



**Пријемни испит за студијски програм
Софтверско инжењерство и вештачка интелигенција 2026**

Шифра задатака:

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

1. Шта представља агенда циљева код уланчавања уназад?
 - a. Списак свих правила поређаних према времену настанка
 - b. **Хијерархијску структуру циљева које треба доказати да би се доказао основни циљ**
 - c. Скуп свих података, без обзира на њихову везу са циљем
 - d. Листу правила која су већ извршена, у редоследу извршавања
 - e. Списак свих могућих корисничких питања без хијерархије
 - n. Не знам
2. Шта омогућава преношење грешке са излаза уназад, са слоја на слој, у *backpropagation* алгоритму?
 - a. **Диференцијабилне функције трансфера и примена правила за извод сложених функција**
 - b. Недиференцијабилна степ функција и случајни избор тежина
 - c. Просечне вредности улаза и фиксне тежине свих веза
 - d. Искључиво повећавање броја излазних неурона
 - e. Замена функције грешке бројем скривених слојева
 - n. Не знам
3. Погледати код који је дат испод. Шта ће се исписати на екрану ако се покрене програм?

```
class Test {  
    public static void main(String[] args) {  
        int x = 10;  
        int y = 4;  
        int z = 2;  
        int rezultat = 0;  
  
        while (x > 0) {  
            rezultat += z;  
            z *= 2;  
            x -= y;  
        }  
        System.out.print(rezultat);  
    }  
}
```

 - a. 2
 - b. 6
 - c. 16
 - d. **14**
 - e. 30
 - n. Не знам
4. Када се нит буди, ако се успава помоћу *sleep()* методе?
 - a. Нит се буди, након што је прошло подразумевано време “успаваности” нити.
 - b. **Нит се буди, након што је прошло време “успаваности”, задато методом *sleep()*.**
 - c. Нит се буди, након што је прошло време “успаваности”, задато *run()* методом.
 - d. Нит се буди, позивом нити од стране главне нити.
 - e. Нит се буди, позивом методе *resume()*.
 - n. Не знам
5. Актор (учесник) у фази прикупљања захтева код упрошћене Ларманове методе представља:
 - a. Тестера софтверског система
 - b. Аналитичара софтверског система
 - c. Доменског експерта
 - d. **Спољног корисника система**
 - e. Пројектанта базе података
 - n. Не знам

6. Погледати код који је дат испод. Шта ће се исписати на екрану ако се покрене програм?

```
class Test {
    public static void main(String[] args) {
        int x = 9;
        int y = 6;
        int z = 3;
        int rezultat = 2;

        while (x > y) {
            rezultat *= z - 2;
            y++;
        }
        System.out.print(rezultat);
    }
}
```

- | | |
|-------------|------------|
| a. 0 | d. 28 |
| b. 2 | e. 54 |
| c. 10 | n. Не знам |

7. Следећим *SQL* наредбама креиране су табеле *STUDENT* и *SMER*, а потом су попуњене подацима:

```
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 '[A-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);
```

Напомене: Слово *N* испред низа знакова који се налазе у једноструким наводницима, означава да је наведени низ знакова *nvarchar* вредност. Сви *SQL* упити који следе односе се на претходно креиране и попуњене табеле.

Погледати *SQL* упит који је дат испод. Шта ће бити приказано у четвртом реду резултата добијеног извршењем наведеног упита?

```
SELECT I#, Ime, Prezime
FROM STUDENT
WHERE Smer# != 10
ORDER BY I#;
```

- | | | | |
|------------------|-----------|-------|---------------------|
| a. 16014 | Мара | Илић | d. Синтаксна грешка |
| b. 14002 | Лаза | Марић | e. 16008 |
| c. Сви студенти, | сортирани | по I# | Мила Јовић |
| | | | f. Не знам |

8. Извршењем упита:

```
SELECT COUNT(*), COUNT(ALL Smer#), COUNT(DISTINCT Smer#)
FROM STUDENT
WHERE Prezime LIKE N'K%';
```

биће приказано:

- | | |
|--|---------------------|
| a. Број студената чије презиме почиње са 'К' | d. Синтаксна грешка |
| b. 5 3 2 | e. 4 3 2 |
| c. 5 4 3 | f. Не знам |

9. Погледати *SQL* упит који је дат испод. Колико ће редова бити приказано у резултату који ће се добити извршењем наведеног упита?

```
SELECT S.I#, S.Ime, S.Prezime  
FROM STUDENT S  
WHERE S.I# IN (  
SELECT I# FROM STUDENT WHERE Smer# IS NULL);
```

- | | |
|------|------------|
| a. 3 | d. 0 |
| b. 4 | e. 3 |
| c. 5 | n. Не знам |

10. Шта ће бити приказано у првом реду резултата који се добија извршавањем *SQL* упит који је дат испод?

```
SELECT Smer#, COUNT(*) AS BrojStudenata  
FROM STUDENT S JOIN SMER M ON S.Smer# = M.S#  
GROUP BY Smer#  
HAVING COUNT(Smer#) > /*==>*/1--;;;  
ORDER BY 2 DESC;
```

- | | |
|---------|------------------------------------|
| a. 8 3 | d. Биће приказана синтаксна грешка |
| b. 11 2 | e. 10 3 |
| c. 11 3 | n. Не знам |

11. Шта ће бити приказано у првом реду резултата који се добија извршавањем *SQL* упит који је дат испод?

```
;WITH BrojStudenata AS (  
SELECT Smer#, COUNT(I#) AS Broj  
FROM STUDENT  
GROUP BY Smer#  
)  
SELECT M.Naziv, ISNULL(B.Broj, 0) AS BrojStudenata  
FROM SMER M  
LEFT JOIN BrojStudenata B ON M.S# = B.Smer#  
ORDER BY BrojStudenata ASC;;;
```

- | | | | |
|------------------------------------|---|--------------------------|---|
| a. Информациони системи | 2 | d. Управљање производњом | 0 |
| b. Управљање пословањем | 3 | e. Информациони системи | 3 |
| c. Биће приказана синтаксна грешка | | n. Не знам | |

12. Шта ће бити приказано у другом реду резултата који се добија извршавањем *SQL* упит који је дат испод?

```
SELECT I#, Ime, Prezime  
FROM STUDENT S  
WHERE EXISTS (  
SELECT 3--(1+2)  
FROM STUDENT X  
WHERE X.Smer# = S.Smer#  
GROUP BY Smer#  
HAVING COUNT(*) > 1  
)  
ORDER BY I#;
```

- | | | | | | |
|----------|------|--------|------------------------------------|------|------|
| a. 14014 | Јова | Киш | d. Биће приказана синтаксна грешка | | |
| b. 15002 | Аца | Костић | e. 16014 | Мара | Илић |
| c. 15014 | Мома | Којић | n. Не знам | | |

13. Шта се мења, а шта остаје фиксно при минимизацији функције коштања Ј у фази померања тежишта K-means алгоритма?

- | | |
|--|---|
| a. Мењају се доделе инстанци кластерима, а тежишта остају фиксна | d. Мењају се вредности улазних атрибута, а доделе остају фиксне |
| b. Мења се број кластера, а тежишта остају фиксна | e. Мењају се тежишта кластера, а доделе инстанци кластерима остају фиксне |
| c. Истовремено се мењају тежишта и доделе инстанци кластерима | n. Не знам |

14. Ако P означава прецизност, а R одзив, који израз представља балансирану $F1$ меру?
- $(P + R) / 2$
 - $P \times R / (P + R)$
 - $2 \times P \times R / (P + R)$
 - $(P + R) / (2 \times P \times R)$
 - $2 \times (P + R) / (P \times R)$
 - Не знам
15. Којој операцији припреме података припада PCA, према презентацији?
- Филтрирању
 - Редукцији димензионалности**
 - Нормализацији
 - Стандардизацији
 - Подели података на тренинг и тест скуп
 - Не знам
16. Која тврдња описује недовољно подударање модела са подацима (*under-fitting*)?
- Модел савршено препознаје тренинг инстанце, али не препознаје ни мало измењене инстанце.
 - Модел има одличне перформансе и на тренинг и на тест скупу.
 - Модел се успешно прилагођава подацима само повећавањем броја кластера.
 - Модел не користи скуп за тренинг, већ искључиво скуп за тестирање.
 - Модел не успева да апроксимира податке и има слабе перформансе чак и на тренинг скупу.**
 - Не знам
17. Погледати код који је дат испод. Који позив конструктора ће успешно креирати инстанцу класе **Pravougaonik**?
- ```
class Pravougaonik {
 private int sirina;
 private int visina;

 public Pravougaonik(int sirina, int visina) {
 this.sirina = sirina;
 this.visina = visina;
 }

 public int povrsina() {
 return sirina * visina;
 }
}
```
- Pravougaonik p = new Pravougaonik();
  - Pravougaonik p = new Pravougaonik(4);
  - Pravougaonik p = Pravougaonik(4, 6);
  - Pravougaonik p = new Pravougaonik(4, 6);**
  - Pravougaonik p = Pravougaonik.new(4, 6);
  - Не знам
18. Уколико се јави потреба да две или више нити деле заједнички ресурс и имају искључиви (ексклузивни) приступ до заједничког ресурса, који се механизам мора обезбедити:
- Перзистенција
  - Активација
  - Пасивизација
  - Синхронизација**
  - Материјализација
  - Не знам
19. Када се Јава програм повезује са системом за управљање базом података он то ради помоћу:
- JDBC API-а и нити
  - JDBC API-а и управљачког програма**
  - JDBC API-а и Јава виртуалне машине
  - JDBC API-а и серверског програма
  - JDBC API-а и перзистентних објеката
  - Не знам
20. Која тврдња тачно разликује улоге параметара *Learning rate* и *Batch mode*?
- Learning rate* одређује максималан број итерација, а *Batch mode* праг грешке.
  - Learning rate* одређује број слојева, а *Batch mode* број неурона.
  - Learning rate* одређује праг укупне грешке, а *Batch mode* максималан број итерација.
  - Learning rate* одређује избор активационе функције, а *Batch mode* избор тест скупа.
  - Learning rate* одређује утицај грешке на промену тежине, а *Batch mode* да ли се тежине мењају за сваки пар (улаз, излаз) или за скуп парова.**
  - Не знам

21. Погледати код који је дат испод. Код не може да се компајлира. Које линије кода садрже грешке?

```
class Automobil {
 String marka = 'Fiat';
 int brojVrata = 4.5;
 double cena = 12000;
 boolean registrovan = 0;
 char klasa = 'B';
}
```

- a. 2, 3, 4, 5
- b. 2, 5
- c. 3, 4, 5

- d. 1, 2, 3, 5
- e. **2, 3, 5**
- n. Не знам

22. Који је препоручени поступак за избор величине подстабла при орезивању класификационог стабла?

- a. Изабрати највеће подстабло, без процене грешке
- b. Изабрати подстабло које има исти број листова као што има атрибута
- c. Изабрати подстабло које даје највећу грешку на тренинг скупу

- d. **Крос валидацијом проценити грешку подстабала различитих величина и изабрати оно са најмањом грешком**
- e. Увек одсећи све терминалне чворове осим једног
- n. Не знам

23. Погледати код који је дат испод. Шта ће се исписати на екрану ако се покрене програм?

```
class Test {
 public static void main(String[] args) {
 String s = "Programiranje";
 System.out.println(s.charAt(3));
 }
}
```

- a. P
- b. r
- c. o

- d. **g**
- e. Догодиће се грешка у извршавању
- n. Не знам

24. Погледати код који је дат испод. Шта ће се исписати на екрану ако се покрене програм?

```
class Test {
 public static void main(String[] args) {
 for (int i = 1; i <= 6; i++) {
 if (i % 3 == 0) {
 continue;
 }
 System.out.print(i);
 }
 }
}
```

- a. 123456
- b. 36
- c. **1245**

- d. 12
- e. 12456
- n. Не знам

25. Адреса сокета састоји се из два дела:

- a. Адресе меморије на којој се налази програм који је генерисао сокет и протокола који користе сокети.
- b. Адресе објекта који је генерисао сокет и порта 8189.
- c. Адресе класе која је генерисала сокет и предефинисаног порта који је генерисан помоћу сокета.

- d. Адресе рачунара који је сервер сокету и броја порта који је клијент сокету.
- e. **Адресе рачунара на коме се налази програм који је генерисао сокет и броја порта који је генерисан помоћу сокета.**
- n. Не знам

26. Прво Боемово правило (*Boehm's first law*) гласи:

- a. Сценарија СК не треба да буду у међусобној интеракцији.
- b. Прављење прототипова значајно смањује могуће грешке код дефинисања захтева и његовог развоја, нарочито код дефинисања корисничког интерфејса.
- c. Недостаци код дефинисања захтева су основни разлог могућег неуспеха у развоју пројекта (програма).

**d. Уколико се не уоче грешке у току дефинисања захтева, исте се веома тешко могу уклонити у каснијим фазама развоја програма.**

e. Захтев за извршење једне или више системских операција се не одиграва континуално него дискретно.

n. Не знам

27. Погледати код који је дат испод. Шта ће се исписати на екрану ако се покрене програм?

```
class Knjiga {
 String naslov;
}

class Test {
 public static void main(String[] args) {
 Knjiga k1 = new Knjiga();
 Knjiga k2 = new Knjiga();
 k1.naslov = "Hamlet";
 k2.naslov = "Makbet";
 Knjiga k3 = k2;
 k3.naslov = "Otello";
 k2 = k1;
 System.out.println(k3.naslov);
 }
}
```

a. Otello

b. Hamlet

c. Makbet

d. null

e. Долази до грешке у време компилације

n. Не знам

28. Ако су садржаји радне меморије и базе знања дати у табели, а механизам за закључивање користи уланчавање унапред, одговорити на следеће питање:

| Радна меморија                                          | База знања                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1) Auto неће да упали<br>2) Napon na akumulatoru = 11 V | <b>PRAVILO 1</b><br>IF Auto неће да упали AND<br>Napon na akumulatoru < 12 V<br>THEN Akumulator је празан<br><br><b>PRAVILO 2</b><br>IF Akumulator је празан<br>THEN Napuni akumulator<br><br><b>PRAVILO 3</b><br>IF Auto неће да упали AND<br>Napon na akumulatoru = 12 V<br>THEN Anlaser је неисправан<br><br><b>PRAVILO 4</b><br>IF Anlaser је неисправан<br>THEN Zameni anlaser |

До којих ће све нових закључака доћи ЕС? Изабрати један одговор који садржи све и само нове чињенице које ће бити додате у радну меморију.

a. Акумулатор је празан

b. Анласер је неисправан; Замени анласер

c. Акумулатор је празан; Замени анласер

d. Акумулатор је празан; Напуни акумулатор;  
Анласер је неисправан

**e. Акумулатор је празан; Напуни акумулатор**

n. Не знам

29. Шта је резултат фазе анализе код упрошћене Ларманове методе развоја софтвера:

- a. База података
- b. Пословна логика**
- c. Архитектура софтверског система
- d. Софтверски патерни
- e. Екранске форме
- n. Не знам

30. Перзистентни оквири су засновани на:

- a. Холивудском принципу**
- b. Принципу независности системских операција
- c. Принципу слабе повезаности класа
- d. Принципу отворено-затворено
- e. Принципу компатибилности класа
- n. Не знам