

Prijemni ispit – Master 2017
Studijski program: Softversko inženjerstvo i računarske nauke

Šifra zadatka

| | | | | | |
|---|---|---|---|---|---|
| 4 | 1 | 7 | 7 | 1 | 2 |
|---|---|---|---|---|---|

| | |
|-----------|--|
| 1. | <p>Pogledati kod koji je dat ispod. Koja od narednih izjava je tačna:</p> <pre>class Osoba { } class OSobA { }</pre> |
| a. | Naziv OSobA nije dozvoljen u Java programskom jeziku |
| b. | Naziv Osoba nije dozvoljen u Java programskom jeziku |
| c. | Nazive Osoba i OSobA Java posmatra kao različite nazive |
| d. | Nazive Osoba i OSobA Java posmatra kao iste nazive |
| e. | Klase Osoba i OSobA su javne klase |
| f. | Ne znam |
| 2. | <p>Pogledati kod koji je dat ispod (broj linije koda je dat uz levu ivicu). Kod ne može da se kompajlira. Koje linije koda SADRŽE sintaksne greške?</p> <pre>1 class Student { 2 STRING ime = "Pera Peric"; 3 double prosek = 8.83 4 int godinaRodjenja = 1993; 5 char pol = "M"; 6 }</pre> |
| a. | 2,3,4,5 |
| b. | 1,2,3 |
| c. | 2,5 |
| d. | 1,2 |
| e. | 2,3,5 |
| f. | Ne znam |
| 3. | <p>Pogledati kod koji je dat ispod. Koliko objekata klase Automobil je inicijalizovano u main metodi?</p> <pre>class Automobil { } class Test { public static void main(String args[]) { Automobil a1 = new Automobil(); Automobil a2 = new Automobil(); Automobil a3; } }</pre> |
| a. | 1 |
| b. | 2 |
| c. | 3 |
| d. | 4 |
| e. | 0 |
| f. | Ne znam |
| 4. | <p>Pogledati kod koji je dat ispod. Koja od narednih izjava je tačna:</p> <pre>class Metode { void vratiApsolutnuNulu () { return -273.15; } }</pre> |
| a. | Metoda vraća -273.15 kao povratnu vrednost |
| b. | Metoda ima jedan parametar |
| c. | Metoda ima dva parametra |
| d. | Tip povratne vrednosti metode je realan broj |
| e. | Kod ne može da se kompajlira |
| f. | Ne znam |

| | |
|----|--|
| 5. | <p>Pogledati kod koji je dat ispod. Šta će metoda vratiti kao povratnu vrednost ako joj se za visinu prosledi 2.0 (dva metra) a za težinu 100.0 (kilograma)?</p> <pre>class BMI { double izracunajBMI (double visina, double tezina){ double rezultat = tezina/visina*visina; return rezultat; } }</pre> |
| a. | Metoda ništa ne vraća kao povratnu vrednost |
| b. | 0.0 |
| c. | 25.0 |
| d. | 100.0 |
| e. | Java prijavljuje grešku |
| f. | Ne znam |
| 6. | <p>Pogledati kod koji je dat ispod. Šta će se desiti kad se pokrene main metoda?</p> <pre>class Osoba { String ime; } class Test { public static void main(String[] args) { Osoba o1 = new Osoba(); Osoba o2 = o1; o2.ime = "Pera"; o1.ime = "Mika"; System.out.println(o2.ime); } }</pre> |
| a. | Na ekranu će se ispisati "Pera" |
| b. | Na ekranu će se ispisati "Mika" |
| c. | Na ekranu će se ispisati "Pera Mika" |
| d. | Na ekranu će se ispisati "Mika Pera" |
| e. | Java prijavljuje grešku |
| f. | Ne znam |
| 7. | <p>Pogledati kod koji je dat ispod. Šta će metoda vratiti kao povratnu vrednost ako joj se kao argument prosledi vrednost 300?</p> <pre>class Provera { boolean proveraIntervala (int a){ if (a < 100 a > 500) return true; else return false; } }</pre> |
| a. | Metoda ništa ne vraća kao povratnu vrednost |
| b. | true |
| c. | false |
| d. | a |
| e. | 0 |
| f. | Ne znam |
| 8. | <p>Pogledati kod koji je dat ispod. Šta će metoda ispisati na ekranu ako se pozove?</p> <pre>class Petlje { void ispisi(){ for (int i=0;i<10;i++) if (i%2 == 0) System.out.println(i); } }</pre> |
| a. | ništa |
| b. | 0 1 2 3 4 5 6 7 8 9 |
| c. | 0 2 4 6 8 |
| d. | 0 2 4 6 8 10 |

| | |
|-----|---|
| e. | beskonačna petlja |
| f. | Ne znam |
| 9. | <p>Pogledati kod koji je dat ispod. Šta će metoda ispisati na ekranu ako se pozove?</p> <pre> class Petlje { void ispisi(){ for (int i=1; i<=3; i++){ System.out.print("["); for (int j=5; j>i; j--) { System.out.print(j); break; } System.out.print("]"); } } } </pre> |
| a. | [5] [5] [5] |
| b. | [5] |
| c. | [5] [4] [3] |
| d. | [5432] [543] [54] |
| e. | [] [] [] |
| f. | Ne znam |
| 10. | <p>Pogledati kod koji je dat ispod. U koliko ciklusa će se izvršiti while petlja?</p> <pre> class Petlje { void izvrsi () { int i = 5; while (i > 0) { i = i - 1; continue; } } } </pre> |
| a. | 5 |
| b. | 4 |
| c. | 3 |
| d. | 1 |
| e. | beskonačna petlja |
| f. | Ne znam |
| 11. | <p>Pogledati kod koji je dat ispod. Šta će metoda vratiti kao povratnu vrednost ako se pozove sa argumentom 55 (kao n)?</p> <pre> class Petlje { int vrati (int n){ int broj = 1; while (broj < n){ broj = broj * 3; } return broj; } } </pre> |
| a. | ništa (beskonačna petlja) |
| b. | 1 |
| c. | 3 |
| d. | 165 |
| e. | 81 |
| f. | Ne znam |

| | |
|-------------------------------------|--|
| | Pogledati kod koji je dat ispod. Šta će se ispisati na ekranu kad se pozove main metoda? |
| | <pre>interface GFigura { public void ispisi(); } class Trougao implements GFigura{ public void ispisi(){System.out.println("trougao");} } class PravougliTrougao extends Trougao implements GFigura{ public void ispisi(){System.out.println("pravougli trougao");} } class Test{ public static void main(String[] args){ GFigura gf = new PravougliTrougao(); gf.ispisi(); } }</pre> |
| 12. | |
| a. | ništa |
| <input checked="" type="radio"/> b. | pravougli trougao |
| c. | trougao |
| d. | trougao pravougli trougao |
| e. | pravougli trougao trougao |
| f. | Ne znam |
| 13. | Zahtevi se kod Larmanove metode razvoja softvera opisuju pomoću: |
| <input checked="" type="radio"/> a. | modela slučaja korišćenja |
| b. | relacionog modela |
| c. | sekvencnih dijagrama |
| d. | dijagrama klasa |
| e. | dijagrama prelaza stanja |
| f. | Ne znam |
| 14. | Iz kojih sekcija se sastoje ugovori kod sistemskih operacija: |
| a. | Operacija, Veza sa SK, Vrednosna ograničenja, Strukturna ograničenja |
| b. | Operacija, Aktori, Lista argumenata, Telo operacije |
| <input checked="" type="radio"/> c. | Operacija, Veza sa SK, Preduslovi, Postuslovi |
| d. | Operacija, Aktori, Preduslovi, Postuslovi |
| e. | Operacija, Veza sa SK, Lista argumenata, Telo operacije |
| f. | Ne znam |
| 15. | Kako glasi princip odvajanja modela od pogleda (Model-View Separation Principle) |
| a. | Aplikaciona logika (model) zna gde se nalazi korisnički interfejs (pogled). |
| <input checked="" type="radio"/> b. | Aplikaciona logika (model) nema znanja o tome gde se nalazi korisnički interfejs (pogled). |
| c. | Aplikaciona logika (model) poziva korisnički interfejs (pogled) preko poslovne logike. |
| d. | Aplikaciona logika (model) se nalazi na korisničkom interfejsu (pogledu). |
| e. | Aplikaciona logika (model) kod arhitekture softverskog sistema se nalazi na sloju ispod korisničkog interfejsa (pogleda). |
| f. | Ne znam |
| 16. | Perzistentni okvir je: |
| a. | skup interfejsa i klasa koji omogućava različitim klasama da imaju svoja pojavljivanja. |
| b. | skup interfejsa i modula koji omogućava perzistentnost grafičkim objektima. |
| c. | skup interfejsa i klasa koji omogućava perzistentnost tabelama iz baze podataka. |
| d. | skup interfejsa i modula koji omogućava perzistentnost transijentnim objektima. |
| <input checked="" type="radio"/> e. | skup interfejsa i klasa koji omogućava perzistentnost objektima različitih klasa. |
| f. | Ne znam |
| 17. | Kako glasi pravilo nezavisnosti scenarija: |
| <input checked="" type="radio"/> a. | Scenarija slučaja korišćenja ne treba da budu u međusobnoj interakciji. |
| b. | Scenarija slučaja korišćenja treba da budu u međusobnoj interakciji. |
| c. | Scenarija slučaja korišćenja treba povezati sa sistemskim operacijama. |
| d. | Scenarija slučaja korišćenja ne treba da budu povezana sa bazom podataka. |
| e. | Scenarija slučaja korišćenja treba da budu povezana sa poslovnom logikom. |
| f. | Ne znam |

| | |
|------------|---|
| 18. | Strukturalna ograničenja su definisana preko: |
| a. | kardinalnosti preslikavanja između sekvencnih dijagrama. |
| b. | kardinalnosti preslikavanja između slučaja korišćenja. |
| c. | kardinalnosti preslikavanja između domenskih klasa (tabela). |
| d. | kardinalnosti preslikavanja između sistemskih operacija. |
| e. | kardinalnosti preslikavanja između vrednosnih ograničenja. |
| f. | Ne znam |
| 19. | Uloga kod asocijacije sadrži: |
| a. | ime, vrednost i operator. |
| b. | ime, preslikavanje i navigaciju. |
| c. | ime, preslikavanje i operator. |
| d. | ime, operand i operator. |
| e. | ime, vrednost i vezu. |
| f. | Ne znam |
| 20. | Soketi omogućavaju da se: |
| a. | povežu programi koji nisu povezani sa bazom podataka. |
| b. | više klijentskih programa poveže sa jednim serverskim programom. |
| c. | jedan program konkurentno izvrši sa drugim programom. |
| d. | klijentski program neposredno poveže sa drugim klijentskim programima. |
| e. | povežu programi sa arhitekturom softverskog sistema. |
| f. | Ne znam |
| 21. | U Javi se nit može napraviti na 2 načina: |
| a. | Realizacijom interfejsa Runnable i proširenjem klase Thread. |
| b. | Realizacijom klase Runnable i proširenjem interfejsa Thread. |
| c. | Realizacijom interfejsa Class i proširenjem klase Thread. |
| d. | Realizacijom interfejsa Execute i proširenjem klase Thread. |
| e. | Realizacijom klase Runnable i proširenjem klase Class. |
| f. | Ne znam |
| 22. | Nit predstavlja: |
| a. | deo klase koja može istovremeno da se izvršava sa drugim klasama. |
| b. | deo programa koji ne može istovremeno da se izvršava sa drugim nitima istog programa. |
| c. | deo programa koji ne može istovremeno da se izvršava sa drugim programima. |
| d. | deo programa koji može istovremeno da se izvršava sa drugim nitima istog programa. |
| e. | deo klase koji omogućava objektima da se istovremeno izvršavaju. |
| f. | Ne znam |
| 23. | Učitavanje upravljačkog programa u Java program se radi preko sledeće naredbe: |
| a. | <code>import java.sql.*;</code> |
| b. | <code>String dbUrl = "jdbc:mysql://127.0.0.1:3306/student";</code> |
| c. | <code>Connection c = DriverManager.getConnection(dbUrl, user, pass);</code> |
| d. | <code>Class.forName("com.mysql.jdbc.Driver");</code> |
| e. | <code>String upit = "SELECT brind, ime, prezime FROM Student";</code> |
| f. | Ne znam |
| 24. | Šta opisuje faza analize u razvoju softverskog sistema? |
| a. | Faza analize opisuje arhitekturu softverskog sistema. |
| b. | Faza analize opisuje ponašanje softverskog sistema. |
| c. | Faza analize opisuje strukturu softverskog sistema. |
| d. | Faza analize opisuje slučajeve korišćenja. |
| e. | Faza analize opisuje logičku strukturu i ponašanje softverskog sistema. |
| f. | Ne znam |

Primer 1. (na ovaj primer odnose se pitanja 25-30) Sledećim naredbama kreirane su dve tabele:

```
CREATE TABLE MESTO(
    Id                int not null primary key,
    Naziv             nvarchar(60) not null,
    BrStanovnika      int null,
    PttBroj           int not null unique);
```

| | |
|-----|--|
| 25. | <p>Upit:</p> <pre>SELECT AVG(convert(float, O.Visina)) FROM OSOBA O LEFT JOIN MESTO as M on (O.IdMestaPreb = M.Id) WHERE O.Visina <= 171 AND O.IdMestaRod != (SELECT Id FROM MESTO WHERE Naziv LIKE 'Beograd');</pre> <p>prikazuje:</p> |
| a. | 166 |
| b. | 167 |
| c. | 167,666666666667 |
| d. | 168 |
| e. | Srednju visinu osoba koje su rođene u Beogradu |
| f. | Ne znam |
| 26. | <p>Upit:</p> <pre>SELECT Ime --, Prezime, Visina, GodRod FROM OSOBA WHERE Prezime NOT LIKE '%ski' AND GodRod LIKE '_ _[^5-7]_';</pre> <p>Prikazuje sledeća imena:</p> |
| a. | Error: Incorrect syntax near 'GodRod' |
| b. | Jana |
| c. | Ništa |
| d. | Stanko, Branko, Janko, Dragan, Milan, Boban, Milica, Jana, Jovana, Brankica |
| e. | Jana, Vanja |
| f. | Ne znam |
| 27. | <p>Upit:</p> <pre>SELECT 2 * (Year(GetDate())) - SUM(R.GodRod) FROM OSOBA R JOIN OSOBA P on (R.IdMestaPreb = P.IdMestaRod) WHERE R.Id = P.Id;</pre> <p>prikazuje:</p> |
| a. | 351 |
| b. | Error: Incorrect syntax near 'Year(GetDate())' |
| c. | 100 |
| d. | 2017 |
| e. | Nijedan od prethodnih odgovora |
| f. | Ne znam |
| 28. | <p>Upit:</p> <pre>SELECT COUNT(*), AVG(Visina), Pol FROM OSOBA WHERE IdMestaPreb IN (SELECT Id FROM MESTO EXCEPT SELECT Id FROM MESTO WHERE Naziv LIKE 'Smederevo') GROUP BY Pol;</pre> <p>prikazuje:</p> |
| a. | Error: Incorrect syntax near 'GROUP BY Pol' |
| b. | Ukupan broj, prosečnu visinu i pol Smederevaca/ki (Smederevci/ke su sve osobe koje su rođene u Smederevu) |
| c. | Koliko ima osoba i kolika im je prosečna visina, razvrstanih po polu, a koji prebivaju u Smederevu |
| d. | Koliko ima osoba i kolika im je prosečna visina, razvrstanih po polu, a koji ne prebivaju u Smederevu |
| e. | Nijedan od prethodnih odgovora |
| f. | Ne znam |

| | |
|-----|---|
| 29. | <p>Upit:</p> <pre>SELECT Ime, Prezime, GodRod FROM OSOBA as O WHERE 'Beograd' = (SELECT M.Naziv FROM MESTO as M WHERE M.Id = O.IdMestaRod) AND GodRod LIKE '[1]_[^7]%' ;</pre> <p>prikazuje:</p> |
| a. | Error: Incorrect syntax near 'GodRod' |
| b. | Tri reda |
| c. | Četiri reda |
| d. | Dva reda |
| e. | Nijedan red |
| f. | Ne znam |
| 30. | <p>Upit:</p> <pre>SELECT O.Id, Ime, Prezime, Naziv, GodRod FROM OSOBA O JOIN MESTO M ON O.IdMestaPreb = M.Id WHERE IdMestaPreb = (SELECT Id FROM MESTO WHERE Naziv LIKE 'Beograd') AND GodRod NOT IN (SELECT DISTINCT GodRod FROM OSOBA WHERE IdMestaRod = (SELECT Id FROM MESTO WHERE Naziv LIKE 'Smederevo')));</pre> <p>prikazuje:</p> |
| a. | Identifikator osobe, ime, prezime, mesto prebivanja (življenja) i godina rođenja osoba koje ŽIVE u Smederevu, a ROĐENE su u onim godinama u kojima su rođeni Beograđani/ke (Beograđani/ke su sve osobe koje su rođene u Beogradu) |
| b. | Identifikator osobe, ime, prezime, mesto prebivanja (življenja) i godina rođenja osoba koje ŽIVE u Beogradu, a NISU ROĐENE su u onim godinama u kojima su rođeni Smederevci/ke (Smederevci/ke su sve osobe koje su rođene u Smederevu) |
| c. | Identifikator osobe, ime, prezime, mesto prebivanja (življenja) i godina rođenja osoba koje su ROĐENE u onim godinama u kojima su rođeni Smederevci/ke (Smederevci/ke su sve osobe koje su rođene u Smederevu) koji imaju različite (DISTINCT) godine od onih koji ŽIVE u Beogradu |
| d. | Error: Incorrect syntax near 'MESTO' |
| e. | Nijedan od prethodnih odgovora |
| f. | Ne znam |