



---

---

**Državni izpitni center**

---

---



M 2 2 2 7 8 1 1 3

JESENSKI IZPITNI ROK

# **RAČUNALNIŠTVO**

---

---

NAVODILA ZA OCENJEVANJE

**Ponedeljek, 29. avgust 2022**

---

---

**SPLOŠNA MATURA**

---

---

Moderirana različica

## IZPITNA POLA 1

Naloga	Točke	Rešitev	Dodatna navodila
1	1	♦ C	
2	2	♦	Zanka in števec 1 točka, inicializacija, branje, izpis 1 točka.
<pre> graph TD     Start([Start]) --&gt; BeriTrapez[/beri N/]     BeriTrapez --&gt; I1[i=1]     I1 --&gt; N0{N=0}     N0 -- da --&gt; Stop([stop])     N0 -- ne --&gt; Izpisi[/izpisi i/]     Izpisi --&gt; I2[i=i*2 N=N-1]     I2 --&gt; N0 </pre>			
3	1	♦ A	
4	1	♦ 5 ali 5.0	
5	1	♦ B	
6	1	♦ B	

Naloga	Točke	Rešitev	Dodatna navodila
7	2	<ul style="list-style-type: none"> <li>◆ R</li> <li>◆ N</li> </ul>	Vsak odgovor 1 točka.
8	2	<pre>for (int i=100; i&gt;=0; i--){ }</pre>	Sintaksa zanke, vključno z inicializacijo in pogojem, 1 točka, sprememba 1 točka.
9	1	<ul style="list-style-type: none"> <li>◆ D</li> </ul>	Vsak odgovor 1 točka.
10	2	<ul style="list-style-type: none"> <li>◆ break</li> <li>◆ return</li> </ul>	Vsak odgovor 1 točka.
11	2	<pre>private static int zadnjaStevka (int x) { return x%10; }</pre>	Glava metode 1 točka, telo metode 1 točka.
12	3	<ul style="list-style-type: none"> <li>◆ static boolean stevke (String x) {  for (int i=0; i&lt;x.length(); i++)  if (x.charAt(i)&lt;'0'    x.charAt(i)&gt;'9') return false;  return true;  }</li> </ul>	Zanka 1 točka, pogoj in vračanje ustrezne vrednosti 1 točka, glava 1 točka.
13	3	<pre>public static String podniz (String niz, char start, char end) {  String rez = "";  for (int i=0; i&lt;niz.length(); i++)  if (niz.charAt(i)&gt;=start &amp;&amp; niz.charAt(i)&lt;=end)  rez += niz.charAt(i);  return rez;  }</pre>	Sprehod čez podani niz 1 točka, filtriranje znakov 1 točka, tvorjenje in vračanje novega niza 1 točka.
14	2	<ul style="list-style-type: none"> <li>◆ XXXXX</li> <li>XXXXO</li> <li>XOXOO</li> <li>XOOXO</li> <li>XOOOX</li> </ul>	Pravilni izpis prve vrstice 1 točka, pravilne ostale vrstice 1 točka.

<b>Naloga</b>	<b>Točke</b>	<b>Rešitev</b>	<b>Dodatna navodila</b>
15	1	♦ C	
16	3	<pre> ♦ static boolean prastevilo(int a){     int d;     for (d=2; d&lt;= a/2; d++){         if (a%d==0) return false;     }     return true; }  int vsota=0; for (int i=0; i&lt;x.length; i++)     if (prastevilo(x[i])) vsota+=x[i]; </pre>	Metoda <code>prastevilo</code> 1 točka, sprehod po tabeli 1 točka, klic metode in računanje vsote 1 točka.
17	3	<pre> ♦ public static boolean vsiRazlicni(int[] x) {     for(int i=0; i&lt;x.length; i++)         for(int j=i+1; j&lt;x.length; j++)             if (x[i]==x[j])                 return false;     return true; } </pre>	Glava metode 1 točka, sprehod po vseh parih, ki jih je treba preveriti, 1 točka, ustrezno vračanje rezultata 1 točka.
18	1	♦ D	
<b>Naloga</b>	<b>Točke</b>	<b>Rešitev</b>	<b>Dodatna navodila</b>
19.1	2	<pre> ♦ public class Igralec {     protected double x, y;     protected String ime;     protected int id;     public static int koliko = 0; } </pre>	Razred z nestatičnimi lastnostmi 1 točka, statična lastnost 1 točka.

<b>Naloga</b>	<b>Točke</b>	<b>Rešitev</b>	<b>Dodatna navodila</b>
<b>19.2*</b>	<b>2</b>	<pre> ♦ public Igralec(double x, double y) {     this(x,y,"Player "+(koliko+1)); } public Igralec(String ime) {     this(0,0,ime); } </pre>	Glavi obeh konstruktorjev 1 točka, uporaba/klic obstoječega konstruktorja 1 točka.

<b>Naloga</b>	<b>Točke</b>	<b>Rešitev</b>	<b>Dodatna navodila</b>
<b>20.1</b>	<b>2</b>	<pre> ♦ public class Vrsta {     private int Stevilka;     private int Zadnji;     private int[] cakajoci = new int[10];      public Vrsta(){         Zadnji=-1;         Stevilka=0;     }      public void prihodStranke(){         Stevilka++;         if (Zadnji&lt;9){             Zadnji++;             cakajoci[Zadnji]=Stevilka;             System.out.println ("V vrsto dodana stranka s številko "+                 Stevilka);             System.out.println ("Število strank v vrsti: "+ (Zadnji +1));         }         else {             System.out.println ("Vrsta je polna, stopite k sosednjem okencu");         }     } } </pre>	Lastnosti in konstruktor 1 točka, metoda prihodStranke 1 točka.

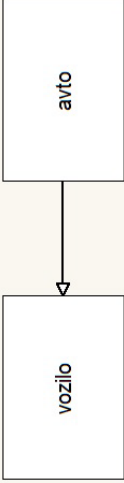
\* Zaradi napake v nalogi vsi kandidati pri tej postavki prejmejo 2 točki.

Naloga	Točke	Rešitev	Dodatna navodila
20.2	2	<pre>♦ public void odhodStranke(){     if (Zadnji&gt;-1) {         System.out.println("Odhod stranke " + cakaJoci[0]);         Zadnji--;         for (int i=0; i&lt;=Zadnji; i++)             cakaJoci[i]=cakaJoci[i+1];         else System.out.println("Vrsta je prazna");     } }</pre>	Premikanje čakajočih 1 točka, preverjanje, ali je vrsta prazna, 1 točka.

Skupno število točk IP 1: 40

**IZPITNA POLA 2**


<b>Naloga</b>	<b>Točke</b>	<b>Rešitev</b>	<b>Dodatna navodila</b>
1	1	♦ register	
2	2	♦ eniški komplement: 11101011 ♦ dvojiški komplement: 11101100	Vsak odgovor 1 točka.
3	1	♦ C	
4	1	♦ B	
5	1	♦ C	
6	2	♦ A, C	Vsak odgovor 1 točka, že en nepravilen odgovor 0 točk.
7	2	♦ $(1000 \times 1000 \times 3 \times 8) / (10 \times 1000000) = 2,4$ s ♦ $1000 \times 1000 \times 3 \times 8$	Vsak odgovor 1 točka.
8	3	♦ 172.16.16.63 ♦ Broadcast ali poplavljanje ali razpršeno oddajanje ♦ 255.255.255.192	Vsak odgovor 1 točka.
9	1	♦ C	
10	2	dve od: ♦ FTP ( <i>File Transfer Protocol</i> ) – prenos datotek ♦ SMTP ( <i>Simple Mail Transfer Protocol</i> ) – pošiljanje pošte ♦ POP3 – prejetanje pošte ♦ IMAP – prejetanje pošte ♦ TELNET ali SSH – oddaljeno povezovanje ♦ HTTPS – varno prenašanje spletnih strani	Vsak pravilen protokol z opisom 1 točka. Upoštevajo so tudi drugi protokoli aplikacijske plasti.
11	1	♦ C	
12	3	♦ A 6 B 3 C 1 D 2 E 2 F 5 G 1	Dve ali tri pravilne povezave 1 točka. Štiri ali pet pravilnih povezav 2 točki. Šest ali sedem pravilnih povezav 3 točke.
13	2	♦ N ♦ R	Vsak pravilen odgovor 1 točka.

Naloga	Točke	Rešitev	Dodatna navodila
14	1	♦ B	
15	1	♦	
16	3	♦ Create table Točka (x int not null, y int not null, barva varchar(20) not null, primary key(x,y)); ♦ Drop table Točka;	Ustvarjanje tabele z atributi in integritetnimi omejitvami 1 točka, primarni ključ 1 točka, brisanje tabele 1 točka.
17	2	♦ B ♦ C	Vsak odgovor 1 točka, že en nepravilen odgovor 0 točk.
18	3	♦ Inventurni_kos(evidencna_st:A10, kat:A3->Kategorija, status:A15, IDprostora:A5->Prostor, zadolzen:A8->Oseba) Oseba(os_stev:A8, ime:A15, priimek:A20) Prostor(IDprostora:A5, opis:A20) Kategorija(kat:A3, opis_kategorije:A20)	Sheme in lastnosti 1 točka, primarni ključ 1 točka, tuji ključ 1 točka. Prizna se tudi rešitev brez tabele Prostor.

Naloga	Točke	Rešitev	Dodatna navodila
19.1	2	♦ Select d.priimek, d.ime From dijak d inner join rezultat r on (d.solaID=r.solaID) and (d.ZapSt=r.zapSt) Inner join tekmovanje t on (r.TID=t.TID) Where r.Mesto=1 and t.ImeTekmovanja='Bralna značka 2018' Order by d.priimek;	Povezovanje 1 točka, filtriranje in urejanje 1 točka.

Naloga	Točke	Rešitev	Dodatna navodila
19.2	2	♦ Select t.ImeTekmovanja, s.ImeSole, count(d.zapSt) as sTekmovalcev From tekmovanje t inner join rezultat r on (t.TID=r.TID) inner join dijak d on (d.solaID=r.solaID) and (d.ZapSt=r.zapSt) inner join sola s on (d.solaID=s.solaID) group by t.ImeTekmovanja,s.ImeSole;	Povezovanje tabel 1 točka, funkcija in združevanje 1 točka.



Naloga	Točke	Rešitev	Dodatna navodila
20.1	4	 <pre> erDiagram     Kategorija   --o{ Igra : "ima"     Igra   --o{ Rezultatigre : "ima"     Rezultatigre   --o{ Igralec : "ima"      Kategorija {         string imeKategorije     }     Igra {         int IgraID PK         int KID FK         string imeIgre     }     Rezultatigre {         int IgraID PFK         int IgralecID PFK         int datum     }     Igralec {         int IgralecID PK         string uporabniškoIme         string geslo         string eMail         int datumRegistracije     } </pre>	<p>Entitetni tip Igralec z atributi in ključem 1 točka, entitetna tipa Kategorija in Igra z atributi in vsemi ključi 1 točka. Rezultatigre z atributi in ključi 1 točka, povezave in števnosti 1 točka.</p>

Skupno število točk IP 2: 40