Prijemni ispit – Master 2019, 2. konkursni rok
Studijski program: Softversko inženjerstvo i računarske nauke

Šifra zadataka | 7 | 7 | 2 | 8 | 4 | 6 |

1.	***Pogledati kod koji je dat ispod. Šta će metoda ispisati na ekranu ako se pozove?***
	`class Petlje {`
	`  void ispisi(){`
	`  for (int i=0;i<10;i++)`
	`  if ( i%2 == 0)`
	`  System.out.println(i);`
	`  }`
	`}`

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*
n.	*Ne znam*

2.	***Pogledati kod koji je dat ispod. Šta će metoda ispisati na ekranu ako se pozove?***
	`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("]");`
	`  }`
	`  }`
	`}`

a.	*[5] [5] [5]*
b.	*[5*
c.	*[5] [4] [3]*
d.	*[5432] [543] [54]*
e.	*[ ] [ ] [ ]*
n.	*Ne znam*

3.	***Pogledati kod koji je dat ispod. U koliko ciklusa će se izvršiti while petlja?***
	`class Petlje {`
	`  void izvrsi (){`
	`  int i = 5;`
	`  while (i > 0){`
	`  i = i - 1;`
	`  continue;`
	`  }`
	`  }`
	`}`

a.	*5*
b.	*4*
c.	*3*
d.	*1*
e.	*beskonačna petlja*
n.	*Ne znam*

4.	***Pogledati kod koji je dat ispod. Šta će metoda vratiti kao povratnu vrednost ako se pozove sa argumentom 55 (kao n)?***
	`class Petlje {`
	`  int vrati (int n){`
	`  int broj = 1;`
	`  while (broj < n){`
	`  broj = broj * 3;`
	`  }`
	`  return broj;`
	`  }`
	`}`

a.	*ništa (beskonačna petlja)*
b.	*1*
c.	*3*
d.	*165*
e.	*81*
n.	*Ne znam*

5.	***Pogledati kod koji je dat ispod. Šta će se ispisati na ekranu kad se pozove main metoda?***
	`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();`
	`  }}`

a.	*ništa*
b.	*pravougli trougao*
c.	*trougao*
d.	*trougao pravougli trougao*
e.	*pravougli trougao trougao*
n.	*Ne znam*
6.	***Zahtevi se kod Larmanove metode razvoja softvera opisuju pomoću:***
a.	*modela slučaja korišćenja*
b.	*relacionog modela*
c.	*sekvencnih dijagrama*
d.	*dijagrama klasa*
e.	*dijagrama prelaza stanja*
n.	*Ne znam*
7.	***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*
c.	*Operacija, Veza sa SK, Preduslovi, Postuslovi*
d.	*Operacija, Aktori, Preduslovi, Postuslovi*
e.	*Operacija, Veza sa SK, Lista argumenata, Telo operacije*
n.	*Ne znam*
8.	***Kako glasi princip odvajanja modela od pogleda (Model-View Separation Principle):***
a.	*Aplikaciona logika (model) zna gde se nalazi korisnički interfejs (pogled).*
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).*
n.	*Ne znam*

9.	***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.*
e.	*skup interfejsa i klasa koji omogućava perzistentnost objektima različitih klasa.*
n.	*Ne znam*
10.	***Kako glasi pravilo nezavisnosti scenarija:***
a.	*Scenarija slučaja korišenja 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.*
n.	*Ne znam*
11.	***Strukturna 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.*
n.	*Ne znam*
12.	***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.*
n.	*Ne znam*
13.	***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.*
n.	*Ne znam*
14.	***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.*
n.	*Ne znam*
15.	***Nit predstavlja:***
a.	*deo klase koji omogućava objektima da se istovremeno izvršavaju.*
b.	*deo klase koja može istovremeno da se izvršava sa drugim klasama.*
c.	*deo programa koji ne može istovremeno da se izvršava sa drugim nitima istog programa.*
d.	*deo programa koji ne može istovremeno da se izvršava sa drugim programima.*
e.	*deo programa koji može istovremeno da se izvršava sa drugim nitima istog programa.*
n.	*Ne znam*
16.	***Učitavanje upravljačkog programa u Java program se radi preko sledeće naredbe:***
a.	*import java.sql.*;*
b.	*String dbUrl = "jdbc:mysql://127.0.0.1:3306/student";*
c.	*Connection c = DriverManager.getConnection(dbUrl, user, pass);*
d.	*Class.forName("com.mysql.jdbc.Driver");*
e.	*String upit = "SELECT brind,ime,prezime FROM Student";*
n.	*Ne znam*
17.	***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.*
n.	*Ne znam*

18.	***Pogledati kod koji je dat ispod. Koja od narednih izjava je tačna:***
	`class Osoba {`
	`}`
	
	`class OSobA {`
	`}`

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*
n.	*Ne znam*

19.	***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?***
	`1 class Student {`
	`2  STRING ime = "Pera Peric";`
	`3  double prosek = 8.83`
	`4  int godinaRodjenja = 1993;`
	`5  char pol = "M";`
	`6 }`

a.	*2,3,4,5*
b.	*1,2,3*
c.	*2,5*
d.	*1,2*
e.	*2,3,5*
n.	*Ne znam*

20.	***Pogledati kod koji je dat ispod. Koliko objekata klase Automobil je inicijalizovano u main metodi?***
	`class Automobil {`
	`}`
	
	`class Test {`
	`  public static void main(String args[]) {`
	`  Automobil a1 = new Automobil();`
	`  Automobil a2 = new Automobil();`
	`  Automobil a3;`
	`  }`
	`}`

a.	*1*
b.	*2*
c.	*3*
d.	*4*
e.	*0*
n.	*Ne znam*

21.	***Pogledati kod koji je dat ispod. Koja od narednih izjava je tačna:***
	
	`class Metode {`
	`  void vratiApsolutnuNulu (){`
	`  return -273.15;`
	`  }`
	`}`

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*
n.	*Ne znam*

22.	***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)?***
	`class BMI {`
	`  double izracunajBMI  (double visina, double tezina){`
	`  double rezultat = tezina/visina*visina;`
	`  return rezultat;`
	`  }`
	`}`

a.	*Metoda ništa ne vraća kao povratnu vrednost*
b.	*0.0*
c.	*25.0*
d.	*100.0*
e.	*Java prijavljuje grešku*
n.	*Ne znam*

23.	***Pogledati kod koji je dat ispod. Šta će se desiti kad se pokrene main metoda?***
	`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);`
	`  }`
	`}`

a.	*Java prijavljuje grešku*
b.	*Na ekranu će se ispisati "Pera"*
c.	*Na ekranu će se ispisati "Pera Mika"*
d.	*Na ekranu će se ispisati "Mika Pera"*
e.	*Na ekranu će se ispisati "Mika"*
n.	*Ne znam*

24.	***Pogledati kod koji je dat ispod. Šta će metoda vratiti kao povratnu vrednost ako joj se kao argument prosledi vrednost 300?***
	`class Provera {`
	`  boolean proveraIntervala (int a){`
	`  if (a < 100 || a > 500)`
	`  return true;`
	`  else`
	`  return false;`
	`  }`
	`}`

a.	*Metoda ništa ne vraća kao povratnu vrednost*
b.	*true*
c.	*false*
d.	*a*
e.	*0*
n.	*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);
-------------------------------------------------------------
```

```
CREATE TABLE OSOBA(
      Id                int not null ,
      Ime               nvarchar(50) not null
                        check (Ime like '[A-Ž]%'),
      Prezime           nvarchar(50) not null
                        check (Prezime like '[A-Ž]%'),
      SredIme           nvarchar(50) null,
      Pol               nvarchar(8) not null
                        check (Pol in
('MUŠKI','ŽENSKI','NEPOZNAT')),
      GodRod            int null
                        check ((GodRod > 1900) OR (GodRod is NULL))
                        default (Year(GetDate())),
      Visina            int null
                        check ((Visina > 0) OR (Visina is NULL)),
      IdMestaRod        int not null,
      IdMestaPreb             int null,
CONSTRAINT PK_Osoba PRIMARY KEY (Id),
CONSTRAINT CK_Osoba_Id CHECK (Id > 0),
CONSTRAINT FK_Osoba_MestoRod FOREIGN KEY (IdMestaRod) references
MESTO(Id),
CONSTRAINT FK_Osoba_Preb FOREIGN KEY (IdMestaPreb) references
MESTO(Id)
);
```

Tabele sadrže sledeće podatke:

MESTO

Id	Naziv	BrStanovnika	PttBroj
11	Smederevo	71327	11300
22	Bor	23713	19300
33	Zrenjanin	59123	23000
44	Beograd	1234789	11000
55	Niš	321321	18000

OSOBA

Id	Ime	Prezime	SredIme	Pol	GodRod	Visina	IdMestaRod	IdMestaPreb
10	Stanko	Vasić	NULL	MUŠKI	1957	171	11	NULL
15	Branko	Jovanović	NULL	MUŠKI	1967	181	22	NULL
20	Janko	Milidrag	NULL	MUŠKI	1977	191	22	11
25	Dragan	Kostić	H.G.	MUŠKI	1957	181	11	44
30	Milan	Vasić	D.	MUŠKI	1967	171	11	44
35	Boban	Kostadinov	D.	MUŠKI	1977	161	44	44
40	Jana	Lazarovski	Sava	ŽENSKI	1987	161	33	11
45	Milica	Lazarević	Saša	ŽENSKI	1977	151	44	22
50	Jovana	Vasić Lazić	NULL	ŽENSKI	1967	161	44	NULL
55	Brankica	Milidrag	NULL	ŽENSKI	1957	171	44	44
60	Vanja	Lazarovski	Mita	NEPOZNAT	1997	172	22	11
65	Jana	Savić	Mita	ŽENSKI	2017	172	22	11

T-SQL funkcije koje su korišćene:

- GetDate() – vraća sistemski (trenutni) datum (date) u formatu: yyyy-mm-dd hh:mm:ss
- Year(date) – vraća ceo broj koji predstavlja godinu u datumu date
- Convert(data_type, expression) – konvertuje izraz expression u tip podatka data_type

25.	***Upit***:
	`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');`
	prikazuje:

a.	*166*
b.	*167*
c.	*167,666666666667*
d.	*168*
e.	*Srednju visinu osoba koje su rođene u Beogradu*
n.	*Ne znam*

26.	***Upit:***
	`SELECT Ime --, Prezime, Visina, GodRod`
	`FROM OSOBA`
	`WHERE`
	`  Prezime NOT LIKE '%ski'`
	`  AND GodRod LIKE '_ _[^5-7]_';`
	Prikazuje sledeća imena:

a.	*Jana*
b.	*Ništa*
c.	*Stanko, Branko, Janko, Dragan, Milan, Boban, Milica, Jana, Jovana, Brankica*
d.	*Jana, Vanja*
e.	*Error: Incorrect syntax near ' GodRod'*
n.	*Ne znam*

27.	***Upit:***
	`SELECT 2 * (Year(GetDate())) - SUM(R.GodRod)`
	`FROM OSOBA R JOIN OSOBA P`
	`  on (R.IdMestaPreb = P.IdMestaRod)`
	`WHERE R.Id = P.Id;`
	prikazuje:

a.	*351*
b.	*Error: Incorrect syntax near ' Year(GetDate())'*
c.	*100*
d.	*2017*
e.	*Nijedan od prethodnih odgovora*
n.	*Ne znam*

28.	***Upit:***
	`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;`
	prikazuje:

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*
n.	*Ne znam*

29.	***Upit:***
	`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]%';`
	prikazuje:

a.	*Error: Incorrect syntax near ' GodRod'*
b.	*Tri reda*
c.	*Četiri reda*
d.	*Nijedan red*
e.	*Dva reda*
n.	*Ne znam*

30.	***Upit:***
	`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')`
	`);`
	prikazuje:

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*