisnici dođu do kraja svog reda matrice. Po završetku simulacije se prikazuje statistički podatak o tome koliki je procenat podignutih knjiga i u tekstualni fajl se upisuje datum simulacije, vrijeme trajanja simulacije i navedeni statistički podatak.
- Nakon upisa u fajl prikazuje se i sadržaj fajla *BIBLIOTEKA-trenutno_vrijeme_u_milisekundama.txt*.

2. **(20 bodova)** Napisati program koji za uneseni broj n kreira n direktorijuma. Broj n i putanja do direktorijuma u kojem se nalaze direktorijumi su argumenti komandne linije. Naziv svakog od kreiranih direktorijuma je redni broj direktorijuma prilikom kreiranja. U sklopu istog programa implementirati mehanizam za praćenje promjena u svim kreiranim direktorijumima. Pod promjenama koje treba pratiti podrazumjeva se dodavanje tekstualnih datoteka u neki od direktorijuma ili u više direktorijuma odjednom. Prilikom dodavanja tekstualne datoteke u direktorijum za svaku od datoteka na konzolu se ispisuje putanja do datoteke i broj samoglasnika (pojedinačno po samoglasniku), te ukupan broj samoglasnika za sve datoteke koje su do tog trenutka dodavane. U sklopu projekta sačuvati i jednu tekstualnu datoteku sa kojom je vršeno testiranje tokom pisanja zadatka te primjer poziva programa (argumenti komandne linije).

3. **(20 bodova)** Napisati jednostavan program koji rekurzivno pronalazi skrivene fajlove u određenim folderima. Tokom pretrage u tekstualni fajl se upisuju apsolutne putanje svih pronađenih skrivenih fajlova u hijerarhiji traženog foldera. Tekstualni fajl se čuva u (*root*) folderu koji se pretražuje. Na kraju fajla upisuje se ukupan broj pronađenih skrivenih fajlova, ukupan broj svih fajlova i procenat skrivenih fajlova u odnosu na ukupan broj. Napraviti JAR fajl koji se koristi za pokretanje programa, pri čemu se kao argumenti prosljeđuje putanja. U komentaru ostaviti naredbe koje se koriste za kreiranje i pokretanje tog JAR fajla.

Vrijeme za rad: 180 minuta