

## Programski jezici - JAVA kolokvijum - mart 2007.

Napisati JAVA aplikaciju za podršku izvođenja testova studenata uz pomoć računara. Potrebno je podržati sledeće vrste pitalica:

- da-ne pitalice — pitanja koja mogu biti tačna ili ne; Npr.

Java omogućava višestruko nasledjivanje implementacije. DA NE

- abcd pitalice — pitanja sa ponuđenim odgovorima; Npr.

Klase u Javi imaju:

- a) Kopi-konstruktor
- b) Konstruktor
- c) Destruktor
- d) Operator dodele

- pitalice sa dopunjavanjem — pitanja u kojima je potrebno dopuniti nedostajuću reč; Npr.

Ključna rec ----- označava nasledjivanje.

Student na svaku pitalicu daje odgovor ili označava da odgovor ne zna.

### I deo

1. Definirati apstraktnu klasu **Pitalica** koja predstavlja baznu klasu za sve konkretne vrste pitalica. Definirati klase **DaNePitalica**, **ABCDPitalica** i **PitalicaSaDopunjavanjem** koje predstavljaju navedene vrste pitalica i koje pamte postavku zadatka i odgovor koji je student dao. Klase snabdeti odgovarajućim konstruktorima i **get**, odnosno **set** metodima. [5]
2. Definirati metod **postavi** kojim se pitalica postavlja i očitava odgovor studenta. [10]
3. Definirati metod **tacnoOdgovoreno** kojim se proverava da li je student dao tačan odgovor i metod **brojPoena** koji izračunava broj poena dobijenih na nekoj pitalici. Za tačan odgovor student dobija dva poena, za odgovor "ne znam" ne dobija poene, a za netačan odgovor gubi jedan poen. Napisati JUnit TestCase koji proverava metod **brojPoena**. [10]
4. U klasi **ABCDPitalica** dodati mogućnost "mešanja" redosleda ponuđenih odgovora kako bi se smanjila mogućnost prepisivanja. [5]
5. Kroz hijerarhiju pitalica definirati metode za učitavanje iz toka (pogodan format ulaza samostalno definirati). Napisati JUnit TestCase koji proverava ispravnost ovih metoda za učitavanje. [15]
6. Napisati klasu **Test** koja predstavlja kolekciju pitalica. Klasa mora da podrži učitavanje pitanja iz datoteke, vršenje testiranja postavljanjem svih pitanja (izmešati redosled pitanja na testu i redosled ponuđenih odgovora u ABCD pitalicama) i prikupljanjem odgovora studenta, kao i prikazivanje izveštaja o rezultatima testiranja. Priložiti datoteku koja sadrži primere sa početka formulacije ovog zadatka. [15]

### II deo

Napraviti grafičku aplikaciju koja vrši testiranje i nakon obavljenog testiranja prikazuje izveštaj o rezultatima testiranja.

1. Omogućiti da se na početku testiranja unese ime studenta koji se testira i ime datoteke koja sadrži test. Dodati meni **File|Open** koji omogućava da se ovo ime datoteke odabere korišćenjem standardnog dijaloga. [8]
2. Omogućiti da se na da-ne pitalice odgovara pritiskanjem dugmeta **Da**, odnosno **Ne**. [7]
3. Omogućiti da se pitalice sa dopunjavanjem prikazuju tako da se na mestu reči koje treba dopuniti postavi polje za unos teksta u koju student upisuje odgovor. [7]
4. Omogućiti da se ABCD pitalice prikazuju tako da se odgovor odabira korišćenjem radio dugmića. Voditi računa o rasporedu komponenti na prozoru. [8]
5. Omogućiti da se nakon popunjavanja svih pitanja iz učitanoog testa korisniku prikaže ukupan broj poena kao i tabela koja sadrži izveštaj o proteklom testiranju, odnosno da se uz tekst svakog pitanja prikaže da li je dati odgovor bio tačan, broj osvojenih poena i ako odgovor nije bio tačan, koji je bio tačan odgovor. [10]