



**Пријемни испит за студијски програм
Софтверско инжењерство и вештачка интелигенција 2025**

Шифра задатака:

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

1. 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 = "Hello";  
        System.out.println(s.charAt(1));  
    }  
}
```

- | | |
|------|------------------------------------|
| A. H | D. o |
| B. e | E. Догодиће се грешка у извршавању |
| C. l | N. Не знам |

2. 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 = 7; int y = 5; int z = 2; int rezultat = 1;  
        while (x >= y) {  
            rezultat *= z - 1;  
            y++;  
        }  
        System.out.print(rezultat);  
    }  
}
```

- | | |
|-------|------------|
| A. 1 | D. 32 |
| B. 8 | E. 64 |
| C. 16 | N. Не знам |

3. Који позив конструктора ће успешно креирати инстанцу класе Kvadrat?

```
class Kvadrat {  
    private int duzinaStranice;  
  
    public Kvadrat(int duzina) {  
        this.duzinaStranice = duzina;  
    }  
    public int povrsina() {  
        return duzinaStranice * duzinaStranice;  
    }  
}
```

- | | |
|---------------------------------------|--------------------------------|
| A. Kvadrat k = new Kvadrat(); | D. Kvadrat k = Kvadrat(5); |
| B. Kvadrat k = new Kvadrat(5); | E. Kvadrat k = Kvadrat.new(5); |
| C. Kvadrat k = new Kvadrat(5, 5); | N. Не знам |

4. 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.

Извршењем упита:

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

биће приказани:

- | | |
|---------------------------------------|--------------------------------|
| A. Сви студенти са смера 10 и 20 | D. Синтаксна грешка |
| B. Студенти: Јова, Аца и Аница | E. Студенти: Лаза, Јова и Мила |
| C. Сви студенти, сортирани по I# | N. Не знам |

5. Извршењем упита:

```
SELECT Ime, Prezime  
FROM STUDENT  
WHERE Prezime LIKE N'K%';
```

биће приказани:

- | | |
|---|---------------------|
| A. Сви студенте чије презиме почиње са 'К' | C. Ништа |
| B. Само студенти у чијем презимену се не налази 'К' | D. Синтаксна грешка |
| | E. Сви студенти |
| | N. Не знам |

6. Извршењем упита:.

```
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#;
```

биће приказано:

- | | |
|---|---|
| A. Називи смерова и бројеви студената по смеру | D. Називе смерова који су без студената |
| B. Ништа | E. Синтаксна грешка |
| C. Називе смерова и број студената по смеру, ако смерови имају студенте | N. Не знам |

7. Извршењем упита:

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

биће приказано:

- A. Седам редова
- B. Ништа
- C. Два реда

- D. Осам редова**
- E. Синтаксна грешка
- N. Не знам

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

```
1 class Osoba {
2     int godine = "trideset";
3     double visina = 1.75;
4     char pol = "M";
5     boolean zaposlen = "da";
6 }
```

- A. 2, 3, 5
- B. 1, 3, 4, 5
- C. 2, 4, 5**

- D. 2, 5
- E. 4, 5
- N. Не знам

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

```
class Zivotinja {
    String vrsta;
}
class Test {
    public static void main(String[] args) {
        Zivotinja z1 = new Zivotinja();
        Zivotinja z2 = new Zivotinja();
        z1.vrsta = "Pas";
        z2.vrsta = "Vrabac";
        Zivotinja z3 = z1;
        z1 = z2;
        System.out.println(z3.vrsta);
    }
}
```

- A. Pas**
- B. Vrabac
- C. null

- D. Долази до грешке у време извршавања
- E. Долази до грешке у време компилације
- N. Не знам

10. Извршењем упита:

```
SELECT S.I#, S.Ime, S.Prezime
FROM STUDENT S
WHERE S.I# NOT IN (
    SELECT I# FROM STUDENT WHERE Smer# IS NOT NULL);
```

биће приказани:

- A. Ништа
- B. Сви студенти без уписаног смера**
- C. Синтаксна грешка

- D. Сви студенти
- E. Сви студенти који су уписали смер
- N. Не знам

11. Извршењем упита:

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

биће приказано:

- | | | | | | |
|----|--------------------------------------|---|----|----------------------|---|
| A. | Софтверско инжењерство | 3 | D. | Синтаксна грешка | |
| B. | Управљање пословањем | 3 | E. | Информациони системи | 3 |
| C. | Све што је претходно наведено | | N. | Не знам | |

12. 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 <= 5; i++) {
            if (i % 2 == 0) {
                continue;
            }
            System.out.print(i);
        }
    }
}
```

- | | | | |
|-----------|------------|----|---------|
| A. | 12345 | D. | 345 |
| B. | 135 | E. | 12 |
| C. | 24 | N. | Не знам |

13. 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 = 7; int y = 3; int z = 1; int rezultat = 0;
        while (x > 0) {
            rezultat += z;
            z += 2;
            x -= y;
        }
        System.out.print(rezultat);
    }
}
```

- | | | | |
|----|----|-----------|----------|
| A. | 1 | D. | 7 |
| B. | 3 | E. | 9 |
| C. | 12 | N. | Не знам |

14. Погледати код који је дат испод. Шта ће метода `proveraVrednosti()` вратити као повратну вредност ако јој се као аргумент проследи вредност 200?

```
class Test {
    static String proveraVrednosti(int a) {
        if (a > 100 && a < 200)
            return "u opsegu";
        else
            return "van opsega";
    }
}
```

- | | | | |
|----|-------------------------------------|-----------|--------------------------------------|
| A. | u opsegu | D. | Долази до грешке у време компилације |
| B. | null | E. | van opsega |
| C. | Долази до грешке у време извршавања | N. | Не знам |

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

```
public class Test {
    public static void main(String[] args) {
        int[] niz = {3, 1, 4, 1, 5};
        for (int i = 1; i < niz.length; i++) {
            niz[i] = niz[i-1] * niz[i];
        }
        System.out.println(niz[4]);
    }
}
```

- A. 5
B. 10
C. 15
D. 120
E. 60
N. Не знам

16. Дата је табела:

Радна меморија	База знања
1) Паркет је нераван 2) Паркет је хрпав на додир 3) На паркету су крупне капљице лака	ПРАВИЛО 1: <u>IF</u> Паркет је нераван <u>OR</u> Паркет има много нијанси боја <u>THEN</u> Паркет је лоше хоблован ПРАВИЛО 2: <u>IF</u> Паркет је хрпав на додир <u>THEN</u> Паркет је лоше лакиран ПРАВИЛО 3: <u>IF</u> Паркет је лоше лакиран <u>AND</u> На паркету су крупне капљице лака <u>THEN</u> Употребљен је лош лак за паркет ПРАВИЛО 4: <u>IF</u> Паркет је лоше лакиран <u>AND</u> На паркету нису крупне капљице лака <u>THEN</u> Фали још један или више слојева лака

Ако је садржај радне меморије и базе знања као у табели и ако се зна да механизам за закључивање користи **уланчавање унапред**, одговорити на следеће питање: До којих ће све закључака доћи ЕС (од понуђених одговора изабрати све нове чињенице тј. закључке који ће бити додати у радну меморију)?

- A. Паркет је лоше хоблован
B. Паркет је лоше лакиран, Употребљен је лош лак за паркет
C. Паркет је лоше хоблован, Паркет је лоше лакиран, Фали још један или више слојева лака
D. Паркет је лоше хоблован, Паркет је лоше лакиран, Употребљен је лош лак за паркет, Фали још један или више слојева лака
E. **Паркет је лоше хоблован, Паркет је лоше лакиран, Употребљен је лош лак за паркет**
N. Не знам

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

```
public class Test {
    public static void main(String[] args) {
        String poruka = "Zdravo svete!";
        String izvod = poruka.substring(7);
        System.out.println(izvod);
    }
}
```

- A. Zdravo
B. Zdravo svete
C. **svete!**
D. vete!
E. avo svete
N. Не знам

18. Који је основни циљ *K-Means* алгоритма?
- A. Максимизација броја кластера
 - B. Расподела инстанци по унапред задатим класама
 - C. Максимизација броја итерација
 - D. Минимизација функције дисторзије (*distortion function*)**
 - E. Редукција димензионалности података
 - N. Не знам
19. *Deadlock* је мултитаскинг проблем који се дешава када:
- A. два процеса заузму ресурсе и међусобно ослобађају ресурсе.
 - B. два процеса ослободе ресурсе и међусобно се чекају да их поново заузму.
 - C. два процеса ослободе ресурсе и међусобно се чекају да други процеси заузму те ресурсе.
 - D. два процеса заузму ресурсе и постепено их ослобађају.
 - E. два процеса заузму ресурсе и међусобно се чекају да ослободе те ресурсе.**
 - N. Не знам
20. Metode `wait()`, `notify()` и `notifyAll()` могу се једино позвати из:
- A. `static` метода
 - B. синхронизованих метода**
 - C. `public` метода
 - D. перзистентних метода
 - E. простих метода
 - N. Не знам
21. Који су основни облици машинског учења?
- A. Ненадгледано, Хеуристичко, Детерминистичко
 - B. Надгледано, Статистичко, Хеуристичко
 - C. Ненадгледано, Полу-надгледано, Детерминистичко
 - D. Надгледано, Мета-learning, Трансфер learning
 - E. Надгледано, Ненадгледано, Учење уз подстицаје**
 - N. Не знам
22. Из чега се састоји тронивојска архитектура:
- A. контролера корисничког интерфејса, пословне логике и складишта података.
 - B. корисничког интерфејса, контролера и складишта података.
 - C. екранске форме, апликационе логике и складишта података.
 - D. корисничког интерфејса, пословне логике и складишта података.
 - E. корисничког интерфејса, апликационе логике и складишта података.**
 - N. Не знам
23. За податке се каже да су конзистентни ако задовољавају:
- A. вредносна и системска органичења дефинисана над тим подацима.
 - B. пројектна и структурна органичења дефинисана над тим подацима.
 - C. пројектна и системска органичења дефинисана над тим подацима.
 - D. вредносна и структурна органичења дефинисана над тим подацима.**
 - E. типска и програмска органичења дефинисана над тим подацима.
 - N. Не знам
24. Шта представља *False Positive (FP)* у класификацији?
- A. Број инстанци за које је модел предвидео да припадају позитивној класи, а заправо припадају негативној**
 - B. Број инстанци које су тачно класификоване као позитивне
 - C. Број инстанци које нису класификоване ни у једну класу
 - D. Број инстанци које су тачно класификоване као негативне
 - E. Број инстанци за које је модел предвидео да припадају негативној класи, а заправо припадају позитивној
 - N. Не знам

25. Који критеријум НИЈЕ типично коришћен за поделу простора атрибута у класификационим стаблима?
- A. *Gini indeks*
 - B. *Cross-entropy*
 - C. *Classification Error Rate*
 - D. *Mean Absolute Error***
 - E. Ниједан од наведених – сви су типични
 - N. Не знам
26. Шта од наведеног НИЈЕ основни контролни параметар *backpropagation* алгоритма?
- A. *Max error*
 - B. *Max iterations*
 - C. *Dropout ratio***
 - D. *Learning rate*
 - E. *Batch mode*
 - N. Не знам
27. Шта карактерише сигмоидну активациону функцију?
- A. Излаз мора бити негативан
 - B. Функција има оштар прекид у средини
 - C. Излаз је ограничен између 0 и 1**
 - D. Користи се само у излазном слоју
 - E. Функција увек даје вредност 1
 - N. Не знам
28. Шта ради механизам за закључивање у експертском систему?
- A. Комбинује знање из базе и чињенице из радне меморије да би створио закључке**
 - B. Управља комуникацијом између експерта и корисника да би дошао до закључка
 - C. Чува све податке о корисницима и њиховим уносима који су од значаја за закључивање
 - D. Визуелно приказује правила система која су се користила у поступку закључивања
 - E. Оптимизује брзину извршавања поступка закључивања у експертском систему
 - N. Не знам
29. Код упрошћене Ларманове методе развоја софтвера **системске операције** описују:
- A. Стање софтверског система
 - B. Перзистентност софтверског система
 - C. Ограничења софтверског система
 - D. Структуру софтверског система
 - E. Понашање софтверског система**
 - N. Не знам
30. Постоје два аспекта пројектовања екранске форме:
- A. пројектовање акција случаја коришћења које се изводе преко екранске форме и пројектовање контролера корисничког интерфејса.
 - B. пројектовање метода екранске форме и пројектовање контролера корисничког интерфејса.
 - C. пројектовање сценарија екранске форме и пројектовање контролера корисничког интерфејса.
 - D. пројектовање сценарија случаја коришћења које се изводе преко екранске форме и пројектовање контролера корисничког интерфејса.
 - E. пројектовање сценарија случаја коришћења који се изводе преко екранске форме и пројектовање метода екранске форме.**
 - N. Не знам