

PROGRAMSKI JEZICI 2 (2246) – PISMENI ISPIT (27.04.2017.)

1. **(30)** Turističke atrakcije mogu se podijeliti na besplatne, one koje se plaćaju i one koje se plaćaju na bazi dobrovoljnog priloga. Svaku turističku atrakciju karakteriše naziv i lokacija na kojoj se nalazi. Istorijski spomenik je turistička atrakcija koja je besplatna i koja sadrži još i opis. One atrakcije koje se plaćaju imaju cijenu ulaznice koja se generiše slučajno prilikom kreiranja atrakcije u opsegu od 10 do 80. Muzej je turistička atrakcija koja može biti i besplatna i plaćati se, u zavisnosti od toga da li je parni ili neparni dan u sedmici. Pored toga, muzej sadrži i fajl koji predstavlja letak sa osnovnim informacijama o muzeju i aktuelnim izložbama. Zabavni park je atrakcija koja se plaća. Crkva je turistička atrakcija kojoj se posjeta plaća na osnovu dobrovoljnog priloga i sadrži atribut koji predstavlja iznos prikupljenih dobrovoljnih priloga. Turista je osoba koja obilazi turističke atrakcije. Turista ima ime, novac, način kretanja (enum, koji može da bude SAMO_U_JEDNOM_REDU, DIJAGONALNO ili KROZ_CIJELU_MATRICU), folder letaka i broj posjećenih mjesta. Prilikom kreiranja turista, slučajno mu se podešava ime, novac, bira se slučajno način kretanja i kreira se folder sa imenom turista. U gradu predstavljenom matricom 50x50 postavljeno je po 20 turističkih atrakcija od svakog tipa. Turisti se takmiče u obilascima turističkih atrakcija u gradu. Takmičenje započinje tako što se na matricu grada na slučajne pozicije postavlja 50 turista koji nastavljaju kretanje po matrici na svoj način. Ukoliko prilikom kretanja naiđu na atrakciju koja ima letak, leci se smještaju u folder turista. U slučaju da se atrakcija plaća, turisti se umanjuje odgovarajući iznos novca. U slučaju da turista više nema novca, prekida kretanje. Za svaku posjećenu atrakciju, turisti se uvećava broj posjećenih atrakcija. Pri nailasku na atrakciju, na konzoli se ispisuju sve informacije o posjećenoj atrakciji, kao i ko ju je posjetio. Kada svi turisti „prešetaju“ matricu do kraja, u zavisnosti od njihovog načina kretanja, takmičenje se završava. Po završetku takmičenja, turisti se prikazuju na komandnoj liniji sortirani po broju sakupljenih letaka. Pored toga, uz svakog turistu se prikazuje koliko je atrakcija obišao od postojećih 80 u procentima.
2. **(25)** Napisati klasu *Stack* koja predstavlja implementaciju steka koji radi sa elementima predstavljenim cjelobrojnim vrijednostima. Ova klasa ima dva atributa: *elements* i *topElementIndex*, gdje *elements* predstavlja niz u kojem se čuvaju elementi steka, dok *topElementIndex* predstavlja indeksnu poziciju vrha steka. Kapacitet steka se zadaje kao argument konstruktora. Klasa posjeduje dvije metode *push* i *pop*, za smještanje i skidanje elemenata sa steka, respektivno. Pored toga, klasa posjeduje i dvije pomoćne metode *isFull* i *isEmpty*, koje provjeravaju da li je stek pun ili prazan, respektivno. Elementi sa steka se mogu skidati, ako stek nije prazan. Elementi se mogu stavljati na stek ako on nije pun. U slučaj praznog ili punog steka potrebno je baciti korisnički-implementiran *StackException*. Neophodno je obezbijediti sinhronizovan pristup steku. *PusherThread* i *PopperThread* koje su niti zadužene za stavljanje i skidanje elemenata sa steka, respektivno. Ove klase imaju po 2 atributa: *stack* (tipa *Stack*) i *number* (tipa *int*), gdje *stack* predstavlja stek s kojim niti rade, a *number* broj elemenata koje *PusherThread* i *PopperThread* niti trebaju smjestiti na stek, odnosno skinuti sa steka, respektivno. Napisati klasu *StackMain* u čijoj će *main* metodi biti instanciran stek proizvoljnog kapaciteta, te niti *PusherThread* i *PopperThread* koje će na stek stavljati i sa steka skidati zadati broj elemenata. Elementi koji se smještaju na stek generišu se proizvoljno u opsegu od 10 do 99. Simulacija punjenja/pražnjenja steka traje jedan minut, pri čemu se punjenje i pražnjenje pokreću naizmjenično. U slučaju da po završetku simulacije stek nije

prazan, objekat steka je potrebno serijalizovati i smjestiti u fajl *stack.ser*. Na početku svake simulacije potrebno je provjeriti da li postoji navedeni fajl i u slučaju da postoji, deserijalizovati stek i njega koristiti u simulaciji.

3. **(15)** Na Moodle-u je postavljen CSV fajl ***quotes*** u kom su podaci razvojeni separatorom **|** umjesto zarezom. Prva kolona predstavlja citat, druga kolona broj komentara vezanih za dati citat, dok treća kolona predstavlja link ka citatu. Napisati jednostavnu konzolnu aplikaciju u kojoj je korisnicima omogućena pretraga citata po ključnoj riječi, kao i prikaz citata dana. U prvom slučaju, korisnik unosi pojam koji želi pretražiti. Citati koji u sebi sadrže unesene ključne riječi se prikazuju na konzoli, sortirani po broju komentara od najviše ka najmanje. Citat dana se prikazuje po pokretanju aplikacije na konzoli i bira se na slučajan način iz podskupa citata koji imaju paran broj komentara.

Napomena: Vrijeme trajanja ispita je 180 minuta. Nakon završenog ispita, zadatke je potrebno *upload*-ovati na *Moodle*, arhivirane u formatu ***broj_indeksa_ime_i_prezime***.