# 

# V. RJEŠENJA ZADATAKA IZ RADNE BILJEŽNICE

## 1. STROJNA OPREMA RAČUNALA

### 1. Elektronički sklopovi računala

1. Jedan bit informacija. Dva stanja elektronske cijevi (ima struje/nema struje) mogu se zapisati kao 1 ili 0.

2. Tranzistor je mali, troši malo električne energije, ne zagrijava se, pouzdan je, relativno ga je jednostavno proizvesti, jeftin je.

3.

REGISTRI

UPRAVLJAČKA JEDINICA

ALU

Izvođenje aritmetičkih i logičkih operacija

Brze ćelije spremnika koje se upotrebljavaju za izvođenje operacija

Upravljanje, kontrola i usklađivanje rada dijelova procesora

4.

M:\SysPrint\Udzbenici\INFO\INFO 5-8 2013\INF8\RB\1 sirovina\1 Stroj oprema računala (Vinko)\slike\von Neumman -prazno.tif

**Procesor**

**Unutarnji spremnik**

**Vanjski spremnici**

**Izlazne jedinice**

**Ulazne jedinice**

2. Ulazno-izlazni pristupi

5. Izlazni uređaji služe za prikazivanje, odnosno predočavanje podataka obrađenih računalom. Podaci se mogu prikazivati u raznim oblicima (npr. tekst, slika, zvuk) i na raznim medijima (npr. zaslon, papir, zvučnik).

6. a) miš, tipkovnica, skener, crtača ploča, mikrofon, videokamera

b) monitor, pisač, crtač, projektor, zvučnik, slušalice

7. Pomoću raznih analogno-digitalnih pretvarača računalo analogne podatke pretvara u digitalni oblik, odnosno binarni zapis.

8. Jedinica bps (bits per second, odnosno bitova u sekundi) upotrebljava se za količinu prenesenih podataka u sekundi, odnosno posredno za brzinu prijenosa podataka.

9. Serijski (sljedni) prijenos podataka.

10. Prednost: prijenos više podataka odjednom (tj. istovremeno). Nedostatak: priključak s mnogo žica, debeli (ili široki) priključni kabel.

11. Specijalizirane sabirnice namijenjene a) proširivanju računala dodatnim karticama, odnosno b) spajanju vanjskih uređaja na računalo.

### 3. Svojstva računala

12. Računalo se može nadograditi hardverski i softverski, čime se mogu znatno proširiti njegove osnovne mogućnosti i tako ih upotrijebiti za rješavanje raznovrsnih zadaća.

13. Uređaji, uključujući i računala, koja nisu kompatibilna rade na osnovama koje međusobno nisu udružive, odnosno uskladive. Obično govorimo o hardverskoj nekompatibilnosti, no ona može biti i softverska.

14. S obzirom na širinu registra današnji se procesori obično dijele na 32-bitne i 64-bitne, iako ima onih s manjom i većim širinom. Širina registra određuje koliko bitova informacija procesor može odjednom primiti i obraditi, što u konačnici utiče na brzinu njegova rada. Što je procesorski registar veći, veća je brzina, odnosno učinkovitost procesora jer odjednom može obraditi više podataka.

15. b), d), e)

16. Mjernom jedinicom Mb/s koja označava količinu podataka u megabitima prenesenu u jednoj sekundi.

17. b)

18. Svi pokrenuti programi i svi otvoreni dokumenti privremeno se čuvaju u RAM-u. Što je RAM veći, računalo može raditi s više programa i dokumenata.

19. 2 TB = 2 · 1 024 = 2 024 MB = 2 024 · 1 024 = 2 097 152 kb.

### DODATNI ZADACI

20. U Colossusu se rabio binarni brojevni sustav koji se pokazao učinkovitiji od dekadskog sustava.

21. Binarni sustav može poslužiti kao osnova za binarno kodiranje događaja u računalu: ima struje = 1, nema struje = 0. U sprezi s Booleovom algebrom i binarnom aritmetikom predstavlja tehnički temelj digitalnog računala. Prednost binarnog sustava je što rabi samo dvije znamenke pa je postupak rada s binarno kodiranim podacima brži, jednostavniji i manje podložan pogreškama.

22. Princip rada elektronske cijevi jednostavan je: ima struje – nema struje. Ta dva stanja mogu se neposredno binarno kodirati: ima struje kao binarno 1, nema struje kao binarno 0.

23. Pouzdanost, relativno jeftina i jednostavna proizvodnja te male dimenzije tranzistora su značajke koje su omogućile konstruiranje manjih i jeftinih digitalnih uređaja, odnosno njihovu masovnu proizvodnju i primjenu u raznim segmentima ljudskog djelovanja.

24. Sklopka može biti u dva stanja: uključeno ili isključeno, odnosno ima struje ili nema struje. Ta dva stanja mogu se predstaviti, tj. binarno kodirati binarnim znamenkama: 1 za stanje uključeno, 0 za stanje isključeno.

25. Tranzistori i sposobnost magnetiziranja materijala bili su temelj razvoja modernih računala. (Magnetiziranje se koristi u vanjskim spremnicima za pohranjivanje podataka.)

26. Male dimenzije, mala potrošnja električne energije, malo zagrijavanje, velika pouzdanost i brzina rada, relativno jednostavna i jeftina proizvodnja.

27. Jedan bit informacija.

28. Integrirani krug povezuje veći broja tranzistora na maloj površini, čime se postiže izvršavanje složenijih operacije. Ostvarenjem više paralelnih sklopova za istu operaciju, moguće je odjednom izvršiti operaciju nad više bitova i tako znatno povećati brzinu izvršavanja složene operacije.

29. Mikroprocesor je elektronički sklop visokog stupnja integracije (sadrži na milijune tranzistora) koji izvodi različite vrste operacija od kojih su temeljne upravljačke i aritmetičko-logičke.

30. Mikroprocesor se sastoji od tri osnovna dijela: registara, upravljačke jedinice i aritmetičko-logičke jedinice. U registrima se (privremeno) čuvaju podaci za obradu, aritmetičko-logička jedinica obrađuje podatke, a upravljačka jedinica izvodi upravljačke procese.

31. Iz registara.

32. Veličina, odnosno širina, registra određuje širinu procesorske riječi. Što je procesorska riječ veća (dulja), to je procesor u stanju primiti i obraditi veću količinu podataka u jedinici vremena.

33. Širina procesorske riječi temeljna je značajka procesora koja utiče ne samo na hardver, nego i na softver računala.

34. Iz glavnog spremnika, odnosno RAM-a, šalju se podaci na obradu procesoru. Budući da je procesor znatno brži od RAM-a, mora čekati da mu RAM proslijedi podatke za obradu. Što RAM može brže poslati podatke procesoru, to će ih procesor prije obraditi. Zato procesor u svome radu ovisi o brzini RAM-a.

35. Računala i danas rade po von Neumannovom principu.

36. Podaci se spremaju u registrima na memorijske lokacije koje imaju svoje adrese.

37. Podaci u računalu razmjenjuju se u obliku električnih impulsa dvije naponske razine. Viša naponska razina predstavlja logičku jedinicu (1 - istina), a niža predstavlja logičku nulu ( 0 - neistina).

38. U digitalnim računalima postoje samo dva logička stanja podataka – logička jedinica (istina) i logička nula (neistina). To je zato jer električni impuls fizički može biti samo u dva stanja – ima struje (viši napon) ili nema struje (niži napon).

39. Sabirnica prikuplja podatke sa svih uređaja priključenih u računalu kako bi oni mogli razmjenjivati podatke i tako međusobno komunicirati.

40. Računalo se u pravilu sastoji od više različitih uređaja. Kada ne bi bilo sabirnica uređaji ne bi mogli međusobno komunicirati pa računalo ne bi moglo funkcionirati kao cjelina.

41. Bez sabirnica ne bi bilo moguće nadograđivati računalo novim uređajima.

42. Ulazni su uređaji specijalizirani za prihvat podataka iz okoline ili od korisnika, pretvaraju ih u digitalni oblik te ih prosljeđuju na obradu središnjoj jedinici. Izlazni uređaji prihvaćaju obrađene podatke izašle iz središnje jedinice i prezentiraju ih na jedan od korisniku razumljivih načina.

43. Može. Primjerice, zaslon osjetljiv na dodir. Zaslon koji je inače izlazni uređaj ovdje ujedno ima ulogu i ulaznog uređaja jer prima i interpretira naredbe korisnika putem dodira njegovih prstiju.

44. Da bi se vanjski uređaj priključio na računalo, računalo mora imati ugrađen odgovarajući ulazno-izlazni sklop koji je u stanju razmjenjivati podatke s vanjskih uređajem.

45.

ULAZNI UREĐAJI

M:\SysPrint\Udzbenici\INFO\INFO 5-8 2013\INF8\RB\1 sirovina\1 Stroj oprema računala (Vinko)\slike\sabirnica.tif

RADNI SPREMNIK

PROCESOR

IZLAZNI UREĐAJ

G L A V N A S A B I R N I C A

1. Korisnik na ulaznom uređaju izvrši neku akciju, npr. pritisne tipku na tipkovnici.

2. Ulazni uređaj taj događaj ili promjenu stanja pretvara u električni signal koji žičanom ili bezičnom vezom putuje do serijskog ili paralelnog ulaznog sklopa u računalu.

3. Ulazni sklop prosljeđuje signal glavnoj sabirnici koja ga odvodi do radnog spremnika i procesora na obradu.

4. Rezultat obrade prosljeđuju se glavnom sabirnicom do podsustava za pohranu podataka ili do izlaznog podsustava.

5. Ako je rezultat obrade izlazni podatak, izlazni sklop ga prosljeđuje odgovarajućem izlaznom uređaju, npr. na zaslonu monitora prikaže se slovo pritisnute tipke na tipkovnici.

46. Paralelni pristup prosljeđuje više električnih impulsa odjednom, dočim ih serijski pristup prosljeđuje jedan za drugim.

47. Paralelni pristup prosljeđuje podatke kroz više vodova istovremeno, dočim ih serijski pristup prosljeđuje samo kroz jedan vod.

48. Budući da paralelni pristup rabi više vodova, paralelni priključak je znatno širi nego serijski priključak.

49. Obično sa stražnje strane kućišta, tj. tamo gdje su ulazno-izlazni sklopovi smješteni na matičnoj ploči.

50. USB priključcima mogu se povezivati razni, različiti uređaji. To omogućava pojednostavljeno, jednoobrazno i jeftinije povezivanja vanjskih uređaja.

51. Jednostavnije je i jeftinije ako se vanjski uređaji priključuju na jednak načina rabeći univerzalan ulazno-izlazni sklop, odnosno USB priključak.

52. Imaju hardverske komponente koje međusobno nisu udružive. To znači da se sklopovlje iz jedne vrste računala ne može rabiti u drugoj vrsti računala.

53. O vrsti procesora ovisi nabavka nekih ostalih dijelova, tj. komponenti računala. Primjerice, o vrsti procesora ovisi nabavka matične ploče.

54. Krešo najprije mora odlučiti koju vrstu procesora želi nabaviti i od kojeg proizvođača (npr. Intel ili AMD).

55. Marijino računalo je staro. (Pentium IV procesori proizvodili su se u razdoblju od 2000. do 2008. godine.)

56. Samo na osnovu brzine procesora ne možemo zaključiti čije računalo brže radi.

57. Petri je poglavito potreban veći radni spremnik, odnosno RAM. To stoga jer obrađuje velike datoteke, a one se tijekom obrade privremeno smještaju u radni spremnik.

58. Sporiji je prijenos podataka s čvrstog diska u radni spremnik. Osim što sabirnica za čvrste diskove ima svoja ograničenja u brzini prijenosa podataka, prijenos dodatno usporava činjenica da je čvrsti disk mehanički uređaj pa se dio vremena dodatno potroši na pokretanje i zaustavljanje pokretnih dijelova (npr. na pokretanje glave za čitanje/pisanje podataka).

### ZADACI S NATJECANJA

1. b) memorija koja procesoru smanjuje vrijeme dohvata instrukcija i podataka

2. a) Procesor može izvršiti 4 milijarde osnovnih operacija u sekundi

## 2. IZRADA PREZENTACIJA

### 1. Oblikovanje prezentacije

1. Matrica slajda služi kao predložak svim slajdovima prezentacije. Svi sadržaji i oblikovanja koja primijenimo na matrici slajda odrazit će se na sve slajdove prezentacije.

2. 

3. Normalno; Prikaz matrice.

4. Slovni ili grafički element koji na jedinstven i prepoznatljiv način predstavlja nekoga ili nešto.

5. Potrebno je označiti slajd ili proizvoljan niz slajdova i na kartici *Umetanje* kliknuti na naredbu *Zaglavlje i podnožje*.

6. Datuma, teksta i broja slajda.

7. U dijaloškome okviru *Postavljanje stranice* u okviru *Numeracija slajdova od*: upišemo 0.

8. Podnožje ćete dodati klikom na *Umetanje > Zaglavlje i podnožje*. U prikazu matrice oblikujte okvire elemenata (datum, tekst i broj slajda).

9. Promjena predloška može uzrokovati potpuno drukčiji raspored rezerviranih mjesta na slajdu.

10. Potrebno je u prikazu matrice umetnuti željenu sliku na prvi slajd, odnosno, prvu matricu.

### 2. Rad s hipertekstom i hipervezama

11. Hiperveze ili linkovi su veze koje nas vode na neko drugo mjesto istog dokumenta, na drugi dokument, na drugu web stranicu ili u PowerPointu na drugi slajd prezentacije.

12.

1. Označimo objekt (tekst/sliku)

2. Na kartici *Umetanje* odaberemo *Hiperveza*

3. U odjeljku *Veza s* biramo vrstu veze

4. Kliknemo na *U redu*

13. Najprije ćemo označiti sliku, zatim odabrati *Umetanje > Hiperveza >Mjestom u ovom dokumentu >* pronaći ćemo željeni slajd i kliknuti *U redu.*

14. Najprije ćemo označiti određeni objekt, zatim odabrati *Umetanje > Hiperveza > Postojećom datotekom ili web stranicom* > u okviru *Pogledaj u* pronaći ćemo željenu prezentaciju i kliknuti *U redu*.

15. Akcijske gumbe. Služe za izvođenje neke akcije ili operacije, a isto tako i za hiperveze.

16. *Početak* ili *Polazni*.

17.



### 3. Izvedba prezentacije

18. Prezentacijski način rada.

19. 

20. Tako da prezentaciju spremimo u obliku projekcije.

21. Najjednostavniji način je uporabom tipkovnice: utipkamo broj željenog slajda i pritisnemo Enter, ili iz skočnog izbornika odaberemo naredbu *Idi na slajd* te klikom na željeni slajd.

22. Najsigurniji način za spremanje prezentacije koja tijekom izvođenja poziva i vanjske datoteke.

23. *Tab Datoteka > Spremanje i slanje > Zapakiraj prezentaciju za CD*. U dijaloškom okviru *Paket za CD*, u odjeljku *Daj naziv CD-u* upisujemo naziv paketa. Za dodavanje prezentacije ili drugih datoteka koje nisu automatski uključene potrebno je kliknuti na gumb *Dodaj* i odabrati datoteke koje želimo dodati. Klikom na gumb *Kopiraj u mapu* spremamo prezentaciju na disk računala.

24. b) tipkom W; f) tipkom B

25. Projekcija na dva monitora – Pogled izlagača. Ukoliko naše računalo ne podržava dva monitora, projekcija na dva monitora neće biti moguća.

26. Možemo, ali je potrebno na računalu koje nema PowerPoint instalirati PowerPointov razglednik. Program će omogućiti pregled i ispis prezentacije, no ne i njezino uređivanje.

### DODATNI ZADACI

27. 

28. Rješenje zadataka pogledajte na slici u radnoj bilježnici priloženoj uz zadatak.

Prilikom pisanja teksta o gradovima, informacije pronađite na internetu. Spremite prezentaciju pod imenom **Gradovi.pptx.**

29. Hiperveze postavljate na način da označite sliku te *odaberete Umetanje > Hiperveza > Mjestom u ovom dokumentu* > odaberite slajd i kliknite *U redu.*

U sljedećim zadacima (30. do 33.) učenici izrađuju kviz s detaljnim uputama napisanim u zadacima. Prilikom rješavanja zadataka učenici koriste hiperveze, akcijske gumbe, zaglavlje i podnožje, umeću glazbu s diska računala i brojeve na slajdove, određuju postavke projekcije, stvaraju Paket za CD te vizualno uređuju prezentaciju.

### ZADACI S NATJECANJA

1. c) Izgled ispisa

2. a) bilješke

3. c) prijeći na sljedeći slajd

4. a) zamračiti ekran

5. a) prijelazom miša preko poveznice

b) klikom miša na poveznicu

c) prijelazom miša ili klikom na poveznicu

6. d)

7. a) Shift + F5

8. c) Matrica jednodijelne stranice

9. c) 9

10. a) Paket za CD

## 3. LOGIČKE OSNOVE RAČUNALA

### Logičke izjave; Zapisivanje logičkih izjava

1. e) Logično da znam rješenje ovog zadatka.

2. c) nije logička izjava

3. a) istinita

4.

a) nedjelja > srijeda

b) slovo H > slovo B

c) nova godina > stara godina

5.

|  |  |  |
| --- | --- | --- |
| izjava A | izjava B | A ILI B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

### DODATNI ZADACI

6. Jest jer možemo utvrditi je li istinita ili lažna.

7. Jer nemožemo utvrditi je li istinita ili lažna - dosadnost je subjektivan pojam, a ne činjenično stanje.

8. Jedan bit informacija.

9. b) Stariji sam od tebe.

10. a) istina; b) istina; c) laž; d) istina; e) laž; f) istina

11. a) laž; b) laž; c) istina; d) laž

12. udžbenik AND informatika

13. pas NOT pekinezer

### ZADACI S NATJECANJA

1. c) Logično je da znam rješenje.

2. b) subota < petak

## 4A. PROGRAMSKI JEZIK PYTHON

### 1. Ponovimo

1.

a)

k = 10

while k<50:

print(k,end=' ')

k+=1

k = 1

while k<20:

print(k,end=' ')

k+=2

for k in range (10,50):

print(k,end=' ')

**for k in range (100,500,4):**

**print((k,end=' ')**

**for k in range (100,500,4):**

**print((k,end=' ')**

b)

for k in range (1,20,2):

print(k,end=' ')

2.

a)

for k in range (10,100):

print(k,end=' ') (

k = 10

while k<100:

print(k,end=' ')

k+=1

b)

for k in range (100,25,-3):

print(k,end=' ') (

k = 100

while k>25:

print(k,end=' ')

k-=3

3.

from math import \*

n=input('Do kojeg broja treba ispisati kvadrate i korijene?')

n=int(n)

for x in range (1,n+1):

print(x,'\t',x\*\*2,'\t', round(sqrt(x),4))

4.

from math import \*

a=input('a='); a=int(a)

b=input('b='); b=int(b)

if a>b:

a,b=b,a

for x in range (a,b+1):

print(x,'\t',x\*\*2,'\t', round(sqrt(x),4))

5.

from math import \*

a=input('a='); a=int(a)

b=input('b='); b=int(b)

k=input('k='); k=int(k)

if a>b:

a,b=b,a

for x in range (a,b+1,k):

print(x,'\t',x\*\*2,'\t', round(sqrt(x),4))

6.

**setpos(100,120)**

postavlja kornjaču u točku s koordinatama (-100, 120)

**position()**

vraća uređeni par brojeva, trenutne koordinate kornjače

**setpos(-100,120)**

postavlja kornjaču u točku s koordinatama (100, 120)

7.

from turtle import \*

title ('Bor');lt(90)

pu(); setposition (-40,-120); pd()

setposition (-40,-20); setposition (-160,-60);

setposition (-40,40); setposition (-140,0);

setposition (-40,80);setposition (-120,60);

setposition (-40,120); setposition (-100,120);

setposition (0,200);

setposition (100,120);setposition (40,120);

setposition (120,60); setposition (40,80);

setposition (140,0); setposition (40,40);

setposition (160,-60); setposition (40,-20);

setposition (40,-120); setposition (-40,-120)

8.

from turtle import \*

def niz(m,a):

x=position()

for k in range (m):

for x in range(4):

fd(a);rt(90)

fd(a);rt(45)

for x in range(2):

fd(a/2); rt(45)

fd(a);rt(135)

lt(45);bk(a);rt(90)

fd(a);lt(90)

pu();setpos(x);pd()

def stepenice(n,m,a):

pocetak=position()

for k in range (n):

niz(m,a); fd(a); rt(45);fd(a/2);lt(45)

pu();setpos(pocetak);rt(90);fd(m\*a);lt(90);pd()

for k in range(n):

red(n,a)

n -=1

def red(n,a):

for k in range(n):

for m in range(2):

fd(a);rt(45);

fd(a/2);rt(135)

rt(45);fd(a/2);lt(45)

pu();rt(45);bk((n-1)\*a/2)

lt(45);fd(a);pd()

title('Stepenice')

lt(90);st()

a=textinput('Duljina brida kocke','a=');a=int(a)

m=textinput('Broj kocaka u redu','m=');m=int(m)

n=textinput('Broj redova kocaka','n=');n=int(n)

stepenice (n,m,a)

9.

from turtle import \*

from random import \*

def boja():

c=randrange(0,256)

z=randrange(0,256)

p=randrange(0,256)

color((0,0,0), (c,z,p))

def krug(r):

boja(); begin\_fill();

pu(); rt(90);fd(r);lt(90);pd();circle(r);pu()

lt(90);fd(r);rt(90);end\_fill()

def niz\_krugova(n,r):

for k in range(n):

krug(r)

rt(90);fd(2\*r);lt(90)

lt(90);fd(2\*n\*r);rt(90)

def piramida\_krugova(n,r):

for k in range(n,0,-1):

niz\_krugova(k,r)

rt(30);fd(2\*r);lt(30)

title('Piramida krugova')

lt(90);st(); colormode(255)

r=textinput('Polumjer','r=');r=int(r)

n=textinput('Broj krugova u prvom redu','n=');n=int(n)

piramida\_krugova(n,r)

10.

from turtle import \*

def trokut(a\_x,a\_y,b\_x,b\_y,c\_x,c\_y):

pu();setpos(a\_x,a\_y);pd()

setpos(b\_x,b\_y);setpos(c\_x,c\_y);setpos(a\_x,a\_y);

def simetrala (x1,y1,x2,y2):

pu();setpos(x1,y1); seth(towards(x2,y2));

a=distance(x2,y2); fd(a/2); rt (90); pd();

fd (200);bk(400)

title('Trokut')

lt(90);st()

a\_x=textinput('Točka A','x=');a\_x=int(a\_x)

a\_y=textinput('Točka A','y=');a\_y=int(a\_y)

b\_x=textinput('Točka B','x=');b\_x=int(b\_x)

b\_y=textinput('Točka B','y=');b\_y=int(b\_y)

c\_x=textinput('Točka C','x=');c\_x=int(c\_x)

c\_y=textinput('Točka C','y=');c\_y=int(c\_y)

trokut(a\_x,a\_y,b\_x,b\_y,c\_x,c\_y)

simetrala (a\_x,a\_y,b\_x,b\_y)

simetrala (b\_x,b\_y,c\_x,c\_y)

simetrala (a\_x,a\_y,c\_x,c\_y)\_\_\_

11.

from turtle import \*

def četverokut(a\_x,a\_y,b\_x,b\_y,c\_x,c\_y, d\_x,d\_y):

pu();setpos(a\_x,a\_y);pd()

setpos(b\_x,b\_y);setpos(c\_x,c\_y)

setpos(d\_x,d\_y);setpos(a\_x,a\_y)

def dijagonala (x1,y1,x2,y2):

pu();setpos(x1,y1); pd();setpos(x2,y2);

title('Četverokut')

lt(90);st()

a\_x=textinput('Točka A','x=');a\_x=int(a\_x)

a\_y=textinput('Točka A','y=');a\_y=int(a\_y)

b\_x=textinput('Točka B','x=');b\_x=int(b\_x)

b\_y=textinput('Točka B','y=');b\_y=int(b\_y)

c\_x=textinput('Točka C','x=');c\_x=int(c\_x)

c\_y=textinput('Točka C','y=');c\_y=int(c\_y)

d\_x=textinput('Točka D','x=');d\_x=int(d\_x)

d\_y=textinput('Točka D','y=');d\_y=int(d\_y)

četverokut(a\_x,a\_y,b\_x,b\_y,c\_x,c\_y,d\_x,d\_y )

dijagonala (a\_x,a\_y,c\_x,c\_y)

dijagonala (b\_x,b\_y,d\_x,d\_y)

### 2. Logičke funkcije u programiranju

12.

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A or** **B** |
| **False** | **False** | **False** |
| **False** | **True** | **True** |
| **True** | **False** | **True** |
| **True** | **True** | **True** |

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A** and **B** |
| **False** | **False** | **False** |
| **False** | **True** | **False** |
| **True** | **False** | **False** |
| **True** | **True** | **True** |

|  |  |
| --- | --- |
| **A** | not **A** |
| **False** | **True** |
| **True** | **False** |

13.

1. not

2. and

3. or

14.

a) print (not 4+9-5<45)= print ( not True)= False

b) print (not 4+9>45)= print (not False)= True

c) print ( 47<23+24 or 23+24==47)= print (False or True)= True

d) print (7==3+4 or 23+24==47)= print ( True or True)= True

e) print (47==23+24 or 23+24==7)= print (True or False)= True

f) print (47>23+24 or 23+22==47)= print ( False or False)= False

g) print (9+5==45 and 9\*5==45)= print ( False and True)= False

h) print (9+5==14 and 9\*5==45 )= print (True and True)= True

i) print ( 9+5==14 and 9\*5==35)= print (True and False)= False

j) print ( 9+5==45 and 9\*5==54)= print (False and False)= False

k) print ((34>29) or ( 3>0 and 5<87 and 77!=11\*7)) = print (True or (True and True and False))= print (True or False)= True

l) print (( 9\*8>45 or 6+6>13) and (not 5==6))= print ((True or False) and (not False))= print (True and True)= True

m) print ( (not 4+2==5+1) or (5/5<1 or 3+7>5) )= print ((not True) or (False or True))= print (False or True)= True

n) print (not(5==6+7 or 45==78-56) and (56-47==19 or 36/4==9)) = print (not (False or False) and (False or True))= print (True and True)= True

15.

n=input('Upiši broj:')

n=int(n)

if n>=10 and n<=99:

print(n,' je dvoznamenkast')

else:

print(n,' nije dvoznamenkast')

16.

uspjeh=input('Upiši uspjeh na dvije decimale:')

uspjeh=float(uspjeh)

if uspjeh>=2.0 and uspjeh<2.50:

print('dovoljan')

elif uspjeh>=2.50 and uspjeh<3.50:

print('dobar')

elif uspjeh>=3.50 and uspjeh<4.50:

print('vrlo dobar')

elif uspjeh>=4.50 and uspjeh<=5.0:

print('odličan')

elif uspjeh<2.0 or uspjeh>5.0:

print('Uspjeh mora biti između 2 i 5.')

17.

godine=input('Upiši godine života:')

godine=int(godine)

if godine<1:

print('novorođenče')

elif godine>=1 and godine<3:

print('jaslice')

elif godine>=3 and godine<7:

print('vrtić')

elif godine>=7 and godine<15:

print('osnovna škola')

elif godine>=15 or godine<19:

print('srednja škola')

elif godine>19:

print('odrasli')

18.

if mj==7 and dan<23:

print ('RAK')

if mj==7 and dan>=23:

print('LAV')

if mj==8 and dan>=22:

print('DJEVICA')

if mj==9 and dan<23:

print ('DJEVICA')

if mj==9 and dan>=23:

print('VAGA')

if mj==10 and dan<23:

print ('VAGA')

if mj==10 and dan>=23:

print('ŠKORPION')

if mj==11 and dan<22:

print ('ŠKORPION')

if mj==11 and dan>=22:

print('STRIJELAC')

if mj==12 and dan<21:

print ('STRIJELAC')

if mj==12 and dan>=21:

print('JARAC')

dan= input('Dan=')

mj= input('Mjesec=')

dan= int (dan)

mj= int (mj)

if mj==1 and dan<21:

print ('JARAC')

if mj==1 and dan>=21:

print('VODENJAK')

if mj==2 and dan<19:

print ('VODENJAK')

if mj==2 and dan>=19:

print('RIBE')

if mj==3 and dan<21:

print ('RIBE')

if mj==3 and dan>=21:

print('OVAN')

if mj==4 and dan<21:

print ('OVAN')

if mj==4 and dan>=21:

print('BIK')

if mj==5 and dan<22:

print ('BIK')

if mj==5 and dan>=22:

print('BLIZANCI')

if mj==6 and dan<22:

print ('BLIZANCI')

if mj==6 and dan>=22:

print('RAK')

19.

n= input('n=')

n= int (n)

for k in range (1,n+1):

if n%k==0:

if k>=10 and k<=99:

print (k, end=' ')

prvi=input ('Koji dan u tjednu je Nova godina?')

if prvi=='ponedjeljak':

početak=0

if prvi=='utorak':

početak=1

if prvi=='srijeda':

početak=2

if prvi=='četvrtak':

početak=3

if prvi=='petak':

početak=4

if prvi=='subota':

početak=5

if prvi=='nedjelja':

početak=6

dan=input ('Dan:')

mjesec=input ('Mjesec:')

godina=input('Godina:')

dan=int(dan)

mjesec=int(mjesec)

godina=int(godina)

broj=0

prijestupna=False

if godina%4==0 and godina%100!=0:

prijestupna=True

elif godina%400==0:

prijestupna=True

print(prijestupna)

if mjesec>1:

broj=broj+31

if mjesec>2:

if prijestupna:

broj=broj+29

else:

broj=broj+28

if mjesec>3:

broj=broj+31

if mjesec>4:

broj+=30

if mjesec>5:

broj+=31

if mjesec>6:

broj+=30

if mjesec>7:

broj+=31

if mjesec>8:

broj+=31

if mjesec>9:

broj+=30

if mjesec>10:

broj+=31

if mjesec>11:

broj+=30

print(broj)

broj=(broj+dan+početak-1)%7

if broj==0:

d='ponedjeljak'

elif broj==1:

d='utorak'

elif broj==2:

d='srijeda'

elif broj==3:

d='četvrtak'

elif broj==4:

d='petak'

elif broj==5:

d='subota'

elif broj==6:

d='nedjelja'

print(dan,'.',mjesec,'. bit će',d,'.')

20.

prvi=input('Koji dan u tjednu je Nova godina?')

if prvi=='ponedjeljak':

početak=0

if prvi=='utorak':

početak=1

if prvi=='srijeda':

početak=2

if prvi=='četvrtak':

početak=3

if prvi=='petak':

početak=4

if prvi=='subota':

početak=5

if prvi=='nedjelja':

početak=6

dan=input ('Dan:')

mjesec=input ('Mjesec:')

godina=input('Godina:')

dan=int(dan)

mjesec=int(mjesec)

godina=int(godina)

broj=0

prijestupna=False

if godina%4==0 and godina%100!=0:

prijestupna=True

elif godina%400==0:

prijestupna=True

if mjesec>1:

broj+=31

if mjesec>2:

if prijestupna:

broj+=29

else:

broj+=28

if mjesec>3:

broj+=31

if mjesec>4:

broj+=30

if mjesec>5:

broj+=31

if mjesec>6:

broj+=30

if mjesec>7:

broj+=31

if mjesec>8:

broj+=31

if mjesec>9:

broj+=30

if mjesec>10:

broj+=31

if mjesec>11:

broj+=30

print(broj)

broj=(broj+dan+početak-1)%7

if broj==0:

d='ponedjeljak'

elif broj==1:

d='utorak'

elif broj==2:

d='srijeda'

elif broj==3:

d='četvrtak'

elif broj==4:

d='petak'

elif broj==5:

d='subota'

elif broj==6:

d='nedjelja'

print(dan,'.',mjesec,'. bit će',d,'.')

### 3. Rekurzivne procedure

21.

def ispis(n):

if n==2:

print(2, end=' ')

else:

ispis(n-2)

print(n,end=' ')

n=input('Do kojeg broja treba ispisati? ')

n=int(n)

if n % 2==0:

ispis(n)

else:

ispis(n-1)

a) Za ispis parnih brojeva manjih od zadanog broja.

b) Da, jer unutar sebe sadrži poziv iste procedure ispis (n).

c)

def ispis(n):

if n==1:

print(1, end=' ')

else:

ispis(n-2)

print(n,end=' ')

n=input('Do kojeg broja treba ispisati? ')

n=int(n)

if n % 2==0:

ispis(n-1)

else:

ispis(n)

22.

def zbroj(n):

if n==2:

return 2

else:

return n+zbroj(n-2)

n=input('Koliko parnih brojeva treba zbrojti? ')

n=int(n)

print('Zbroj prvih',n,' parnih prirodnih brojeva je',zbroj(2\*n))

OKRENI STRANICU

23.

def zbroj(n):

if n==1:

return 1

else:

return n+zbroj(n-2)

n=input('Koliko neparnih brojeva treba zbrojti?')

n=int(n)

print('Zbroj prvih',n,' neparnih prirodnih brojeva je',zbroj(2\*n-1))

24.

def umnožak(n):

if n==2:

return 2

else:

return n\*umnožak(n-2)

n=input('Upiši n: ')

n=int(n)

if n%2==0:

print('Umnožak parnih brojeva manjih od',n,'je',umnožak(n-2))

else:

print('Umnožak parnih brojeva manjih od',n,'je',umnožak(n-1))

25.

def piramida (n,a):

if n==1:

kvadrat(a)

else:

niz\_kvadrata(n,a);pu()

#pomak na početak sljedećeg reda

lt(90);fd(n\*a-a);rt(90);fd(a);

rt(90); fd(a/2);lt(90); pd()

piramida(n-1,a)

title('Piramida kvadrata')

lt(90);colormode(255)

a=textinput('Stranica','a=')

n=textinput('Broj kvadrata','n=')

a=int(a); n=int(n);

piramida(n,a)

from turtle import \*

from random import \*

def boja():

c= randrange(0,256)

z= randrange(0,256)

p= randrange(0,256)

color(c,z,p)

def kvadrat (a):

boja()

begin\_fill()

for k in range(4):

fd(a); rt(90)

end\_fill()

def niz\_kvadrata (n,a):

if n==1:

kvadrat(a)

else:

kvadrat(a)

rt(90);fd(a);lt(90)

niz\_kvadrata(n-1,a)

26.

from turtle import \*

from random import \*

def boja():

c= randrange(0,256)

z= randrange(0,256)

p= randrange(0,256)

color(c,z,p)

def trokut (a):

boja()

begin\_fill()

for k in range(3):

fd(a); lt(120)

end\_fill()

def niz\_trokuta (n,a):

if n==1:

trokut(a)

else:

trokut(a)

fd(a);

niz\_trokuta(n-1,a)

def piramida (n,a):

if n==1:

trokut(a)

else:

niz\_trokuta(n,a);pu()

# pomak na početak sljedećeg reda

bk(n\*a-a);lt(60);fd(a);rt(60); pd()

piramida(n-1,a)

title('Piramida trokuta')

colormode(255)

a=textinput('Stranica','a=')

n=textinput('Broj trokuta','n=')

a=int(a); n=int(n);

piramida(n,a)

27.

from turtle import \*

from random import \*

def boja():

c= randrange(0,256)

z= randrange(0,256)

p= randrange(0,256)

color(c,z,p)

def kvadrat (a):

boja()

begin\_fill()

for k in range(4):

fd(a); rt(90)

end\_fill()

def niz\_kvadrata (n,a):

if n==1:

kvadrat(a)

else:

kvadrat(a)

rt(90);fd(a);lt(90)

niz\_kvadrata(n-1,a)

def spirala (a, korak, kut, granica):

if a<granica:

kvadrat(a); fd (a); lt(kut);

spirala(a+korak,korak, kut, granica)

title('Spirala kvadrata')

lt(90);colormode(255)

a=textinput('Stranica','a=')

korak=textinput('Korak','korak=')

kut=textinput('Kut','kut=')

granica=textinput('Granica','granica=')

a=int(a); korak=int(korak); kut=int(kut); granica=int(granica);

spirala(a,korak, kut, granica)

28.

from turtle import \*

def zvijezda (a):

if a<2:

return True

for k in range (5):

fd (a);rt(144);

zvijezda(a/3)

title ('Zvijezda')

color ("blue");lt(90)

a=textinput('Veličina','a=');a=int(a)

zvijezda (a)

29.

from turtle import \*

def stranica(n,a):

if n==0:

fd(a)

return True

stranica(n-1,a/3);lt(60)

stranica(n-1,a/3);rt(120)

stranica(n-1,a/3);lt(60)

stranica(n-1,a/3)

title ('Pahulja')

colormode (255);lt(90)

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

for k in range(3):

stranica(n,a);rt(120)

30.

from turtle import \*

def stranica(n,a):

if n==0:

fd(a)

return True

stranica(n-1,a/2);rt(90);stranica(n-1,a/2);lt(90);stranica(n-1,a/2)

title ('Kvadrat')

colormode (255);lt(90)

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

for k in range(4):

stranica(n,a);rt(90)

31.

from turtle import \*

def H(n,a):

if n==0:

return True

lt(90);fd(a);rt(90);fd(a)

H(n-1,a/2); bk(2\*a)

H(n-1,a/2); fd(a);rt(90);fd(2\*a);lt(90);fd(a)

H(n-1,a/2); bk(2\*a)

H(n-1,a/2); fd(a);lt(90);fd(a);rt(90)

title ('Slovo H')

colormode (255);lt(90)

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

H(n,a)

32.

from turtle import \*

from math import \*

def stranica(n,a):

if n==0:

fd(a)

return True

rt(45);stranica(n-1,a/sqrt(2))

lt(90);stranica(n-1,a/sqrt(2))

rt(45)

title ('Levijev fraktal')

colormode (255);rt(90)

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

for k in range(3):

stranica(n,a);lt(90)

33.

from turtle import \*

from math import \*

def stranica(n,a):

if n==0:

fd(a)

return True

rt(45);stranica(n-1,a/sqrt(2))

lt(90);stranica(n-1,a/sqrt(2))

rt(45)

title ('Levijev fraktal')

colormode (255);rt(90)

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

for k in range(4):

stranica(n,a);lt(90)

34.

from turtle import \*

def stranica(n,a,kut):

if n==0:

fd(a)

return True

stranica(n-1,a/2,kut);rt(90-kut/2)

stranica(n-1,a/2,kut);lt(180-kut)

stranica(n-1,a/2,kut);rt(90-kut/2)

stranica(n-1,a/2,kut)

title ('Cesaro fraktal')

colormode (255);lt(90)

a=textinput('Veličina','a=');a=int(a)

kut=textinput('Kut','kut=');kut=int(kut)

n=textinput('Dubina rekurzije','n=');n=int(n)

for k in range(4):

stranica(n,a, kut);rt(90)

35.

from turtle import \*

from random import \*

def boja():

c=randrange(0,256)

z=randrange(0,256)

p=randrange(0,256)

color((0,0,0),(c,z,p))

def kvadrat(a):

boja();begin\_fill()

for k in range(4):

fd(a);rt(90)

end\_fill()

def podijeli\_kvadrat(n,a):

if n==0:

kvadrat(a)

return True

podijeli\_kvadrat(n-1,a/3)

pu();rt(90);fd(2\*a/3);lt(90);pd()

podijeli\_kvadrat(n-1,a/3)

pu();lt(90);fd(a/3);rt(90);fd(a/3);pd()

podijeli\_kvadrat(n-1,a/3)

pu();lt(90);fd(a/3);rt(90);fd(a/3);pd()

podijeli\_kvadrat(n-1,a/3)

pu();rt(90);fd(2\*a/3);lt(90);pd()

podijeli\_kvadrat(n-1,a/3)

pu();lt(90);fd(2\*a/3);rt(90); bk(2\*a/3);pd()

title ('Podijeli kvadrat')

colormode (255);lt(90)

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

podijeli\_kvadrat(n,a)

36.

from turtle import \*

from random import \*

def boja():

c=randrange(0,256)

z=randrange(0,256)

p=randrange(0,256)

color((0,0,0),(c,z,p))

def trokut(a):

boja();begin\_fill()

rt(30)

for k in range(3):

fd(a);rt(120)

end\_fill()

lt(30)

def podijeli\_trokut(n,a):

if n==0:

trokut(a)

return True

podijeli\_trokut(n-1,a/2)

pu();rt(90);fd(a/2);lt(90);pd()

podijeli\_trokut(n-1,a/2)

pu();lt(90);fd(a/2);rt(120);fd(a/2);lt(30);pd()

podijeli\_trokut(n-1,a/2)

pu();rt(30);bk(a/2);lt(30);pd()

title ('Podijeli trokut')

colormode (255);lt(90)

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

podijeli\_trokut(n,a)

37.

from turtle import \*

from random import \*

def boja():

c=randrange(0,256)

z=randrange(0,256)

p=randrange(0,256)

color((0,0,0),(c,z,p))

def L(a):

boja()

begin\_fill()

fd(a);rt(90);fd(a/2);rt(90)

fd(a/2);lt(90)

fd(a/2);rt(90);fd(a/2);rt(90)

fd(a);rt(90)

end\_fill()

def veliki\_L (n,a):

if n==0:

L (a)

return True

veliki\_L (n-1,a/2)

fd(a);rt(90)

veliki\_L (n-1,a/2)

pu();lt(90);bk(a);rt(90);fd(a);rt(180);pd()

veliki\_L (n-1,a/2)

pu();fd(a/2);rt(90);fd(a/4)

lt(90);fd(a/4);rt(90);pd()

veliki\_L (n-1,a/2)

pu();bk(a/4);lt(90);fd(a/4);rt(90);pd()

title ('Slovo L')

colormode (255);lt(90)

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

veliki\_L (n,a)

38.

from turtle import \*

from random import \*

def boja():

c=randrange(0,256)

z=randrange(0,256)

p=randrange(0,256)

color((0,0,0),(c,z,p))

def pola\_šesterokuta(a):

boja();begin\_fill()

rt(30);fd(a);rt(60); fd(a);rt(60);fd(a);rt(120);fd(2\*a);rt(90)

end\_fill()

def lik(n,a):

if n==0:

pola\_šesterokuta(a)

return True

rt(90);fd(a/2);lt(90)

lik (n-1,a/2)

rt(90);fd(3\*a/2);rt(150)

lik (n-1,a/2)

rt(90);fd(a);lt(150)

lik (n-1,a/2)

rt(90);fd(a);lt(150)

lik (n-1,a/2)

rt(90);fd(a);rt(150)

title ('Pola šesterokuta')

colormode (255);lt(90)

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

lik(n,a)

### 4. Grafičke kontrole

39. a) Grafičke kontrole omogućuje modul **tkinter**. Želimo li koristiti grafičke kontrole na početku programa modul unosimo naredbom: **from tkinter import \***

b) Grafičke kontrole stavljamo u zasebni prozor. Napiši naredbu kojom ćeš zadati prozor za grafičke kontrole imena **prozor**: **prozor = Tk()**

c) Grafičke kontrole unutar prozora smještamo u okvire.  
1. Napiši naredbu kojom ćeš u prozoru **prozor** zadati okvir **glavni**. **glavni=Frame(prozor)**

2. Napiši naredbu koja zadaje veličinu okvira 1 red i 1 stupac. **glavni.grid(columns=1,rows=1)**

d) Napiši naredbu koja zadaje naredbeni gumb **gumb** u okviru **glavni** za crtanje pravokutnika, koji poziva potprogram **pravokutnik gumb=Button(glavni, text="Pravokutnik",command=pravokutnik)**

### e) Postavi naredbeni gumb u 1. red i 1. stupac. gumb.grid(column=1, row=1)

40.

from turtle import \*

from tkinter import \*

def pravokutnik():

for k in range(2):

fd(50);rt(90); fd(100);rt(90);

prozor = Tk()

prozor.title('Gumb')

title('Pravokutnik')

colormode(255);st();lt(90);width(4)

glavni=Frame(prozor)

glavni.grid(columns=1,rows=1)

gumb=Button(glavni, text="Pravokutnik",command=pravokutnik)

gumb.grid(column=1, row=1)

41.

from turtle import \*

from tkinter import \*

def kružnica():

pu();rt(90);fd(20);lt(90);pd();circle(20);pu();rt(90);bk(20);lt(90)

def pomak():

pu();rt(90);fd(40);lt(90);pd()

prozor = Tk()

prozor.title('Gumbi')

title('Niz kružnica')

lt(90);

glavni=Frame(prozor)

glavni.grid(columns=1,rows=2)

gumb=Button(glavni, text="Kružnica",command=kružnica)

gumb.grid(column=1, row=1)

gumb=Button(glavni, text="Pomak",command=pomak)

gumb.grid(column=1, row=2)

42.

from turtle import \*

from tkinter import \*

from random import \*

def boja():

c=randrange(0,256)

z=randrange(0,256)

p=randrange(0,256)

color((0,0,0),(c,z,p))

def trokut():

rt(30); boja();begin\_fill()

for k in range(3):

fd(40);rt(120)

end\_fill();lt(30)

def desno():

pu();rt(90);fd(40);lt(90);pd()

def lijevo():

pu();lt(90);fd(40);rt(90);pd()

def gore():

pu();rt(30);fd(40);lt(30);pd()

def brisi():

reset();lt(90)

prozor = Tk()

prozor.title('Gumbi')

title('Piramida trokuta')

colormode(255);lt(90);

glavni=Frame(prozor)

glavni.grid(columns=1,rows=5)

gumb1=Button(glavni, text="Trokut",command=trokut)

gumb1.grid(column=1, row=1)

gumb2=Button(glavni, text="Desno",command=desno)

gumb2.grid(column=1, row=2)

gumb3=Button(glavni, text="Lijevo",command=lijevo)

gumb3.grid(column=1, row=3)

gumb4=Button(glavni, text="Gore",command=gore)

gumb4.grid(column=1, row=4)

gumb5=Button(glavni, text="Briši",command=brisi)

gumb5.grid(column=1, row=5)

43.

from tkinter import \*

from turtle import \*

def kvadrat():

a=int(duljina.get())

for k in range(4):

fd(a);rt(90)

prozor = Tk()

prozor.title('Kontrole')

title('Kvadrat')

st();lt(90)

glavni = Frame(prozor)

glavni.grid(column=2, row=2)

#upis duljine stranice

t1=Label(glavni, text="Duljina")

t1.grid(column=1, row=1)

duljina=Entry(glavni)

duljina.grid(column=2, row=1)

#gumb za crtanje kvadrata

gumb=Button(glavni, text="Kvadrat", command=kvadrat)

gumb.grid(column=1, row=3)

44.

from tkinter import \*

from turtle import \*

from random import \*

def boja():

c=randrange(0,256); z=randrange(0,256)

p=randrange(0,256); color((0,0,0),(c,z,p))

def osmerokut():

a=int(duljina.get()); boja();begin\_fill()

for k in range(8):

fd(a);rt(45)

end\_fill()

def pomak():

a=int(duljina.get()); pu();rt(135)

for k in range(3):

fd(a);lt(45)

pd()

def vrati():

a=int(duljina.get()); pu();lt(135)

for k in range(3):

fd(a);rt(45)

pd()

def red():

a=int(duljina.get())

pu();fd(a);rt(45);fd(a);lt(90); fd(a);rt(45);pd()

def briši():

reset();lt(90)

prozor = Tk(); prozor.title('Kontrole')

title('Mreža osmerokuta'); colormode(255);st();lt(90)

glavni = Frame(prozor); glavni.grid(column=2, row=7)

#upis duljine stranice

t1=Label(glavni, text="Duljina") t1.grid(column=1, row=1)

duljina=Entry(glavni); duljina.grid(column=2, row=1)

#gumbi za crtanje

gumb1=Button(glavni, text="Osmerokut", command=osmerokut)

gumb1.grid(column=1, row=2)

gumb2=Button(glavni, text="Boja", command=boja)

gumb2.grid(column=1, row=3)

gumb3=Button(glavni, text="Pomak", command=pomak)

gumb3.grid(column=1, row=4)

gumb4=Button(glavni, text="Vrati", command=vrati)

gumb4.grid(column=1, row=5)

gumb5=Button(glavni, text="Red", command=red)

gumb5.grid(column=1, row=6)

gumb6=Button(glavni, text="Briši", command=briši)

gumb6.grid(column=1, row=7)

45.

from tkinter import \*

def izračun():

a = float(duljina.get())

b = float(širina.get())

h = float(visina.get())

P=(2\*a+2\*b)\*h

iznos=P\*float(cijena.get())

porez1= round(iznos\*float(porez.get())/100,2)

r1.config(text = 'Površina za bojanje: ' + str(P) + ' kv. m')

r2.config(text = 'Iznos: ' + str(iznos) + ' kn')

r3.config(text = 'Porez: ' + str(porez1) + ' kn')

prozor = Tk()

prozor.title("Program za soboslikare")

glavni = Frame(prozor)

glavni.grid(column=3, row=9)

#upis podataka

t1=Label(glavni, text="Duljina:", font=('Arial',12,'bold'))

t1.grid(column=1, row=1)

duljina=Entry(glavni, font=('Arial',12,'bold'))

duljina.grid(column=2, row=1)

t2=Label(glavni, text="Širina:", font=('Arial',12,'bold'))

t2.grid(column=1, row=2)

širina=Entry(glavni, font=('Arial',12,'bold'))

širina.grid(column=2, row=2)

t3=Label(glavni, text="Visina:", font=('Arial',12,'bold'))

t3.grid(column=1, row=3)

visina=Entry(glavni, font=('Arial',12,'bold'))

visina.grid(column=2, row=3)

t4=Label(glavni, text="Cijena:", font=('Arial',12,'bold'))

t4.grid(column=1, row=4)

cijena=Entry(glavni, font=('Arial',12,'bold'))

cijena.grid(column=2, row=4)

t5=Label(glavni, text="Porez u %:", font=('Arial',12,'bold'))

t5.grid(column=1, row=5)

porez=Entry(glavni, font=('Arial',12,'bold'))

porez.grid(column=2, row=5)

#ispis rezultata

r1=Label(glavni, font=('Arial',12,'bold'))

r1.grid(column=1, row=6, columnspan=2)

r2=Label(glavni, font=('Arial',12,'bold'))

r2.grid(column=1, row=7, columnspan=2)

r3=Label(glavni, font=('Arial',12,'bold'))

r3.grid(column=1, row=8, columnspan=2)

#gumb za pretvorbu

gumb=Button(glavni, font=('Arial',12,'bold'),text="Izračunaj", command=izračun)

gumb.grid(column=3, row=9)

### 5. Primjena programiranja u matematici

46.

from tkinter import \*

from turtle import \*

from random import \*

from math import \*

def kružnica(r):

pu();rt(90);fd(r);lt(90); pd();circle(r);pu();rt(90);bk(r);lt(90)

def pravokutnik():

a=int(duljina.get()); b=int(širina.get())

for k in range(2):

fd(b);rt(90);fd(a);rt(90)

r\_opisana=sqrt(a\*\*2+b\*\*2)/2

pu();rt(90);fd (a/2);lt(90);fd(b/2);pd()

color('red');kružnica(r\_opisana);pd()

setpos(0,0)

t3.config(text="Opseg o= "+str(2\*(a+b)))

t4.config(text="Površina P= "+str(a\*b))

t5.config(text="Dijagonala d= "+str(2\*r\_opisana))

t6.config (text= "Polumjer kružnice r= "+str(r\_opisana))

prozor = Tk(); prozor.title('Stranice')

title('Pravokutnik');lt(90); colormode(255);st();

glavni = Frame(prozor); glavni.grid(column=1, row=7)

#upis

t1=Label(glavni, text="a="); t1.grid(column=1, row=1)

duljina = Entry(glavni); duljina.grid(column=2, row=1)

t2=Label(glavni, text="b="); t2.grid(column=1, row=2)

širina = Entry(glavni); širina.grid(column=2, row=2)

#opseg i površina

t3=Label(glavni); t3.grid(column=1, row=3, columnspan=2)

t4=Label(glavni); t4.grid(column=1, row=4,columnspan=2)

t5=Label(glavni); t5.grid(column=1, row=5,columnspan=2)

t6=Label(glavni); t6.grid(column=1, row=6,columnspan=2)

#gumb za crtanje

gumb=Button(glavni, text="Crtaj", command=pravokutnik)

gumb.grid(column=3, row=7)

47.

from tkinter import \*

from turtle import \*

from random import \*

from math import \*

def kružnica(r):

pu();rt(90);fd(r);lt(90); pd();circle(r);pu();rt(90);bk(r);lt(90)

def kvadrat():

a=int(duljina.get())

for k in range(4):

fd(a);rt(90)

r\_opisana=a\*sqrt(2)/2

r\_upisana=a/2

pu();rt(90);fd (a/2);lt(90);fd(a/2);pd()

color('red');kružnica(r\_opisana);pd()

setpos(0,0)

pu();rt(90);fd (a/2);lt(90);fd(a/2);pd()

color('blue');kružnica(r\_upisana);pd(); bk(a/2)

t2.config(text="Opseg o= "+str(4\*a))

t3.config(text="Površina P= "+str(a\*a))

t4.config(text="Dijagonala d= "+str(round(a\*sqrt(2),4)))

t5.config (text= "Opisana kružnica R= "+str(round(r\_opisana,4)))

t6.config (text= "Upisana kružnica r= "+str(round(r\_upisana,4)))

prozor = Tk(); prozor.title('Podaci')

title('Kvadrat');lt(90); colormode(255);st();

glavni = Frame(prozor); glavni.grid(column=1, row=7)

#upis

t1=Label(glavni, text="a="); t1.grid(column=1, row=1)

duljina = Entry(glavni); duljina.grid(column=2, row=1)

#opseg i površina

t2=Label(glavni); t2.grid(column=1, row=2, columnspan=2)

t3=Label(glavni); t3.grid(column=1, row=3, columnspan=2)

t4=Label(glavni); t4.grid(column=1, row=4,columnspan=2)

t5=Label(glavni); t5.grid(column=1, row=5,columnspan=2)

t6=Label(glavni); t6.grid(column=1, row=6,columnspan=2)

#gumb za crtanje

gumb=Button(glavni, text="Crtaj", command=kvadrat)

gumb.grid(column=3, row=7)

48.

from tkinter import \*

from turtle import \*

from random import \*

from math import \*

def kružnica(r):

pu();rt(90);fd(r);lt(90); pd();circle(r);pu();rt(90);bk(r);lt(90)

def romb():

e=int(d1.get())

f=int(d2.get())

a=sqrt(e\*\*2+f\*\*2)

t3.config(text="Stranica a= "+str(a))

t4.config(text="Opseg o= "+str(4\*a))

t5.config(text="Površina P= "+str(e\*f/2))

A=position(); color('red');rt(90);fd(e);C=position();bk(e/2);lt(90);

fd(f/2);D=position();bk(f);B=position()

pu();setpos(A);pd(); color("black");

setpos(B);setpos(C);setpos(D);setpos(A)

prozor = Tk()

prozor.title('Stranice')

title('Romb');lt(90)

colormode(255);st();

glavni = Frame(prozor); glavni.grid(column=1, row=6)

#upis

t1=Label(glavni, text="e="); t1.grid(column=1, row=1)

d1 = Entry(glavni); d1.grid(column=2, row=1)

t2=Label(glavni, text="f="); t2.grid(column=1, row=2)

d2 = Entry(glavni); d2.grid(column=2, row=2)

#opseg i površina

t3=Label(glavni); t3.grid(column=1, row=3, columnspan=2)

t4=Label(glavni); t4.grid(column=1, row=4,columnspan=2)

t5=Label(glavni); t5.grid(column=1, row=5,columnspan=2)

#gumb za crtanje

gumb=Button(glavni, text="Crtaj", command=romb)

gumb.grid(column=3, row=6)

49.

from turtle import \*

from random import \*

def vektor(a\_x,a\_y,b\_x,b\_y,ime1,ime2):

pu();setpos(a\_x,a\_y);pd();točka(4,ime1)

seth(towards(b\_x,b\_y));setpos(b\_x,b\_y);točka(4,ime2)

strelica()

def F (a\_x,a\_y,b\_x,b\_y,e\_x,e\_y):

pu(); setpos(a\_x,a\_y);d=distance(b\_x,b\_y);kut=towards(b\_x,b\_y)

pu(); setpos(e\_x,e\_y); seth(kut);fd(d)

return xcor(),ycor()

def točka(r,ime):

pu();rt(90);fd(r);lt(90);pd();circle(r);pu();lt(90);fd(r);rt(90);pd()

pu();kut=heading();seth(90);bk(20);write(ime);fd(20);seth(kut);pd()

def strelica():

rt(180); rt(15);fd(15);bk(15);lt(30)

fd(15);bk(15);rt(15);rt(180)

title('Vektori')

colormode(255);st();lt(90);

a\_x=textinput('Točka A','x=');a\_x=int(a\_x)

a\_y=textinput('Točka A','y=');a\_y=int(a\_y)

b\_x=textinput('Točka B','x=');b\_x=int(b\_x)

b\_y=textinput('Točka B','y=');b\_y=int(b\_y)

e\_x=textinput('Točka E','x=');e\_x=int(e\_x)

e\_y=textinput('Točka E','y=');e\_y=int(e\_y)

vektor(a\_x,a\_y,b\_x,b\_y,'A','B')

color('red')

f\_x,f\_y=F (a\_x,a\_y,b\_x,b\_y,e\_x,e\_y)

vektor(e\_x,e\_y,f\_x,f\_y,'E','F')

50.

from turtle import \*

from random import \*

from math import \*

def boja():

c=randrange(0,256); z=randrange(0,256)

p=randrange(0,256); color((0,0,0),(c,z,p))

def kvadrat(a):

boja();begin\_fill()

for k in range(4):

fd(a);rt(90)

end\_fill()

def pravokutni\_trokut (a,b):

C=position(); rt(90); fd(a); B=position()

bk(a);lt(90);fd (b);A=position();setpos(B)

c=sqrt(a\*a+b\*b); pu(); setpos(C); seth(90); pd()

lt (90); kvadrat (b)

rt (180); kvadrat (a)

pu(); setpos(A); seth (towards (B)); lt (90); pd()

kvadrat (c)

title ('Pitagorin poučak')

colormode (255);lt(90)

a=textinput('Kateta','a=');a=int(a)

b=textinput('Kateta','b=');b=int(b)

pravokutni\_trokut(a,b)

51.

from tkinter import \*

from turtle import \*

from random import \*

from math import \*

def boja():

c=randrange(0,256)

z=randrange(0,256)

p=randrange(0,256)

color(c,z,p)

def koordinatni\_sustav():

#x\_os

reset(); pu();setpos(-250,-250); pd(); bk(10);fd(605);strelica();lt(90);pu(); bk(20);write('x');

fd(20); rt(90); bk(25);pd()

for k in range(60):

lt(90);fd(5);bk(10);fd(5);rt(90);bk(10)

#y\_os

pu();setpos(-250,-250); pd();lt(90);bk(10);fd(605)

strelica();bk(10);pu();rt(90);fd(10);lt(90)

write('y');lt(90);fd(10);rt(90);bk(15);pd()

for k in range(60):

lt(90);fd(5);bk(10);fd(5);rt(90);bk(10)

def strelica():

rt(180); rt(15);fd(15);bk(15);lt(30); fd(15);bk(15);rt(15);rt(180)

def graf():

a=float(a1.get()); x1=0;y1=sqrt(a\*x1)

pu();setpos(-250+10\*x1,-250+10\*y1);pd()

for x in range(1,50):

y=sqrt(a\*x); setpos(-250+10\*x,-250+10\*y)

title('Funkcija drugog korijena')

st();lt(90);colormode(255);speed(0)

koordinatni\_sustav()

prozor = Tk()

prozor.title("Parametri funkcije")

glavni = Frame(prozor)

glavni.grid(column=2, row=3)

#upis

t1=Label(glavni, text="a=")

t1.grid(column=1, row=1)

a1 = Entry(glavni)

a1.grid(column=2, row=1)

#gumb za crtanje

Button(glavni, text="Crtaj", command=graf).grid(column=1, row=3)

Button(glavni, text="Boja", command=boja).grid(column=2, row=3)

52**.**

from tkinter import \*

from turtle import \*

from random import \*

from math import \*

def boja():

c=randrange(0,256)

z=randrange(0,256)

p=randrange(0,256)

color(c,z,p)

def koordinatni\_sustav():

#x\_os

reset(); bk(300);fd(595);strelica();lt(90);pu(); bk(20);write('x');

fd(20); rt(90); bk(25);pd()

for k in range(55):

lt(90);fd(5);bk(10);fd(5);rt(90);bk(10)

setpos(0,0);seth(90)

#y\_os

bk(300);fd(595); strelica();bk(10);pu();rt(90);fd(10);lt(90)

write('y');lt(90);fd(10);rt(90);bk(15);pd()

for k in range(55):

lt(90);fd(5);bk(10);fd(5);rt(90);bk(10)

def strelica():

rt(180); rt(15);fd(15);bk(15);lt(30)

fd(15);bk(15);rt(15);rt(180)

def graf():

width(1)

a=float(a1.get())

b=float(b1.get())

x1=-27;y1=a\*x1+b

pu();setpos(10\*x1,10\*y1);pd()

for x in range(-27,28):

y=a\*x+b

setpos(10\*x,10\*y)

title('Linearna funkcija'); st();lt(90);colormode(255);speed(0);width(1)

koordinatni\_sustav()

prozor = Tk()

prozor.title("Parametri funkcije")

glavni = Frame(prozor)

glavni.grid(column=2, row=3)

#upis

t1=Label(glavni, text="a=")

t1.grid(column=1, row=1)

a1 = Entry(glavni)

a1.grid(column=2, row=1)

t2=Label(glavni, text="b=")

t2.grid(column=1, row=2)

b1 = Entry(glavni)

b1.grid(column=2, row=2)

#gumb za crtanje

Button(glavni, text="Crtaj", command=graf).grid(column=1, row=3)

Button(glavni, text="Boja", command=boja).grid(column=2, row=3)

### 6. Primjena programiranja u fizici i kemiji

53.

.

from tkinter import \*

from turtle import \*

from random import \*

def graf():

v=float(v1.get());t=float(t1.get())

s1=v\*t

x=-250+t; y=-250+v

width(1)

begin\_fill()

pu();setpos(-250,-250);pd()

setpos(x,-250);setpos(x,y);setpos(-250,y)

end\_fill()

s.config (text='Prijeđeni put: '+str(s1)+' m')

def boja():

c=randrange(0,256)

z=randrange(0,256)

p=randrange(0,256)

color(c,z,p);width(3)

def koordinatne\_osi():

#v\_os

reset();pu();setpos(-250,-250);pd();lt(90);fd(570);strelica();lt(90);

pu(); fd(20);write('v(m/s)',font=('Arial',11,'bold')); bk(20); rt(90);

bk(25);pd()

for k in range(550,0,-10):

if k%50==0:

width(3);

pu(); lt(90);fd(30); rt(90); bk(8)

write(k,font=('Arial',11,'bold'))

fd(8);rt(90);fd(30);lt(90);pd()

else:

width(1)

lt(90);fd(5);bk(10);fd(5);rt(90);pu();bk(10);pd()

setpos(-250,-250);seth(0)

#t\_os

bk(20);fd(530); strelica();bk(10);pu();lt(90);bk(20)

write('t(s)',font=('Arial',11,'bold'));fd(20);rt(90);bk(15);pd()

seth(90);setpos(220,-250)

for k in range(470,0,-10):

if k%50==0:

width(3)

pu();bk(25);lt(90);fd(8);rt(90);write(k,font=('Arial',11,'bold'))

rt(90);fd(8);lt(90);fd(25)

else:

width(1)

pd();seth(90);fd(5);bk(10);pu();fd(5);rt(90); bk(10);lt(90)

def strelica():

rt(180); rt(15);fd(15);bk(15);lt(30)

fd(15);bk(15);rt(15);rt(180)

*(nastavak 53. zadatka)*

prozor = Tk()

prozor.title('Kontrole')

title('Jednoliko pravocrtno gibanje')

colormode(255);st();lt(90)

glavni = Frame(prozor)

glavni.grid(column=3, row=4)

#upis brzine

k1=Label(glavni, text="Brzina")

k1.grid(column=1, row=1)

v1 = Entry(glavni)

v1.grid(column=2, row=1)

#upis vremena

k2=Label(glavni, text="Vrijeme")

k2.grid(column=1, row=2)

t1 = Entry(glavni)

t1.grid(column=2, row=2)

#ispis prijeđenog puta

s = Label(glavni)

s.grid(column=2, row=3)

#gumb za crtanje grafa

gumb1=Button(glavni, text="Crtaj", command=graf)

gumb1.grid(column=1, row=4)

gumb2=Button(glavni, text="Boja", command=boja)

gumb2.grid(column=2, row=4)

koordinatne\_osi()

54.

from tkinter import \*

from turtle import \*

from random import \*

def odredi\_otpor():

R=0

for i in range(1,9):

r=int(o[i].get())

if r!=0:

R+=1/r

R=round(1/R,4)

otpor.config(text="Ukupni otpor: R= "+str(R)+' '+u'\N{GREEK CAPITAL LETTER OMEGA}')

strujni\_krug()

def strujni\_krug():

pu();setpos(-250,50);pd();fd(50);lt(90)

for i in range(1, 9):

r=int(o[i].get())

if r!=0:

otpornik(r)

rt(180);fd(130);setpos(-220,-80)

begin\_fill()

fd(10);rt(90);fd(8);rt(90);fd(20);rt(90);fd(8)

rt(90);fd(10);end\_fill()

pu();rt(90); fd(12);rt(90); pd();fd(20);bk(40);fd(20);lt(90)

setpos(-250,-80);setpos(-250,50)

pu();lt(90);fd(25);lt(90);fd(20); pd(); fd(40); pu()

bk(130); lt(90);pd(); fd(70);lt(90);pd()

def otpornik(ot):

bk(40);rt(90);fd(15);rt(90)

for k in range(2):

fd(50);rt(90);fd(30);rt(90)

pu();fd(30);lt(90);fd(10);lt(90);pd()

ispis=str(ot)+' '+u'\N{GREEK CAPITAL LETTER OMEGA}'

color('blue');write(ispis,font=('Arial',12,'bold'));color('black')

pu();lt(90);fd(25);lt(90);fd(20); pd(); fd(40); pu(); bk(130); lt(90);pd(); fd(70);lt(90);pd()

prozor = Tk()

prozor.title('Otpor')

title('Paralelni spoj otpornika');st();

glavni = Frame(prozor)

glavni.grid(column=3, row=10)

*(nastavak 54. zadatka)*

prozor = Tk()

prozor.title('Otpor')

title('Paralelni spoj otpornika');st();

glavni = Frame(prozor)

glavni.grid(column=3, row=10)

# upis otpora

t = []

o = []

t.append(glavni)

o.append(glavni)

for x in range(1, 9):

t.append(Label(glavni, text='Otpor '+str(x)))

t[x].grid(row = x, column = 1)

o.append(Entry(glavni))

o[x].grid(row = x, column = 2)

# gumb za crtanje sheme

gumb=Button(glavni, text="Crtaj", command=odredi\_otpor)

gumb.grid(column=1, row=11)

# izračunavanje ukupnog otpora

otpor=Label(glavni)

otpor.grid(column=2, row=10)

55.

from tkinter import \*

from turtle import \*

def atom(simbol):

if simbol=='C':

color('black','blue')

else:

color('black','yellow')

begin\_fill()

pu();rt(90);fd(25);lt(90); pd();circle(25);pu();rt(90);bk(25);lt(90)

end\_fill()

def CH2():

atom('C');pu();fd(25);pd();fd(30); pu();fd(25);pd()

atom('H');pu();bk(160);pd();atom('H'); pu();fd(25);pd();fd(30);

pu();fd(25);pd()

def alken (n):

CH2();

pu();rt(90);fd(22); lt(90); fd (8); pd()

rt(90);fd(30);bk(30);lt(90);pu();bk(12);pd();

rt(90);fd(30);pu();fd(25);lt(90);fd(6);pd()

atom('C'); pu();fd(25);pd();fd(30); pu();fd(25);pd()

atom('H');pu();bk(80);pd()

rt(90);pu();fd(25);pd();fd(30);pu();fd(25);lt(90)

for k in range(n-2):

CH2();

rt(90);pu();fd(25);pd();fd(30);pu();fd(25);lt(90)

atom('H')

o\_spoju(n)

def o\_spoju(n):

nazivi=['Eten','Propen','Buten','Penten','Heksen','Hepten','Okten']

ime=nazivi[n-2]

broj\_vodika=2\*n

broj\_ugljika=n

masa=12\*broj\_ugljika+1.008\*broj\_vodika

maseni\_c=12\*broj\_ugljika/masa

maseni\_h=1.008\*broj\_vodika/masa

postotak\_vodika=int(maseni\_h\*10000)/100

postotak\_ugljika=int(maseni\_c\*10000)/100

pu();setpos(-250,150);color('red')

masa='Relativna molekulska masa: '+str(masa)

postotak\_ugljika='Maseni udio ugljika je '+str(postotak\_ugljika)+'%.'

postotak\_vodika='Maseni udio vodika je '+str(postotak\_vodika)+'%.'

write(ime,font=('Arial',16,'bold'));pu();bk(30)

color('Sienna')

write(masa,font=('Arial',16,'bold')); pu();bk(30)

write(postotak\_ugljika,font=('Arial',16,'bold')); pu();bk(30)

write(postotak\_vodika,font=('Arial',16,'bold')); pu();bk(30)

title('Alkeni')

st();pu();bk(300);lt(90);bk(100);pd()

n=textinput('Broj atoma ugljika','n=');n=int(n)

alken(n)

56.

from tkinter import \*

from turtle import \*

def atom(simbol):

if simbol=='C':

color('black','blue')

else:

color('black','yellow')

begin\_fill()

pu();rt(90);fd(25);lt(90); pd();circle(25);pu();rt(90);bk(25);lt(90)

end\_fill()

def CH2():

atom('C');pu();fd(25);pd();fd(30); pu();fd(25);pd()

atom('H');pu();bk(160);pd();atom('H'); pu();fd(25);pd();fd(30);pu();fd(25);pd()

def alkin (n):

atom('C'); pu();fd(25);pd();fd(30); pu();fd(25);pd()

atom('H');pu();bk(80);pd()

pu();rt(90);fd(22); lt(90); fd (10); pd()

rt(90);fd(30);bk(30);lt(90);pu();bk(10);pd();

rt(90);fd(30);bk(30);lt(90);pu();bk(10);pd();

rt(90);fd(30);bk(30);lt(90);pu()

fd(10);rt(90);pu();fd(50);lt(90)

atom('C');rt(90); pu();fd(25);pd();fd(30); pu();fd(25);pd();lt(90)

for k in range(n-2):

CH2();

rt(90);pu();fd(25);pd();fd(30);pu();fd(25);lt(90)

atom('H')

o\_spoju(n)

def o\_spoju(n):

nazivi=['Etin','Propin','Butin','Pentin','Heksin','Heptin','Oktin']

ime=nazivi[n-2]

broj\_vodika=2\*n-2

broj\_ugljika=n

masa=12\*broj\_ugljika+1.008\*broj\_vodika

maseni\_c=12\*broj\_ugljika/masa

maseni\_h=1.008\*broj\_vodika/masa

postotak\_vodika=int(maseni\_h\*10000)/100

postotak\_ugljika=int(maseni\_c\*10000)/100

pu();setpos(-250,150);color('red')

masa='Relativna molekulska masa: '+str(masa)

postotak\_ugljika='Maseni udio ugljika je '+str(postotak\_ugljika)+'%.'

postotak\_vodika='Maseni udio vodika je '+str(postotak\_vodika)+'%.'

write(ime,font=('Arial',16,'bold'));pu();bk(30)

color('Sienna')

write(masa,font=('Arial',16,'bold')); pu();bk(30)

write(postotak\_ugljika,font=('Arial',16,'bold')); pu();bk(30)

write(postotak\_vodika,font=('Arial',16,'bold')); pu();bk(30)

title('Alkini')

st();pu();bk(300);lt(90);bk(100);pd()

n=textinput('Broj atoma ugljika','n=');n=int(n)

alkin(n)

57.

from tkinter import \*

from turtle import \*

def atom(simbol):

if simbol=='C':

color('black','blue')

elif simbol=='O':

color('black','red')

else:

color('black','yellow')

begin\_fill()

pu();rt(90);fd(25);lt(90); pd();circle(25);pu();rt(90);bk(25);lt(90)

end\_fill()

def CH2():

atom('C');pu();fd(25);pd();fd(30); pu();fd(25);pd()

atom('H');pu();bk(160);pd();atom('H'); pu();fd(25);pd();fd(30);pu();fd(25)

def alkohol (n):

atom('H');

rt(90);pu();fd(25);pd();fd(30);pu();fd(25);lt(90)

for k in range(n):

CH2();rt(90);pu();fd(25);pd();fd(30);pu();fd(25);lt(90)

atom('O');pu();rt(45);fd(25);pd();fd(30);pu();fd(25);pd();atom('H')

o\_spoju(n)

def o\_spoju(n):

nazivi=['Metanol', 'Etanol','Propanol','Butanol',

'Pentanol','Heksanol','Heptanol','Oktanol']

ime=nazivi[n-1]

broj\_vodika=2\*n+2

broj\_ugljika=n

broj\_kisika=1

masa=12\*broj\_ugljika+1.008\*broj\_vodika+16\*broj\_kisika

maseni\_c=12\*broj\_ugljika/masa

maseni\_h=1.008\*broj\_vodika/masa

maseni\_o=16\*broj\_kisika/masa

postotak\_vodika=int(maseni\_h\*10000)/100

postotak\_ugljika=int(maseni\_c\*10000)/100

postotak\_kisika=int(maseni\_o\*10000)/100

pu();setpos(-250,150);color('red')

masa='Relativna molekulska masa: '+str(masa)

postotak\_ugljika='Maseni udio ugljika je '+str(postotak\_ugljika)+'%.'

postotak\_vodika='Maseni udio vodika je '+str(postotak\_vodika)+'%.'

postotak\_kisika='Maseni udio kisika je '+str(postotak\_kisika)+'%.'

write(ime,font=('Arial',16,'bold'));pu();setpos(-250,120)

color('Sienna')

write(masa,font=('Arial',16,'bold')); pu();setpos(-250,90)

write(postotak\_ugljika,font=('Arial',16,'bold')); pu();setpos(-250,60)

write(postotak\_vodika,font=('Arial',16,'bold')); pu();setpos(-250,30)

write(postotak\_kisika,font=('Arial',16,'bold')); pu();setpos(-250,0)

title('Alkoholi')

st();pu();bk(300);lt(90);bk(100);pd()

n=textinput('Broj atoma ugljika','n=');n=int(n)

alkohol(n)

### 7. Ponovimo što smo naučili

n=input('n=')

n=int(n)

k=input('k=')

k=int(k)

for m in range(100,1000):

if m%n==0 or m%k==0:

print(m,end=' ')

58.

59.

|  |  |
| --- | --- |
| Ulaz | Izlaz |
| n=2 | 1 1 |
| n=5 | 1 1 2 3 5 |
| n=10 | 1 1 2 3 5 8 13 21 34 55 |

Zbroj svakog člana niza od trećeg člana jednak je zbroju prethodna dva člana niza, a prva dva člana su 1.

Kakve je vrste potprogram fibonacci(n)? Rekurzivni potprogram.

Po kojim naredbama unutar potprograma to zaključuješ? Ako je n<3 program vraća 1 (return 1), a inače vraća zbroj koji daju pozivi potprograma fibonacci(n-2) i fibonacci(n-1).

60.

from turtle import \*

def stranica(n,a):

if n==0:

fd(a)

return True

stranica(n-1,a/3);lt(60)

stranica(n-1,a/3);rt(60)

stranica(n-1,a/3)

title ('Šesterokut')

colormode (255);lt(90)

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

for k in range(6):

stranica(n,a);rt(60)

61.

from turtle import \*

from random import \*

def boja():

c=randrange(0,256)

z=randrange(0,256)

p=randrange(0,256)

color((0,0,0),(c,z,p))

def stranica(n,a):

if n==0:

fd(a)

return True

stranica(n-1,a/3);lt(60)

stranica(n-1,a/3);rt(60)

stranica(n-1,a/3)

title ('Šesterokuti')

colormode (255);lt(90)

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

for m in range(6):

boja()

begin\_fill()

for k in range(6):

stranica(n,a);rt(60)

end\_fill()

stranica(n,a);lt(60)

stranica(n,a);rt(180);

boja()

begin\_fill()

for k in range(6):

stranica(n,a);rt(60)

end\_fill()

62.

from turtle import \*

from tkinter import \*

from random import \*

def boja():

c=randrange(0,256)

z=randrange(0,256)

p=randrange(0,256)

color((0,0,0),(c,z,p))

def šesterokut():

a=int(duljina.get())

boja();begin\_fill()

for k in range(6):

fd(a);rt(60)

end\_fill()

def desno():

a=int(duljina.get())

pu();rt(120);fd(a);lt(60); fd(a);lt(60);pd()

def lijevo():

a=int(duljina.get())

pu();lt(120);fd(a);rt(60); fd(a);rt(60);pd()

def gore():

a=int(duljina.get())

pu();fd(a);rt(60);fd(a);lt(60);pd()

def brisi():

reset();lt(90)

prozor = Tk()

prozor.title('Gumbi')

title('Piramida šesterokuta')

colormode(255);lt(90);

glavni=Frame(prozor)

glavni.grid(columns=1,rows=6)

t1=Label(glavni,text="a=")

t1.grid(column=1, row=6)

duljina=Entry(glavni)

duljina.grid(column=2, row=6)

gumb1=Button(glavni, text="Šesterokut",command=šesterokut)

gumb1.grid(column=1, row=1)

gumb2=Button(glavni, text="Desno",command=desno)

gumb2.grid(column=1, row=2)

gumb3=Button(glavni, text="Lijevo",command=lijevo)

gumb3.grid(column=1, row=3)

gumb4=Button(glavni, text="Gore",command=gore)

gumb4.grid(column=1, row=4)

gumb5=Button(glavni, text="Briši",command=brisi)

gumb5.grid(column=1, row=5)

### 8. Dodatni zadaci

from turtle import \*

def stranica(n,a):

if n==0:

fd(a)

return True

stranica(n-1,a/2);rt(90)

stranica(n-1,a/3);rt(180)

stranica(n-1,a/3);rt(90)

stranica(n-1,a/2)

title ('Pahuljica')

colormode (255);lt(90)

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

for m in range(4):

stranica(n,a);lt(90)

63.

from turtle import \*

def stranica(n,a):

if n==0:

fd(a)

return True

stranica(n-1,a/2);rt(90)

stranica(n-1,a/3);rt(180)

stranica(n-1,a/3);rt(90)

stranica(n-1,a/2)

title ('Pahuljica')

colormode (255);lt(90)

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

for m in range(4):

stranica(n,a);rt(90)

64.

65.

from turtle import \*

def stranica(n,a):

if n==0:

fd(a)

return True

stranica(n-1,a/2);lt(120)

stranica(n-1,a/5);rt(180)

stranica(n-1,a/5);lt(120)

stranica(n-1,a/5);rt(180)

stranica(n-1,a/5);lt(120)

stranica(n-1,a/2)

title ('Pahuljica')

colormode (255);

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

for m in range(3):

stranica(n,a);lt(120)

66.

from turtle import \*

def stranica(n,a):

if n==0:

fd(a)

return True

stranica(n-1,a/2);lt(120)

stranica(n-1,a/5);rt(180)

stranica(n-1,a/5);lt(120)

stranica(n-1,a/5);rt(180)

stranica(n-1,a/5);lt(120)

stranica(n-1,a/2)

title ('Pahuljica')

colormode (255);

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

lt(60)

for m in range(3):

stranica(n,a);rt(120)

67.

from turtle import \*

def kvadrat(a):

color("black");begin\_fill()

for k in range(4):

fd(a);rt(90)

end\_fill()

def kvadrati(n,a):

if n==0:

kvadrat(a)

return True

for k in range(4):

for m in range(2):

kvadrati (n-1,a/3);fd(a/3)

fd(a/3);rt(90)

title ('Kvadrat')

colormode (255);

a=textinput('Veličina','a=');a=int(a)

n=textinput('Dubina rekurzije','n=');n=int(n)

kvadrati(n,a)

68.

from turtle import \*

def kružnica(x,y):

pu();setpos (x,y);rt(90);fd(20);lt(90); pd();circle(20)

title ('Muhara')

# stapka

pu(); setpos (-320,0); pd(); setpos (-280,-40);setpos (-240,-60)

setpos (-200,-80); setpos (-160,-80); setpos (-140,-80)

setpos (-140,0);setpos (-60,20); setpos (0,20); setpos (20,-40)

setpos (20,-80); setpos (20,-180); setpos (0,-220);setpos (-20,-260)

setpos (-40,-300); setpos (-80,-320); setpos (-140,-320)

setpos (-200,-300);setpos (-220,-280);setpos (-200,-240)

setpos (-180,-200);setpos (-160,-160);setpos (-140,-140)

setpos (-140,-80)

# klobuk

pu(); setpos (-320,0); pd(); setpos (-120,80);setpos (20,80)

setpos (180,0); setpos (100,-60);setpos (20,-80)

pu(); setpos (-320,0); pd();

setpos (-300,60);setpos (-260,140); setpos (-200,220)

setpos (-160,260);setpos (-100,280);setpos (-40,280); setpos (20,240)

setpos (120,160);setpos (180,0)

# pjege

kružnica (-140,200); kružnica (-220,100); kružnica (-20,220)

kružnica (50,150); kružnica (100,80); kružnica (-100,115)

69.

c=input('Koliko gori crveno (u sekundama)?'); c=int(c)

ž=input('Koliko gori žuto (u sekundama)?'); ž=int(ž)

z=input('Koliko gori zeleno (u sekundama)?'); z=int(z)

minute=input('Broj minuta u kojima se traži broj izmjene svjetala:'); minute=int(minute)

h=input('U koliko sati Ivica kreće?'); h=int(h)

m=input('U koliko minuta Ivica kreće?'); m=int(m)

x=input('U koliko minuta Ivica ima do semafora?'); x=int(x)

n=int (minute\*60/(c+ž+z))

br\_sekundi=h\*3600+(m+x)\*60

ostatak = br\_sekundi %(c+ž+z)

print ('U', minute,'minuta svjetla na semaforu izmijene se', n, 'puta.')

if ostatak >= c+ž:

print('Ivica neće čekati na semaforu.')

else:

ceka= c+ž-ostatak

print('Ivica će čekati na semaforu', ceka, 'sekundi.')

## 4B. PROGRAMSKI JEZIK SMALL BASIC

### 1. Ponovimo

1.

a) Ispis programa:

**a=-7**

**While x<= -1**

**TextWindow.WriteLine (2\*x)**

**x=x+1**

**EndWhile**

-14

-12

-10

-8

-6

-4

-2

b) Ispis programa:

**For a= 100 To 5 Step -5**

**TextWindow.WriteLine (a+ "--->"+ a/5)**

**EndFor**

100 ---> 20

95 ---> 19

90 ---> 18

…

5 --->5

c) Ispis programa:

**m=2**

**While m<= 20**

**TextWindow.WriteLine (m)**

**m=m+4**

**EndWhile**

2

6

10

14

18

d) Ispis programa:

**For b= 0 To 100 Step 10**

**TextWindow.WriteLine (b)**

**EndFor**

0

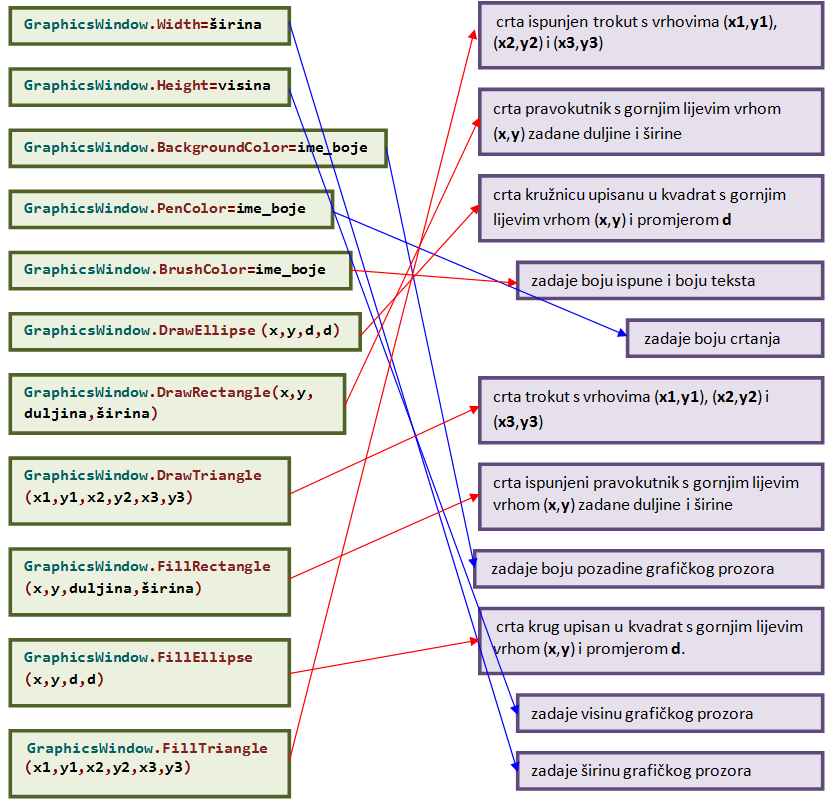
10

20

…

100

2.



3.

GraphicsWindow.Title="TABLICA KVADRATA I KORIJENA"

GraphicsWindow.Height=500

GraphicsWindow.Width=300

GraphicsWindow.DrawLine (30,0,30,500)

GraphicsWindow.DrawLine (120,0,120,500)

GraphicsWindow.DrawLine (0,30,300,30)

GraphicsWindow.DrawBoundText (10,10,20,"x")

GraphicsWindow.DrawBoundText (50,10,50, "kvadrat")

GraphicsWindow.DrawBoundText (150,10,100, "korijen")

**For** k= 1 **To** 30

GraphicsWindow.DrawBoundText (10,15+k\*15,20,k)

GraphicsWindow.DrawBoundText (70,15+k\*15,50,Math.Power (k,2) )

GraphicsWindow.DrawBoundText (150,15+k\*15,200,Math.SquareRoot(k))

**EndFor**

4.

GraphicsWindow.Title="TABLICA KVADRATA I KORIJENA"

GraphicsWindow.Height=500

GraphicsWindow.Width=300

GraphicsWindow.DrawLine (30,0,30,500)

GraphicsWindow.DrawLine (120,0,120,500)

GraphicsWindow.DrawLine (0,30,300,30)

GraphicsWindow.DrawBoundText (10,10,20,"x")

GraphicsWindow.DrawBoundText (50,10,50, "kvadrat")

GraphicsWindow.DrawBoundText (150,10,100, "korijen")

TextWindow.Write ("m=")

m=TextWindow.ReadNumber()

TextWindow.Write ("n=")

n=TextWindow.ReadNumber()

**If** m>n **Then**

t=m

m=n

n=t

**EndIf**

**For** k= m **To** n

GraphicsWindow.DrawBoundText (10,15+(k-m+1)\*15,20,k)

GraphicsWindow.DrawBoundText (70,15+(k-m+1)\*15,50,Math.Power (k,2) )

GraphicsWindow.DrawBoundText (150,15+(k-m+1)\*15,200,Math.SquareRoot(k))

**EndFor**

5.

GraphicsWindow.Title="Pravokutnik"

GraphicsWindow.Height=500

GraphicsWindow.Width=300

TextWindow.Write ("a=")

a=TextWindow.ReadNumber()

TextWindow.Write ("b=")

b=TextWindow.ReadNumber()

o=2\*(a+b)

p=a\*b

d=Math.SquareRoot (Math.Power (a,2)+Math.Power (b,2))

GraphicsWindow.DrawRectangle(10,10,a,b)

GraphicsWindow.DrawLine (10,10,10+a,10+b)

GraphicsWindow.DrawLine (10,10+b,10+a,10)

GraphicsWindow.DrawBoundText (10,b+15,200,"opseg="+o)

GraphicsWindow.DrawBoundText (10,b+30,500,"površina="+p )

GraphicsWindow.DrawBoundText (10,b+45,500,"dijagonala="+d)

6. a)

GraphicsWindow.Show()

GraphicsWindow.DrawRectangle(100,100,50,50)

GraphicsWindow.DrawRectangle(150,150,50,50)

b)

GraphicsWindow.Show()

GraphicsWindow.DrawRectangle(0,100,50,50)

GraphicsWindow.DrawRectangle(50,50,50,50)

c)

GraphicsWindow.Show()

GraphicsWindow.DrawEllipse(50,70,90,90)

GraphicsWindow.DrawEllipse(140,70,90,90)

GraphicsWindow.DrawEllipse(230,70,90,90)

d)

GraphicsWindow.Show()

**For** k= 50 **To** 150 **Step** 50

GraphicsWindow.DrawTriangle(k,50,k+25,5,k+50,50)

GraphicsWindow.DrawTriangle(k,50,k+25,95,k+50,50)

**EndFor**

e)

GraphicsWindow.Show()

GraphicsWindow.DrawLine(50,50,75,5)

GraphicsWindow.DrawLine(100,50,75,5)

GraphicsWindow.DrawLine(100,50,75,95)

GraphicsWindow.DrawLine(50,50,75,95)

GraphicsWindow.DrawLine(60,50,75,25)

GraphicsWindow.DrawLine(90,50,75,25)

GraphicsWindow.DrawLine(90,50,75,75)

GraphicsWindow.DrawLine(60,50,75,75)

7.

a)

GraphicsWindow.BrushColor="red"

GraphicsWindow.FillEllipse(50,50,160,160)

GraphicsWindow.BrushColor="white"

GraphicsWindow.FillRectangle(80,115,100,30)

GraphicsWindow.BrushColor="blue"

GraphicsWindow.FillRectangle (50,50,160,170)

graphicsWindow.BrushColor="white"

GraphicsWindow.FillRectangle (80,80,20,110)

GraphicsWindow.FillRectangle (100,125,60,20)

GraphicsWindow.FillRectangle (160,80,20,110)

b)

c)

GraphicsWindow.BrushColor="blue"

GraphicsWindow.FillRectangle(0,0, 160,170)

GraphicsWindow.BrushColor="white"

GraphicsWindow.FillTriangle (80,20,40,70,120,70)

GraphicsWindow.FillRectangle(70,70,20,80)

d)

GraphicsWindow.DrawLine (50,100, 150,0)

GraphicsWindow.DrawLine (150,0, 250,100)

GraphicsWindow.DrawLine (250,100, 150,200)

GraphicsWindow.DrawLine (150,200, 50,100)

GraphicsWindow.BrushColor="yellow"

GraphicsWindow.FillTriangle (80,100, 150,30, 150, 170)

GraphicsWindow.FillTriangle (220,100, 150,30, 150, 170)

e)

GraphicsWindow.BrushColor="red"

GraphicsWindow.FillEllipse (50,50,200,200)

GraphicsWindow.BrushColor="blue"

GraphicsWindow.FillEllipse (80,80,140,140)

GraphicsWindow.BrushColor="red"

GraphicsWindow.FillTriangle (80,90,100,70,220,200)

GraphicsWindow.FillTriangle (200,220,80,90,220,200)

f)

GraphicsWindow.BrushColor="red"

GraphicsWindow.FillEllipse (50,50,200,200)

GraphicsWindow.BrushColor="blue"

GraphicsWindow.FillEllipse (80,80,140,140)

GraphicsWindow.BrushColor="red"

GraphicsWindow.FillTriangle (80,90,100,70,220,200)

GraphicsWindow.FillTriangle (200,220,80,90,220,200)

GraphicsWindow.FillTriangle (80,200,200,70,100,220)

GraphicsWindow.FillTriangle (220,90,200,70,100,220)

g)

GraphicsWindow.BrushColor="red"

GraphicsWindow.FillTriangle (160,30,10,290,310,290)

GraphicsWindow.BrushColor="white"

GraphicsWindow.FillTriangle (160,70,45,265,265,265)

GraphicsWindow.BrushColor="red"

GraphicsWindow.FillEllipse (140,100,30,30)

GraphicsWindow.BrushColor="Yellow"

GraphicsWindow.FillEllipse (140,160,30,30)

GraphicsWindow.BrushColor="green"

GraphicsWindow.FillEllipse (140,220,30,30)

h)

GraphicsWindow.BrushColor="red"

GraphicsWindow.FillTriangle (150,30,10,290,310,290)

GraphicsWindow.BrushColor="white"

GraphicsWindow.FillTriangle (150,70,45,265,265,265)

GraphicsWindow.BrushColor="black"

GraphicsWindow.FillRectangle (95,185,110,10)

GraphicsWindow.FillRectangle (95,225,110,10)

**For** x= 105 **To** 200 **Step** 20

GraphicsWindow.FillTriangle(x,175,x+10,175,x+5,155)

GraphicsWindow.FillRectangle(x,175,10,80)

**EndFor**

8.

GraphicsWindow.Title="Snjegović"

GraphicsWindow.Width=320

GraphicsWindow.BackgroundColor="LightSkyBlue"

GraphicsWindow.BrushColor="white"

GraphicsWindow.FillEllipse (80,280,160,160)

GraphicsWindow.FillEllipse (100,190,120,120)

GraphicsWindow.FillEllipse (120,130,80,80)

GraphicsWindow.BrushColor="sienna"

*'desna ruka*

GraphicsWindow.FillRectangle (200,225,60,10)

GraphicsWindow.FillTriangle (200,235,260,235,200,245)

GraphicsWindow.FillTriangle (260,235,260,225,300,200)

GraphicsWindow.FillTriangle (225,225,235,225,230,200)

*'lijeva ruka*

GraphicsWindow.FillRectangle (45,235,60,10)

GraphicsWindow.FillTriangle (45,235,105,235,105,225)

GraphicsWindow.FillTriangle (45,235,45,245,5,210)

GraphicsWindow.FillTriangle (70,245,80,245,75,275)

*'oči*

GraphicsWindow.BrushColor="black"

GraphicsWindow.FillEllipse (130,150,20,20)

GraphicsWindow.FillEllipse (170,150,20,20)

*'gumbi*

GraphicsWindow.FillEllipse (155,220,15,15)

GraphicsWindow.FillEllipse (155,260,15,15)

GraphicsWindow.FillEllipse (155,300,15,15)

GraphicsWindow.FillEllipse (155,340,15,15)

*'šešir*

GraphicsWindow.FillRectangle (130,90,60,40)

GraphicsWindow.PenWidth=7

GraphicsWindow.DrawLine (95,110,110,127)

GraphicsWindow.DrawLine (110,127,130,127)

GraphicsWindow.DrawLine (210,127, 225,110)

GraphicsWindow.DrawLine (190,127,210,127)

*'nos*

GraphicsWindow.BrushColor="OrangeRed"

GraphicsWindow.FillTriangle (160,165, 160,185, 200,185)

*'metla*

GraphicsWindow.BrushColor="Chocolate"

GraphicsWindow.FillRectangle(260,140,12,300)

GraphicsWindow.FillRectangle(226,120,80,20)

**For** x= 235 **To** 300 **Step** 12

GraphicsWindow.DrawLine (x,100,x,120)

**EndFor**

9.

GraphicsWindow.Title="Kvadrat"

GraphicsWindow.Height=500

GraphicsWindow.Width=300

TextWindow.Write ("a=")

a=TextWindow.ReadNumber()

GraphicsWindow.Height=a\*2

GraphicsWindow.Width=a\*2

ru=a/2

ro=a/2\*Math.SquareRoot(2)

GraphicsWindow.DrawRectangle(a/2,a/2,a,a)

GraphicsWindow.DrawEllipse (a/2,a/2,ru\*2,ru\*2)

GraphicsWindow.DrawEllipse (a-ro,a-ro,ro\*2,ro\*2)

10.

GraphicsWindow.Show()

GraphicsWindow.Width=260

GraphicsWindow.Height= 600

GraphicsWindow.BrushColor="Sienna"

GraphicsWindow.FillRectangle (50,30,160,380)

GraphicsWindow.BrushColor="Gray"

GraphicsWindow.FillRectangle (110,410,40,160)

GraphicsWindow.PenColor="black"

GraphicsWindow.BrushColor="red"

GraphicsWindow.FillEllipse (90,65,80,80)

GraphicsWindow.DrawEllipse (90,65,80,80)

GraphicsWindow.BrushColor="yellow"

GraphicsWindow.FillEllipse (90,180 ,80,80)

GraphicsWindow.DrawEllipse (90,180,80,80)

GraphicsWindow.BrushColor="green"

GraphicsWindow.FillEllipse (90,295 ,80,80)

GraphicsWindow.DrawEllipse (90,295,80,80)

11.

GraphicsWindow.Show()

GraphicsWindow.Title="Semafor"

GraphicsWindow.Width=260

GraphicsWindow.Height= 600

GraphicsWindow.BrushColor="Sienna"

GraphicsWindow.FillRectangle (50,30,160,380)

GraphicsWindow.BrushColor="Gray"

GraphicsWindow.FillRectangle (110,410,40,160)

GraphicsWindow.PenColor="black"

x=1

**While** x>0

*'crveno*

GraphicsWindow.BrushColor="red"

GraphicsWindow.FillEllipse (90,65,80,80)

GraphicsWindow.DrawEllipse (90,65,80,80)

GraphicsWindow.BrushColor="black"

GraphicsWindow.FillEllipse (90,180 ,80,80)

GraphicsWindow.FillEllipse (90,295 ,80,80)

Program.Delay (1000)

*' žuto*

GraphicsWindow.BrushColor="black"

GraphicsWindow.FillEllipse (90,65,80,80)

GraphicsWindow.BrushColor="yellow"

GraphicsWindow.FillEllipse (90,180 ,80,80)

GraphicsWindow.DrawEllipse (90,180,80,80)

GraphicsWindow.BrushColor="black"

GraphicsWindow.FillEllipse (90,295 ,80,80)

GraphicsWindow.DrawEllipse (90,295,80,80)

Program.Delay (500)

*'zeleno*

GraphicsWindow.BrushColor="black"

GraphicsWindow.FillEllipse (90,65,80,80)

GraphicsWindow.FillEllipse (90,180 ,80,80)

GraphicsWindow.BrushColor="green"

GraphicsWindow.FillEllipse (90,295 ,80,80)

GraphicsWindow.DrawEllipse (90,295,80,80)

Program.Delay (1000)

**EndWhile**

### 2. Logičke funkcije u programiranju

12.

|  |  |  |
| --- | --- | --- |
| A | B | A **OR** B |
| *laž* | *laž* | *laž* |
| *laž* | *istina* | *istina* |
| *istina* | *laž* | *istina* |
| *istina* | *istina* | *istina* |

|  |  |  |
| --- | --- | --- |
| A | B | A **AND** B |
| *laž* | *laž* | *laž* |
| *laž* | *istina* | *laž* |
| *istina* | *laž* | *laž* |
| *istina* | *istina* | *istina* |

13.

|  |  |  |
| --- | --- | --- |
| A | B | A **OR** B |
| *3<5* | *5=7* | *istina* |
| *3<7-4* | *7<7* | *laž* |
| *3<=7-4* | *11<8+3* | *istina* |
| *15>7+8* | *11<=8+3* | *istina* |
| *15>=7+8* | *15<>20-5* | *istina* |
| *14< 6+9* | *15=20-5* | *istina* |

|  |  |  |
| --- | --- | --- |
| A | B | A **AND** B |
| *3<5* | *5=7* | *laž* |
| *3<7-4* | *7<7* | *laž* |
| *3<=7-4* | *11<8+3* | *laž* |
| *15>7+8* | *11<=8+3* | *laž* |
| *15>=7+8* | *15<>20-5* | *laž* |
| *14< 6+9* | *15=20-5* | *istina* |

14.

a) (4+5=9 **AND** 8+3<11) **OR** (5+9=15 **AND** 7-7=0) = (istina **AND** laž) **OR** (laž **AND** istina)= laž **OR** laž = laž

b) (4+5=9 **OR** 8+3<11) **AND** (5+9=15 **OR** 7-7=0) = (istina **OR** laž) **AND** (laž **OR** istina)= istina **AND** istina = istina

c) (4+5=9 **OR** 8+3<11) **OR** (5+9=15 **OR** 7-7=0) = (istina **OR** laž) **OR** (laž **OR** istina)= istina **OR** istina = istina

d) (4+5=9 **AND** 8+3<=11) **OR** (5+9<15 **AND** 7-7=0) = (istina **AND** istina) **OR** (istina **AND** istina)= istina **OR** istina = istina

e) (4+5=9 **AND** 8+2<11) **AND** (5+9<15 **AND** 7-7=0) = (istina **AND** istina) **AND** (istina **AND** istina)= istina **AND** istina = istina

f) (4+5=9 **OR** 8+3<11) **AND** (5+9=15 **OR** 7-7=0) = (istina **OR** istina) **AND** (laž **OR** istina)= istina **AND** istina = istina

15.

TextWindow.Write ("Upiši broj:")

a= TextWindow.ReadNumber()

**If** a>=10 And a<100 **Then**

TextWindow.WriteLine (a+" je dvoznamenkast broj.")

**EndIf**

16.

TextWindow.Write ("Upiši broj:")

a= TextWindow.ReadNumber()

**If** a<10 **Then**

TextWindow.WriteLine (a+" je jednoznamenkast broj.")

**ElseIf** a>=10 And a<100 **Then**

TextWindow.WriteLine (a+" je dvoznamenkast broj.")

**ElseIf** a>=100 And a<1000 **Then**

TextWindow.WriteLine (a+" je troznamenkast broj.")

**ElseIf** a>=1000 And a<10000 **Then**

TextWindow.WriteLine (a+" je četveroznamenkast broj.")

**Else**

TextWindow.WriteLine (a+" ima više od četiri znamenke.")

**EndIf**

17. Napiši program koji upisuje dva datuma (dan i mjesec) iste godine. Program treba ispisati broj dana između ta dva datuma.

TextWindow.Write ("Upiši 1. dan ")

dan1= TextWindow.ReadNumber()

TextWindow.Write ("Upiši 1. mjesec ")

mjesec1= TextWindow.ReadNumber()

TextWindow.Write ("Upiši 2. dan ")

dan2= TextWindow.ReadNumber()

TextWindow.Write ("Upiši 2. mjesec ")

mjesec2= TextWindow.ReadNumber()

**If** mjesec1>mjesec2 **Then**

t= mjesec1

mjesec1=mjesec2

mjesec2=t

t=dan1

dan1=dan2

dan2=t

**EndIf**

broj\_dana=0

**If** mjesec1=mjesec2 **Then**

broj\_dana=Math.Abs (dan2-dan1)

**Else**

**If** mjesec1=1 Or mjesec1=3 Or mjesec1=5 Or mjesec1=7 Or mjesec1=8 Or mjesec1=10 Or mjesec1=12 **Then**

broj\_dana=broj\_dana+31-dan1+1

**ElseIf** mjesec1=4 Or mjesec1=6 Or mjesec1=9 Or mjesec1=11 **Then**

broj\_dana=broj\_dana+30-dan1+1

**Else**

broj\_dana=broj\_dana+28-dan1+1

**EndIf**

TextWindow.WriteLine(broj\_dana)

**For** mjesec=mjesec1+1 **To** mjesec2-1

**If** mjesec=1 Or mjesec=3 Or mjesec=5 Or mjesec=7 Or mjesec=8 Or mjesec=10 Or mjesec=12 **Then**

broj\_dana=broj\_dana+31

**ElseIf** mjesec=4 Or mjesec=6 Or mjesec=9 Or mjesec=11 **Then**

broj\_dana=broj\_dana+30

**Else**

broj\_dana=broj\_dana+28

**EndIf**

**EndFor**

broj\_dana=broj\_dana+dan2

**EndIf**

TextWindow.WriteLine ("Izmeðu "+dan1+"."+mjesec1+ " i " +dan2+"."+mjesec2+" prošlo je "+ broj\_dana+ " dana.")

*(nastavak 17. zadatka)*

**If** mjesec1>mjesec2 **Then**

t= mjesec1

mjesec1=mjesec2

mjesec2=t

t=dan1

dan1=dan2

dan2=t

**EndIf**

broj\_dana=0

**If** mjesec1=mjesec2 **Then**

broj\_dana=Math.Abs (dan2-dan1)

**Else**

**If** mjesec1=1 Or mjesec1=3 Or mjesec1=5 Or mjesec1=7 Or mjesec1=8 Or mjesec1=10 Or mjesec1=12 **Then**

broj\_dana=broj\_dana+31-dan1+1

**ElseIf** mjesec1=4 Or mjesec1=6 Or mjesec1=9 Or mjesec1=11 **Then**

broj\_dana=broj\_dana+30-dan1+1

**Else**

broj\_dana=broj\_dana+28-dan1+1

**EndIf**

TextWindow.WriteLine(broj\_dana)

**For** mjesec=mjesec1+1 **To** mjesec2-1

**If** mjesec=1 Or mjesec=3 Or mjesec=5 Or mjesec=7 Or mjesec=8 Or mjesec=10 Or mjesec=12 **Then**

broj\_dana=broj\_dana+31

**ElseIf** mjesec=4 Or mjesec=6 Or mjesec=9 Or mjesec=11 **Then**

broj\_dana=broj\_dana+30

**Else**

broj\_dana=broj\_dana+28

**EndIf**

**EndFor**

broj\_dana=broj\_dana+dan2

**EndIf**

TextWindow.WriteLine ("Između "+dan1+"."+mjesec1+ " i " +dan2+"."+mjesec2+" prošlo je "+ broj\_dana+ " dana.")

18.

TextWindow.Write ("Upiši stranicu prvog pravokutnika:")

a=TextWindow.ReadNumber()

TextWindow.Write ("Upiši stranicu prvog pravokutnika:")

b=TextWindow.ReadNumber()

TextWindow.Write ("Upiši stranicu drugog pravokutnika:")

c=TextWindow.ReadNumber()

TextWindow.Write ("Upiši stranicu drugog pravokutnika:")

d=TextWindow.ReadNumber()

**if** a<c **Then**

t=a

a=c

c=t

**EndIf**

**if** b<d **Then**

t=b

b=d

d=t

**EndIf**

GraphicsWindow.BrushColor=GraphicsWindow.GetRandomColor()

GraphicsWindow.FillRectangle (0,0,a,b)

GraphicsWindow.BrushColor=GraphicsWindow.GetRandomColor()

GraphicsWindow.FillRectangle ((a-c)/2, (b-d)/2,c,d)

19.

TextWindow.Write ("Broj učenika u razredu:")

n= TextWindow.ReadNumber()

br1=0

br2=0

br3=0

br4=0

br5=0

**For** k= 1 **To** n

TextWindow.Write ("Broj bodova "+k+". učenika:")

bodovi = TextWindow.ReadNumber()

**If** bodovi<50 **Then**

br1=br1+1

**ElseIf** bodovi<65 **Then**

br2=br2+1

**ElseIf** bodovi<80 **Then**

br3=br3+1

**ElseIf** bodovi<90 **Then**

br4=br4+1

**Else**

br5=br5+1

**EndIf**

**EndFor**

TextWindow.WriteLine ("Odličnih:"+br5)

TextWindow.WriteLine ("Vrlo dobrih:"+br4)

TextWindow.WriteLine ("Dobrih:"+br3)

TextWindow.WriteLine ("Dovoljnih:"+br2)

TextWindow.WriteLine ("Nedovoljnih:"+br1)

20.

GraphicsWindow.Height=200

GraphicsWindow.Width=200

GraphicsWindow.Title="Točkice"

TextWindow.Write("a=")

a= TextWindow.ReadNumber()

GraphicsWindow.DrawRectangle (100-a/2, 100-a/2, a ,a)

**For** k= 1 **To** 10000

x= Math.GetRandomNumber (200)

y= Math.GetRandomNumber (200)

**If** x>100-a/2 And x<100+a/2 AND y>100-a/2 AND y<100+a/2 **then**

boja="red"

**Else**

boja="blue"

**EndIf**

GraphicsWindow.SetPixel(x,y,boja)

**EndFor**

21. a)

GraphicsWindow.Height=600

GraphicsWindow.Width=600

GraphicsWindow.Title="Točkice"

GraphicsWindow.DrawRectangle (150, 150,300,300)

pogodak=0

**While** pogodak<1000

x= Math.GetRandomNumber (600)

y= Math.GetRandomNumber (600)

**If** x>150 And x<450 AND y>150 AND y<450 **Then**

boja=GraphicsWindow.GetRandomColor()

GraphicsWindow.SetPixel(x,y,boja)

pogodak=pogodak+1

**EndIf**

**EndWhile**

b)

GraphicsWindow.Height=600

GraphicsWindow.Width=600

GraphicsWindow.Title="Točkice"

TextWindow.Write("a=")

a= TextWindow.ReadNumber()

GraphicsWindow.DrawRectangle (300-a/2, 300-a/2, a ,a)

pogodak=0

**While** pogodak<1000

x= Math.GetRandomNumber (600)

y= Math.GetRandomNumber (600)

**If** x>300-a/2 And x<300+a/2 AND y>300-a/2 AND y<300+a/2 **Then**

boja=GraphicsWindow.GetRandomColor()

GraphicsWindow.SetPixel(x,y,boja)

pogodak=pogodak+1

**EndIf**

**EndWhile**

22.

GraphicsWindow.Title="Histogram ocjena iz informatike"

GraphicsWindow.Width=600

GraphicsWindow.Height= 530

GraphicsWindow.FontBold="true"

GraphicsWindow.FontSize=14

GraphicsWindow.DrawLine (60,470, 60,30)

GraphicsWindow.DrawLine (60,30,50,40)

GraphicsWindow.DrawLine (60,30,70,40)

GraphicsWindow.DrawBoundText (20,20,50,"broj")

y=1

**For** k= 430 **To** 50 **Step** -20

GraphicsWindow.DrawLine (50,k, 70,k)

**For** m= 70 **To** 550 **Step** 10

GraphicsWindow.DrawLine (m,k, m+3,k)

**EndFor**

GraphicsWindow.DrawBoundText (20,k-10,20,y)

y=y+1

**EndFor**

GraphicsWindow.DrawLine (40,450, 570,450)

GraphicsWindow.DrawLine (570,450,560,440)

GraphicsWindow.DrawLine (570,450,560,460)

GraphicsWindow.DrawBoundText (550,460,50,"ocjene")

GraphicsWindow.DrawBoundText (100,460,80,"1")

GraphicsWindow.DrawBoundText (180,460,80,"2")

GraphicsWindow.DrawBoundText (260,460,80,"3")

GraphicsWindow.DrawBoundText (340,460,80,"4")

GraphicsWindow.DrawBoundText (420,460,80,"5")

TextWindow.Write ("Koliko ima učenika je pisalo ispit?")

br=TextWindow.ReadNumber()

**For** k= 1 **To** br

TextWindow.Write ("Ocjena " + k + ". tog učenika:")

ocjena=TextWindow.Read()

**If** ocjena=1 **Then**

br1=br1+1

**ElseIf** ocjena=2 **Then**

br2=br2+1

**ElseIf** ocjena=3 **Then**

br3=br3+1

**ElseIf** ocjena=4 **Then**

br4=br4+1

**Else**

br5=br5+1

**EndIf**

**EndFor**

GraphicsWindow.BrushColor=GraphicsWindow.GetRandomColor()

GraphicsWindow.FillRectangle (80,450-20\*br1,50, 20\*br1)

GraphicsWindow.BrushColor=GraphicsWindow.GetRandomColor()

GraphicsWindow.FillRectangle (160,450-20\*br2,50, 20\*br2)

GraphicsWindow.BrushColor=GraphicsWindow.GetRandomColor()

GraphicsWindow.FillRectangle (240,450-20\*br3,50, 20\*br3)

GraphicsWindow.BrushColor=GraphicsWindow.GetRandomColor()

GraphicsWindow.FillRectangle (320,450-20\*br4,50, 20\*br4)

GraphicsWindow.BrushColor=GraphicsWindow.GetRandomColor()

GraphicsWindow.FillRectangle (400,450-20\*br5,50, 20\*br5)

### 3. Potprogrami

23. Potprogrami nam služe za pojednostavljivanje složenog programa, jer nam omogućuju rastavljanje programa na manje dijelove, čime tijek programa postaje pregledniji. Pomoću njih možemo zadavati naredbe koje treba izvršiti kad se dogodi neki događaj u programu, npr. klik mišem, pritisak neke tipke na tipkovnici, klik gumba ili upis teksta u tekstualni okvir.

Potprogram započinje riječju **Sub** iza koje slijedi ime potprograma. Potprogram završava s **EndSub**. Potprogram pozivamo navodeći ime potprograma.

24. U tablici su zadani opisi pojedinih naredbi za rad mišem i tipkovnicom. Dopuni dijelove naredbi koje nedostaju.

|  |  |
| --- | --- |
| **Opis naredbe** | **Naredba** |
| daje trenutnu **y** koordinatu miša u grafičkom prozoru | GraphicsWindow.MouseY |
| pokazuje pokazivač miša | Mouse.ShowCursor() |
| izvršava zadni potprogram kada je pritisnuta bilo koja tipka na tipkovnici | GraphicsWindow.KeyDown |
| izvršava zadni potprogram kada je miš pomaknut | GraphicsWindow.MouseMove |
| daje rezultat *istina* ako je pritisnuta lijeva tipka miša, a inače *laž* | Mouse.IsLeftButtonDown |
| izvršava zadni potprogram kada je kliknuta tipka miša | GraphicsWindow.MouseDown |
| daje trenutnu **x** koordinatu miša u grafičkom prozoru | GraphicsWindow.MouseX |
| daje zadnju pritisnutu tipku na tipkovnici | GraphicsWindow.LastKey |
| daje rezultat *istina* ako je pritisnuta desna tipka miša, a inače *laž* | Mouse.IsRightButtonDown |

25.

GraphicsWindow.Height=200

GraphicsWindow.Width=200

GraphicsWindow.FontSize=18

GraphicsWindow.Title="Tipka miša"

GraphicsWindow.MouseDown=ispiši

**Sub** ispiši

**If** Mouse.IsLeftButtonDown **Then**

GraphicsWindow.DrawBoundText (10,100,200,"Kliknuta lijeva tipka")

**Else**

GraphicsWindow.DrawBoundText (10,100,200,"Kliknuta desna tipka")

**EndIf**

**EndSub**

26.

GraphicsWindow.Height=300

GraphicsWindow.Width= 300

GraphicsWindow.Title="Crtanje pomoću tipkovnice"

GraphicsWindow.PenWidth=10

GraphicsWindow.KeyDown=crtaj

x= GraphicsWindow.Width/2

y= GraphicsWindow.Height/2

GraphicsWindow.SetPixel (x,y, "black")

**Sub** crtaj

x1=x

y1=y

**If** GraphicsWindow.LastKey="I" **then**

y=y-10

**ElseIf** GraphicsWindow.LastKey="M" **then**

y=y+10

**ElseIf** GraphicsWindow.LastKey="J" **then**

x=x-10

**ElseIf** GraphicsWindow.LastKey="K" **then**

x=x+10

**ElseIf** GraphicsWindow.LastKey="B" **then**

GraphicsWindow.PenColor=GraphicsWindow.GetRandomColor()

**EndIf**

GraphicsWindow.DrawLine (x1,y1,x,y)

**EndSub**

GraphicsWindow.Height=400

GraphicsWindow.Width=400

GraphicsWindow.Title="Likovi"

Mouse.ShowCursor()

GraphicsWindow.MouseDown=crtaj

**Sub** crtaj

**If** Mouse.IsLeftButtonDown **Then**

kvadrat()

**ElseIf** Mouse.IsRightButtonDown **Then**

pravokutni\_trokut()

**EndIf**

**EndSub**

**Sub** kvadrat

GraphicsWindow.BrushColor=GraphicsWindow.GetRandomColor()

x=GraphicsWindow.MouseX

y=GraphicsWindow.MouseY

GraphicsWindow.FillRectangle (x,y 20,20)

**EndSub**

**Sub** pravokutni\_trokut

GraphicsWindow.BrushColor=GraphicsWindow.GetRandomColor()

x=GraphicsWindow.MouseX

y=GraphicsWindow.MouseY

GraphicsWindow.FillTriangle (x,y,x+20,y,x,y+20)

**EndSub**

27.

28.

|  |  |
| --- | --- |
| **Opis** | **Naredba** |
| zadaje gumb u grafičkom prozoru na mjestu (50,10), a na gumbu je ispisan tekst **pravokutnik** | Controls.AddButton (50,10, “pravokutnik“) |
| zadaje okvir s tekstom na mjestu (300,210) | Controls.AddTextBox (300,210) |
| daje tekst koji je ispisan na gumbu **kontrola** | Controls. GetButtonCaption (kontrola) |
| postavlja širinu na 30 i visinu na 70 kontrole **kontrola** | Controls SetSize (kontrola, 30,70) |
| izvršava potprogram **crtanje** kada je kliknut bilo koji gumb u grafičkom prozoru | Controls.ButtonClicked= crtanje |
| daje ime zadnjeg kliknutog gumba | Controls.LastClickedButton |
| daje ime zadnjeg okvira za tekst u koji upisan tekst | Controls.LastTypedTextBox |

29.

GraphicsWindow.Height=300

GraphicsWindow.Width=400

GraphicsWindow.Title="Kružnica"

gumb= Controls.AddButton ("kružnica", 20,50)

Controls.ButtonClicked = kružnica

**Sub** kružnica

GraphicsWindow.DrawEllipse (150,50,200,200)

**EndSub**

30

GraphicsWindow.Height=300

GraphicsWindow.Width=500

GraphicsWindow.Title="Zid"

mala= Controls.AddButton ("Mala cigla", 10,50)

velika= Controls.AddButton ("Velika cigla", 10,90)

gore = Controls.AddButton ("Sljedeći red", 10,130)

Controls.ButtonClicked = zidaj

x=100

y= GraphicsWindow.Height-80

**Sub** zidaj

**If** Controls.LastClickedButton=velika **Then**

GraphicsWindow.BrushColor="red"

GraphicsWindow.FillRectangle (x,y,100,50)

GraphicsWindow.DrawRectangle (x,y,100,50)

x=x+100

**ElseIf** Controls.LastClickedButton=mala **Then**

GraphicsWindow.BrushColor="Maroon"

GraphicsWindow.FillRectangle (x,y,50,50)

GraphicsWindow.DrawRectangle (x,y,50,50)

x=x+50

**ElseIf** Controls.LastClickedButton=gore **Then**

x=100

y=y-50

**EndIf**

**EndSub**

31.

GraphicsWindow.Height=500

GraphicsWindow.Width=500

GraphicsWindow.Title="Piramida trokuta"

trokut= Controls.AddButton ("Trokut", 10,50)

gore = Controls.AddButton ("Sljedeći red", 10,90)

briši= Controls.AddButton ("Briši", 10,130)

Controls.ButtonClicked = piramida

x=100

x1=100

y= GraphicsWindow.Height-50

**Sub** piramida

**If** Controls.LastClickedButton=trokut **Then**

GraphicsWindow.BrushColor=GraphicsWindow.GetRandomColor()

GraphicsWindow.FillTriangle (x,y,x+50,y, x+25,y-40)

GraphicsWindow.DrawTriangle (x,y,x+50,y, x+25,y-40)

x=x+50

**ElseIf** Controls.LastClickedButton=gore **Then**

x=x1+25

y=y-40

x1=x

**ElseIf** Controls.LastClickedButton=briši **Then**

GraphicsWindow.BrushColor="White"

GraphicsWindow.FillRectangle (100,0,GraphicsWindow.Width-100, GraphicsWindow.Height)

x=100

x1=100

y= GraphicsWindow.Height-50

**EndIf**

**EndSub**

32.

GraphicsWindow.Height=300

GraphicsWindow.Width=500

GraphicsWindow.Title="Ispis teksta"

GraphicsWindow.DrawBoundText (5,30,65, "Tekst:")

tekst1= Controls.AddTextBox (70,30)

GraphicsWindow.DrawBoundText (5,70,65, "Boja teksta:")

boja1= Controls.AddTextBox (70,70)

Controls.SetSize (boja1, 50,30)

GraphicsWindow.DrawBoundText (5,110,65, "Veličina teksta:")

veličina1= Controls.AddTextBox (70,110)

Controls.SetSize (veličina1, 50,30)

GraphicsWindow.DrawBoundText (5,150,65, "Boja podloge")

pozadina1= Controls.AddTextBox (70,150)

Controls.SetSize (pozadina1, 50,30)

gumb = Controls.AddButton ("Ispis", 50, 190)

Controls.ButtonClicked= ispiši

**Sub** ispiši

boja= Controls.GetTextBoxText (boja1)

tekst=Controls.GetTextBoxText (tekst1)

veličina=Controls.GetTextBoxText (veličina1)

pozadina= Controls.GetTextBoxText (pozadina1)

GraphicsWindow.BrushColor=boja

GraphicsWindow.FontSize= veličina

GraphicsWindow.BackgroundColor=pozadina

GraphicsWindow.DrawBoundText (200,100,300,tekst)

**EndSub**

33.

GraphicsWindow.Title="Računske operacije"

GraphicsWindow.FontSize=16

GraphicsWindow.BrushColor="black"

prvi= Controls.AddButton ("Prvi broj", 10,10)

drugi= Controls.AddButton ("Drugi broj", 10,60)

operacija = Controls.AddTextBox (10,110)

račun= Controls.AddButton ("Izračunaj", 10,160)

brisanje= Controls.AddButton ("Briši", 10, 210)

Controls.ButtonClicked=ispis

**Sub** ispis

**If** Controls.LastClickedButton= prvi **Then**

a= Math.GetRandomNumber(1000)

GraphicsWindow.DrawBoundText (120,10,200, a)

**ElseIf** Controls.LastClickedButton= drugi **Then**

b= Math.GetRandomNumber(1000)

GraphicsWindow.DrawBoundText (120,60,200, b)

**elseif** Controls.LastClickedButton=brisanje **then**

briši()

**Else**

op= Controls.GetTextBoxText (operacija)

**If** op="+" **Then**

rezultat=a+b

**ElseIf** op="-" **Then**

rezultat=a-b

**ElseIf** op="\*" **Then**

rezultat=a\*b

**Elseif** op="/" **Then**

rezultat=a/b

**EndIf**

**If** op="+" or op="-" or op="\*" or op="/" **Then**

GraphicsWindow.DrawBoundText (150,160,500, a+" " + op+ " " +b+"="+rezultat)

**Else**

GraphicsWindow.DrawBoundText (150,160,500,"Operacija nije dobro zadana.")

**EndIf**

**EndIf**

**EndSub**

**Sub** briši

GraphicsWindow.BrushColor="White"

GraphicsWindow.FillRectangle (120,5, 500, 350)

GraphicsWindow.BrushColor="black"

**EndSub**

34.

GraphicsWindow.Title=" Kvadrat u 16 milijuna boja"

GraphicsWindow.FontSize=14

GraphicsWindow.BrushColor="red"

GraphicsWindow.DrawBoundText(10,10,100,"crvena=")

c1= Controls.AddTextBox (70,10)

Controls.SetSize (c1,60, 30)

GraphicsWindow.BrushColor="green"

GraphicsWindow.DrawBoundText(10,60,100,"zelena=")

z1= Controls.AddTextBox (70,60)

Controls.SetSize (z1,60, 30)

GraphicsWindow.BrushColor="blue"

GraphicsWindow.DrawBoundText(10,110,100,"plava=")

p1= Controls.AddTextBox (70,110)

Controls.SetSize (p1,60, 30)

GraphicsWindow.BrushColor="black"

GraphicsWindow.DrawBoundText(10,160,100,"a=")

a1= Controls.AddTextBox (70,160)

Controls.SetSize (a1,60, 30)

GraphicsWindow.BrushColor="black"

gumb= Controls.AddButton ("Kvadrat", 35,200)

Controls.ButtonClicked=crtaj

**Sub** crtaj

c= Controls.GetTextBoxText(c1)

z=Controls.GetTextBoxText(z1)

p= Controls.GetTextBoxText(p1)

a= Controls.GetTextBoxText(a1)

GraphicsWindow.BrushColor=GraphicsWindow.GetColorFromRGB(c,z,p)

GraphicsWindow.FillRectangle (200,50,a,a)

**EndSub**

### 4. Primjena programiranja u matematici

35.

GraphicsWindow.Title="Pravokutnik"

GraphicsWindow.BrushColor="black"

GraphicsWindow.FontSize=14

GraphicsWindow.DrawBoundText(10,10,40,"a=")

a1= Controls.AddTextBox (35,10)

Controls.SetSize (a1,60, 30)

GraphicsWindow.DrawBoundText(10,50,40,"b=")

b1= Controls.AddTextBox (35,50)

Controls.SetSize (b1,60, 30)

gumb= Controls.AddButton ("Pravokutnik", 35,90)

Controls.ButtonClicked=crtaj\_pravokutnik

**Sub** crtaj\_pravokutnik

a= Controls.GetTextBoxText(a1)

b= Controls.GetTextBoxText(b1)

d=Math.SquareRoot(a\*a+b\*b)

r\_opisana=d/2

x1=GraphicsWindow.Width/3

y1= GraphicsWindow.Height/5

GraphicsWindow.DrawRectangle(x1,y1,a,b)

x\_središte = x1+a/2

y\_središte = y1+b/2

GraphicsWindow.PenColor="red"

GraphicsWindow.DrawLine (x\_središte,y\_središte,x1,y1)

GraphicsWindow.DrawEllipse (x\_središte-r\_opisana, y\_središte-r\_opisana, 2\*r\_opisana, 2\*r\_opisana)

GraphicsWindow.PenColor="blue"

GraphicsWindow.DrawLine (x1,y1,x1+a,y1+b)

GraphicsWindow.DrawLine (x1,y1+b,x1+a,y1)

o=2\*(a+b)

p=a\*b

GraphicsWindow.DrawBoundText(10,130,409,"d="+d)

GraphicsWindow.DrawBoundText(10,170,409,"o="+o)

GraphicsWindow.DrawBoundText(10,210,409,"P="+p)

**EndSub**

36.

GraphicsWindow.Title="Kvadrat"

GraphicsWindow.BrushColor="black"

GraphicsWindow.FontSize=14

GraphicsWindow.DrawBoundText(10,10,40,"a=")

a1= Controls.AddTextBox (35,10)

Controls.SetSize (a1,60, 30)

gumb= Controls.AddButton ("Kvadrat", 35,50)

Controls.ButtonClicked=crtaj\_kvadrat

**Sub** crtaj\_kvadrat

a= Controls.GetTextBoxText(a1)

r\_opisana=a\* Math.SquareRoot(2)/2

r\_upisana=a/2

x1=GraphicsWindow.Width/3

y1= GraphicsWindow.Height/5

GraphicsWindow.DrawRectangle(x1,y1,a,a)

x\_središte = x1+a/2

y\_središte = y1+a/2

GraphicsWindow.PenColor="red"

GraphicsWindow.DrawLine (x\_središte,y\_središte,x1,y1+a)

GraphicsWindow.DrawEllipse (x\_središte-r\_opisana, y\_središte-r\_opisana, 2\*r\_opisana, 2\*r\_opisana)

GraphicsWindow.PenColor="blue"

x\_p=x1+a/2

y\_p=y1+a

GraphicsWindow.DrawLine (x\_središte,y\_središte,x\_p,y\_p)

GraphicsWindow.DrawEllipse (x\_središte-r\_upisana, y\_središte-r\_upisana, 2\*r\_upisana, 2\*r\_upisana)

**EndSub**

37.

GraphicsWindow.Title="Romb"

GraphicsWindow.BrushColor="black"

GraphicsWindow.FontSize=14

GraphicsWindow.DrawBoundText(10,10,40,"e=")

e1= Controls.AddTextBox (35,10)

Controls.SetSize (e1,60, 30)

GraphicsWindow.DrawBoundText(10,60,40,"f=")

f1= Controls.AddTextBox (35,60)

Controls.SetSize (f1,60, 30)

gumb= Controls.AddButton ("Romb", 135,50)

Controls.ButtonClicked=crtaj\_romb

**Sub** crtaj\_romb

e= Controls.GetTextBoxText(e1)

f= Controls.GetTextBoxText(f1)

xs=GraphicsWindow.Width/2

ys= GraphicsWindow.Height/2

x1=xs

x3=xs

x2=xs-e/2

x4=xs+e/2

y1=ys-f/2

y3=ys+f/2

y2=ys

y4=ys

a=Math.SquareRoot ((e\*e+f\*f)/4)

GraphicsWindow.DrawLine(x1,y1,x2,y2)

GraphicsWindow.DrawLine(x2,y2,x3,y3)

GraphicsWindow.DrawLine(x3,y3,x4,y4)

GraphicsWindow.DrawLine(x1,y1,x4,y4)

GraphicsWindow.PenColor="blue"

GraphicsWindow.DrawLine(x1,y1,x3,y3)

GraphicsWindow.DrawLine(x2,y2,x4,y4)

opseg = 4\*a

površina=e\*f/2

GraphicsWindow.DrawBoundText (20,130,250, "Duljina stranice: a="+a)

GraphicsWindow.DrawBoundText (20,150,150, "Opseg: o="+opseg)

GraphicsWindow.DrawBoundText (20,170,150, "Površina: P="+površina)

**EndSub**

38.

GraphicsWindow.Title="Jednakokračan trapez"

GraphicsWindow.BrushColor="black"

GraphicsWindow.FontSize=14

GraphicsWindow.DrawBoundText(10,10,40,"a=")

a1= Controls.AddTextBox (35,10)

Controls.SetSize (a1,60, 30)

GraphicsWindow.DrawBoundText(10,50,40,"b=")

b1= Controls.AddTextBox (35,50)

Controls.SetSize (b1,60, 30)

GraphicsWindow.DrawBoundText(10,90,40,"c=")

c1= Controls.AddTextBox (35,90)

Controls.SetSize (c1,60, 30)

gumb= Controls.AddButton ("Jednakokračan trapez", 35,130)

Controls.ButtonClicked=crtaj\_trapez

**Sub** crtaj\_trapez

a= Controls.GetTextBoxText(a1)

b= Controls.GetTextBoxText(b1)

c= Controls.GetTextBoxText(c1)

z=(a-c)/2

v=Math.SquareRoot(b\*b-z\*z)

x=250

y= 50+v

GraphicsWindow.DrawLine (x,y,x+a,y)

GraphicsWindow.DrawLine (x+a,y, x+z+c,y-v)

GraphicsWindow.DrawLine (x+z+c,y-v, x+z,y-v)

GraphicsWindow.DrawLine (x+z,y-v, x,y)

GraphicsWindow.PenColor="red"

GraphicsWindow.DrawLine (x+z,y-v, x+z,y)

GraphicsWindow.DrawLine (x+z+c,y-v, x+z+c,y)

o=a+c+2\*b

p=(a+c)/2\*v

GraphicsWindow.DrawBoundText(10,170,409,"o="+o)

GraphicsWindow.DrawBoundText(10,210,409,"P="+p)

**EndSub**

39.

GraphicsWindow.Title="Funkcija drugog korijena"

GraphicsWindow.FontSize=20

koordinatni\_sustav()

a\_1= Controls.AddTextBox (5,5)

Controls.SetSize (a\_1,50,35)

gumb= Controls.AddButton ("CRTAJ", 70,5)

red=50

Controls.ButtonClicked=crtaj\_funkciju

**Sub** koordinatni\_sustav

GraphicsWindow.Top=0

GraphicsWindow.Left=0

GraphicsWindow.Height=700

GraphicsWindow.Width=1000

*'x os*

GraphicsWindow.DrawLine (0,350,950,350)

GraphicsWindow.DrawLine (950,350,940,340)

GraphicsWindow.DrawLine (950,350,940,360)

**For** x= 10 **To** 940 **Step** 10

GraphicsWindow.DrawLine (x,345,x,355)

**EndFor**

GraphicsWindow.DrawBoundText (945,360,20,"x")

*'y os*

GraphicsWindow.DrawLine (500,10,500,700)

GraphicsWindow.DrawLine (500,10,490,20)

GraphicsWindow.DrawLine (500,10,510,20)

**For** y= 20 **To** 690 **Step** 10

GraphicsWindow.DrawLine (495, y, 505,y)

**EndFor**

GraphicsWindow.DrawBoundText (530,10,20,"y")

**EndSub**

**Sub** crtaj\_funkciju

GraphicsWindow.PenColor=GraphicsWindow.GetRandomColor()

a= Controls.GetTextBoxText (a\_1)

x=00

y= Math.SquareRoot(a\*x)

x\_koor=10\*x+500

y\_koor=350-10\*y

**For** x= 1 **To** 48

raniji\_x= x\_koor

raniji\_y=y\_koor

y=Math.SquareRoot(a\*x)

x\_koor=10\*x+500

y\_koor=350-10\*y

GraphicsWindow.DrawLine(raniji\_x,raniji\_y,x\_koor,y\_koor)

**EndFor**

GraphicsWindow.BrushColor=GraphicsWindow.PenColor

GraphicsWindow.DrawBoundText (10,red,400, "y= korijen("+a+"\*x)")

red=red+30

**EndSub**

.

### 5. Primjena programiranja u fizici i kemiji

40.

GraphicsWindow.Title="Jednoliko pravocrtno gibanje"

GraphicsWindow.Height=700

GraphicsWindow.Width=750

GraphicsWindow.Top=0

GraphicsWindow.Left=0

GraphicsWindow.DrawBoundText (5,15,30,"v=")

v1= Controls.AddTextBox (30,10)

Controls.SetSize (v1,50,30)

GraphicsWindow.DrawBoundText (5,50,30,"t=")

t1= Controls.AddTextBox (30,45)

Controls.SetSize (t1,50,30)

gumb= Controls.AddButton ("v-t graf",10,90)

Controls.ButtonClicked = crtaj\_graf

y\_teksta=120

**Sub** crtaj\_graf

koordinatni\_sustav()

v=Controls.GetTextBoxText (v1)

t= Controls.GetTextBoxText (t1)

GraphicsWindow.BrushColor=GraphicsWindow.GetRandomColor()

GraphicsWindow.FillRectangle (150,600-v,t,v)

GraphicsWindow.DrawLine (150,600-v, 150+t,600-v)

s= v\*t

GraphicsWindow.DrawBoundText(10,y\_teksta,200, "v= "+v+ " m/s")

GraphicsWindow.DrawBoundText(10,y\_teksta+20,200, "t= "+t+ " s")

GraphicsWindow.DrawBoundText(10,y\_teksta+40,200, "Prijeđeni put: ")

GraphicsWindow.DrawBoundText(10,y\_teksta+60,200,s +" m")

y\_teksta=y\_teksta+80

**EndSub**

*(nastavak 40. zadataka)*

**Sub** koordinatni\_sustav

*'v os*

GraphicsWindow.DrawLine (150,10,150, 620)

GraphicsWindow.DrawLine (150,10,140,20)

GraphicsWindow.DrawLine (150,10,160,20)

GraphicsWindow.DrawBoundText (90,10,100,"v (m/s)")

**For** y= 600 **To** 50 **Step** -10

**If** Math.Remainder (y,50)=0 **Then**

GraphicsWindow.PenWidth=2

GraphicsWindow.DrawBoundText (110,y-8,100,600-y)

**Else**

GraphicsWindow.PenWidth=1

**EndIf**

GraphicsWindow.DrawLine (140,y,160,y)

**EndFor**

*'x os*

GraphicsWindow.DrawLine (150,600,700, 600)

GraphicsWindow.DrawLine (700,600,690,590)

GraphicsWindow.DrawLine (700,600,690,610)

GraphicsWindow.DrawBoundText (680,610,100,"t (s)")

**For** x= 160 **To** 650 **Step** 10

**If** Math.Remainder (x,50)=0 **Then**

GraphicsWindow.PenWidth=2

GraphicsWindow.DrawBoundText (x-10,615,50,x-150)

**Else**

GraphicsWindow.PenWidth=1

**EndIf**

GraphicsWindow.DrawLine (x,590,x,610)

**EndFor**

**EndSub**

41.

GraphicsWindow.Title="Paralelni spoj otpornika"

GraphicsWindow.Width=1000

GraphicsWindow.FontSize=14

**for** k= 1 **to** 10

GraphicsWindow.DrawBoundText (10,15+(k-1)\*40, 50, "R"+k+"=")

okvir [k] = Controls.AddTextBox (40,10+(k-1)\*40)

Controls.SetSize (okvir[k], 30,30)

**Endfor**

otpor=Controls.AddButton ("Ukupni otpor", 20, 410)

Controls.ButtonClicked=crtaj\_paralelni

**Sub** crtaj\_paralelni

**For** k= 1 **To** 10

r [k] = Controls.GetTextBoxText (okvir [k])

**EndFor**

GraphicsWindow.DrawLine (100,50,150,50)

x=150

y= 50

r\_ukupni=0

**For** k= 1 **To** 10

**If** r [k]<>"" **Then**

GraphicsWindow.DrawLine (x,y,x,y+30)

GraphicsWindow.DrawRectangle (x-15,y+30,30,50)

GraphicsWindow.FontName="ARIAL"

GraphicsWindow.DrawBoundText (x+20,y+50,40,r[k])

GraphicsWindow.FontName="SYMBOL"

GraphicsWindow.DrawBoundText(x+40,y+50,20,"W")

GraphicsWindow.DrawLine (x,y+80,x,y+110)

GraphicsWindow.DrawLine (x,y,x+80,y)

x=x+80

r\_ukupni=r\_ukupni+1/r[k]

**EndIf**

**EndFor**

GraphicsWindow.DrawLine (x,y,x,y+110)

GraphicsWindow.DrawLine (x,y+110,130,y+110)

GraphicsWindow.DrawLine (120,y+90,120,y+130)

GraphicsWindow.PenWidth=8

GraphicsWindow.DrawLine (130,y+100,130,y+120)

GraphicsWindow.PenWidth=2

GraphicsWindow.DrawLine (120,y+110,100,y+110)

GraphicsWindow.DrawLine (100,y+110,100,y)

r\_ukupni=1/r\_ukupni

GraphicsWindow.FontName="ARIAL"

GraphicsWindow.DrawBoundText (100,200,350,"Ukupni otpor: R="+r\_ukupni)

GraphicsWindow.FontName="SYMBOL"

GraphicsWindow.DrawBoundText (250,200,20,"W")

**EndSub**

42.

GraphicsWindow.Title="Alkeni"

GraphicsWindow.top=0

GraphicsWindow.Left=0

GraphicsWindow.Height=600

GraphicsWindow.Width=1000

GraphicsWindow.DrawBoundText (10,10,200, "Broj atoma vodika:")

n\_1 = Controls.AddTextBox (210,10)

Controls.SetSize (n\_1,60,25)

gumb = Controls.AddButton ("Formula",500,10)

Controls.ButtonClicked=crtaj\_formulu

**Sub** crtaj\_formulu

*' obriši raniji crtež*

GraphicsWindow.BrushColor="white"

GraphicsWindow.FillRectangle (0,50, 1000,600)

n=Controls.GetTextBoxText (n\_1)

x=40

y=350

GraphicsWindow.BrushColor="yellow"

GraphicsWindow.BrushColor="blue"

atom()

GraphicsWindow.DrawLine (x,y-25,x,y-75)

y=y-100

GraphicsWindow.BrushColor="yellow"

atom()

y=y+100

GraphicsWindow.DrawLine (x,y+25,x,y+75)

y=y+100

atom()

y=y-100

x=x+100

*'dvostruka veza*

GraphicsWindow.DrawLine (x-75,y-10,x-25,y-10)

GraphicsWindow.DrawLine (x-75,y+10,x-25,y+10)

GraphicsWindow.BrushColor="blue"

atom()

GraphicsWindow.DrawLine (x,y-25,x,y-75)

y=y-100

GraphicsWindow.BrushColor="yellow"

atom()

y=y+100

x=x+100

GraphicsWindow.DrawLine (x-75,y,x-25,y)

**For** k= 1 **To** n-2

ch2()

**EndFor**

atom ()

GraphicsWindow.BrushColor="black"

o\_spoju ()

**EndSub**

*(nastavak 42. zadatka)*

**Sub** o\_spoju

**If** n=2 **Then**

GraphicsWindow.DrawBoundText (20,50, 100, "ETEN")

**ElseIf** n=3 **Then**

GraphicsWindow.DrawBoundText (20,50, 100, "PROPEN")

**ElseIf** n=4 **Then**

GraphicsWindow.DrawBoundText (20,50, 100, "BUTEN")

**ElseIf** n=5 **Then**

GraphicsWindow.DrawBoundText (20,50, 100, "PENTEN")

**ElseIf** n=6 **Then**

GraphicsWindow.DrawBoundText (20,50, 100, "HEKSEN")

**ElseIf** n=7 **Then**

GraphicsWindow.DrawBoundText (20,50, 100, "HEPTEN")

**ElseIf** n=8 **Then**

GraphicsWindow.DrawBoundText (20,50, 100, "OKTEN")

**EndIf**

h=2\*n

GraphicsWindow.DrawBoundText (20,80,300, "BROJ ATOMA VODIKA:"+h)

masa= 12\*n+1.008\*h

GraphicsWindow.DrawBoundText (20,110,300, "RELATIVNA MOLEKULSKA MASA:"+ masa)

maseni\_c =12\*n/masa

maseni\_h =1.008\*h/masa

maseni\_c = (Math.Round (maseni\_c\*10000))/10000

maseni\_h = (Math.Round (maseni\_h\*10000))/10000

postotak\_c = Math.Round(maseni\_c\*10000)/100

postotak\_h = Math.Round(maseni\_h\*10000)/100

GraphicsWindow.DrawBoundText (20,140,700, "Maseni udio vodika:"+ maseni\_h+"="+ postotak\_h+"%")

GraphicsWindow.DrawBoundText (20,170,700, "Maseni udio ugljika:"+ maseni\_c+"="+ postotak\_c+"%")

**EndSub**

**Sub** atom

GraphicsWindow.FillEllipse(x-25,y-25,50,50)

GraphicsWindow.DrawEllipse(x-25,y-25,50,50)

**EndSub**

**Sub** ch2

GraphicsWindow.BrushColor="blue"

atom()

GraphicsWindow.DrawLine (x,y-25,x,y-75)

y=y-100

GraphicsWindow.BrushColor="yellow"

atom()

y=y+100

GraphicsWindow.DrawLine (x,y+25,x,y+75)

y=y+100

atom()

y=y-100

GraphicsWindow.DrawLine (x+25,y,x+75,y)

x=x+100

**EndSub**

43.

GraphicsWindow.Title="Alkini"

GraphicsWindow.top=0

GraphicsWindow.Left=0

GraphicsWindow.Height=600

GraphicsWindow.Width=1000

GraphicsWindow.DrawBoundText (10,10,200, "Broj atoma ugljika:")

n\_1 = Controls.AddTextBox (210,10)

Controls.SetSize (n\_1,60,25)

gumb = Controls.AddButton ("Formula",500,10)

Controls.ButtonClicked=crtaj\_formulu

**Sub** crtaj\_formulu

*' obriši raniji crtež*

GraphicsWindow.BrushColor="white"

GraphicsWindow.FillRectangle (0,50, 1000,600)

n=Controls.GetTextBoxText (n\_1)

x=40

y=350

GraphicsWindow.BrushColor="yellow"

GraphicsWindow.BrushColor="blue"

atom()

GraphicsWindow.DrawLine (x,y-25,x,y-75)

y=y-100

GraphicsWindow.BrushColor="yellow"

atom()

y=y+100

x=x+100

*'trostruka veza*

GraphicsWindow.DrawLine (x-85,y-20,x-15,y-20)

GraphicsWindow.DrawLine (x-75,y,x-25,y)

GraphicsWindow.DrawLine (x-85,y+20,x-15,y+20)

GraphicsWindow.BrushColor="blue"

atom()

x=x+100

GraphicsWindow.DrawLine (x-75,y,x-25,y)

**If** n> 2 **then**

**For** k= 1 **To** n-2

ch2()

**EndFor**

**EndIf**

GraphicsWindow.BrushColor="yellow"

atom ()

GraphicsWindow.BrushColor="black"

o\_spoju ()

**EndSub**

*(nastavak 43. zadatka)*

**Sub** o\_spoju

**If** n=2 **Then**

GraphicsWindow.DrawBoundText (20,50, 100, "ETIN")

**ElseIf** n=3 **Then**

GraphicsWindow.DrawBoundText (20,50, 100, "PROPIN")

**ElseIf** n=4 **Then**

GraphicsWindow.DrawBoundText (20,50, 100, "BUTIN")

**ElseIf** n=5 **Then**

GraphicsWindow.DrawBoundText (20,50, 100, "PENTIN")

**ElseIf** n=6 **Then**

GraphicsWindow.DrawBoundText (20,50, 100, "HEKSIN")

**ElseIf** n=7 **Then**

GraphicsWindow.DrawBoundText (20,50, 100, "HEPTIN")

**ElseIf** n=8 **Then**

GraphicsWindow.DrawBoundText (20,50, 100, "OKTIN")

**EndIf**

h=2\*n-2

GraphicsWindow.DrawBoundText (20,80,300, "BROJ ATOMA VODIKA:"+h)

masa= 12\*n+1.008\*h

GraphicsWindow.DrawBoundText (20,110,300, "RELATIVNA MOLEKULSKA MASA:"+ masa)

maseni\_c =12\*n/masa

maseni\_h =1.008\*h/masa

maseni\_c = (Math.Round (maseni\_c\*10000))/10000

maseni\_h = (Math.Round (maseni\_h\*10000))/10000

postotak\_c = Math.Round(maseni\_c\*10000)/100

postotak\_h = Math.Round(maseni\_h\*10000)/100

GraphicsWindow.DrawBoundText (20,140,700, "Maseni udio vodika:"+ maseni\_h+"="+ postotak\_h+"%")

GraphicsWindow.DrawBoundText (20,170,700, "Maseni udio ugljika:"+ maseni\_c+"="+ postotak\_c+"%")

**EndSub**

**Sub** atom

GraphicsWindow.FillEllipse(x-25,y-25,50,50)

GraphicsWindow.DrawEllipse(x-25,y-25,50,50)

**EndSub**

**Sub** ch2

GraphicsWindow.BrushColor="blue"

atom()

GraphicsWindow.DrawLine (x,y-25,x,y-75)

y=y-100

GraphicsWindow.BrushColor="yellow"

atom()

y=y+100

GraphicsWindow.DrawLine (x,y+25,x,y+75)

y=y+100

atom()

y=y-100

GraphicsWindow.DrawLine (x+25,y,x+75,y)

x=x+100

**EndSub**

## 5. BAZE PODATAKA

1. Excelova tablica može sadržavati najviše 1 048 576 redaka i najviše 16 384 stupaca), podržava manje vrsta podataka, pojedina ćelija može sadržavati najviše 32 767 znakova, Excelove tablice se ne mogu međusobno povezivati, tj. stavljati u relaciju.

2. Sustav za upravljanje bazom podataka je program za organiziranje baze podataka i za rad s podacima u njoj. Omogućuje npr. unošenje, uređivanje, prikazivanje, pretraživanje, sortiranje podataka u bazi te služi za nadgledanje ispravnosti podataka i osiguranje njihove zaštite.

3. Zapis (slog) je redak u tablici baze podataka u koji se upisuje pojedini subjekt ili objekt. Objedinjuje sva njegova svojstva opisana poljima tablice.

4. Relacijske baze podataka omogućavaju međusobno povezivanje tablica i programsko združivanje podataka iz više njih.

5. U tablicama podataka.

6. Tablicama i obrascima (te određenom vrstom upita).

7. Za upisivanje cijena u polje kao vrstu podatka odabrat ćemo valuta.

8. Tijekom privremenog ili trajnog spremanja, baza podataka rezervira u spremniku prostor sukladno maksimalnoj širini tekstualnog polja, neovisno o stvarno upisanim podacima. Što je najveća dopuštena širina tekstnog polja manja, to će baza zauzeti manje prostora u spremniku.

9. Indeksno polje je stupac u tablici podataka po kojem namjeravamo redati, sortirati podatke tijekom uporabe baze. Nakon što se polje označi kao indeksno, sortiranje podataka po tom polju izvodi se znatno brže.

10. a) s dupliciranjem vrijednosti (*Duplikati U redu*)

11. Upravitelj bazom podataka prijavit će grešku, jer u polju s glavnim ključem ne smije biti dupliciranih vrijednosti.

12. Tipkom Escape.

13. Zapisi koji zadovoljavaju kriterij filtriranja će se prikazati, a ostali ne. U slučaju ovih potonjih to ne znači da su uklonjeni (izbrisani) iz tablice podataka.

14. Obrasci olakšavaju upis podataka jer možemo npr. odrediti redosljed upisa u polja, oblikovati polja i tekst u njima (odrediti poziciju i veličinu polja, odrediti vrstu, veličinu i boju teksta itd.), kontrolirati ispravnost upisa podataka u pojedino polje itd.

15. Pomoću *Čarobnjaka za obrasce* koji u tri koraka vodi kroz kreiranje obrasca.

16. Za uređivanje obrasca potrebno je odabrati način rada pod nazivom *Prikaz dizajna*.

17. Odaberemo *Prikaz dizajna*, kliknemo na praznu površinu obrasca, u okviru *List svojstava* kliknemo na karticu *Sve* i za svojstvo *Opis* upišemo željeni tekst.

18. Najprije uključimo način rada *Prikaz dizajna* i označimo polje *Cijena*. Zatim u *Listu svojstava* odaberemo: *Prednja boja*: #ED1C24 (ili slično); *Stražnja boja*: #FFF200 (ili slično); *Pravnanje teksta*: Desno.

### DODATNI ZADACI

20. U tablici podataka po žuto obojanom čelu trenutno aktivnog retka, a u upisom obrascu po crnom trokutiću u čelu trenutno aktivnog retka.

21. To znači da upravo upisujemo ili uređujemo neko od polja trenutno aktivnog zapisa, odnosno retka.

23. To ne znači da smo pokušali upisati pogrešnu vrstu podatka. Polje ID je tipa *Samonumeriranje* pa u njega nije moguće ručno upisivati podatke.

24. Podatke ćemo sortirati po polju *Vrsta*.

25. Najprije sortiramo podatke po polju *Godina* i to u obrnutom redosljedu (opcija *Silazno*). Zatim u polju *Izvođač* kliknemo na jedno od polja koje sadrži ime Gotan Project. U alatnoj vrpci u grupi *Rasporedi i filtriraj* kliknemo na gumb *Odabir* te iz padajućeg popisa odaberemo opciju *Jednako "Gotan Project"*.

27. U načinu rada *Prikaz obrasca* obrazac se ne može uređivati. Takav način rada predstavlja oblik zaštite od korisnikova hotimičnog ili nehotimičnog mijenjanja oblika i strukture obrasca.

28. To je traka s gumbima za navigaciju kroz zapise: skok na prvi, prijašnji, idući ili zadnji zapis. Osim toga, tu se nalazi gumb za upis novoga zapisa, podatak o tome na kojemu se zapisu trenutačno nalazimo i koliki je ukupan broj zapisa u tablici, indikator stanja uključenosti filtra te okvir za pretraživanje.

29. Prednost je što su polja s podacima pregledno prikazana i što taj prikaz možemo uređivati po našoj želji. Nedostatak je što se prikazuju podaci samo jednoga zapisa.

30. Na slici je prikazan način rada *Prikaz dizajna* koji omogućuje uređivanje svih komponenti izgleda obrasca. Osim što možemo urediti veličinu obrasca, boju njegove pozadine i sl., možemo uređivati i svaku grafičku kontrolu zasebno – njihovu veličinu i poziciju, boju, oblik i veličinu pisma, izvor podataka itd.

31. U *Listu svojstava* potrebno je odabrati: *Prednja boja*: #0072BC (ili slično), *Debljina fonta*: Podebljano (ili slično).

### ZADACI S NATJECANJA

1. c) plošni model

2. b) Obrazac (*Form*)

3. a) Pivot Table

4. a) popularnost, grad i ime

b) popularna imena koja počinju slovom Z

5. b) .accdb; c) .mdb

6. b) Automatski brojač redaka koji osigurava da svaki redak u tablici dobije jedinstveni broj

7. b) indeksiranje

8. c) ograničena veličinom prostora tvrdog diska

9. a) 18; b) 4; c) 3

10. b) slogu

11. a) s dupliciranjem vrijednosti (*Duplicates OK*)

12. b) polje

13. b) Da racionalno trošimo memoriju.

14. b) definiciju kriterija za prikaz podataka

15. a) Database Management System

16. b) relacijska baza podataka

## 6. IZRADA WEB-MJESTA

**1. Ponavljanje**

1. Web-stranica je dinamična tj. sadržaj web stranice se može mijenjati ili nadopunjavati i nakon prvog objavljivanja. Uz tekst, na web stranicu možemo smjestiti slike, video zapise i ostale multimedijalne sadržaje. Za razliku od web-stranica, pisani mediji su statični tj. jednom objavljen (otisnut) sadržaj se ne može mijenjati ili nadopunjavati. Sadržaj pisanih medija je ograničen na tekst i fotografije, bez ostalih multimedijalnih sadržaja. Primjer: web stranica škole i školski časopis koji izlazi 2 puta godišnje.

2. Web-stranica je hipertekstualni multimedijalni HTML dokument smješten na internetu, na nekom javnom poslužitelju.

3. Elementi web-stranice su tekst, slike, animacije, video zapisi. Tekst možemo odmah upisati na web-stranicu a ostale elemente umetnemo na željeno mjesto na web-stranici, pri čemu web-stranica pamti točnu putanju do mjesta gdje je pospremljen neki multimedijalni sadržaj. Npr. slike za web-stranicu možemo pospremiti u podmapu *slike* glavne mape u kojoj smo kreirali našu web-stranicu.

4. Obje slike prikazuju poveznicu (hiperlink), a pokazivač miša poprimio je znak šake (šapica). *Slika 1. a)* pokazuje tekst koji je poveznica, a *slika 1. b)* pokazuje sliku kao poveznicu.

5. .html je nastavak web-stranice napravljene u nekom od programa za izradu web-stranica. Web-mjesto predstavlja internetsku lokaciju koja objedinjuje više tematski međusobno povezanih web-stranica. Web-stranice jednog web-mjesta organizirane su hijerarhijski. Najčešće zadano ime početne stranice web-mjesta je index.html ili default.html.

6. Na slici 6. nedostaje naslov web-stranice u web-pregledniku.

7. Uljez je Google. Google je tražilica a ostali navedeni pojmovi su web-pretraživači.

8. Rješenje: tekst koji nedostaje u zaokruženom dijelu slike je: http://www.vlada.hr. To je točna Internet lokacija web-mjesta Vlada Republike Hrvatske.

|  |
| --- |
| 6_8zad |

9. Microsoft Expression Web kao program za je vizualni HTML uređivač a pokrećemo ga

10. Tri pogleda u Expression Webu: *Design*, *Split*, *Code*.

11.

|  |  |
| --- | --- |
| 1. Traka s izbornicima | 6. Izbornik prikaza |
| 2. Alatna traka | 7. Okno mapa |
| 3. Kartica pojedine web-stranice | 8. Okno svojstva |
| 4. Točka unosa | 9. Okno s HTML naredbama i stilovima |
| 5.Radni prostor | 10. Statusna traka |

12. Svaka web-stranica sastoji se od više HTML blokova a osnovni su: <html>, <head>, <body>. Svaki od navedenih HTML blokova ima i pripadajući HTML tag za zatvaranje: </html>, </head>, </body>.

13. Rješenje za sliku:

|  |
| --- |
| 6_13zad |

Drugi način za kreiranje nove web-stranice koristeći izbornik: *File* > *New* > *Page…*

14.

|  |
| --- |
| a) Slične mogućnosti oblikovanja teksta dostupne su u MS Wordu. |
| b) Na naslovnoj traci web-preglednika se prikazuje naslov stranice tj. tekst koji se nalazu između <title> i </title> HTML tagova web-stranice. Naslov web-stranice i naziv web stranice pod kojim je spremljena ne moraju biti isti. |
| c) Podaci u dijaloškom prozoru *Accessibility Properties* potrebni su kao informacija korisniku u slučaju da se slike iz nekog razloga ne prikaže u Internet pregledniku. U tom slučaju, korisniku se prikazuje alternativni tekst koji ga može informirati o slici koja se nije učitala. |
| d) Poveznice nam služe za navigaciju kroz hipertekstualne dokumente tj. za povezivanje web-stranica međusobno i povezivanje na druge dokumente (fotografije, tekstualne dokumente, multimediju…) |
| e) Kartica Site View > Hyperlinks |
| f) Web-stranice se mogu ispisivati ali u pravilu nisu namijenjene za ispis na pisaču. Zbog prevelikih slika i ostalih sadržaja web-stranice, prilikom ispisa na pisaču može doći do razlamanja stranice. Neke web-stranice imaju posebnu verziju (uglavnom tekst, bez slika) koja je prilagođena za ispis. |
| g) Osim promjene boje, pozadinu web-stranice možemo odrediti tako da kao pozadinu web-stranice postavimo sliku. S postavljanjem slika kao pozadine web-stranica treba biti umjeren jer velike slike uvelike povećavaju vrijeme učitavanja web-stranice i mogu odvratiti korisnika. Kao pozadinske slike obično se postavljaju slike malih dimenzija, u .gif ili .png formatu, koje se zatim automatski slažu jedna do druge i tako čine privid jedne velike slike. Na taj način preglednik brzo učita malu sliku i vrijeme učitavanja web-stranice je kratko. |

2. Timska izrada web-mjesta

15. Rješenje križaljke:

|  |  |
| --- | --- |
| **Okomito:** | **Vodoravno:** |
| 1. domena | 2. timu |
| 3. index | 6. website |
| 4. url | 8. struktura |
| 5. hijerarhija | 9. dizajner |
| 7. autor | 11. poveznica |
| 10. član | 12. slika |
| 13. ie |  |

2.1. Izrada web-mjesta u Expression Webu

16. Na slici 5. prikazana je skica budućeg web-mjesta. Na njoj su vidljive sve web-stranice, točan naziv svake web-stranice i njihov međusoban odnos (hijerarhija).

17. Načini kreiranja web-mjesta u Expression Webu:

|  |
| --- |
| Način 1: Odabirom izbornika *Site > New Site…* |
| Način 2: Klikom na ikonicu new_site.png na alatnoj traci. |

18. Rješenje:

|  |
| --- |
| Način 1: 6_18zad |
| Način 2: Upotrebom kombinacije tipki Ctrl + Shift + N na tipkovnici. |

19. Predloškom je određen dizajn, stil slova, područje navigacijskoga izbornika i slika pa se time znatno skraćuje vrijeme izrade web-mjesta i uređivanje web-stranica. Osim toga, stranice dijele zajednički izgled (dizajn), što doprinosi vizualnome identitetu web-mjesta.

20. Označeni dijelovi prozora *New*, slika 6:

|  |
| --- |
| 1. Odabrani predložak naziva *Organization 3* |
| 2. Predpregled (*Prieview*) tj. izgled predloška u pregledniku |
| 3.Mjesto na lokalnom disku računala gdje je pospremljeno web-mjesto |
| 4. Naziv web-mjesta |

21. Rješenje (zaokružena ikona): 6_21zad

22. Datoteka *master.dwt*  je datoteka predloška u Expression Webu. U datoteci predloška su definirane sve postavke izgleda web-mjesta na osnovu kojih nastaju sve ostale pripadajuće web-stranice. Nastavak *.dwt* označava *Dynamic Web Template.* Datoteka *default.html* je početna stranica tj. datoteka koju web-preglednik učitava kada posjetimo neko web-mjesto. Takva početna stranica još se naziva *home* ili *indeksna stranica* i najčešće se sprema pod nazivom *index.html*.

23. Postoje 3 osnovne vrste poveznica:  
1) poveznice na vanjsku lokaciju tj. na drugu web-stranicu ili dokument   
2) poveznice na mjesto trenutne web-stranice (npr. na vrh, sredinu ili podnožje web-stranice)  
3) poveznice na neku novu web-stranicu dokument koju ćemo kreirati.

24. Ikona označava mapu web-mjesta (*Site*).

25. Razlike prilikom spremanja:

|  |
| --- |
| Save – sprema sve promjene u trenutnom dokumentu. |
| Save As… - sprema dokument pod novim nazivom. |
| Save All – sprema sve promjene na svim trenutno otvorenim dokumentima. |

26. Slika 7. prikazuje uspješno završen postupak osvježavanja 8 datoteka na web poslužitelju.

27. .gif je nastavak ­­ slikovne datoteke napravljene u nekom od programa za izradu animacija. Dvije ili više sličica prikazuju se jedna za drugom u određenom vremenskom slijedu. Easy Gif animator je program za animiranje statičkih slika.

28. Na kvalitetu animacije utječe kvaliteta slika koje čine animaciju, broj kadova (*frames*) u animaciji i primijenjeni efekti na animaciji.

29. Lokalno web-mjesto je skup web-stanica pohranjenih na korisničkom računalu.

30. Lokalno web-mjesto potrebno je objaviti (web-publiciranje) na javno dostupan web-poslužitelj. Takva lokacija na javnom web-poslužitelju ima svoj URL te takvo web-mjesto postaje dostupno svim korisnicima koji imaju pristup internetu.

31. Na slici 8. prikazan je odabir postavki programa Expression Web za objavljivanje web-mjesta. Do navedene opcije dolazimo odabirom *Site > Publishing Settings…*

32. Standardni način prijenosa podataka sa lokalnog računala na web-poslužitelj je upotreba nekog od FTP (*File Transfer Protocol*) programa, pri čemu podatke kopiramo sa lokalnog računala na udaljeni poslužitelj.

33.

|  |  |
| --- | --- |
| 6_33zad-lijevo | 6_33zad-desno2 |
| 6_33zad-desno5 |
| 6_33zad-desno3 |
| 6_33zad-desno4 |
| 6_33zad-desno1 |

ZADACI S NATJECANJA

1. b) Prikaz > Izvor (View > Source)

2. c) www

3. c) .jpeg i. gif

4. b) referencu naše web stranice

5. d) Sustav koji slovne adrese računala pretvara u brojčane IP.

6. a) Slikovna mapa

7. b) FTP Enterprise

8. c) FTP

9. b) Bookmarks

10. d) index.html

11. b) POP

12. a) web adresa određenog resursa na internetu.

13.

|  |  |
| --- | --- |
| </body> | Oznaka kraja tijela stranice |
| <title> | Oznaka početka naslova stranice |
| <head> | Oznaka početka zaglavlja stranice |
| <body> | Oznaka početka tijela stranice |
| </head> | Oznaka kraja zaglavlja stranice |

14. a) Mb/s

15. a) korisničko-poslužiteljski odnos računala

16.

|  |
| --- |
| Početak novog odlomka: <p> |
| Prelazak u novi red: <br> |

17. Rješenje: <font style="font-family:Verdana; color:blue; font-size:6px;">

18. c) mjesto u trenutnom dokumentu.

## 7. ODNOS PREMA INTERNETU

**1. Kritički odnos prema internetu**

1. Naziv pojma internet potječe iz engleskoga jezika i u njemu se piše Internet. U hrvatskome je u praksi potvrđeno pisanje i Internet i internet. Iako je internet jedna i jedinstvena mreža, ipak nije riječ o određenoj tvrtki koja bi imala i svojega osnivača, sjedište, osoblje te taj naziv ne treba pisati velikim slovom. Internet označuje tehnologiju i način komuniciranja i stoga taj naziv treba pisati malim slovom.

2. a) http://www.google.hr/

3. Na slici 1A prikazana je domena *www.google.de*  tj. početna Google stranica za Njemačku, pri čemu je u postavkama definirano da rezultati pretraživanja budu na hrvatskom jeziku.  
Slika 1B prikazuje domenu *www.google.hr* tj. početnu Google stranicu za Hrvatsku. Rezultati pretraživanja se prikazuju na hrvatskom jeziku.

4. Web-stranicama se pristupa pomoću: d) http protokola.

5. Za pretraživanje interneta najčešće se koriste­ tražilice. Pri sastavljanju upita koristimo se logičkim operatorima i filtrima. Rezultate pretraživanja tražilice pružaju velikom brzinom.

Web-katalozi omogućavaju pronalaženje informacija razvrstanih prema određenim kategorijama. Metatražilice pokreću pretraživanje istovremenom upotrebom većega broja tražilica.

6. Osnovne karakteristike nevidljivog web-a: to su web-mjesta i stranice koje ostaju van dohvata poznatih tražilica tj. ne indeksiraju se na tražilicama pa ih tražilice ne prikazuju u rezultatima pretrage. Neki od razloga zašto se takve stranice ne indeksiraju: web-stranice su „zakopane“ duboko i na njih ne vode poveznice sa drugih web-stranica, stranice se stvaraju dinamički tj. ne postoje dok korisnik ne obavi neku aktivnost (npr. pretraživanje baze podataka po kriterijima, nakon čega se tek generira HTML stranica sa rezultatima), web-mjesta i stranice koje traže prijavu korisnika za pregledavanje sadržaja… Procjenjuje se da je nevidljivi web barem deset puta veći od javno dostupnog web-a.

8. Web-stranica na slici 2. pripada u skupinu tzv. Wikipedija web-stranica. Takve stranice mogu stvarati i uređivati svi korisnici što povećava mogućnost netočnih ili nepotpunih informacija. Informacije koje pronađemo na wikipediji preporučljivo je provjeriti na barem još jednom izvoru (npr. enciklopediji, web-stranicama neke relevantne ustanove ili institucije).

9. Povežite pojmove:

**.com**

**.hr**

**.edu**

**.ca**

**.at**

**.org**

**.gov**

Komercijalne domene

Domena države - Hrvatska

Obrazovne ustanove

Domena države - Kanada

Domena države - Austrija

Organizacijske domene

Domene državnih institucija

10. Elementi koji povećavaju vjerodostojnost informacija su starost informacije tj. dan i godinu objave, ime ili naziv autora ili izvora informacije, poziciju rezultata pretrage na tražilicama, tematski sadržaj i mogućnost povrede autorskih prava.

11. Jedan od značajnih načina je i provjera vršne adrese (URL-a) na kojoj je objavljena neka web-stranica. To postižemo tzv. skraćivanjem pune adrese gledajući vršnu adresu. Nepoznata vršna adresa ukazuje na mogućnost netočnih informacija na podstranici.

12. Znak ~ (tilda)u adresi web-stranice ukazuje nam na mogućnost da se radi o privatnoj web-stranici korisnika neke vršne institucije. Npr. učenik može imati svoju web-stranicu oblika www.skola.hr/~ucenik. Informacije na takvim stranicama poželjno je provjeriti na još nekim izvorima.

13. Engleski nazivi:  
a) web-stranica: web-page  
b) web-mjesto: web-site .

14. Znak ©(copyright) na slici daje nam do znanja da je slika vjerojatno zaštićena kao nečije djelo. O uporabi takvih dokumenata potrebno se dodatno informirati.

15. Rezultati pretraživanja na web-stranici http://scholar.google.hr/scholar su specfičniji i odnose se na studije, stručne radove i akademska djela gdje se spominje pojam „velebitska degenija“. Takve izvore informacija možemo uglavnom smatrati vjerodostojnima.   
Pretraživanje pojma velebitska degenija u tražilici prikazuje nam rezultate onako koko ih je tražilica indeksirala po važnosti, sukladno vlastitim definiranim kriterijima.

2. Internetski projekti

16. Točni odgovori:   
a) Projekt je rad u grupi s ciljem rješavanju nekog problema  
d) Osobe koje sudjeluju na projektu čine projektni tim

17. Spajanje pojmova:

**Sponzorirani projekti**

**Internetski projekti**

**Volonterski projekti**

**Komercijalni projekti**

Financirani su projekti od strane nekog ulagača (jedne osobe ili više njih, neke tvrtke ili ustanove).

Koriste internet kao medij i sredstvo komunikacije. Obično se radi o povezivanju timova iz više zemalja.

Dostupni su osobama koje za svoj rad na projektu ne traže materijalnu naknadu.

Sudjeluju osobe koje imaju materijalni ili drugi interes. Znakoviti za tvrtke koje pozivaju pojedince na suradnju.

19. Zaokružite istinite tvrdnje za pojam Internetski oblak:  
a) Način suradnje i dijeljenje podataka i informacija s drugim korisnicima  
c) U Internet oblacima postoje besplatne usluge i one koje se naplaćuju  
e) Mi (korisnici) sa svoje strane ne trebamo ni prostor, ni struju, ni klimatizaciju, ni osoblje koje će sve to održavati.

20. *Google Drive* je jedna od usluga Internet oblaka koja nam omogućuje kreiranje dokumenata (dokumenti, slike, prezentacije,grafovi… ), spremanje dokumenata, dijeljenje tih dokumenata s ostalim korisnicima.

21. Slika 4. Prikazuje mežusobnu komunikaciju, razmjenu i pohranu podataka, dijeljenje podataka između korisnika. Na oblak se korisnici mogu spojiti koristići ne samo osobna računala već i druge uređaje koji imaju mogućnost spajanja na internet (mobilne telefone, tablete…).

22. Neki od najpoznatijih oblaka: Google Apps, SkyDrive, Amazon Cloud Drive, Dropbox Dropbox, iCloud, AppNexus, GoGrid, Salesforce.

3. Školski Internet projekti

23. Članovi školskog tima su učenici škole, u pravilu učenici jednog školskog razreda.

24. Rad u timu predstavlja zajedničko nastojanje grupe da ostvari zadani cilj. Bez međusobne komunikacije, članovi tima ne bi mogli razumjeti koje zadatke moraju obaviti pa bi ubrzo došlo do zastoja u radu tima.

25. Mogući neuspjeh projekta: nerealni ciljevi (nešto što je teško ili gotovo nemoguće ostvariti), premalo vremena, premalo članova, loša međusobna komunikacija članova tima, loša organizacija i raspodjela vremena.

26. Povezivanje pojmova:

**Voditelj tima**

**Promatrač**

**Kontrolor**

**Menadžer vremena**

Financirani su projekti od strane nekog ulagača (jedne osobe ili više njih, neke tvrtke ili ustanove).

Koriste internet kao medij i sredstvo komunikacije. Obično se radi o povezivanju timova iz više zemalja.

Dostupni su osobama koje za svoj rad na projektu ne traže materijalnu naknadu.

Sudjeluju osobe koje imaju materijalni ili drugi interes. Znakoviti za tvrtke koje pozivaju pojedince na suradnju.

DODATNI ZADACI

33. Uljez je ikonag.png koja predstavlja Google uslugu pretraživanja tj. početnu Google stranicu. Ostale ikone u nizu su ikone različitih internet preglednika.

ZADACI S NATJECANJA

1. Odaberi točno napisanu e-mail adresu: e) [mirko.bukic@skole.hr](mailto:mirko.bukic@skole.hr).

2. c) Serverska veza

3. c) „Rage Against the Machine“.

4. c) Serverska veza

5. c) usmjernik (router).

6. d) Windows Security Essentials.

7. a) Use All (Koristi sve).

8. a) privremeno pohranjivanje nedovršene pošte.

9. a) ACTA (Anti-Counterfeiting Trade Agreement)- multinacionalni ugovor uspostave međunarodnih standarda za provedbu prava intelektualnog vlasništva.

10. c) web katalog.

11. ARPANET

12. c) gav

13. IP adrese koje davatelj internetskih usluga dodjeljuje računalu iz grupe adresa pri svakom spajanju na Internet zovu se: dinamičke IP adrese.

14. Sigurna verzija http protokola je: https.

15. a) Wide.