

# RAČUNALNIŠTVO

**Predmetni izpitni katalog za splošno matura** ■

---

Predmetni izpitni katalog se uporablja od spomladanskega roka **2009**, dokler ni določen novi. Veljavnost kataloga za leto, v katerem bo kandidat opravljal matura, je navedena v Maturitetnem izpitnem katalogu za splošno matura za tisto leto.



# VSEBINA

1. Uvod	5
2. Izpitni cilji	6
3. Zgradba in vrednotenje izpita	8
3.1 Shema izpita	8
3.2 Tipi nalog in vrednotenje	8
3.3 Merila vrednotenja izpita in posameznih izpitnih delov	10
4. Izpitne vsebine	11
4.1 Osnove informatike	11
4.2 Programski jeziki in programiranje	12
4.3 Arhitektura in organizacija računalniških sistemov	14
4.4 Računalniške komunikacije	15
5. Kandidati s posebnimi potrebami	16
6. Literatura	17
7. Primeri izpitnih nalog	18
8. Seznam uporabljenih kratic	20



# 1. UVOD

**P**redmetni izpitni katalog za splošno maturo iz računalništva je namenjen kandidatom, ki želijo opravljati maturitetni izpit iz tega predmeta po končani tehniški gimnaziji.

Katalog določa:

- izpitne cilje,
- cilje preverjanja,
- shemo preverjanja,
- način ocenjevanja,
- izpitne vsebine,
- sestavo testa.

Pri izpitnih ciljeh sloni katalog na vsebinah predmeta računalništvo, ki se poučuje v 2., 3. in 4. letniku tehniške gimnazije.

## 2. IZPITNI CILJI

### Splošni izpitni cilji

Kandidat naj razvije:

- poznavanje in razumevanje osnovnih načel računalništva;
- poznavanje uporabe računalnikov na raznih področjih;
- sposobnost formuliranja problemov in načrtovanja rešitev;
- sposobnost razumevanja računalniških organizacij in sistemov, z aparaturno, aplikacijsko in sistemsko programsko opremo vred;
- poznavanje računalniških komunikacij;
- poznavanje razlik med programskimi jeziki in med programskim orodjem;
- veščine uporabe programskega orodja.

### Cilji preverjanja

1) Kandidat naj bo usposobljen, da opiše in pojasni:

- uporabo računalnikov v različnih aplikacijah,
- potrebo po podatkovnih strukturah in osnovnih programskih konstrukcijah ter njihovo uporabo,
- sistematični razvoj rešitev problemov in tehnike, primerne za uporabo v takih rešitvah,
- gradnike aparaturne opreme, njihove funkcije in povezave,
- gradnike sistemske programske opreme, njihove funkcije in povezave,
- osnovne pojme iz informatike.

2) Kandidat naj pokaže sposobnost:

- analize problema in njegove priprave za računalniško obdelavo,
- izbire ustreznih tehnik in načel za razvoj struktur in algoritmov, primernih za rešitev problema,
- analize delovanja in sodelovanja posameznih gradnikov aparaturne opreme,
- analize delovanja in sodelovanja posameznih delov sistemske programske opreme,
- analize sodelovanja aparaturne in sistemske programske opreme,
- uporabe orodja iz informatike.

3) Kandidat naj zna uporabiti znanje, kar pokaže z:

- uporabo znanja o programski in aparaturni opremi za učinkovito reševanje nalog,
- analizo ponujenih rešitev nalog,
- izdelavo seminarske – projektne naloge.

Posamezni izpitni cilji imajo naslednjo težo:

<b>Cilj</b>	<b>Teža cilja</b>
1	20 %–30 %
2	40 %–60 %
3	20 %–30 %

Navedeni odstotki veljajo za pisni izpit, pri seminarski – projektni nalogi pa se upoštevajo le smiselno.

## 3. ZGRADBA IN VREDNOTENJE IZPITA

### 3.1 SHEMA IZPITA

#### Pisni del

Izpitna pola	Čas reševanja	Delež pri oceni	Ocenjevanje	Pripomočki
1	110 minut	45 %	zunanje	nalivno pero ali kemični svinčnik
2	70 minut	30 %	zunanje	

Med reševanjem Izpitne pole 1 in Izpitne pole 2 je predviden 30-minutni odmor.

#### Seminarska – projektna naloga

	Delež pri oceni	Ocenjevanje	Pripomočki
Seminarska – projektna naloga (lahko tudi več dijakov skupaj) se izdelava v 4. letniku pod mentorjevim vodstvom.	25 %	notranje pri javnem zagovoru	vse, kar potrebujejo pri izdelavi seminarske – projektne naloge

### 3.2 TIPI NALOG IN VREDNOTENJE

Izpit	Tipi nalog	Vrednotenje
Izpitna pola 1	16 nalog izbirnega tipa: – 8 nalog iz poglavja Arhitektura in organizacija računalniških sistemov – 8 nalog iz poglavja Programski jeziki in programiranje	za vsako nalogo 2 točki
	4 naloge s kratkimi odgovori: – vse iz poglavja Programski jeziki in programiranje	za vsako nalogo 2 točki
	4 strukturirane naloge: – vse iz poglavja Programski jeziki in programiranje	za vsako nalogo 4 točke
Izpitna pola 2	10 nalog izbirnega tipa: – 6 nalog iz poglavja Informatika – 4 naloge iz poglavja Računalniške komunikacije	za vsako nalogo 2 točki
	4 naloge s kratkimi odgovori: – vse iz poglavja Informatika	za vsako nalogo 2 točki
	2 strukturirani nalogi: – obe iz poglavja Informatika	za vsako nalogo 4 točke



Izpit	Tipi nalog	Vrednotenje
Seminarska – projektna naloga	<p>Seminarska – projektna naloga naj bo sestavljena iz naslednjih delov: projekta, dokumentacije in predstavitve oziroma zagovora.</p> <p>Dokumentacija vsebuje:</p> <ul style="list-style-type: none"> <li>– naslovni list, ki ga izda šola naslov (ime avtorja, mentorji, šola) definicija naloge</li> <li>– povzetek in ključne besede (slovenščina in angleščina)</li> <li>– seznam kratic in okrajšav</li> <li>– vsebinsko kazalo</li> <li>– UVOD</li> <li>– JEDRO</li> <li>– ZAKLJUČEK</li> <li>– literatura</li> <li>– stvarno kazalo (možnost)</li> <li>– priloge (možnost)</li> </ul>	Seminarska – projektna naloga se oceni z največ 50 točkami.

## **3.3 MERILA VREDNOTENJA IZPITA IN POSAMEZNIH IZPITNIH DELOV**

### **■ Izpitni poli**

Vsak pravilen odgovor na postavljeno vprašanje prinaša v izpitni poli navedeno število točk. Zunanji ocenjevalec za pravilne odgovore dodeli ustrezno število točk.

### **■ Seminarska – projektna naloga**

Seminarska – projektna naloga se ocenjuje po naslednjih merilih:

#### **A) OCENA SEMINARSKE – PROJEKTNE NALOGE (25 točk)**

1. Predstavitev in analiza problema (5 točk)
2. Načrt izvedbe (5 točk)
3. Vsebinska izvedba seminarske – projektne naloge – verifikacija in validacija (10 točk)
4. Testiranje (5 točk)

#### **B) OCENA DOKUMENTACIJE (15 točk)**

1. Popolnost dokumentacije (5 točk)
2. Kakovost vsebine posameznih točk (10 točk)

#### **C) OCENA ZAGOVORA SEMINARSKE – PROJEKTNE NALOGE (10 točk)**

1. Sestava predstavitve (5 točk)
2. Jezik in nastop na predstavitvi (5 točk)

Seminarska – projektna naloga se oceni z največ 50 točkami.

### **■ Ocenjevanje**

Točke, dosežene pri pisnem izpitu, in točke, prejete za seminarsko – projektno nalogo, se preoblikujejo v skupno točkovno oceno (STO) po naslednji formuli:

$$\text{STO} = \frac{\text{doseženo št. točk}_{(\text{iz. p. 1})}}{\text{maks. št. točk}_{(\text{iz. p. 1})}} \times 45 \% + \frac{\text{doseženo št. točk}_{(\text{iz. p. 2})}}{\text{maks. št. točk}_{(\text{iz. p. 2})}} \times 30 \% \\ + \frac{\text{doseženo št. točk}_{(\text{sem. nal.})}}{\text{maks. št. točk}_{(\text{sem. nal.})}} \times 25 \%$$

Državna maturitetna komisija za splošno maturo na predlog Državne predmetne komisije za splošno maturo iz računalništva določi merila, kako se iz uspeha pri pisnem izpitu in projektu, izraženega v skupni točkovni oceni, določi uspeh na lestvici od nezadostno (1) do odlično (5). Način pretvorbe je enak za spomladanski in jesenski rok.

## 4. IZPITNE VSEBINE

### 4.1 OSNOVE INFORMATIKE

#### ■ VSEBINA, POJMI

#### ■ CILJI

##### 4.1.1 TEMELJNI POJMI

I. Kandidat naj pozna opredelitev pojmov: podatek, informacija, življenjski cikel, entiteta, atribut, proces, dogodek, objekt, aplikacija, informacijski sistem.

II. Kandidat naj razume pretvorbe med številskimi sestavi.

##### 4.1.2 INFORMACIJSKI SISTEMI

I. Kandidat naj opiše informacijski sistem, določi njegove komponente, opiše temeljni proces in upravljavski proces.

II. Kandidat razloži zgradbo informacijskega sistema in poda kategorizacijo informacijskih sistemov.

##### 4.1.3 TEHNOLOŠKI VIDIK INFORMACIJSKIH SISTEMOV

II. Kandidat naj opiše funkcije orodja za računalniško podprto programsko inženirstvo (CASE), prototipno razvojno okolje, generacije programskih jezikov (s poudarkom na jezikih 4. generacije in na vizualnem razvoju aplikacij), sisteme za upravljanje podatkovnih zbirk, smeri razvoja informacijske tehnologije (vpogled v najaktualnejše dogajanje).

##### 4.1.4 ODLOČITVENI SISTEMI

I. Kandidat naj formulira pojem informacije kot podlage za sprejemanje odločitev.

II. Predstavi vmesnik človek–stroj, razloži kategorizacijo, pozna in razume odločitvene tabele.

##### 4.1.5 EKONOMSKI IN DRUŽBENI VIDIKI INFORMATIKE

I. Kandidat naj navede vpliv informatike na ljudi in organizacije.

II. Pojasni ekonomske vidike, opiše vpliv na družbo, zaposlovanje, zaščito in varovanje osebnih podatkov.

##### 4.1.6 METODE SISTEMSKÉ ANALIZE

III. Kandidat naj pozna, razume in pripravi opis zahtev z uporabo diagramov toka podatkov, podatkovnega slovarja in specifikacije prvinskih procesov.

##### 4.1.7 METODE SISTEMSKÉGA NAČRTOVANJA

II. Kandidat naj pozna in razume uporabo strukturnega načrtovanja, ki temelji na rezultatih systemske analize in je podlaga za izvedbo, razloži pojma enovitost in povezanost modulov.

##### 4.1.8 DATOTEKE

I. Kandidat naj loči zapis (record), datoteko (file) kot organizirano zbirko zapisov enakih ali različnih dolžin.

II. Pozna načine dostopov do posameznih zapisov, operacije nad zapisi, vrste organizacij datotek, zaščito in arhiviranje datotek (neurejene, zaporedne, razpršene, indeksirane datoteke).

##### 4.1.9 PODATKOVNE ZBIRKE

I. Kandidat naj našteje temeljne značilnosti, vrste zbirk (hierarhične, mrežne, relacijske, objektne).

II. Predstavi zgradbo podatkovne zbirke.

III. Predlaga in uporabi orodje za delo s podatkovnimi zbirkami.

##### 4.1.10 RELACIJSKI PODATKOVNI MODEL

II. Našteje in uporabi osnovne operacije nad relacijami (relacijska algebra).

III. Kandidat naj uporabi relacije, določi relacijske sheme, načrtuje konceptualne in logične sheme podatkovnih zbirk za enostavne primere iz okolja.

## **4.2 PROGRAMSKI JEZIKI IN PROGRAMIRANJE**

### ■ VSEBINA, POJMI

### ■ CILJI

#### **4.2.1 PROGRAMSKI JEZIK**

- I. Kandidat razvrsti programske jezike po ravneh (strojni jezik – zbirni jezik – višji programski jeziki).
- II. Razume različne programske paradigme (postopkovno programiranje, objektno usmerjeno programiranje).

#### **4.2.2 ALGORITEM**

- I. Kandidat razloži vlogo programskega jezika kot sredstva za komunikacijo med razvijalcem programske opreme in strojno opremo.
- I. Kandidat predstavi osnovne zahteve za algoritem (nedvoumnost in elementarnost korakov ter končnost).
- II. Prikaže načrtovanje algoritmov in njihovo pretvorbo v programe.
- III. Kandidat predstavi preproste algoritme v obliki diagrama poteka ali psevdokode.

#### **4.2.3 OSNOVNI PODATKOVNI TIPI**

- I. Kandidat našteje osnovne podatkovne tipe in opiše njihovo predstavitev v računalniku (cela števila, realna števila, znaki, logične vrednosti).
- II. Kandidat razloži pojem podatkovnega tipa (vsak podatek pripada točno določenemu tipu, s tipom so določene zaloga vrednosti in operacije, ki jih lahko izvajamo).
- II. Predstavi različne načine kodiranja števil in znakov.
- III. Kandidat uporabi enostavne podatkovne tipe pri pisanju programov in metod.

#### **4.2.4 STAVKI**

- II. Kandidat pozna in razume delovanje osnovnih stavkov programskega jezika.
- II. Opiše pravila za pisanje izrazov v priredilnem stavku (vrste operatorjev, prioritete operatorjev, skladnost podatkovnih tipov).
- III. Kandidat sledi izvajanju bloka stavkov.
- III. Kandidat prikaže delovanje stavkov, ki so v nekem programskem jeziku na voljo za uresničitev algoritma.
- III. Oceni prednosti in slabosti izbire različnih stavkov za izvedbo ponavljanja in izbire.
- III. Kandidat zapiše programske konstrukte (zaporedje, ponavljanje, izbira) z različnimi stavki.

#### **4.2.5 METODE**

- II. Kandidat opiše vrste metod, opiše osnovne pojme povezane z njimi (formalni parametri, dejanski parametri, lokalne spremenljivke, tip rezultata, deklaracija metode, klic metode).
- III. Kandidat razume smiselnost uporabe metod pri pisanju programov, opiše različne načine prenosa parametrov (po vrednosti, po referenci), razloži dostopnost spremenljivk (lokalne, globalne).
- III. Kandidat iz dane metode ugotovi, kaj ta naredi. V enostavni metodi odkrije in popravi sintaktične in logične napake, napiše in uporabi metode za dobro definirano nalogo.

#### 4.2.6 RAZREDI IN OBJEKTI

- II. Kandidat razume pojem razreda (opis zgradbe objekta) in pojem objekta (konkreten primerek nekega razreda).
- II. Kandidat razloži zgradbo objektov (atributi in metode).
- II. Opiše postopek generiranja objektov (operator `new`), njihovo predstavitev v pomnilniku in vlogo konstruktorja.
- II. Kandidat razume pojem ograjevanja (enkapsulacije) ter se zaveda lokalnosti ograjenih metod in atributov. Pozna vlogo vmesnika objekta za interakcijo z drugimi objekti.
- II. Pojasni pomen dostopnih določil `public`, `protected` in `private` ter koncept skrivanja vsebine objekta (atributov, metod).
- II. Razloži razliko med statičnimi spremenljivkami in spremenljivkami objekta ter statičnimi metodami in metodami objekta.
- III. Uporabi objekte pri pisanju preprostih programov.
- III. Uporabi vnaprej pripravljene razrede in metode.

#### 4.2.7 TABELE IN NIZI

- I. Kandidat pozna način sestavljanja podatkov v tabele in nize.
- II. Kandidat razloži postopek kreiranja tabele, njeno predstavitev v pomnilniku in način dostopa do posameznih elementov.
- II. Razloži postopek kreiranja nizov, njihovo predstavitev v pomnilniku in uporabo metod iz razredov za delo z nizi.
- II. Razloži in prikaže povezovanje različnih sestavljenih podatkovnih tipov v kompleksnejše strukture (npr. tabele objektov in objekti, katerih atributi so tabele in nizi).
- III. Uporabi sestavljene podatkovne strukture pri pisanju programov.

#### 4.2.8 DEDOVANJE

- II. Kandidat razume osnovne koncepte dedovanja (nadrazred, podrazred, podedovani atributi in metode, možnost deklaracije dodatnih atributov v podrazredu).
- II. Opiše postopek redefinicije podedovanih metod.
- II. Razloži naloge konstruktorja podrazreda, ko ima konstruktor nadrazreda argumente.
- III. Uporabi koncept dedovanja pri pisanju programov.

#### 4.2.9 PROGRAM

- II. Kandidat opiše zgradbo programa, ki je sestavljen iz več razredov.
- III. Kandidat napiše program za enostavnejši algoritem (na primer: delo z objekti, izdelava in uporaba metod, delo s sestavljenimi podatkovnimi tipi, urejanje in iskanje podatkov, izdelava programa na osnovi že izdelanih metod).

#### 4.2.10 VMESNIKI

- I. Pozna vlogo in možnosti uporabe vmesnikov.
- II. Razloži, razume razlike med razredom in vmesnikom.
- II. Zna uporabiti vmesnik v enostavnem programu.

#### 4.2.11 OBRAVNAVANJE IZJEM

- I. Kandidat pozna pojem programske izjeme.
- II. Razloži mehanizem proženja, lovljenja in obravnavanja izjem.

## **4.3 ARHITEKTURA IN ORGANIZACIJA RAČUNALNIŠKIH SISTEMOV**

### ■ VSEBINA, POJMI

### ■ CILJI

- 4.3.1 ZGRADBA RAČUNALNIKA**
- I. Kandidat pozna in našteje gradnike računalniškega sistema.
  - II. Opiše in razume naloge osnovnih enot računalniškega sistema: procesor (aritmetično logično enoto, registre, vodila in krmilno enoto), pomnilnik (predpomnilnik, primarni – glavni pomnilnik, sekundarni – pomožni pomnilnik, arhivski pomnilnik), vhodno-izhodne enote (tipkovnica, zaslon, miška, tiskalnik).
  - II. Kandidat razloži funkcijo procesorja, pomnilnika, vhodno-izhodnih enot, krmilne enote, vlogo registrov, razloge za uporabo več različnih pomnilnikov, utemelji uvrstitev magnetnih diskov med pomožne pomnilnike in vhodno-izhodne enote.
- 4.3.2 UKAZ**
- II. Kandidat razloži pojme: operacija, operand, naslov in način naslavljanja, opiše tipe operandov: celo nepredznačeno in predznačeno število, število v pomični vejici, oktet-znak, četverko in bit, loči naslove: fizične, logične, navidezne.
  - III. Opiše in uporabi načine naslavljanja: vsebovano, takojšnje, indeksno, neposredno, posredno.
  - II. Kandidat opiše razliko med kompleksnim in enostavnim ukazom, razliko med procesorji CISC in RISC ter med arhitekturama register–register in pomnilnik–pomnilnik.
- 4.3.3 RAČUNSKÉ OPERACIJE**
- II. Kandidat razume seštevanje, odštevanje in komplementiranje binarno kodiranih celih števil, načine preverjanja pravilnosti informacije (prečna in vzdolžna pariteta, kontrolna vsota).
  - II. Razume pojma preliv in prenos pri operacijah s števili.
- 4.3.4 INTERPRETACIJA UKAZOV**
- II. Kandidat razloži in prepozna zaporedje ukazov, vejitev, klic podprograma in vračanje iz podprograma, prekinitvev (aparaturno in programsko), past.
  - II. Razume dejstvo, da vsebina v pomnilniku dobi pomen šele z izvajanjem programa.
  - III. Kandidat zapiše ali popravi sled za krajši program v zbirnem jeziku, ki rešuje enostavno nalogo.
- 4.3.5 SLOJI REALIZACIJE RAČUNALNIŠKEGA SISTEMA**
- I. Kandidat našteje sloje (sloj elektronskega vezja, sloj logičnega vezja, sloj strojnih ukazov, sloj operacijskega sistema).
  - II. Kandidat vse sloje opiše.
- 4.3.6 FUNKCIJE OPERACIJSKEGA SISTEMA**
- I. Kandidat navede razlike med enouporabniškim in večuporabniškim operacijskim sistemom.
  - I. Opiše naloge operacijskega sistema in naloge njegovih gradnikov.
  - II. Razume posebne zahteve operacijskih sistemov za delo v realnem času.
  - II. Razloži delovanje vhodno-izhodnega podsistema, datotečnega (diskovni) podsistema. Pozna razlike med posli, procesi in opravili.
  - III. Kandidat naj uporabi ukaze operacijskega sistema za izvedbo najpogostejših opravil uporabnika (oblikovanje, premeščanje in odstranjevanje datotek in imenikov), določi pravice uporabnikov ter zaščito datotek in imenikov.

## **4.4 RAČUNALNIŠKE KOMUNIKACIJE**

### ■ VSEBINA, POJMI

### ■ CILJI

#### **4.4.1 RAZLOGI ZA POVEZOVANJE RAČUNALNIKOV**

- I. Kandidat našteje možnosti elektronskega komuniciranja (elektronska pošta, konference).
- II. Opiše postopek izmenjave podatkov, pojasni vlogo specializiranih skupnih vozlišč (datotečni strežnik, podatkovni strežnik, tiskalniki).

#### **4.4.2 OSNOVNE TOPOLOGIJE RAČUNALNIŠKIH OMREŽIJ**

- II. Kandidat razloži osnovne značilnosti, prednosti in slabosti osnovnih topologij (topologija vodila, zvezde, obroča) in sestavljenih topologij (zvezda-obroč, drevo).

#### **4.4.3 TEHNIKE DOSTOPA DO SKUPNEGA PRENOSNEGA MEDIJA**

- II. Kandidat opiše osnovne pristopne mehanizme, ki so značilni za posamezne topologije omrežij (CSMA, CSMA/CD, obroč z žetonom (Token Ring), FDDI, ATM).

#### **4.4.4 SEDEMSLOJNI MODEL ISO/OSI RAČUNALNIŠKIH KOMUNIKACIJ**

- I. Kandidat pozna definicijo protokola.
- II. Opiše pomen protokolov in zagovarja njihovo potrebnost, opiše osnovne značilnosti posameznega sloja (plasti) in funkcije, ki jih opravlja (fizični sloj, sloj zveze, mrežni sloj, prenosni sloj, sloj seje, predstavitevni sloj, aplikacijski sloj), ter naprave za izvajanje teh funkcij.

#### **4.4.5 PROTOKOLNI MODEL TCP/IP IN STORITVE**

- II. Kandidat opiše protokolni model TCP/IP in posamezne protokole v njem, navede omrežja, ki temeljijo na protokolnem modelu TCP/IP (naslavljanja, domene).
- III. Kandidat uporabi in prilagodi storitve (povezava na oddaljeni računalnik, prenos datotek, elektronska pošta).

#### **4.4.6 INTERNET**

- II. Kandidat pozna arhitekturo interneta in razume njegove sociološke vidike.
- III. Kandidat uporabi ponujane storitve na internetu.

## 5. KANDIDATI S POSEBNIMI POTREBAMI

Zakon o maturi v 4. členu določa, da kandidati opravljajo maturo pod enakimi pogoji. Kandidatom s posebnimi potrebami, ki so bili usmerjeni v izobraževalne programe z odločbo o usmeritvi, v utemeljenih primerih pa tudi drugim kandidatom (poškodba, bolezen), se lahko glede na vrsto in stopnjo primanjkljaja, ovire oziroma motnje prilagodi način opravljanja mature in način ocenjevanja znanja.

Možne so naslednje prilagoditve:

1. opravljanje mature v dveh delih, v dveh zaporednih rokih;
2. podaljšanje časa opravljanja maturitetnega izpita (tudi odmorov, možno je več krajših odmorov);
3. prilagojena oblika izpitnega gradiva (npr. Braillova pisava, povečava, kjer je prevod vprašanj nemogoč, zapis izpitnega gradiva na disketi ...);
4. poseben prostor;
5. prilagojena delovna površina (dodatna osvetlitev, možnost dviga ...);
6. uporaba posebnih pripomočkov (Braillov pisalni stroj, ustrezna pisala, folije za pozitivno risanje ...);
7. izpit s pomočnikom (npr. pomočnik bralec ali pisar);
8. uporaba računalnika;
9. prirejeni ustni izpit in izpit slušnega razumevanja (oprostitev, branje z ustnic, prevajanje v znakovni jezik);
10. prilagoditev opravljanja praktičnega dela maturitetnega izpita (npr. prilagoditev opravljanja seminarske naloge, vaj);
11. prilagojen način ocenjevanja (npr. napake, ki so posledica kandidatove motnje, se ne upoštevajo, pri ocenjevanju zunanji ocenjevalci sodelujejo s strokovnjaki za komunikacijo s kandidati s posebnimi potrebami).



## 6. LITERATURA

Pri pripravi na splošno maturo kandidati uporabljajo učbenike in učna sredstva, ki jih je potrdil Strokovni svet Republike Slovenije za splošno izobraževanje. Potrjeni učbeniki in učna sredstva so zbrani v **Katalogu učbenikov za srednjo šolo**, ki je objavljen na spletni strani Zavoda Republike Slovenije za šolstvo [www.zrss.si](http://www.zrss.si).

### 1. Osnove informatike

Priporočena literatura za učitelje:

M. Vintar, A. Kovačič: *Načrtovanje in izgradnja informacijskih sistemov*, DZS, Ljubljana 1994

T. Mohorič: *Uvod v podatkovne baze*, BI-TIM d. o. o., Ljubljana 1995

Priporočena literatura za učitelje in dijake:

J. Gras, J. Jaklič: *Baze podatkov*, Ekonomska fakulteta Ljubljana, Ljubljana 1996

### 2. Programski jeziki in programiranje

Priporočena literatura za učitelje:

U. Mesojedec, B. Fabjan: *Java 2: Temelji programiranja*, Pasadena, Ljubljana 2004, ISBN 961-6361-30-9

Priporočena literatura za učitelje in dijake:

J. Farrell: *Java Programming, Third Edition*, Thomson Course Technology, 2006, ISBN 0-619-21319-1

### 3. Arhitektura in organizacija računalniških sistemov

Priporočena literatura za učitelje:

D. Zazula: *Operacijski sistemi*, FRI Maribor, Maribor 1997

D. Kodek: *Arhitektura računalniških sistemov*, BI-TIM, d. o. o., Ljubljana 2000

Priporočena literatura za učitelje in dijake:

T. Rugelj: *Osnove mikrokontrolerjev na primeru MC6803*, Metra d. o. o., Ljubljana 1992

### 4. Računalniške komunikacije

Priporočena literatura za učitelje:

T. Vidmar: *Računalniška omrežja in storitve*, Atlantis, Ljubljana 1997

Priporočena literatura za učitelje in dijake:

J. Györkös, A. Živkovič: *Metode komuniciranja – elektronsko komuniciranje*, FRI Maribor, Maribor 1998

## 7. PRIMERI IZPITNIH NALOG

### Primer naloge izbirnega tipa

Kaj je značilno za sprotno potrjevanje pri nepovezavnih protokolih?

(2 točki)

- A Pošiljatelj oddaja pakete drugega za drugim brez premora, dokler ne pošlje celotnega sporočila.
- B Pošiljatelj oddaja pakete v vnaprej določenih časovnih presledkih.
- C Pošiljatelj odda vse pakete naenkrat.
- D Pošiljatelj odda naslednji paket šele takrat, ko sprejme potrditev predhodnega paketa.
- E Pošiljatelj odda določeno število paketov in čaka na potrditev.

Rešitev: D

### Primer naloge s kratkimi odgovori

Naštejte vsaj štiri možne nepravilnosti na diagramih toka podatkov.

(2 točki)

---

---

---

---

---

---

---

---

---

---

Rešitve:

povezava (pod. tok) med zunanjima entitetama (izvor/izvor, izvor/ponor)

povezava (pod. tok) med zunanjo entiteto (izvor/ponor) in podatkovno shrambo (zbirko)

povezava (pod. tok) med dvema podatkovnima shrambama (zbirdkama)

proces le ponor/izvor podatkov

nepoimenovan podatkovni tok (razen ob povezavi s pod. shrambo)

neuravnoteženost diagramov toka podatkov

gradnik (proces, podatkovni tok, podatkovna shramba, izvor/ponor) ni poimenovan (proces dodatno še oštevilčen)

podatkovni tok izhaja iz ali kaže v »nič« (ni elementa na eni strani)

preveč procesov na eni ravni

preveč vhodno-izhodnih tokov pri nekem procesu

4 pravilni odgovori 2 točki

2, 3 pravilni odgovori 1 točka

## Primer strukturirane naloge

Sprogramirajte razred `Delavec`, v katerem bo vsak delavec predstavljen kot objekt z atributi matična številka (celo število, ki je enolični identifikator delavca), priimek (niz znakov), ime (niz znakov) in število ur (celo število, ki pove, koliko ur je delavec prebil na delu). Konstruktor naj omogoča inicializacijo vseh atributov razen števila ur, ki se vpiše naknadno z metodo `vpisiStUr`. Za izpis vseh atributov objekta napišite metodo `izpisiVse`.

*deklaracija atributov (1 točka)*

*deklaracija konstruktorja (1 točka)*

*deklaracija metode vpisiStUr (1 točka)*

*metoda za izpis vseh atributov (1 točka)*

Rešitev:

```
public class Delavec
{
    private int matStev;
    private String priimek;
    private String ime;
    private int stUr;

    Delavec(int ms, String p, String i)
    {
        matStev=ms;
        priimek=p;
        ime=i;
    }

    public void vpisiStUr(int u)
    {
        stUr=u;
    }

    public void izpisiVse()
    {
        System.out.println("Maticna stevilka: "+matStev);
        System.out.println("Priimek in ime: "+priimek+' '+ime);
        System.out.println("Število ur: "+stUr);
    }
}
```

## 8. SEZNAM UPORABLJENIH KRATIC

CASE	Computer Aided Software Engineering
CISC	Complex Instruction Set Computer
RISC	Reduced Instruction Set Computer
CSMA	Carrier Sense Multiple Access
CSMA/CD	Carrier Sense Multiple Access Collision Detect
FDDI	Fiber Distributed Data Interface
ATM	Asynchronous Transfer Mode
TCP/IP	Transmition Control Protocol/Internet Protocol
ISO/OSI	International Standards Organization/Open System Interconnect