



Prijemni ispit za studijski program Softversko inženjerstvo i veštačka inteligencija 2023

Шифра задатака:

1	2	3	4	6	9
---	---	---	---	---	---

1. Pogledati kod koji je dat ispod. Šta će se ispisati na ekranu ako se pokrene program?

```
class Test {  
    public static void main(String[] args) {  
        int sum = 0;  
  
        for(int i = 0; i < 4; i++) {  
            sum += i;  
        }  
        System.out.print(sum);  
    }  
}
```

- | | |
|-------------|------------|
| A. 10 | D. 7 |
| B. 6 | E. 12 |
| C. 4 | N. Ne znam |

2. Koji poziv konstruktora će uspešno kreirati instancu klase Pas?

```
class Pas {  
    private String ime;  
  
    public Pas(String ime) {  
        this.ime = ime;  
    }  
  
    public String dohvatiIme() {  
        return ime;  
    }  
}
```

- | | |
|--|---------------------------------|
| A. Pas mojPas = new Pas(); | D. Pas mojPas = Pas("Max"); |
| B. Pas mojPas = new Pas("Max"); | E. Pas mojPas = Pas.new("Max"); |
| C. Pas mojPas = new Pas("Max", 4); | N. Ne znam |

3. 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 = new Osoba();  
        o1.ime = "Anja";  
        o2.ime = "Nadja";  
        Osoba o3 = o1;  
        o1 = o2;  
        System.out.println(o3.ime);  
    }  
}
```

- | | |
|---------------------------|---|
| A. Ispisuje "Nadja" | D. Dolazi do greške u vreme izvršavanja |
| B. Ispisuje "Anja" | E. Dolazi do greške u vreme kompilacije |
| C. Ispisuje null | N. Ne znam |

4. Pogledati kod koji je dat ispod. Šta će se ispisati na ekranu ako se pokrene program?

```
public class Main{
    public static void main(String[] args) {
        String poruka = "Dobar dan!";
        String izvod = poruka.substring(6, 9);
        System.out.println(izvod);
    }
}
```

- | | | | |
|----|--------------|----|---------|
| A. | "dan!" | D. | "an" |
| B. | "dan" | E. | "ar d" |
| C. | "an!" | N. | Ne znam |

5. Pogledati kod koji je dat ispod. Šta će se ispisati na ekranu kad se pozove main metoda?

```
class Vozilo {
    void ispisi(){System.out.println("vozilo");}
}

class Motor extends Vozilo{
    void ispisi(){System.out.println("motor");}
}

class Automobil extends Vozilo{}

class Test{
    public static void main(String[] args){
        Automobil a = new Automobil();
        a.ispisi();
    }
}
```

- | | | | |
|----|-----------------|----|--------------------------------------|
| A. | "motor" | D. | Dolazi do greške u vreme izvršavanja |
| B. | "automobil" | E. | Dolazi do greške u vreme kompilacije |
| C. | "vozilo" | N. | Ne znam |

6. Koja od navedenih karakteristika je tačna za nenadgledano mašinsko učenje?

- A. Ne može identifikovati skrivene obrasce u podacima
- B. Nemamo informaciju o željenoj izlaznoj vrednosti**
- C. Koristi predefinisane kategorije
- D. Primenjuje se isključivo za probleme klasifikacije
- E. Predviđa samo numeričke izlaze
- N. Ne znam

7. Koja od navedenih karakteristika je tačna za meru AUC (eng. *Area Under the Curve*)?

- A. AUC se koristi samo za evaluaciju regresionih modela
- B. AUC ne uzima u obzir tačna predviđanja negativne klase (*True Negative*)
- C. Klasifikator sa vrednošću AUC > 0.5 se smatra odličnim
- D. Za metodu slučajnog izbora važi da je AUC = 0.5**
- E. Vrednost za AUC se kreće u intervalu od -1 do 1
- N. Ne znam

8. Kako se prevazilazi problem K-Means algoritma uslovljen nasumičnim (random) inicijalnim izborom težišta?

- A. Korišćenjem Elbow (lakat) metode
- B. Primenom višestruke nasumične inicijalizacije**
- C. Povećanjem broja klastera na maksimalnu vrednost
- D. Dodeljivanjem svih opservacija istom klasteru pri inicijalizaciji
- E. Dodavanjem šuma u podatke pre pokretanja algoritma
- N. Ne znam

9. U arhitekturi ekspertnog sistema, osnovne tri komponente su:

- A. **baza znanja, radna memorija i mehanizam za zaključivanje**
- B. kratkoročna i dugoročna memorija i mehanizam za zaključivanje
- C. baza znanja, radna memorija i rasuđivanje
- D. kratkoročna i dugoročna memorija i rasuđivanje
- E. baza znanja, radna memorija i mehanizam za objašnjavanje
- N. Ne znam

10. Koja je svrha mehanizma za zaključivanje tj. šta sadrži?

- A. sadrži podatke iz domena i omogućava korišćenje tih podataka
- B. sadrži formalizovano domensko znanje, najčešće u formi pravila
- C. sadrži formalizovane početne činjenice o problemu i zaključke
- D. **sadrži algoritam za zaključivanje i izvodi zaključke na osnovu znanja i činjenica**
- E. sadrži strategiju za rešavanje konflikta
- N. Ne znam

11. Koje su tri ključne uloge u procesu razvoja ekspertnog sistema?

- A. softverski inženjer, korisnik
- B. softverski inženjer, ekspert, korisnik
- C. inženjer znanja, korisnik
- D. inženjer znanja, softverski inženjer, korisnik
- E. **inženjer znanja, ekspert, korisnik**
- N. Ne znam

12. Kako se može izbeći pretreniravanje (eng. *overfitting*) neuronske mreže?

- A. Smanjenjem količine podataka
- B. Povećanjem broja ulaza
- C. Povećanjem složenosti modela
- D. **Primenom tehnike ranog zaustavljanja**
- E. Korišćenjem pažljivo odabranih podataka u skupu podataka za trening
- N. Ne znam

13. Adresa soketa sastoji se iz dva dela:

- A. adrese memorije na kojoj se nalazi program koji je generisao soket i protokola koji koriste soketi
- B. adrese objekta koji je generisao soket i porta 8189
- C. adrese klase koja je generisala soket i predefinisanoj porta koji je generisan pomoću soketa
- D. adrese računara koji je server soketu i broja porta koji je klijent soketu
- E. **adrese računara na kome se nalazi program koji je generisao soket i broja porta koji je generisan pomoću soketa**
- N. Ne znam

14. 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 klase koji omogućava objektima da se istovremeno izvršavaju
- E. **deo programa koji može istovremeno da se izvršava sa drugim nitima istog programa**
- N. Ne znam

15. Koja se metoda poziva da odblokira nit koja je blokirana *wait()* metodom:

- A. *inform()* metoda
- B. *stop()* metoda
- C. *sleep()* metoda
- D. ***notify()* metoda**
- E. *check()* metoda
- N. Ne znam

16. Na osnovu navedene naredbe:

```
String dbUrl="jdbc:mysql://143.126.127.3:3306/student";
```

se može zaključiti da je MySQL SUBP podignut na portu:

- A. 143
 - B. 126
 - C. 127
 - D. 3
 - E. 3306**
 - N. Ne znam
17. Prvo Boemovo pravilo (eng. *Boehm's first law*) glasi:
- A. Scenarija SK ne treba da budu u međusobnoj interakciji
 - B. Pravljenje prototipova značajno smanjuje moguće greške kod definisanja zahteva i njegovog razvoja, naročito kod definisanja korisničkog interfejsa
 - C. Nedostaci kod definisanja zahteva su osnovni razlog mogućeg neuspeha u razvoju projekta (programa)
 - D. Ukoliko se ne uoče greške u toku definisanja zahteva, iste se veoma teško mogu ukloniti u kasnijim fazama razvoja programa**
 - E. Zahtev za izvršenje jedne ili više sistemskih operacija se ne odigrava kontinualno nego diskretno
 - N. Ne znam
18. Kako glasi pravilo nezavisnosti scenarija:
- A. Scenarija slučajja korišćenja ne treba da budu u međusobnoj interakciji**
 - B. Scenarija slučajja korišćenja treba da budu u međusobnoj interakciji
 - C. Scenarija slučajja korišćenja treba povezati sa sistemskim operacijama
 - D. Scenarija slučajja korišćenja ne treba da budu povezana sa bazom podataka
 - E. Scenarija slučajja korišćenja treba da budu povezana sa poslovnom logikom
 - N. Ne znam
19. Šta opisuje faza analize u razvoju softverskog sistema?
- A. Faza analize opisuje ponašanje softverskog sistema
 - B. Faza analize opisuje strukturu softverskog sistema
 - C. Faza analize opisuje logičku strukturu i ponašanje softverskog sistema**
 - D. Faza analize opisuje slučajjeve korišćenja
 - E. Faza analize opisuje arhitekturu softverskog sistema
 - N. Ne znam
20. Konceptualni model opisuje:
- A. Konceptualne klase domena problema**
 - B. Konceptualne prelaze stanja domena problema
 - C. Konceptualnu strukturu ograničenja domena problema
 - D. Konceptualne veze između elemenata domena problema
 - E. Konceptualna preslikavanja između slogova domena problema
 - N. Ne znam
21. U toku izvršenja sistemske operacije nad strukturom softverskog sistema podaci moraju da ostanu konzistentni, odnosno moraju da budu zadovoljena:
- A. vrednosna i strukturna ograničenja definisana nad tokovima podataka
 - B. vrednosna i strukturna ograničenja definisana nad podacima**
 - C. ograničenja definisana nad klasama i objektima
 - D. strukturna ograničenja definisana nad metodama klase
 - E. perzistentna ograničenja definisana nad podacima
 - N. Ne znam

22. Preduslovi kod sistemskih operacija u fazi analize kod uprošćene Larmanove metode:
- A. ukazuje na to šta će se desiti, pre nego što se sistemski operacija izvrši
 - B. ukazuje na to šta je i kako je trebalo da se desi, kako bi sistemski operacija mogla da se izvrši
 - C. ukazuje na to šta će se desiti, pošto se izvrši sistemski operacija
 - D. ukazuje na to kako se nešto desilo, kako bi sistemski operacija mogla da se izvrši, a ne šta je trebalo da se desi
 - E. ukazuju na to šta je trebalo da se desi, kako bi sistemski operacija mogla da se izvrši, a ne kako se to desilo**
 - N. Ne znam
23. Šta je rezultat faze projektovanja kod uprošćene Larmanove metode razvoja softvera:
- A. Baza podataka
 - B. Slučajevi korišćenja
 - C. Softverski paterni
 - D. Arhitektura softverskog sistema**
 - E. Korisnički interfejs
 - N. Ne znam
24. Perzistentni okviri su zasnovani na:
- A. Principu nezavisnosti sistemskih operacija
 - B. Holivudskom principu**
 - C. Principu slabe povezanosti klasa
 - D. Principu otvoreno-zatvoreno
 - E. Principu kompatibilnosti klasa
 - N. Ne znam
25. Relaciona baza podataka je:
- A. skup integrisanih datoteka sa minimalnom redundansom koje su međusobno povezane preko primarnih i spoljnih ključeva i promenljive u vremenu
 - B. skup vremenski promenljivih relacija
 - C. skup tabela u kojima se trajno čuvaju ažurabilni podaci o entitetima iz razmatranog sistema
 - D. sve što je prethodno navedeno**
 - E. sistem integrisanih binarnih i tekstualnih datoteka koji se čuvaju na jednom uređaju
 - N. Ne znam
26. Oznaka NULL se u relacijskim sistemima za upravljanje bazama podataka koristi da bi označila:
- A. trenutno nepoznatu vrednost
 - B. neprimenljivo svojstvo
 - C. sve što je prethodno navedeno**
 - D. specijalnu nedostajuću vrednost
 - E. ništa od prethodno navedenog
 - N. Ne znam
27. Pošto je relacija skup, a svaka tabela nije skup, koji uslov(e) tabela mora da zadovolji da bi bila relacija:
- A. sve vrste u tabeli moraju da budu različite
 - B. redosled vrsta u tabeli je proizvoljan
 - C. redosled kolona u tabeli je proizvoljan
 - D. sve vrednosti polja u tabeli su atomske, odnosno nije dozvoljeno da vrednosti nekih polja u tabeli budu tabele
 - E. sve što je prethodno navedeno**
 - N. Ne znam

28. Prilikom izvršavanja SQL upita (eng. *query*) sistem za upravljanje bazama podataka izvršava sledeće korake:

- A. validacija upita, optimizacija upita, generisanje plana izvršenja, izvršenje plana
- B. validacija upita, parsiranje upita, optimizacija upita, generisanje optimizovanog plana izvršenja upita, izvršenje upita
- C. parsiranje upita, optimizacija upita, generisanje plana izvršenja upita, validacija plana izvršenja, izvršenje optimizovanog i validiranog plana
- D. parsiranje upita, validacija upita, generisanje plana izvršenja upita, optimizacija plana izvršenja, izvršenje optimizovanog plana**
- E. ništa od prethodno navedenog
- N. Ne znam

29. U sledećim SQL naredbama kreirane su tabele STUDENT i SMER, a potom su popunjene podacima:

```
CREATE TABLE SMER(  
  S#          tinyint not null primary key,  
  Naziv      nvarchar(24) not null);
```

```
CREATE TABLE STUDENT(  
  I#          smallint not null primary key,    -- број индекса  
  Ime        nvarchar(7) not null,  
  Prezime    nvarchar(7) not null check (Prezime LIKE '[А-Ш]%' ),  
  Smer#      tinyint null foreign key references SMER(S#));
```

```
INSERT INTO SMER VALUES  
  (10, N'Софтверско инжењерство'), (20, N'Информациони системи'),  
  (80, N'Управљање производњом'), (60, N'Управљање пословањем');
```

```
INSERT INTO STUDENT VALUES  
  (17002, N'Ана', N'Костић', NULL), (17014, N'Ана', N'Марић', NULL),  
  (17008, N'Анка', N'Анић', NULL), (16002, N'Аница', N'Барић', 10),  
  (16014, N'Мара', N'Илић', 20), (16008, N'Мила', N'Јовић', 60),  
  (15002, N'Аца', N'Костић', 10), (15014, N'Мома', N'Којић', 20),  
  (15008, N'Јова', N'Кун', 60), (14002, N'Лаза', N'Марић', 60),  
  (14014, N'Јова', N'Киш', 10);
```

Napomena: Slovo N ispred niza znakova koji se nalaze u jednostrukim navodnicima označava da je navedeni niz znakova nvarchar vrednost.

Dati su sledeći upiti:

<p>Y-1:</p> <pre>SELECT I#, Ime, Prezime, Smer# FROM (SELECT S#, Naziv FROM SMER WHERE Naziv LIKE N'Софтверско инжењерство') SM INNER JOIN STUDENT ST ON Smer# = SM.S# ORDER BY I#;</pre>	<p>Y-2:</p> <pre>SELECT I#, Ime, Prezime, Smer# FROM STUDENT WHERE Smer# = (SELECT S# FROM SMER WHERE S# NOT IN (80, 10, 20)) ORDER BY I#;</pre>
<p>Y-3:</p> <pre>SELECT I#, Ime, Prezime, Smer# FROM STUDENT S JOIN SMER ST ON S.Smer# = ST.S# WHERE Naziv LIKE N'Софтверско инжењерство' ORDER BY 1 DESC;</pre>	<p>Y-4:</p> <pre>SELECT I#, Ime, Prezime, Smer# FROM STUDENT ST WHERE EXISTS (SELECT /* * */ 'X' FROM SMER SM WHERE ST.Smer# = SM.S# AND Naziv LIKE N'Софтверско инжењерство');</pre>

Od navedenih upita, navedite one koji vraćaju isti rezultat u relacijskom smislu:

- A. Y-1, Y-2
- B. Y-2, Y-4
- C. među navedenim upitima ne postoje one koji vraćaju isti rezultat
- D. Y-1, Y-3, Y-4**
- E. Y-2, Y-3
- N. Ne znam

30. Izvršavanjem upita:

```
SELECT S.I#, S.Ime, S.Prezime, Naziv Naziv
FROM STUDENT S
FULL OUTER JOIN SMER MR ON Smer# = MR.S#
ORDER BY S# DESC;
```

U poslednjem redu biće prikazano:

- A. 17014 Ана Марић NULL
- B. 14002 Лаза Марић Управљање пословањем
- C. 14014 Јова Киш Софтверско инжењерство
- D. Ništa, zato što je upit sintaksno neispravan
- E. NULL NULL NULL Управљање производњом
- N. Ne znam