



ISPITNI KATALOG PRIPREMILI:

DR GORAN ŠUKOVIĆ

Prirodno-matematički fakultet, Univerzitet Crne Gore

MLADEN JANKOVIĆ

JU Gimnazija „Slobodan Škerović”, Podgorica

ANA ARANITOVIĆ

JU Gimnazija „Slobodan Škerović”, Podgorica

MILKA STANIŠIĆ

Ispitni centar, Podgorica



SADRŽAJ

1. UVOD 4

2. PRAVILA 5

3. OPŠTI CILJEVI ISPITA 6

4. STRUKTURA ISPITA 7

5. ISPITNI PROGRAM 8

6. PRIMJER TESTA SA SHEMOM ZA BODOVANJE 13

Primjer testa 26

Rješenja 21

7. LITERATURA 30

1.

UVOD

Maturski ispit predstavlja standardizovanu eksternu provjeru usvojenosti neophodnih standarda znanja na kraju gimnazijskog obrazovanja.

Na osnovu Zakona o gimnaziji („Sl. list RCG“, br.64/02, 49/07 i „Sl. List CG“, br. 45/10, 39/13 i 47/17) i Pravilnika o načinu, postupku i vremenu polaganja maturalnog ispita u gimnaziji („Sl. list CG“, 34/09, 19/10, 81/10, 60/11, 65/12, 50/13 i 43/19) maturalni ispit se polaže eksterno. Stručne, razvojne, organizacione i administrativne poslove, vezane za realizaciju ispita obavlja Ispitni centar.

Provjeravaju se znanja, sposobnosti i vještine koje se zasnivaju na ključnim djelovima Predmetnog programa a treba da ih posjeduju učenici na kraju četvrtog razreda gimnazije.

U Ispitnom katalogu navedeni su opšti ciljevi ispita, ispitni program sa područjima i ciljevima, primjer testa sa šemom za bodovanje i literatura.

Ispitni katalog namijenjen je učenicima i nastavnicima.



2.

PRAVILA.

Ispit iz Programiranja i programskih jezika sastoji se iz dva dijela. U prvom dijelu ispita učenici rješavaju test. Drugi dio ispita predstavlja praktičan rad na računaru.

Ispitni materijal biće zapakovan u posebne sigurnosne vrećice (PVC), koje će se otvoriti pred učenicima neposredno prije početka ispita.

Na ispitu nije dozvoljeno:

- lažno predstavljanje
- otvaranje ispitnih zadataka prije dozvoljenog vremena
- ometanje drugih učenika
- prepisivanje od drugog učenika
- korišćenje nedozvoljenog pribora
- oštećenje šifre na test-knjižici
- nepoštovanje znaka za završetak ispita

Dozvoljen pribor je: grafitna olovka i gumica i hemijska olovka.

Tokom ispita **nije dopuštena** upotreba elektronskih uređaja (odnosi se na digitrone, mobilne telefone, tablete, muzičke i video-plejere, fleš-diskove, diskete, eksterne hard-diskove, kompakt diskove i sl), osim onih koji su dio ispitnog materijala. Učenikov rad mora biti napisan hemijskom olovkom.

Maturalni ispit iz Programiranja i programskih jezika treba da ustanovi da li učenici mogu da:

- Kreiraju sopstvene stilove
- Znaju kako da uključe biblioteke/funkcije u svoju veb-stranicu
- Kreiraju veb-stranice primjenom HTML-a, CSS-a
- Primijene metode za rad s datotekama u pisanju programa
- Primijene rekurzivne algoritme u pisanju programa
- Primijene metode za rad s datotekama u pisanju programa
- Projektuju i implementiraju aplikaciju (npr. kreiranje veb-sajta sa opisima dva algoritma sortiranja)
- Pišu programe u kojima koriste višedimenzionalne nizove
- Primijene grafičke funkcije u izradi programa
- Povežu ugrađene klase i funkcije i metode sa objektno-orijentisanim programiranjem
- Uporede vremena traženja elementa u neuređenoj i sortiranoj listi
- Opišu algoritme za sortiranje: zamjenom elemenata (exchange sort), izborom najmanjeg elementa (selection sort), zamjenom susjednih elemenata (bubble sort), umetanjem (insertion sort), sjedinjavanjem (merge sort), brzim sortiranjem (quick sort, razvrstavanjem (bucket sort) i uporede algoritme sortiranja i izračuna njihovu složenost
- Primijene algoritme obilaska drveta (preorder, inorder, postorder) na konkretnom primjeru drveta, olančane liste
- Opišu osnovne klase za kreiranje grafičkog korisničkog interfejsa
- Naprave aplikaciju sa nekoliko osnovnih elemenata grafičkog korisničkog interfejsa i nekoliko događaja
- Napišu SQL upit za dodavanje podataka, za izmjenu određenih podataka u tabeli i za brisanje podataka iz tabele prema zadatom kriterijumu
- Primjenjuju algoritme kriptovanja

Ciljevi ispita su:

- provjera informatičkih znanja i vještina stečenih u toku školovanja u matematičkoj gimnaziji koji su definisani kroz obrazovne standarde predmeta i ovim katalogom
- provjera opšte sposobnosti korišćenja znanja i vještina u rješavanju zadataka
- poznavanje računarskih i informatičkih pojmova i osnovne terminologije
- upotreba informatičkih znanja u rješavanju praktičnih problema
- uočavanje i analiza problema i izbor odgovarajućeg načina rješavanja
- podsticanje različitih načina rješavanja problema, kreativnost i fleksibilnost razmišljanja
- povezivanje znanja iz različitih oblasti informatike
- provjera znanja i vještina neophodnih za nastavak školovanja

4.

STRUKTURA ISPITA

Ispit se sastoji iz dva dijela: izrade testa i praktičnog rada za računarom. Ispit traje 150 minuta.

Ispit sadrži dvije vrste zadataka:

- zadatke višestrukog izbora
- zadatke otvorenog tipa

Kod zadataka otvorenog tipa rješenje može biti:

- kratak odgovor – rješenje je riječ, prosta rečenica ili broj do koga se može doći nakon nekoliko povezanih koraka
- duži odgovor – do krajnjeg rezultata se dolazi rješavanjem više zahtjeva ili primjenom složenijih postupaka.

Maksimalan broj bodova na ispitu je između 60 i 70. Zadatke bodujemo tako što svaki nosi od 1 do 7 bodova u zavisnosti od toga da li ima ili nema djelimičnog bodovanja. Netačno riješen ili neuraden zadatak ne donosi negativne poene.

U zadacima višestrukog izbora, tačno riješen zadatak donosi 1 bod.

U zadacima otvorenog tipa boduje se postavka zadatka, postupak rješavanja i tačan rezultat. Zadatke ovog tipa bodujemo tako što svaki nosi od 1 do 7 bodova.

TIP ZADATKA	Broj bodova
Višestruki izbor	15-20%
Kratak odgovor	10-15%
Duži odgovor	15-20%
Zadaci otvorenog tipa - programiranje	45-55%

Nastavno gradivo podijeljeno je na sljedeće oblasti (područja):

- P1 – Veb dizajn
- P2 – Datoteke
- P3 – Objektno -orijentisano programiranje
- P4 – Složenost algoritama
- P5 – Rekurzivni algoritmi
- P6 – Algoritmi sortiranja
- P7 – 2D grafika i programi sa GUI
- P8 – Višedimenzionalni nizovi
- P9 – Apstraktne strukture podataka
- P10 – Baze podataka
- P11 – Kriptografija

4. STRUKTURA ISPITA

Tabelom je data zastupljenost sadržaja u strukturi ispita.

Redni broj	Test (do 50 min)	Rad na računaru (do 100 min)
P1	0-10%	15-20%
P2	0-5%	0-5%
P3	0-5%	15-20%
P4	5-15%	0-5%
P5	0-5%	0-5%
P6	5-15%	0-10%
P7	0-10%	15-20%
P8	5-10%	5-10%
P9	5-15%	0-10%
P10	5-10%	0-15%
P11	5-10%	0-10%
Zastupljenost	50-55%	45-50%



P1 VEB DIZAJN

SADRŽAJ:

Dodavanje teksta, slike, tabela, lista na veb stranicu, css stilovi, TextBox, Button, Crtanje kruga, pravougaonika

UČENIK TREBA DA:

- Kreira veb stranicu sa odeređenim elementima i da im funkciju
- Crta geometrijsko tijelo sa određenim karakteristikama

P2 DATOTEKE

SADRŽAJ:

Čitanje iz datoteke i upisivanje u datoteku

UČENIK TREBA DA:

- Pročita podatke iz datoteke ili u nju upiše

P3 OBJEKTNO ORJENTISANO PROGRAMIRANJE

SADRŽAJ:

Klasa, Metod, Konstruktor, Pristupne metode, Potklasa i natklasa, Hijerahija klasa

UČENIK TREBA DA:

- Piše konstruktor klase
- Implementira primjere metoda sa dodatnim parametrima
- Koristi specijalne metode za rad s objektima
- Kreira potklase date klase
- Piše programe u kojima primjenjuju objektno-orijentisani način programiranja

P4 SLOŽENOST ALGORITAMA

SADRŽAJ:

Složenost algoritma, Notacija veliko O, Linearno traženje

UČENIK TREBA DA:

- Izvrši kategorizaciju složenosti algoritama prema notaciji veliko O
- Uporedi vremena traženja elementa u neuređenoj i sortiranoj listi

P5 REKURZIVNI ALGORITMI

SADRŽAJ:

Rekurzivni algoritmi

UČENIK TREBA DA:

- Izvodi rekurzivnu relaciju i uslove prekida za dati problem
- Piše programe u kojima primjenjuju rekurziju

P6 ALGORITMI SORTIRANJA

SADRŽAJ:

Algoritam sortiranja:

- Zamjenom elemenata (exchange sort)
- Izborom najmanjeg elementa (selection sort)
- Zamjenom susjednih elemenata (bubble sort)
- Umetanjem (insertion sort)
- Sjedinjavanjem (merge sort)
- Brzim sortiranjem (quick sort),
- Razvrstavanjem (bucket sort)

UČENIK POKAZUJE DA UMIJE DA:

- Opiše i implementira svaki od navedenih algoritama i upoređuje njihovu složenost

P7 2D GRAFIKA I PROGRAMI SA GUI

SADRŽAJ:

Grafički prozor, Pero/olovka, Algoritam crtanja pravilnog n-tougla, Metode za crtanje linije, pravougaonika, kruga, Boja olovke/pera, Grafički korisnički interfejs (GUI), Dugme, Polje za unos podataka, Lista, Principi kreiranja GUI

UČENIK POKAZUJE DA UMIJE DA:

- Napiše program koji primjenjuje grafičke funkcije
- Kreira aplikaciju sa nekoliko osnovnih elemenata grafičkog korisničkog interfejsa i nekoliko događaja
- Upotrijebi klasu za kreiranja liste elemenata na grafičkom korisničkom interfejsu

P8 VIŠEDIMENZIONALNI NIZOVI

SADRŽAJ:

Matrice, determinante, tabelarni prikaz

UČENIK TREBA DA:

- Kreira tabele primjenom 2D niza
- Implementira programe u kojim koriste višedimenzionalne nizove

P9 APSTRAKTNE STRUKTURE PODATAKA

SADRŽAJ:

Drvo, binarno drvo, algoritmi obilaska drveta, dodavanje i brisanje elementa iz binarnog drveta

UČENIK TREBA DA:

- Primijeni algoritme obilaska drveta (preorder, inorder, postorder) na konkretnom primjeru drveta
- Dodaje i briše elemente iz binarnog drveta

P10 BAZE PODATAKA

SADRŽAJ:

Entitet, Atribut, Model Entiteti-veze, Relacioni model podataka, Primarni i strani ključ, Veza jedan-više, Veza više-više, Tabela, SQL

UČENIK POKAZUJE DA UMIJE DA:

- Piše SQL upite za dodavanje, izmjenu, brisanje, selekciju podataka...

P11 KRIPTOGRAFIJA

SADRŽAJ:

Cezarovo kriptovanje, kriptovanje sa pomakom, Vigenерово kriptovanje, Kriptovanja sa javnim ključem

UČENIK POKAZUJE DA UMIJE DA:

- Na određenom tekstu primjenjuje neki od algoritama kriptovanja



VRIJEME RADA: 150 MINUTA

U ZADACIMA OD 1 DO 7 ZAOKRUŽI BROJ ISPRED TAČNOG ODGOVORA. RJEŠENJA PRENESI NA LIST ZA ODGOVORE.

1. Dat je sljedeći metod:

```
public int g(int x){
    if (x <= 1)
        return -1;
    else
        if (x <=12)
            return 4 + g(x-3);
        else
            return 6 + g(x-5);
}
```

Koju će vrijednost vratiti ovaj metod ako je ulazni argument (parametar) metoda 17?

1. 1
2. 17
3. 21
4. 23

1 bod

2. Dat je sljedeći metod:

```
public int met(int n){
    for (int i = 2; i<n; i++){
        for (int j = i; j<n; j++){
            System.out.println(i+j*j);
        }
    }
}
```

Ukupan broj operacija aritmetičkih operacija može se predstaviti izrazom:

1. $O(1)$
2. $O(n)$
3. $O(n \log n)$
4. $O(n^2)$

1 bod

5
PRIMJER
TESTA

6. Primjer testa

3. Koji od sljedećih algoritama sortiranja brojeva, u prosjeku, ima složenost $O(n \log n)$?

1. Exchange sort
2. Quick sort
3. Bubble sort
4. Insertion sort

1 bod

4. Koji od sljedećih html elemenata prikazuje naslov najvećim fontom?

1. h6
2. head
3. h1
4. heading

1 bod

5. Jednom rečenicom opisati šta izračunava sljedeći program:

```
public static int main(Strings [] args){
    int i=1, c=0, a=0;
    int [][] x = new int [10][5];
    for (i = 1; i <= 50; i++){
        a= i;
        while(a>0){
            c += a%10;
            a= a/10;
        }
        x[i%10][i/10]=c;
    }
}
```

Odgovor:

1 bod

6. Šta će desiti ako prevedemo sljedeći kod i pokrenemo program?

```
public class Test {
    public static void aa (int i, int j) {
        System.out.println("Crna");
    }

    public static void aa (int x, int y) {
        System.out.println("Gora");
    }

    public static void main (String [] args) {
        Test t = new Test();
        t.aa(5,10);
    }
}
```

1. štampaće riječ "Crna"
2. štampaće riječ "Gora"
3. greška pri prevođenju ("compilation error")
4. greška pri izvršavanju ("runtime error")

1 bod

7. Olančana lista sadrži 1000 međusobno različitih cijelih brojeva, Neka je x proizvoljan broj iz liste. Koristimo linearno traženje i svi elementi su jednako vjerovatni. Koliko je, u prosjeku, potrebno operacija poređenja da pronađemo broj x u listi?

1. 1000
2. 500
3. 34
4. 11

1 bod

6. Primjer testa

8. Tabela prikazuje prvu iteraciju algoritma Bubble sort. Parovi elemenata koji se upoređuju su uokvireni. Popuniti odgovarajuću tabelu za drugu iteraciju.

14, 33, 27, 35, 10	Početni niz
14, 33, 27, 35, 10	$33 > 14$, nema zamjene
14, 33, 27, 35, 10	$27 < 33$, zamjena mjesta
14, 27, 33, 35, 10	$35 > 33$, nema zamjene.
14, 27, 33, 35, 10	$10 < 35$, zamjena mjesta
14, 27, 33, 10, 35	Niz na kraju prve iteracije

Odgovor:

14, 27, 33, 10, 35	Početni niz
	Niz na kraju prve iteracije

1 bod

9. Algoritam supstitucije (Cezarovo kodiranje) koristi sljedeći par azbuka:

Polazna : ABCDEFGHIJKLMNOPQRSTUVWXYZ
Kodiranje: XYZABCDEFGHIJKLMNPOQRSTUVWXYZ

Napisati kod koji se dobija od teksta "PODGORICA"?

Odgovor: 1 bod

10. Kako se naziva tekst koji se dobije kada se otvoreni tekst šifrira pomoću dogovorenog tajnog ključa?

Odgovor: 1 bod

11. Red je implementiran pomoću jednostruko-olančane liste.

Data su sljedeća tvrđenja:

- a) Ako se dodavanje u red obavlja umetanjem čvora na početak liste, tada se uklanjanje iz reda mora obaviti brisanjem čvora sa kraja liste
- b) Ako se dodavanje u red obavlja umetanjem čvora na kraj liste, tada se uklanjanje iz reda mora obaviti brisanjem čvora sa početka liste

Šta je od navedenog tačno:

- 1. Samo tvrđenje a)
- 2. Samo tvrđenje b)
- 3. Oba tvrđenja a) i b)
- 4. Nijedno od tvrđenja

1 bod

12. Šta treba da piše na linijama u datom kodu da bi program nacrtao prozor dimenzija 150x100 piksela.

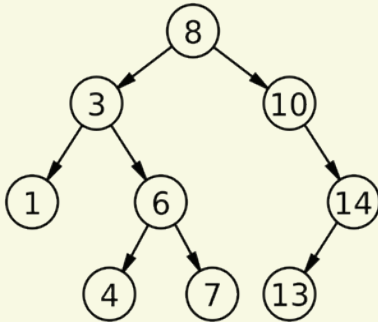
```
import javax.swing.JFrame;
public class Wintest {
    public static void main (String [] args) {
        JFrame frm = new _____();
        frm._____ (150, 100);
        frm._____ (true);
    }
}
```

- 1. JFrame, setVisible, setOn
- 2. JFrame, setSize, setVisible
- 3. JFrame, setVisible, setSize
- 4. Window, setSize, point

1 bod

6. Primjer testa

13. Dato je sljedeće binarno drvo traženja.



1. Nacrtati izgled drveta poslije dodavanja broja 5.
2. Nacrtati izgled drveta koje se dobije brisanjem čvora 6 iz drveta sa slike.
3. Napisati niz koji se dobije prefiksnim obilaskom (preorder) drveta sa slike.

3 boda (1+1+1)

14. Data je sljedeća baza podataka.

1. Šta će biti rezultat sljedećih SQL upita:
 - a. SELECT Ime FROM Student WHERE Starost > 20
 - b. SELECT NazivSmera FROM Smer, Student WHERE Starost > 20 and Smer.SifSmera = Student.SifSmera
2. Napisati upit koji vraća imena nastavnika koji predaju predmet P1
3. Napisati upit koji vraća imena i starost svih studenata koji su položili ispit Matematika

Relaciona BP				
Student				
BrInd	Mlb	Ime	Starost	SifSmer
152/97	16309723331981	Ana	19	01
223/95	?	Mirko	21	01
021/94	11145276418976	Zoran	20	?
429/97	01387976418976	Zora	?	02
003/94	23456786418976	Miloš	22	01

Prijava			
BrInd	SifPred	DatumPol	Ocena
152/97	P1	18.3.97	9
223/95	P2	21.4.96	8
021/94	P1	?	10
003/94	P3	30.5.97	6
021/94	P3	1.7.97	10

Nastavnik			
SN	ImeNast	Zvanje	SifPred
nn1	Pavle	Red. Prof.	P1
Nn2	Zdravko	Docent	P2
Nn3	Mira	Asistent	P1
nn4	Zoran	?	?

Smer		
SifSmer	NazivSmera	SN
01	Informacioni sistemi	Nn1
02	Menadžment	nn2

Predmet		
SifPred	NazivPred	BrojČasova
P1	Baze podataka	6
P2	Matematika	4
p3	Ekonomija	?

6 bodova (1+1+2+2)

PRAKTIČNI DIO – PROGRAMIRANJE

15. Kreirati projekat TestProba.

a) Kreirati klasu Kartica koja modelira bankovnu karticu i koja ima sljedeće private članice: korisnik (tipa String), brojKartice (string dužine 16), mjesecVazenja (cio broj od 1 do 12), godinaVazenja (četvorocifren cio broj) i iznos (realan broj).

1 bod

b) U klasi Kartica napisati odgovarajuće getere i setere.

1 bod

c) Napisati konstruktor sa 3 argumenta koji zadaje brojKartice, mjesec i godinu važenja kartice. U konstruktoru provjeriti da li je mjesec odgovarajući, i ako nije postaviti vrijednost na 1.

1 bod

d) Napisati metod double stednja(int g, double kamata) koji vraća koliki bi iznos korisnik dobio poslije g godina ako bi oročio iznos sa kartice sa godišnjom kamatom od k% (npr. ako je iznos 100 eura, i kamata je 10%, posle prve godine imao bi $10+10\% 100=110$ eura, poslije druge godine $110+10\% 110=121$, itd.).

2 boda

e) Iz klase Kartice izvesti klasu KreditnaKartica koja ima još jednu članicu prekoracenje (realan broj).

1 bod

f) U klasi KreditnaKartica napisati metod void podigniNovac(double d) koji sa kartice podiže iznos d, ako nije došlo do prekoračenja, ili ne mijenja iznos i štampa poruku "Nedostaje vam qqq eura", gdje qqq označava koliko eura ne ostaje da bi se ostvarilo podizanje novca.

2 boda

g) U klasi TestKartica napisati metod int zamijeni(int n) koji učitava prirodan broj n iz intervala [1,99999] koji ne sadrži cifru 0 i vraća broj koji se dobije kada prva i posljednja cifra zamijene mjesta.

2 boda

h) U klasi TestKartica napisati metod int redMaks(int [][] a) koji vraća indeks onog reda matrice a koji sadrži najviše brojeva koji su djeljivi sa 3. Ako ima više takvih redova, vratiti najmanji indeks. Ako nema takvih redova, vratiti -1.

2 boda

i) Kreirati klasu TestKartica u čijem metodu main kreirati po jedan objekat klase Kartica i KreditnaKartica i poziva metode stednja, podigniNovac, zamijeni i redMaks.

2 boda

16. Kreirati prozor koji ima 2 tekstualna polja i dugme sa tekстом “Pretvori”. U prvom tekstualnom polju unosi se temperatura u formatu “b C” ili “b K”, gdje je b cio broj, C predstavlja stepene Celzijusa a K stepene Kelvina. Klikom na dugme Pretvori, unesena temperatura se konvertuje u odgovarajuću temperaturu izraženu stepenima Farenhajta i štampa se u drugom tekstualnom polju u formatu “b F”. Između broja i oznaka stepena nalazi se tačno jedan simbol blanko. Veza između Kelvina, Farenhajta i Celzijusa je data sa: $C=9/5*(F-32)$, $K = C + 273.15$. Nije potrebno provjeravati validnost ulaznih podataka.

5 bodova

17. Kreirati fajlove test.html i test.css. Izgled stranice test.html prikazan je na slici. Fajl test.css sadrži CSS svojstva koja opisuju izgled svih elemenata u fajlu test.html. Svi elementi za koje nisu data svojstva imaju podrazumijevane vrijednosti.

Granice tabele

Koristite CSS svojstvo border-collapse

IME	OBRAZOVANJE	GRAD
Jana	Osnovno	Podgorica
Ema	Srednje	Bar
Ivan	Fakultet	Berane
Lana	Fakultet	Pljevlja
Pero	Srednja	Ulcinj

Tekst “Granice tabele” je h2. Svi stringovi u tabeli imaju font koji je 120% veći od podrazumijevanog fonta. Tekst u tabeli je udaljen 8px od lijeve ivice polja tabele. Ivice tabele su plave boje i debljine 2px. Tekst Podgorica je link koji void ka lokaciji <http://podgorica.me/>, Prvi, treći i peti red tabele imaju boju podloge gray i boju teksta white. Tabela po širini zauzima cjelokupan prostor. Smanjenjem širine prozora mijenja i širinu tabele.

6 bodova

ZADATAK	RJEŠENJE	BODOVANJE										
1.	3											
2.	4											
3.	2											
4.	3											
5.	Matrica koja sadrži sume cifara svih prirodnih n brojeva od 1 do 50.											
6.	3											
7.	1											
8.	<table border="1"> <tr> <td>14, 27, 33, 10, 35</td> <td>Početni niz</td> </tr> <tr> <td>14, 27, 33, 10, 35</td> <td>27 > 14, nema zamjene</td> </tr> <tr> <td>14, 27, 33, 10, 35</td> <td>33 > 27, nema zamjene</td> </tr> <tr> <td>14, 27, 33, 10, 35</td> <td>33 > 10, zamjena mjesta</td> </tr> <tr> <td>14, 27, 33, 10, 35</td> <td>Niz poslije prve iteracije</td> </tr> </table>	14, 27, 33, 10, 35	Početni niz	14, 27, 33, 10, 35	27 > 14, nema zamjene	14, 27, 33, 10, 35	33 > 27, nema zamjene	14, 27, 33, 10, 35	33 > 10, zamjena mjesta	14, 27, 33, 10, 35	Niz poslije prve iteracije	Potpuno ispravna tabela – 1 bod
14, 27, 33, 10, 35	Početni niz											
14, 27, 33, 10, 35	27 > 14, nema zamjene											
14, 27, 33, 10, 35	33 > 27, nema zamjene											
14, 27, 33, 10, 35	33 > 10, zamjena mjesta											
14, 27, 33, 10, 35	Niz poslije prve iteracije											
9.	MLADLOFZX	1 bod za ispravan tekst										
10.	Kriptovani tekst	1 bod										
11.	false	1 bod										
12.	3	1 bod										
13.	<p>a) čvor 5 je desni sin čvora 4 b) prvo rješenje: na mjesto čvora 6 može doći 1, čiji je desni sin 7 drugo rješenje: na mjesto čvora 6 može doći 7, čiji je lijevi sin 1 c) 8, 3, 1, 6, 4, 7, 10, 14, 13</p>	<p>a) i b) – po 1 bod za potpuno ispravne nacrtane slike ili tekst c) 1 bod za potpuno ispravan redosljed</p>										
14.	<table border="1"> <tr> <td>a) 1.</td> <td>a) 2.</td> </tr> <tr> <td>Ime</td> <td>NazivSmera</td> </tr> <tr> <td>Mirko</td> <td>Infomacioni sistemi</td> </tr> <tr> <td>Miloš</td> <td></td> </tr> </table> <p>b) SELECT ImeNast FROM Nastavnik WHERE SifPred = 'P1' c) SELECT Ime, Starost FROM Student, Prijava, Predmet WHERE Predmet.NazivPred = 'P1' and Predmet.SifraPred = Prijava.SifraPred Prijava.Brind = Student.Brind</p>	a) 1.	a) 2.	Ime	NazivSmera	Mirko	Infomacioni sistemi	Miloš		<p>a) 1: 1 bod za tačan odgovor (u obliku tabele ili samo Mirko i Miloš) a) 2: 1 bod za tačan odgovor (u obliku tabele ili samo "Informacioni sistemi") b) 2 boda za ispravan upit (moguće je koristiti prefiks, npr. Nastavnik.ImeNast, ili alijas, npr. FROM Nastavnik N, pa u WHERE N.SifPred) 1 bod ako je upit ispravan ali nije uočeno da je P1 string. c) 2 boda za ispravan upit (moguće je koristiti prefiks ili alijas) 1 bod ako ima neku od sljedećih grešaka: nije uočeno da je P1 string; nisu sve 3 tabele povezane preko zajedničkih kolona, već samo dvije od njih</p>		
a) 1.	a) 2.											
Ime	NazivSmera											
Mirko	Infomacioni sistemi											
Miloš												

ZADATAK 15.

```
public class Kartica {
    private String korisnik, brojKartice;
    private int mjesecVazenja, godinaVazenja;
    private double iznos;

    public Kartica(String brojKartice, int mjesecVazenja, int
godinaVazenja) {
        this.brojKartice = brojKartice;
        if (mjesecVazenja > 12 || mjesecVazenja < 1) {
            this.mjesecVazenja = 1;
        }
        else {
            this.mjesecVazenja = mjesecVazenja;
        }
        this.godinaVazenja = godinaVazenja;
    }
    /**
     * @return the korisnik
     */
    public String getKorisnik() {
        return korisnik;
    }
    /**
     * @param korisnik the korisnik to set
     */
    public void setKorisnik(String korisnik) {
        this.korisnik = korisnik;
    }
    /**
     * @return the brojKartice
     */
    public String getBrojKartice() {
        return brojKartice;
    }
    /**
     * @param brojKartice the brojKartice to set
     */
    public void setBrojKartice(String brojKartice) {
        this.brojKartice = brojKartice;
    }
    /**
     * @return the mjesecVazenja
     */
    public int getMjesecVazenja() {
        return mjesecVazenja;
    }
    /**
     * @param mjesecVazenja the mjesecVazenja to set
     */
}
```

```

public void setMjesecVazenja(int mjesecVazenja) {
    this.mjesecVazenja = mjesecVazenja;
}
/**
 * @return the godinaVazenja
 */
public int getGodinaVazenja() {
    return godinaVazenja;
}
/**
 * @param godinaVazenja the godinaVazenja to set
 */
public void setGodinaVazenja(int godinaVazenja) {
    this.godinaVazenja = godinaVazenja;
}

/**
 * @return the iznos
 */
public double getIznos() {
    return iznos;
}
/**
 * @param iznos the iznos to set
 */
public void setIznos(double iznos) {
    this.iznos = iznos;
}

double stednja (int godina, int kamata) {
    return iznos * Math.pow(1 + kamata/100.0, godina);
}
}

```

```

public class KreditnaKartica extends Kartica {

    public KreditnaKartica(String brojKartice, int
mjesecVazenja, int godinaVazenja) {
        super(brojKartice, mjesecVazenja,
godinaVazenja);
    }

    double prekoracenje;

    /**
     * @return the prekoracenje
     */

```

```
public double getPrekoracenje() {
    return prekoracenje;
}

/**
 * @param prekoracenje the prekoracenje to set
 */
public void setPrekoracenje(double prekoracenje) {
    this.prekoracenje = prekoracenje;
}

void podigniNovac(double d) {
    double x = getIznos();
    double r = x - d + prekoracenje;
    if (0 <= r) {
        x -= d;
    }
    else {
        System.out.println("Nedostaje vam " + (-1)*r + " eura");
    }
}
}
```

```
public class TestKartica {

    public static int zamijeni(int n) {
        int r = n;
        int poslednja = n %10, stepen = 1, sredina = 0;
        r = r /10; // uklanjamo poslednju cifru
        // odredimo sredinu i prvu cifru
        while (r >= 10) {
            sredina = r %10 * stepen + sredina;
            stepen = stepen * 10;
            r = r/10;
        }
        // na kraju ciklusa r je prva cifra
        stepen *= 10;
        // izracunamo rezultat
        r = stepen * poslednja + sredina * 10 + r;
        return r;
    }

    public static int redMaks(int [][] a) {
        int maks = -1, t = 0, ind = -1;
        for (int i = 0; i < a.length; i++) {
            t = 0;
```

```
        for (int j = 0; j < a[i].length; j++) {
            if (a[i][j] % 3 == 0) {
                t++;
            }
        }
        if (t > maks) {
            maks = t;
            ind = i;
        }
    }
    if (0 == maks)
        ind = -1;
    return ind;
}

public static void main(String[] args) {
    Kartica a = new Kartica("1234", 5, 2024);
    Kartica b = new Kartica("1004", 7, 2026);
    a.setIznos(100);
    a.setPrekoracenje(300);
    b.setIznos(100);
    b.setPrekoracenje(300);
    a.podigniNovac(500);
    b.podigniNovac(58);
    System.out.println("Kartica a " + a.getIznos());
    System.out.println("Kartica b " + b.getIznos());
}
}
```

- a) Kreiranje klase Kartica – 1 bod
- b) kreiranje getera i setera – 1 bod.
- c) Ispravan konstruktor – 1 bod
- d) Ispravan metod *stednja* – 2 boda (moguće je koristiti ciklus ili direktno primjenom formule)
- e) Kreiranje klase KreditnaKartica – 1 bod
- f) Ispravan metod *podigniNovac* – 2 boda
- g) Ispravan metod *zamjeni* – 2 boda
- h) Metod *redMax* – 2 boda (1 bod prolazak kroz matricu+1 bod vraćanje reda)
- i) Klasa *TestKartica* sa pozivima metoda – 2 boda

ZADATAK 16.

```
import java.awt.FlowLayout;
import java.awt.Graphics;
import java.awt.LayoutManager;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class Podloga extends JPanel implements
ActionListener{

    JTextField ulaz, izlaz;
    JButton konverzija;

    public Podloga () {
        ulaz = new JTextField(10);
        izlaz = new JTextField(10);
        konverzija = new JButton("Konverzija");
        konverzija.addActionListener(this);

        add(ulaz);
        add(izlaz);
        add(konverzija);
    }

    @Override
    public void actionPerformed(ActionEvent arg0) {
        String inp = ulaz.getText();
        int i = inp.indexOf(" ");
        String jedinicaMjere = inp.substring(i+1);
        double temperatura = Double.parseDouble(inp.substring(1, i));
        if (jedinicaMjere.equalsIgnoreCase("K")) {
            temperatura = 5.0/9 * (temperatura - 273.15) + 32;
        }
        else {
            temperatura = 5.0/9 * temperatura + 32;
        }
        izlaz.setText(temperatura + "");
    }

    import java.awt.HeadlessException;

    import javax.swing.JFrame;

    public class Prozor extends JFrame {
```

```
public Prozor() throws HeadlessException {
    setTitle("Konverzija");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(300, 200);
    Podloga gg = new Podloga();
    setContentPane(gg);
    setVisible(true);
}

// Testiranje prozora
```

```
public static void main(String[] args) {
public static void main(String[] args) {

Prozor p = new Prozor();

}
```

Kreiranje prozora – 1 bod
Ispravan metod kao reakcija na klik – 1 bod
Pravilno učitavanje podataka – 1 bod
Pravilno konvertovanje iz Kelvina – 1 bod
Pravilno konvertovanje iz Celzijusa – 1 bod

ZADATAK 17.

```
Fajl test.html
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-
scale=1">
<meta charset="UTF-8">
<link rel="stylesheet" href="test.css" >
</head>
<body>

<h1>Granice tabele</h1>
<p>Koristite CSS svojstvo border-collapse</p>

<table>
  <tr>
    <th>Ime</th>
    <th>Obrazovanje</th>
    <th>Grad</th>
  </tr>
```

```
<tr>
  <td>Jana</td>
  <td>Osnovno</td>
  <td><a href='http://podgorica.me/'>Podgorica</a></td>
</tr>
<tr>
  <td>Ema</td>
  <td>Srednje</td>
  <td>Bar</td>
</tr>
<tr>
  <td>Ivan</td>
  <td>Fakultet</td>
  <td>Berane</td>
</tr>
<tr>
  <td>Lana</td>
  <td>Fakultet</td>
  <td>Pljevlja</td>
</tr>
<tr>
  <td>Pero</td>
  <td>Srednja</td>
  <td>Ulcinj</td>
</tr>
</table>

</body>
</html>
```

```
Fajl test.css
table {
  border-collapse: collapse;
  border-spacing: 0;
  width: 100%;
  border: 2px solid blue;
}
table * {

font-size:120%;
}
th, td {
```

```

text-align: left;
padding-left: 8px;
border: 2px solid blue;
}

tr:nth-child(even) {
background-color: gray;
color: white;
}

th {
text-align: center;
}

```

Predloženo rješenje ne koristi CSS selektore klasa i id. Moguća su i druga rješenja.

Kreirana tabela bez stila – 1 bod

link "Podgorica" – 0.5 boda

granice tabele – 0.5 boda

padding – 0.5 boda

promjena izgleda neparnih redova – 1 bod

centriran naslovni red tabele – 0.5 boda

naslov i tekst ispred tabele – 0.5 boda

širina tabele – 0.5 bodova

veličina fonta u tabeli – 0.5 boda

ispravno povezivanje html i CSS – 0.5 boda



8

LITERATURA

1. Nikola Klem – Informatika, Udžbenik za prvi razred gimnazije, ZUNS 2006.
2. Milan Čabarkapa, Nevenka Spalević – “Metodička zbirka zadataka iz programiranja sa rešenjima u Pascal-u”, CET Biblioteka, 2007.
3. Dragan Urošević – “Algoritmi u programskom jeziku C” (+ disketa), Mikro knjiga, 1996.
4. Milan Vugdelija – “Programiranje i programiranje”, Sova Beograd, 2001.
5. Bruce Eickel – Misliti na Javi, prevod 4. izdanja Mikro knjiga, 2007.
6. Hebert Schildt – Java J2SE 5: kompletan priručnik, Mikro knjiga, 2006.



www.iccg.co.me

