



---

---

**Državni izpitni center**

---

---



JESENSKI IZPITNI ROK

# **RAČUNALNIŠTVO**

---

---

NAVODILA ZA OCENJEVANJE

**Petek, 29. avgust 2025**

---

---

**SPLOŠNA MATURA**

---

---

## IZPITNA POLA 1

Naloga	Točke	Rešitev	Dodatna navodila
1	1	◆ D	
2	1	◆ B	
3	2	◆	Branje in izpis – 1 točka, ustrezná zanka – 1 točka.
		<pre> graph TD     Start([začetek]) --&gt; Read[/Vhod: prvo, drugo/]     Read --&gt; AssignPrvo[prvo =  prvo ]     AssignPrvo --&gt; Decision{prvo &gt; 0}     Decision -- true --&gt; Print[/Izhod: drugo/]     Print --&gt; DecPrvo[prvo = prvo - 1]     DecPrvo --&gt; Decision     Decision -- false --&gt; End([konec]) </pre>	
4	2	◆ final double konst=1.23456; System.out.format("konst = %.2f", konst);	Deklaracija konstante – 1 točka, izpis na dve decimalni – 1 točka.

<b>Naloga</b>	<b>Točke</b>	<b>Rešitev</b>	<b>Dodatna navodila</b>
5	1	♦ x%5==0	
6	1	♦ c1 = D	
7	2	♦ 2 2 2 3 5	Prve tri vrednosti – 1 točka, vse pravilne – 2 točki.
8	2	♦ for(int i=30; i<=330; i=i+3) System.out.println(i);	Zanka z ustreznimi mejami – 1 točka, izpis želenih števil – 1 točka.
9	2	♦ new Scanner, catch (Exception e)	Vsak manjkajoči izraz – 1 točka.
10	1	♦ B	
11	3	♦ static boolean aliVsebuje(int x, int n){ while (n>9) { if (n%10==x) return true; n=n/10; } if (n==x) return true; else return false; }	Glava metode – 1 točka, zanka s preverjanjem števk – 1 točka, ustrezno vračanje rezultata – 1 točka.
12	2	♦ 15 5	Vsaka vrednost – 1 točka.
13	2	♦ char [][]tab = new char[5][5]; for (int i=0; i<tab.length; i++){ for (int j=0; j<tab.length; j++){ if (i % 2 == 0) if (j % 2 == 0) tab[i][j] = '*'; else tab[i][j] = ' '; else if (j % 2 == 1) tab[i][j] = '*'; else tab[i][j] = ' '; System.out.print(tab[i][j] + " "); } System.out.println(); }	Pravilni pogoji za polnjenje tabele – 1 točka, zanki in izpis tabele – 1 točka.

Naloga	Točke	Rešitev	Dodatna navodila
14	2	<pre> public static int istolezni(double tabela[]) {     int stevec=0;     for(int i=0; i&lt;tabela.length; ++i)         if (tabela[i]==tabela[tabela.length-1-i])             ++stevec;     return stevec; } </pre>	Števec, vračanje in sprehod po tabeli – 1 točka, primerjava ustreznih elementov – 1 točka.
15	1	<pre> String []besede = niz1.split(" "); </pre>	
16	2	<pre> String niz = "Danes je lep dan"; int stevec = 0; niz.toLowerCase(); for (int i = 0; i&lt;niz.length(); i++)     if (niz.charAt(i) = 'a'    niz.charAt(i) = 'e'            niz.charAt(i) = 'i'    niz.charAt(i) = 'o'            niz.charAt(i) = 'u')         stevec ++; </pre>	Šteje – 1 točka, sprehod po nizu –1 točka.
17	3	<pre> public static boolean kontrolaRK(String rk) {     if (rk.substring(0,2).equals("RK") != true    rk.length() != 14)         return false;     String znak1 = Integer.toString((int) rk.charAt(0) - 'A' + 10);     String znak2 = Integer.toString((int) rk.charAt(1) - 'A' + 10);     String zaTest =         rk.substring(4, rk.length()) + znak1 + znak2 + rk.substring(2, 4);     return (Long.parseLong(zaTest) % 97 == 0); } </pre>	Kontrola prvih dveh znakov in dolžine niza – 1 točka, pretvorba obeh črk v številski niz – 1 točka, pravilno zlaganje niza, pretvorba v long in test ostanka ter vračanje – 1 točka.
18	2	♦ 10 3 20 3 20 3	Pravilne vrednosti za nestatične lastnosti – 1 točka, pravilne za statične lastnosti – 1 točka.

<b>Naloga</b>	<b>Točke</b>	<b>Rešitev</b>	<b>Dodatna navodila</b>
19.1	2	<pre> public Dijak(String ime, String priimek, int starost){     this.ime = ime;     this.Priimek = priimek;     this.Starost = starost; }  public static void main(String[] args) {     Dijak []dijaki = {         new Dijak("Tone", "Kos", 16),         new Dijak("Tine", "Urh", 18),         new Dijak("Meta", "As", 17),     }; } </pre>	Konstruktor – 1 točka, tabela z objekti – 1 točka.
19.2	2	<pre> public static Dijak[] uredi (Dijak[] dijaki){     for(int i=0; i&lt; dijaki.length-1; i++)         for(int j=0; j &lt; dijaki.length-1; j++)             if(dijaki[j].ime &gt; dijaki[j+1].ime)                 {                     Dijak t = dijaki[j];                     dijaki[j] = dijaki[j+1];                     dijaki[j+1] = t;                 }     return dijaki;} </pre>	Glava metode urejanje – 1 točka, metoda urejanja – 1 točka.

<b>Naloga</b>	<b>Točke</b>	<b>Rešitev</b>	<b>Dodatna navodila</b>
20.1	4	<pre> public static int vecinski(int[] tab) {     for(int i=0; i&lt;tab.length; ++i) {         int st = 0;         for(int j=0; j&lt;tab.length; ++j)             if (tab[j]==tab[i])                 ++st;         if (st&gt;tab.length/2)             return tab[i];     }     return -1; } </pre>	Glava metode – 1 točka, štetje ponovitev enega elementa – 1 točka, ustrezen pregled vseh možnih kandidatov za večinski element – 1 točka, ustrezno vračanje rezultata – 1 točka.

Skupno število točk IP 1: 40

## IZPITNA POLA 2

Naloga	Točke	Rešitev	Dodatna navodila
1	2	♦ kontrolno (krmilina) enota, registrov	Vsak pravilen odgovor – 1 točka.
2	1	♦ B	
3	2	♦ vzdolžna: 0 1, prečna: 1 1 1	Vsaka pravilna pariteta – 1 točka.
4	1	♦ cat pozdrav.txt matura.txt > obvestilo.txt	
5	1	♦ chmod 574 vaja1.txt ali chmod u=rx,g=rwx,o=r vaja1.txt	
6	2	♦ mkdir podatki touch podatki/danes.txt	Vsak odgovor – 1 točka; upoštevamo tudi druge načine ustvarjanja datoteke.
7	1	♦ D	
8	1	♦ B	
9	1	♦ A	
10	3	♦ M: 255.255.255.128 ali M: 255.255.255.128 ♦ ID: 192.168.4.0 ID: 192.168.4.128 ♦ B: 192.168.4.127 B: 192.168.4.255	Vsak pravilna alineja – 1 točka.
11	2	♦ Osmiško: 2572 ♦ Dvojiško: 0101 0111 1010	Vsak pravilen odgovor – 1 točka.
12	3	♦ A – 1, B – 2, C – 3	Vsaka pravilna povezava – 1 točka.
13	2	dve od: ♦ diagram objektov (angl. <i>object diagram</i> ) ♦ diagram aktivnosti (angl. <i>activity</i> ) ♦ diagram zaporedja (angl. <i>sequence diagram</i> ) ♦ diagram stanja (angl. <i>statechart, state diagram, state machine</i> )	Vsak pravilen odgovor – 1 točka.
14	1	♦ A	
15	2	♦ B, E	Vsak pravilen odgovor – 1 točka, že en napačen odgovor – 0 točk.
16	3	♦ neresnično ♦ resnično ♦ neresnično	Vsak pravilen odgovor – 1 točka.
17	1	♦ ALTER TABLE Komentarji ADD datum DATE NOT NULL;	

Naloga	Točke	Rešitev	Dodatna navodila
18	3	<pre> Create table Test (   tID int primary key,   datum date not null); create table Naloga (   tID int not null,   nID int not null,   besedilo varchar(30) not null,   tocke int not null,   primary key (tID,nID),   foreign key (tID) references Test(tID)); </pre>	<p>Tabela Test – 1 točka,  tabela Naloga s primarnim ključem – 1 točka,  tuji ključ tabele Naloga – 1 točka.</p>
19.1	4	<pre> graph TD     Član((Član))     Knjižničar((Knjižničar))          subgraph knjižnica         registracija((registracija))         iskanje_knjig((iskanje knjig))         vračilo_knjig((vračilo knjig))         izposoja_knjig((izposoja knjig))         obveščanje((obveščanje o zamujenih knjigah))         upravljanje_zalog((upravljanje zalog))     end          Član --- registracija     Član --- iskanje_knjig     Član --- izposoja_knjig     Knjižničar --- vračilo_knjig     Knjižničar --- obveščanje     Knjižničar --- upravljanje_zalog          vračilo_knjig -.-&gt; &lt;&lt;include&gt;&gt;  obveščanje     izposoja_knjig -.-&gt; &lt;&lt;extend&gt;&gt;  obveščanje </pre>	<p><b>Dodatna navodila</b>  Akterji in sistem – 1 točka,  primeri uporabe – 1 točka,  povezave med akterji in primeri uporabe – 1 točka,  povezave med primeri uporabe – 1 točka.</p>

<b>Naloga</b>	<b>Točke</b>	<b>Rešitev</b>	<b>Dodatna navodila</b>
<b>20.1</b>	<b>3</b>	<pre> ♦ select Dijak.priimek, Dijak.ime from Dijak inner join Oddelek on (Dijak.oid=Oddelek.oid) where Oddelek.imeRazrednika='Miha' and Oddelek.priimekRazrednika='Novak' order by Dijak.priimek; </pre>	Povezovanje tabel – 1 točka, filtriranje – 1 točka, razvrščanje – 1 točka.
<b>20.2</b>	<b>1</b>	<pre> ♦ select Dijak.oid, Dijak.ime, Dijak.priimek, max(Dijak.datumROjstva) from Dijak group by Dijak.oid; </pre>	

**Skupno število točk IP 2: 40**