

Programski jezici 2

- 11.09.2024. -

- (30 bodova)** Napisati jednostavnu simulaciju rada na jednom projektu po *Scrum* metodologiji za određenu kompaniju. Projekat se sastoji od 3 sprinta, a tim ima 5 članova. Za svakog člana čuva se ime, prezime i godine rada u kompaniji. Ovi podaci se mogu *hard* kodovati prilikom instanciranja objekata. Svaki član tima ima odgovarajuću ulogu: 1 je *Product Owner*, 1 *Scrum master* i 3 člana su *developer-i*. Članovi trebaju biti niti. Za *Product Owner-a* čuva se opis. Simulacija se odvija na sljedeći način:

- Na početku svakog sprinta *Scrum Master* i *Product owner* kreiraju po 5 zadataka koji se smještaju u određenu kolekciju (*backlog*) po prioritetu, tako da se zadaci sa najvišim prioritetom prvi preuzimaju iz kolekcije. Svaki zadatak ima polja: naslov, opis, prioritet (1-5) i vrstu (enumeracija: *task, story, bug*).
- Developer-i* rade istovremeno i preuzimaju zadatak po zadatak iz kolekcije i rade na njima, dok se ne isprazni kolekcija Trajanje rada na svakom zadatku je 1 - 3 s (odrediti na slučajan način).
- Na kraju sprinta *Scrum master* u svojoj metodi kreira *txt* fajl sa imenom *sprint_#.txt* (# je broj sprinta) i upisuje sve podatke o svakom zadatku, ime i prezime *developer-a* koji je radio na tom zadatku i trajanje rada na tom zadatku (jedan red u fajlu je jedan zadatak).

Nakon završetka jednog sprinta, počinje sljedeći. Nakon završetka posljednjeg sprinta na konzolu se ispisuje ukupno vrijeme trajanja svakog sprinta i spisak svih *developer-a* (ime, prezime) i sa ukupnim brojem urađenih zadataka tokom svih sprintova. Po potrebi modifikovati osnovne klase kako bi se mogli izračunati traženi podaci, a dozvoljeno je i definisanje novih klasa. Na konzoli ispisati samo one poruke koje će olakšati praćenje i pregled simulacije.

- (20 bodova)** Napraviti klasu *Film* koja sadrži atribute: naziv filma, žanr i godina objavlјivanja. Napisati aplikaciju koja na samom početku učitava podatke o filmovima iz CSV fajla dostupnog na *Moodle* stranici predmeta. Aplikacija u prvoj fazi kreira stablo direktorijuma (svi listovi stabla su na dubini 5), a na svakom nivou se nalaze od 2 do 4 direktorijuma. Naziv direktorijuma je u formatu **redni_broj_dubine_redni_broj_u_nivou_naziv_filma**. U svakom direktorijumu se čuva po jedan film iz učitane CSV datoteke. Naziv datoteke je **naziv_filma.txt**, a podaci o filmu se čuvaju tako što se upisuje jedan atribut u jednu liniju fajla. Nakon prve faze, korisnik na konzoli ima mogućnost pretrage po nekom od polja (naziv filma, žanr ili godina objavlјivanja). Pretraga je **Like pretraga** (dio teksta može da se poklapa) za svako polje. Ako se više polja koristi u pretrazi, oni se kombinuju sa logičkim **AND**. Aplikacija bi nakon pretrage trebala da ispiše listu sa putanjom do fajlova u kojima su zapisani filmovi koji ispunjavaju date uslove. Korisnik putem argumenata komandne linije unosi putanju na kojoj se kreira stablo direktorijuma.
- (20 bodova)** Koristeći poznate klase iz Java kolekcija kreirati klasu koja predstavlja stek, u sklopu koje je atribut klasa Knjiga (atributi: naslov, autor, broj stranica, godina izdanja). Potrebno je implementirati dvije niti:
 - Nit za dodavanje knjige u stek.
 - Nit za uklanjanje knjige iz steka.

Obe niti treba da mogu raditi konkurentno, tj. istovremeno, uz pravilno sinhronizovano upravljanje deljenim resursima. Nakon toga pokrenuti simulaciju dodavanja i uklanjanja knjiga iz steka, pri čemu:

- Nit za dodavanje pravi pauzu od 250ms a nit za brisanje 400ms nakon svake operacije.
- Podaci o knjigama koje se dodaju u stek treba da budu nasumično generisani.

- Simulacija traje tačno jednu minutu.

Specifikacije:

- Ako je stek prazan, nit za uklanjanje ne treba da čeka, već će preskočiti operaciju uklanjanja.
- U svakom trenutku treba ispisati na konzolu događaj dodavanja ili uklanjanja knjige sa steka.
- Nakon završetka svih operacija, potrebno je u tekstualnu datoteku ispisati sve knjige koje su ostale na steku.

Vrijeme za rad: 180 minuta