



Šifra kandidata:

**Državni izpitni center**



M 2 2 1 7 8 1 1 1

SPOMLADANSKI IZPITNI ROK

# RAČUNALNIŠTVO

==== Izpitna pola 1 ====

**Petek, 3. junij 2022 / 90 minut**

*Dovoljeno gradivo in pripomočki:  
Kandidat prinese nalivno pero ali kemični svinčnik.  
Konceptni list je na perforiranem listu, ki ga kandidat pazljivo iztrga.*

**SPLOŠNA MATURA**

## NAVODILA KANDIDATU

**Pazljivo preberite ta navodila.**

**Ne odpirajte izpitne pole in ne začenjajte reševati nalog, dokler vam nadzorni učitelj tega ne dovoli.**

Prilepite kodo oziroma vpišite svojo šifro (v okvirček desno zgoraj na tej strani).

Izpitna pola vsebuje 20 nalog. Število točk, ki jih lahko dosežete, je 40. Za posamezno nalogo je število točk navedeno v izpitni poli.

Rešitve pišite z nalivnim peresom ali s kemičnim svinčnikom v izpitno polo v za to predvideni prostor **znotraj okvirja**. Kadar je smiselno, narišite skico, čeprav je naloga ne zahteva, saj vam bo morda pomagala k pravilni rešitvi. Pišite čitljivo. Če se zmotite, napisano prečrtajte in rešitev zapišite na novo. Nečitljivi zapisi in nejasni popravki bodo ocenjeni z 0 točkami. Osnutki rešitev, ki jih lahko napišete na konceptni list, se pri ocenjevanju ne upoštevajo.

Zaupajte vase in v svoje zmožnosti. Želimo vam veliko uspeha.

*Ta pola ima 16 strani, od tega 1 prazno.*



V sivo polje ne pišite. V sivo polje ne pišite. V sivo polje ne pišite. V sivo polje ne pišite. V sivo polje ne pišite. V sivo polje ne pišite. V sivo polje ne pišite. V sivo polje ne pišite. V sivo polje ne pišite. V sivo polje ne pišite.



M 2 2 1 7 8 1 1 1 0 3

### Konceptni list



### Konceptni list

Empty rectangular area for writing the concept list.



M 2 2 1 7 8 1 1 1 0 5

1. Obkrožite črko pred trditvijo, ki je resnična za algoritem.

- A Algoritem je postopek, ki vodi do rešitve nekega problema.
- B Algoritem je problem, ki ga rešimo z nekim programom.
- C Program, ki ga za rešitev problema zaženemo na računalniku, imenujemo algoritem.
- D Algoritem je funkcija, če vrača neko vrednost.

(1 točka)

2. Narišite simbol, ki ga v diagramih poteka uporabimo za vhodno-izhodne operacije.

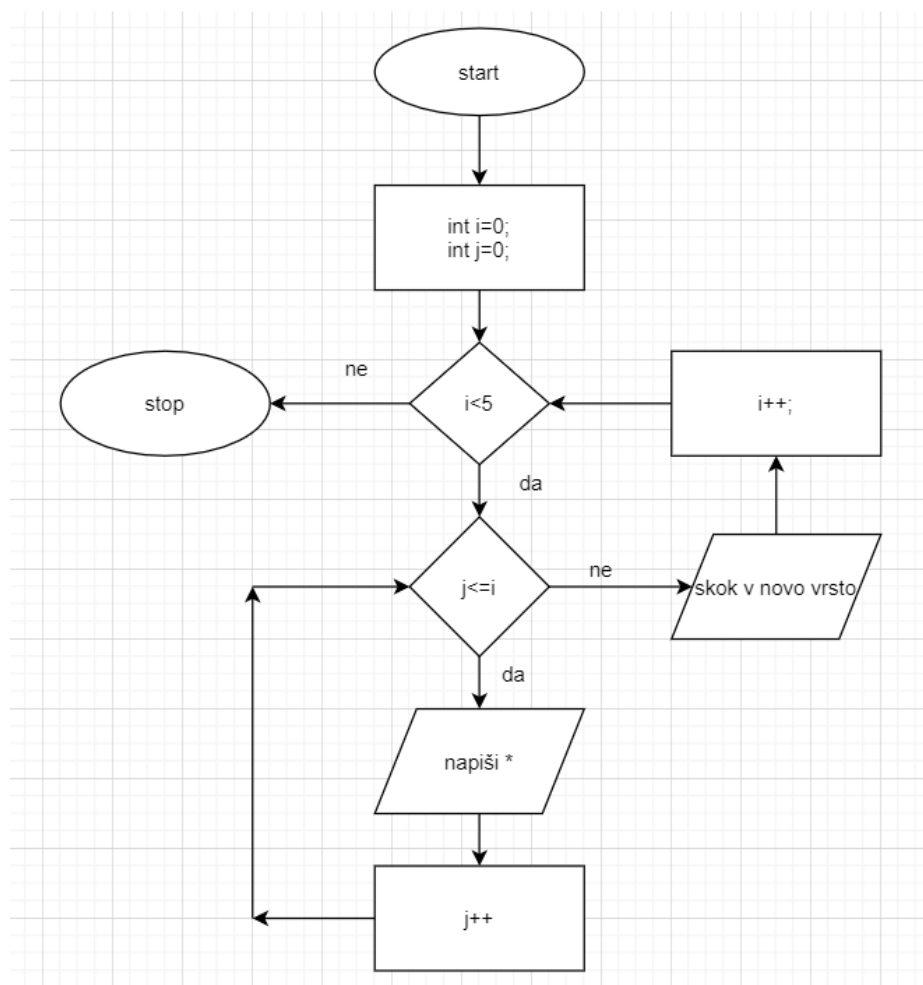
(1 točka)

3. Program, ki izvorno kodo, napisano v programskem jeziku, prevede v vmesno kodo ali v izvršljivo obliko, se imenuje \_\_\_\_\_.

(1 točka)



4. Kaj izpiše diagram poteka?



---

---

---

---

---

(2 točki)



M 2 2 1 7 8 1 1 1 0 7

5. Zapišite oznako primitivnega tipa podatka v programskem jeziku Java za

cela števila \_\_\_\_\_

logično vrednost \_\_\_\_\_

(1 točka)

6. Razred `Igralec` ima statično lastnost razreda `IgralnaKocka` z imenom `kocka`.  
Obkrožite črko pred resnično trditvijo.

- A Če imamo v programu več objektov razreda `Igralec`, potem ima vsak tak objekt svoj objekt razreda `IgralnaKocka`.
- B Če imamo v programu več objektov razreda `Igralec`, potem si vsi delijo en objekt razreda `IgralnaKocka`.
- C Lastnost `kocka` mora biti zaščiten (angl. *protected*).
- D Do lastnosti `kocka` lahko dostopamo le znotraj razreda `Igralec` in v njegovih izpeljanih razredih.

(1 točka)

7. Kaj izpiše izsek programa?

```
int x=100, y=8;
do {
    x=x/y;
    y=y+1;
    System.out.println(x + " " +y);
} while (x>10);
```

\_\_\_\_\_

\_\_\_\_\_

(2 točki)

8. Obkrožite črko pred opisom, ki je resničen za preoblaganje metod.

- A Java ne omogoča preoblaganja metod.
- B Metoda je preobložena, če je večkrat definirana in ima pri tem vsaka definicija metode različen tip rezultata, ime in parametri pa so pri vseh enaki.
- C Metoda je preobložena, če je večkrat definirana in ima pri tem vsaka definicija metode enako ime, mora pa imeti različen seznam parametrov od vseh drugih. Tip rezultata metode ni pomemben.
- D Metoda je preobložena, če je večkrat definirana in ima pri tem vsaka definicija metode različno ime, parametri pa so poljubni. Tip rezultata metode ni pomemben.

(1 točka)



9. Deklarirani in inicializirani sta celoštevilski spremenljivki  $x$  in  $y$ . V katerem primeru je smiselna uporaba bloka `try - catch`?

A

```
try {  
    System.out.println("Rezultat " + (x / y));  
} catch (RuntimeException e) {  
    System.out.println("Napaka");  
}
```

B

```
try {  
    System.out.println("Rezultat " + (x + y));  
} catch (RuntimeException e) {  
    System.out.println("Napaka");  
}
```

C

```
try {  
    System.out.println("Rezultat " + (x - y));  
} catch (RuntimeException e) {  
    System.out.println("Napaka");  
}
```

D

```
try {  
    System.out.println("Rezultat " + (x && y));  
} catch (RuntimeException e) {  
    System.out.println("Napaka");  
}
```

(1 točka)









14. Zapišite metodo `kolikoPozitivnih`, ki vrne število pozitivnih elementov celoštevilске tabele. Tabela naj bo podana kot parameter metode.

---

---

---

---

---

---

---

---

---

---

(2 točki)

15. Kaj izpiše izsek programa?

```
int a = 1;
Integer b = 1;
Integer c = new Integer(1);

System.out.println("a == b "+(a==b));
System.out.println("a == c "+(a==c));
System.out.println("b == c "+(b==c));
```

---

---

---

(3 točke)



## 16. Kaj izpiše program?

```
public class MyClass {
    public static class Razred1 {

        public int x = 4;
        private int y = 2;

        Razred1 () {
            x -= 1;
            y += 1;
        }
        Razred1 (int i) {
            this();
            y += x++;
            if (metoda()) System.out.print("x" + x + " y" + y);

        }
        private boolean metoda(){
            return (y%2==x%2);
        }
    }

    public static void main(String args[]) {
        Razred1 x = new Razred1(1);

    }
}
```

(2 točki)

## 17. Dopolnite spodnjo kodo z zanko do while, ki tvori naključna realna števila iz intervala [0, 1) in tvorjenje preneha, ko je zadnje naključno število večje od 0,75.

```
import java.util.Random;
public class MyClass {
    public static void main(String args[]) {
        Random rand = new Random();
```

```

    }
}
```

(2 točki)



18. Metoda `test_3_9` vrne `true`, če so v tabeli `tab` zapisane le številke 3 ali 9, sicer vrne `false`.  
Dopišite manjkajočo kodo metode `test_3_9`.

```
public boolean test_3_9(int[] tab) {
```

---

---

---

---

---

---

---

---

---

---

(2 točki)





