

7. Zadaci

7.1. Rešeni zadaci

1. Napisati program za izračunavanje rastojanja i direkcionog ugla, ako su date koordinate tačaka A(x₁, y₁) i B(x₂, y₂).

Računati po formulama:

$$\Delta x = x_2 - x_1, \Delta y = y_2 - y_1, R = \sqrt{(\Delta x)^2 + (\Delta y)^2},$$
$$\phi = \begin{cases} \arctg \frac{\Delta y}{\Delta x}, & (\Delta x > 0, \Delta y \geq 0), \\ \pi + \arctg \frac{\Delta y}{\Delta x}, & (\Delta x < 0), \\ 2\pi + \arctg \frac{\Delta y}{\Delta x}, & (\Delta x > 0, \Delta y < 0), \\ \frac{\pi}{2}, & (\Delta x = 0, \Delta y > 0), \\ \frac{3\pi}{2}, & (\Delta x = 0, \Delta y < 0). \end{cases}$$

Slučaj $\Delta x = \Delta y = 0$ ne razmatrati. Slučaj $|\Delta x| < 10^{-6}$ tretirati kao $\Delta x = 0$. Ugao dobijen u radijanima pretvoriti u stepene, minute i sekunde.

Rezultate štampati u obliku:

X1	Y1	X2	Y2	R	STEP	MIN	SEC
xxx.xxx	xxx.xxx	xxx.xxx	xxx.xxx	xxx.xxx	xxx.xxx	xxx.xxx	xxx.xxx
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Rešenje:

```
import java.io.*;
import java.text.*;
```

```
public class Rastojanje
{
```

```
public static void main(String[] args) throws Exception
{
    BufferedReader ulaz = new BufferedReader(new InputStreamReader(System.in));
    System.out.println("Unesite broj koordinata:");
    int n = Integer.parseInt(ulaz.readLine());
    double[] x1 = new double[10];
    double[] y1 = new double[10];
    double[] x2 = new double[10];
    double[] y2 = new double[10];

    System.out.println("\nUnesite koordinate:");
    for(int i = 1; i <= n; i++)
    {
        System.out.print("x1["+i+"]=");
        x1[i] = Double.parseDouble(ulaz.readLine());
        System.out.print("y1["+i+"]=");
        y1[i] = Double.parseDouble(ulaz.readLine());
        System.out.print("x2["+i+"]=");
        x2[i] = Double.parseDouble(ulaz.readLine());
        System.out.print("y2["+i+"]=");
        y2[i] = Double.parseDouble(ulaz.readLine());
    }

    System.out.println("\nX1\tY1\tX2\tY2\tR\tSTEP\tMIN\tSEC");
    System.out.println("=====\n");

    DecimalFormat df=new DecimalFormat("###,##0.00");

    for(int i = 1; i <= n; i++)
    {
        double dx = x2[i]-x1[i];
        double dy = y2[i]-y1[i];
        double r = Math.sqrt(dx*dx+dy*dy);
        double fi = 0.0;
        if(Math.abs(dx)<1.0e-6)
            if(dy>0)
                fi = Math.PI/2;
            else
                fi = 3*Math.PI/2;
        else if(dx<0)
            fi = Math.PI+Math.atan(dy/dx);
        else if(dy<0)
            fi = 2*Math.PI+Math.atan(dy/dx);
        else
            fi = Math.atan(dy/dx);

        double step = fi*180.0/Math.PI;
        int step1=(int) step;
    }
}
```

7. Zadaci

```
double amin=(step-step1)*60.0;
int min1=(int) amin;
double sec=(amin-min1)*60.0;
int isec=(int) sec;
```

```
        System.out.println(x1[i]+"\\t"+y1[i]+"\\t"+x2[i]+"\\t"+y2[i]+"\\t"+df.format(r)+"\\t"+step1+"\\t"
+min1+"\\t"+isec);
    }
}
```

Izlaz:

Unesite broj koordinata:
5

Unesite koordinate:

```
x1[1]=1.5
y1[1]=1.0
x2[1]=3.0
y2[1]=3.0
x1[2]=-3.0
y1[2]=2.0
x2[2]=2.0
y2[2]=5.0
x1[3]=-2.0
y1[3]=1.5
x2[3]=-8.0
y2[3]=6.5
x1[4]=-3.0
y1[4]=-5.0
x2[4]=-11.0
y2[4]=-24.0
x1[5]=2.0
y1[5]=-3.0
x2[5]=7.77
y2[5]=-8.88
```

X1	Y1	X2	Y2	R	STEP	MIN	SEC
1.5	1.0	3.0	3.0	2.50	53	7	48
-3.0	2.0	2.0	5.0	5.83	30	57	49
-2.0	1.5	-8.0	6.5	7.81	140	11	39
-3.0	-5.0	-11.0	-24.0	20.62	247	9	58
2.0	-3.0	7.77	-8.88	8.24	314	27	32

2. Sastaviti program za tabeliranje vrednosti koeficijenta za izračunavanje torzione konstante pravougaonog preseka:

$$\alpha = \frac{1}{3} \frac{b}{a} - \frac{64}{\pi^5} \sum_{n=0}^{+\infty} \frac{\tanh \frac{\pi(2n+1)b}{2a}}{(2n+1)^5},$$

gde je a manja stranica pravougaonika, za $b/a = 1(0.1)10.9$. Proces sumiranja za svako α prekinuti kada opšti član reda postane manji od 10^{-3} . Izlazna lista neka izgleda:

TABELA TORZIONE KONSTANTE

B/A	0.0	0.1	0.2	...	0.9
1	x.xxx	x.xxx	x.xxx	...	x.xxx
2	x.xxx	x.xxx	x.xxx	...	x.xxx
⋮	⋮	⋮	⋮	...	⋮
10	x.xxx	x.xxx	x.xxx	...	x.xxx

Rešenje:

```
import java.text.*;

public class Torzija
{
    public static void main(String[] args)
    {
        double[] Alfa = new double[11];
        double ba, p, a, s;
        int n;

        System.out.println("TABELA TORZIONE KONSTANTE\n");
        System.out.println("B/A      0.1      0.2      0.3      0.9");

        double eps=1.0e-4;
        for(int i=1;i<=10;i++)
        {
            for(int j=1;j<=10;j++)
            {
                ba = i+(j-1)*0.1;
                n = 0;
                p = (2*n+1)*Math.PI*ba/2;
                a = ((Math.exp(p)- Math.exp(-p))/ (Math.exp(p) +Math.exp(-p)))/Math.pow(2*n+1, 5);
                s = a;

                while (Math.abs(a)<=eps)
                {
                    n = n + 1;
                    p = (2*n+1)*Math.PI*ba/2;
                    a += ((Math.exp(p)- Math.exp(-p))/ (Math.exp(p) +Math.exp(-p)))/Math.pow(2*n+1, 5);
                }
            }
        }
    }
}
```

7. Zadaci

```
        s +=a;
    }

    Alfa[j]=ba/3-64.0*a/Math.pow(Math.PI, 5)*s;
}

System.out.print(i);
DecimalFormat df=new DecimalFormat("###,###0.000");
for(int j=1;j<=10;j++)
{
    System.out.print(" "+ df.format(Alfa[j]));
}
System.out.print("\n");
}
}
}
```

Izlaz:

TABELA TORZIONE KONSTANTE

B/A	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
1	0.157	0.182	0.209	0.238	0.267	0.298	0.329	0.361	0.394	0.426
2	0.459	0.492	0.525	0.558	0.591	0.624	0.658	0.691	0.724	0.758
3	0.791	0.824	0.857	0.891	0.924	0.957	0.991	1.024	1.057	1.091
4	1.124	1.157	1.191	1.224	1.257	1.291	1.324	1.357	1.391	1.424
5	1.457	1.491	1.524	1.557	1.591	1.624	1.657	1.691	1.724	1.757
6	1.791	1.824	1.857	1.891	1.924	1.957	1.991	2.024	2.057	2.091
7	2.124	2.157	2.191	2.224	2.257	2.291	2.324	2.357	2.391	2.424
8	2.457	2.491	2.524	2.557	2.591	2.624	2.657	2.691	2.724	2.757
9	2.791	2.824	2.857	2.891	2.924	2.957	2.991	3.024	3.057	3.091
10	3.124	3.157	3.191	3.224	3.257	3.291	3.324	3.357	3.391	3.424

3. U klasi Complex za rad sa kompleksnim brojevima definisana su dva javna člana realan i imaginaran tipa double. Napisati funkcije članice klase Complex:

- `public void stampa()` – za štampanje kompleksnog broja,
- `public Complex mnozi(double k)` – za množenje kompleksnog broja skalarom k,
- `public double apsolutna()` – za određivanje apsolutne vrednosti kompleksnog broja,
- `public Complex saberi(Complex c)` – za sabiranje kompleksnih brojeva,
- `public Complex mnozi(Complex c)` – za množenje kompleksnih brojeva,
- `public Complex oduzmi(Complex c)` – za oduzimanje kompleksnih brojeva.

U glavnom programu za unete kompleksne brojeve $z1 = 3 + 4i$, $I = i$, $z6 = 2 + i$ i konstantu $k = 7$ izračunati $z2 = k * z1$, $z3 = z1 + z2$, $z4 = z1 * z2$, $z5 = I^2$, $z6 * z6$, $z7 = z1 - z2$.

Rešenje:

```
/**
 * Zadatak sastavio
 * Dr.-Ing. Karl-Heinz Lehner, Katedra za inženjersku informatiku, Ruhr-Univerzitet Bochum
 */
public class Complex
{
    public double realan;
    public double imaginaran;
    public void stampa()
    {
        if(imaginaran==0)
        {
            System.out.println(realan);
        }
        else
        {
            System.out.println(realan+" "+imaginaran+"*i");
        }
    }

    public Complex mnozi(double k)
    {
        Complex rezultat=new Complex();
        rezultat.realan=k*this.realan;
        rezultat.imaginaran=k*this.imaginaran;
        return rezultat;
    }

    public double apsolutna()
    {
        double rezultat=realan*realan+imaginaran*imaginaran;
        return Math.sqrt(rezultat);
    }

    public Complex saberi(Complex c)
    {
        Complex z=new Complex();
        z.realan=this.realan+c.realan;
        z.imaginaran=this.imaginaran+c.imaginaran;
        return z;
    }

    public Complex mnozi(Complex c)
    {
        Complex z=new Complex();
        z.realan=this.realan*c.realan-this.imaginaran*c.imaginaran;
```

7. Zadaci

```
z.imaginaran=this.realan*c.imaginaran+this.imaginaran*c.realan;
return z;
}

public Complex oduzmi(Complex c)
{
    Complex z=new Complex();
    z.realan=this.realan-c.realan;
    z.imaginaran=this.imaginaran-c.imaginaran;
    return z;
}
}

public class Test{

public static void main(String[] args)
{
    Complex z1=new Complex();
    z1.realan=3;
    z1.imaginaran=4;
    System.out.print("z1= ");
    z1.stampa();

    Complex z2=z1.mnozi(7);
    System.out.print("z2=z1*7= ");
    z2.stampa();

    Complex z3=z1.saberi(z2);
    System.out.print("z3=z1+z2= ");
    z3.stampa();

    Complex z4=z1.mnozi(z2);
    System.out.print("z4=z1*z2= ");
    z4.stampa();

    Complex I=new Complex();
    I.realan=0;
    I.imaginaran=1;
    Complex z5=I.mnozi(I);
    System.out.print("z5=I^2= ");
    z5.stampa();

    Complex z6=new Complex();
    z6.realan=2;
    z6.imaginaran=1;
    System.out.print("koren iz 2+i= ");
    (z6.mnozi(z6)).stampa();

    Complex z7=z1.oduzmi(z2);
```

```
System.out.print("z7=z1-z2= ");
z7.stampa();
    }
}
```

Izlaz:

```
z1= 3.0+4.0*i
z2=z1*7= 21.0+28.0*i
z3=z1+z2= 24.0+32.0*i
z4=z1*z2= -49.0+168.0*i
z5=I^2= -1.0
koren iz 2+i= 3.0+4.0*i
z7=z1-z2= -18.0+-24.0*i
```

4. Napisati klasu Vektor koja sadrži dva javna člana x i y tipa double i funkcije članice:

- `public void stampa()` – za štampanje elemenata vektora,
- `public Vektor saberi(Vektor v)` – za sabiranje elemenata vektora,
- `public Vektor mnozi(double a)` – za množenje elemenata vektora skalarom,
- `public double mnozi(Vektor v)` – za sabiranje dva vektora,
- `public boolean ortogonalan(Vektor v)` – za ispitivanje ortogonalnosti vektora v,
- `public double duzina()` – za određivanje dužine vektora.

U glavnom programu za unete elemente vektora $v1 = \{3, 4\}$ i $v2 = \{5, 6\}$ izračunati $v3 = v1 + v2$, $v4 = v1 * v2$, ispitati ortogonalnost vektora v1 i v2 i naći dužinu vektora v1.

Rešenje:

```
/**
 * Zadatak sastavio
 * Dr.-Ing. Karl-Heinz Lehner, Katedra za inženjersku informatiku, Ruhr-Univerzitet Bochum
 */
public class Vektor
{
    public double x;
    public double y;

    public void stampa()
    {
        System.out.println("x="+x);
        System.out.println("y="+y);
    }

    public Vektor saberi(Vektor v)
    {
        Vektor sum=new Vektor();
        sum.x=this.x+v.x;
    }
}
```


7. Zadaci

```
        sum.y=this.y+v.y;
        return sum;
    }

    public Vektor mnozi(double a)
    {
        Vektor prod=new Vektor();
        prod.x=a*this.x;
        prod.y=a*this.y;
        return prod;
    }

    public double mnozi(Vektor v)
    {
        double sum;
        sum=this.x*v.x+this.y*v.y;
        return sum;
    }

    public boolean isOrtho(Vektor v)
    {
        return (this.mnozi(v)==0);
    }

    public double duzina()
    {
        return Math.sqrt(this.mnozi(this));
    }

    public static void main(String[] args)
    {
        Vektor v1=new Vektor();
        v1.x=3;
        v1.y=4;
        System.out.println("Vektor v1:");
        v1.stampa();

        Vektor v2=new Vektor();
        v2.x=5;
        v2.y=6;
        System.out.println("Vektor v2:");
        v2.stampa();

        Vektor v3=new Vektor();
        v3.x=v1.x+v2.x;
        v3.y=v1.y+v2.y;
        System.out.println("Zbir vektora v1 i v2 je");
        v3.stampa();
    }
}
```

```
Vektor v4=v1.saberi(v2);
System.out.println("Zbir vektora v1 i v2 je");
v4.stampa();

System.out.println("Skalarni proizvod vektora v1 i v2 je "+v1.mnozi(v2));
System.out.println("Da li je v1 ortogonalan? "+v1.isOrtho(v2));

System.out.println("Duzina vektora v1 je "+v1.duzina());
    }
}
```

Izlaz:

```
Vektor v1:
x=3.0
y=4.0
Vektor v2:
x=5.0
y=6.0
Zbir vektora v1 i v2 je
x=8.0
y=10.0
Zbir vektora v1 i v2 je
x=8.0
y=10.0
Skalarni proizvod vektora v1 i v2 je 39.0
Da li je v1 ortogonalan? false
Duzina vektora v1 je 5.0
```

5. Napisati klasu Vektor3 koja sadrži privatni član val[3] tipa double i funkcije članice:

- `public Vektor3(double x1, double x2, double x3)` – konstruktor klase Vektor3,
- `public void stampa()` – za štampanje elemenata vektora,
- `public Vektor3 saberi(Vektor3 v)` – za sabiranje elemenata vektora,
- `public Vektor3 mnozi(double a)` – za množenje elemenata vektora skalarom,
- `public double mnozi(Vektor3 v)` – za sabiranje dva vektora,
- `public boolean ortogonalan(Vektor3 v)` – za ispitivanje ortogonalnosti vektora v,
- `public double duzina()` – za određivanje dužine vektora.

U glavnom programu za unete elemente vektora $v1 = \{1, 2, 3\}$ i $v2 = \{8, -1, -2\}$ izračunati $v3 = v1 + v2$, $v4 = v1 * v2$, ispitati ortogonalnost vektora v1 i v2 i naći dužinu vektora v1.

Rešenje:

```
/**
 * Zadatak sastavio
 * Dr.-Ing. Karl-Heinz Lehner, Katedra za inženjersku informatiku, Ruhr-Univerzitet Bochum
 */
```

7. Zadaci

```
public class Vektor3
{
    private double val[]=new double[3];

    public Vektor3(double x1, double x2, double x3)
    {
        val[0]=x1;
        val[1]=x2;
        val[2]=x3;
    }

    public Vektor3(double[] x)
    {
        val=x;
    }

    public Vektor3 saberi(Vektor3 v)
    {
        double[] sum=new double[3];
        for(int i=0;i<3;i++)
        {
            sum[i]=this.val[i]+v.val[i];
        }
        return new Vektor3(sum);
    }

    public void stampa()
    {
        for(int i=0;i<3;i++)
        {
            System.out.println("val["+i+"]="+val[i]);
        }
    }

    public Vektor3 mnozi(double a)
    {
        double x[]=new double[3];
        for(int i=0;i<3;i++)
        {
            x[i]=a*val[i];
        }
        return new Vektor3(x);
    }

    public double mnozi(Vektor3 v)
    {
        double sum=0;
        for(int i=0;i<3;i++)
```

```
{
    sum+=this.val[i]*v.val[i];
}
return sum;
}

public boolean ortogonalan(Vektor3 v)
{
    return (mnozi(v)==0);
}

public double duzina()
{
    return Math.sqrt(mnozi(this));
}

public static void main(String[] args)
{
    Vektor3 v1=new Vektor3(1,2,3);

    double[] x=new double[]{4,5,6};
    Vektor3 v2=new Vektor3(x);
    Vektor3 v3=v1.saberi(v2);
    v3.stampa();

    v1=new Vektor3(new double[]{1,2,3});
    v2=new Vektor3(new double[]{8,-1,-2});
    System.out.println("Skalarni proizvod vektora v1 i v2 je "+v1.mnozi(v2));
    System.out.println("Vektor v1 je ortogonalan? "+v1.ortogonalan(v2));

    System.out.println("Duzina vektora v1 je "+v1.duzina());
}
}
```

Izlaz:

```
val[0]=5.0
val[1]=7.0
val[2]=9.0
Skalarni proizvod vektora v1 i v2 je 0.0
Vektor v1 je ortogonalan? true
Duzina vektora v1 je 3.7416573867739413
```

6. Napisati klasu VektorN koja sadrži privatni član val[] tipa double i funkcije članice:

- public void stampa() – za štampanje elemenata vektora,
- public VektorN saberi(VektorN v) – za sabiranje elemenata vektora.

7. Zadaci

U glavnom programu za unete elemente vektora $v1 = \{1, 2, 3, 4\}$, $v2 = \{5, 6, 7, 8\}$ i $v4 = \{10, 20, 30, 40\}$ izračunati $v3 = v1 + v2$, $v3 + v4$.

Rešenje:

```
/**
 * Zadatak sastavio
 * Dr.-Ing. Karl-Heinz Lehner, Katedra za inženjersku informatiku, Ruhr-Univerzitet Bochum
 */
public class VektorN
{
    private double val[];
    public VektorN(double[] x)
    {
        val=x;
    }

    VektorN saberi(VektorN v)
    {
        int velicina = val.length;
        double[] sum=new double[velicina];
        for(int i=0; i<velicina; i++)
        {
            sum[i]=this.val[i]+v.val[i];
        }
        return (new VektorN(sum));
    }

    void stampa()
    {
        int velicina=val.length;
        for(int i=0; i<velicina; i++)
        {
            System.out.println("i= "+i+" val= "+val[i]);
        }
    }

    public static void main(String[] args)
    {
        double[] x1={1,2,3,4};
        double[] x2={5,6,7,8};
        VektorN v1=new VektorN(x1);
        VektorN v2=new VektorN(x2);

        VektorN v3=v1.saberi(v2);
        v3.stampa();
        VektorN v4=new VektorN(new double[] {10,20,30,40});
        v3=v3.saberi(v4);
        v3.stampa();
    }
}
```

Izlaz:

```
i= 0 val= 6.0
i= 1 val= 8.0
i= 2 val= 10.0
i= 3 val= 12.0
i= 0 val= 16.0
i= 1 val= 28.0
i= 2 val= 40.0
i= 3 val= 52.0
```

7. Izračunati integral

$$I = \int_1^3 \frac{\arctan \frac{x^2 + 1}{x}}{1.2 e^x + \cos(x)} dx ,$$

Simpsonovim pravilom za $n = 20$. Podintegralnu funkciju izračunati potprogramom. Izlazna lista neka izgleda

VREDNOST INTEGRALA = xx.xxx.

Rešenje:

```
import java.text.*;

public class Integral
{

    public static double f1( double x)
    {
        return Math.atan((x*x+1)/x)/(1.2*Math.exp(x)+Math.cos(x));
    }

    public static double integral(double a, double b,int n)
    {
        double sum=0;
        double h=(b-a)/n;
        for(int i=0;i<n;i=i+2)
            sum=sum+(f1(a+i*h)+4*f1(a+(i+1)*h)+f1(a+(i+2)*h))*h/3;
        return sum;
    }

    public static void main(String[] args)
    {
        DecimalFormat df=new DecimalFormat("###,###0.000");
        System.out.println("VREDNOST INTEGRALA = "+df.format(integral(1,3,20)));
    }
}
```

7. Zadaci

Izlaz:

VREDNOST INTEGRALA = 0.304

8. Napisati program za realizaciju Gaussovog metoda eliminacije sa izborom glavnog elementa.

Rešenje:

```
import java.text.*;

public class Gauss
{
    void eliminacija(double[][] M)
    {
        int n=M.length;
        System.out.println("Dimenzija matrice A je "+n);
        int i, j, k;
        double mnozi;

        for(k=0; k<n-1; k++)
        {
            for(i=k+1; i<n; i++)
            {
                mnozi=M[i][k]/M[k][k];
                for(j=1; j<n+1; j++)
                {
                    M[i][j]=M[i][j]-mnozi*M[k][j];
                }
            }
        }
        return;
    }

    public double[] vrati(double[][] M)
    {
        int n=M.length;
        double s;
        double[] x=new double[n];
        x[n-1]=M[n-1][n]/M[n-1][n-1];
        for(int i=n-2; i>=0; i--)
        {
            s=M[i][n];
            for(int j=i+1; j<n; j++)
            {
                s=s-M[i][j]*x[j];
            }
        }
    }
}
```

```
        x[i]=s/M[i][i];
    }
    return (x);
}

void stampa(double[] x)
{
    int n=x.length;
    for(int i=0;i<n;i++)
        System.out.println("x["+i+"]="+x[i]);
}

void formatiranaStampa(double[] x)
{
    int n=x.length;
    DecimalFormat df=new DecimalFormat("###,##0.00");

    for(int i=0;i<n;i++)
        System.out.println("x["+i+"]="+df.format(x[i]));
}

public static void main(String[] args)
{
    Gauss g=new Gauss();

    double[][] A=
        {
            {3, 1, 6, 2},
            {2, 1, 3, 7},
            {1, 1, 1, 4}
        };
    g.eliminacija(A);
    double[] x=g.vrati(A);
    System.out.println("Resenje:");
    g.stampa(x);
    System.out.println("Formatirano resenje na dve decimale:");
    g.formatiranaStampa(x);
}
}
```

Izlaz:

Dimenzija matrice A je 3

Resenje:

x[0]=19.0

x[1]=-6.9999999999999982

x[2]=-8.0

7. Zadaci

Formatirano resenje na dve decimale:

x[0]=19.00

x[1]=-7.00

x[2]=-8.00

9. Napisati program za rešavanje jednačine $e^x - 10x = 0$ metodom sečice. Početna rešenja x_0 i x_1 i ϵ dati su na ulazu. Na izlazu štampati krajnje rešenje.

Rešenje:

```
import java.text.*;
public class Secica
{
    public static double f( double x)
    {
        return Math.exp(x)-10*x;
    }

    public static void sec(double eps, double x0,double x1)
    {
        double p2,p1,p0;
        p0=x0;
        p1=x1;
        p2=p1-f(p1)*(p1-p0)/(f(p1)-f(p0));

        while (Math.abs(p2-p1)>eps)
        {
            p0=p1;
            p1=p2;
            p2=p1-f(p1)*(p1-p0)/(f(p1)-f(p0));
        }

        DecimalFormat df=new DecimalFormat("###,###0.000");

        System.out.println("Resenje: "+df.format(p2));
    }

    public static void main(String[] args)
    {
        sec(1.0e-5, 2.3, 3.4);
    }
}
```

Izlaz:

Resenje: 3.577

10. Napisati program za primenu metode Runge-Kutta.

Rešenje:

```
import java.text.*;
public class Kutt
{
    public static double f1(double t, double y)
    {
        return -y + t + 1;
    }

    public static void runge(double a, double b, double vrednost, double korak)
    {
        double t, w, k1, k2, k3, k4;
        t = a;
        w = vrednost;
        DecimalFormat df = new DecimalFormat("###,##0.00");
        for(int i = 0; i < (b - a)/korak; i++)
        {
            k1 = korak * f1(t, w);
            k2 = korak * f1(t + korak/2, w + k1/2);
            k3 = korak * f1(t + korak/2, w + k2/2);
            k4 = korak * f1(t + korak, w + k3);
            w = w + (k1 + 2 * k2 + 2 * k3 + k4)/6;
            t = a + i * korak;
            System.out.print(df.format(t) + "\t" + df.format(w) + "\n");
        }
    }

    public static void main(String[] args)
    {
        runge(0, 1, 1, 0.1);
    }
}
```

Izlaz:

```
0.00 1.00
0.10 1.01
0.20 1.02
0.30 1.04
0.40 1.07
0.50 1.11
0.60 1.15
0.70 1.20
0.80 1.25
0.90 1.31
```

7. Zadaci

11. Sastaviti metodu za izračunavanje vrednosti funkcije $E(x)$ definisane pomoću reda:

$$E(x) = -a - \ln x - \sum_{k=1}^{+\infty} \frac{(-1)^k \cdot x^k}{k \cdot k!}$$

sa tačnošću ε . Proces prekinuti kada opšti član reda postane manji od ε . Konstanta a ima vrednost $a = 0.5772157$. U glavnom programu korišćenjem metoda tabelirati vrednost funkcije $E(x)$ za $x = 0.5(0.1)1.5$ i $\varepsilon = 10^{-6}$ u obliku:

Tabela vrednosti funkcije	
x	E(x)
0.5	x.xxxxxxx
⋮	⋮

Rešenje:

$$a_k = \frac{(-1)^k \cdot x^k}{k \cdot k!}$$

$$a_{k+1} = \frac{(-1)^{k+1} \cdot x^{k+1}}{(k+1) \cdot (k+1)!} = \frac{(-1)^k \cdot x^k}{k \cdot k!} \cdot \frac{-k \cdot x}{(k+1)^2} = -a_k \cdot \frac{k \cdot x}{(k+1)^2}$$

```
import java.text.*;
public class Funkcija
{
    public double E(double x, double eps)
    {
        double a = 0.5772157;
        int k = 1;
        double a0 = -x;
        double s = a0;

        while(Math.abs(a0) > eps)
        {
            a0 = -a0 * (k * x)/Math.pow(k + 1, 2.0);
            s += a0;
            k++;
        }

        return a - Math.log(x) - s;
    }

    public static void main(String[] args)
    {
        Funkcija f=new Funkcija();
        System.out.println("Tabela vrednosti funkcije\n");
    }
}
```

```
System.out.println("x      E(x)");

double eps = 1.0E-6;
DecimalFormat df=new DecimalFormat("###,###0.000");

for(double x = 0.5; x <= 1.5; x += 0.1)
    System.out.println(df.format(x)+"\t"+df.format(f.E(x, eps)));
}
```

Izlaz:

Tabela vrednosti funkcije

x	E(x)
0.500	1.714
0.600	1.609
0.700	1.528
0.800	1.465
0.900	1.415
1.001	1.374
1.100	1.340
1.200	1.313
1.300	1.290
1.400	1.271

12. Neka jednačina $f(x) = 0$ ima koren $x = a$ u intervalu (α, β) , pri čemu je $f(\alpha) \cdot f(\beta) < 0$. Napisati program za nalaženje korena $x = a$ sa tačnošću ε , korišćenjem metoda polovljenja intervala. Na izlazu štampati $k, (x_k, y_k), f(z_k)$, za $k = 0, 5, 10, \dots$ i za poslednje k , pri kome je postignuta zahtevana tačnost. Na kraju tabele štampati vrednost izračunatog korena u obliku:

$A = \pm x.xxxx$ (SA TACNOSCU EPS = 0.xE - xx)

Za testiranje programa uzeti

$$F(x) = e^x - 2(x-1)^2, (\alpha, \beta) = (-0.5, 1.0), \varepsilon = 10^{-12}.$$

Rešenje:

```
import java.text.*;
public class Pola
{
    public double f(double x)
    {
        return Math.exp(x) - 2 * Math.pow(x - 1, 2.0);
    }
}
```

7. Zadaci

```
public static void main(String[] args)
{
    Pola f1 = new Pola();
    double alfa = -0.5;
    double beta = 1.0;
    double eps = 0.1E-11;
    int k = 0;
    double x0 = alfa;
    double y0 = beta;
    double z0 = (x0 + y0)/2;
    double a;

    System.out.println("k"+"\\t"+"x0"+"\\t"+"y0"+"\\t"+"f(z0)");
    DecimalFormat df=new DecimalFormat("###,###0.000");
    System.out.println(df.format(k)+"\\t"+df.format(x0)+"\\t"+df.format(y0)+"\\t"+df.format(f1.f(z0)));

    k++;

    if(f1.f(z0) * f1.f(x0) == 0)
        a=z0;
    else if(f1.f(z0) * f1.f(x0) < 0)
        y0 = z0;
    else
        x0 = z0;

    while(Math.abs(y0-x0) >= eps && f1.f(z0) * f1.f(x0) != 0)
    {
        z0 = (x0 + y0)/2;

        if(f1.f(z0) * f1.f(x0) < 0)
            y0 = z0;
        else
            x0 = z0;

        if(k/5*5 == k)
            System.out.println(df.format(k)+"\\t"+df.format(x0)+"\\t"+df.format(y0)+"\\t"+df.format(f1.f(z0)));

        k ++;
    }

    if(f1.f(z0) * f1.f(x0) == 0)
        a = z0;
    else
        a = (x0 + y0)/2;

    System.out.println("A="+df.format(a)+"\\t (SA TACNOSCU EPS="+eps+"");
}
}
```

Izlaz:

```

k      x0      y0      f(z0)
0.000 -0.500  1.000  0.159
5.000  0.203  0.227  0.058
10.000 0.213  0.213 -0.003
15.000 0.213  0.213  7.047
20.000 0.213  0.213 -0.001
25.000 0.213  0.213  8.935
30.000 0.213  0.213  5.373
35.000 0.213  0.213  5.880
40.000 0.213  0.213  0.001
A=0.213 (SA TACNOSCU EPS=1.0E-12)
    
```

13. Napisati metodu za izračunavanje vrednosti određnog integrala koristeći Simpsonovu uopštenu formulu. Metodu primeniti za izračunavanje vrednosti integrala $\int_0^{\pi/2} \sin x \, dx$. Za početni interval integraljenja uzeti $h = \frac{\pi}{8}$, a zatim deliti interval na pola, do vrednosti $h = \frac{\pi}{1024}$. Štampati listu sledećeg izgleda:

h	Vrednost integrala
$\frac{\pi}{8}$	xx.xxxxxx
$\frac{\pi}{16}$	xx.xxxxxx
⋮	⋮
$\frac{\pi}{1024}$	xx.xxxxxx

Rešenje:

```

import java.text.*;
public class Integral
{
    public double fun(double x)
    {
        return Math.sin(x);
    }
    public double Simpson(double a, double b, double h)
    {
        long n = Math.round((b - a)/(2 * h));
        double s1 = 0;
        double s2 = 0;
    }
}
    
```

7. Zadaci

```
for(int i = 1; i <= n; i++)
    s1 += fun(a + (2*i-1) * h);

for(int i = 1; i <= n - 1; i++)
    s2 += fun(a + 2 * i * h);

return h/3 * (fun(a) + 4 * s1 + 2 * s2 + fun(b));
}

public static void main(String[] args)
{
    Integral f = new Integral();
    System.out.println("H      Vrednost integrala");

    double a = 0;
    double b = Math.PI/2;

    for(int i = 3; i <= 10; i++)
    {
        double k = i;
        double h = Math.PI/Math.pow(2,k);
        System.out.println("PI/"+Math.pow(2,k)+"\t"+f.Simpson(a, b, h));
    }
}
}
```

Izlaz:

H	Vrednost integrala
PI/8.0	1.0001345849741938
PI/16.0	1.0000082955239678
PI/32.0	1.0000005166847064
PI/64.0	1.0000000322650007
PI/128.0	1.0000000020161286
PI/256.0	1.0000000001260012
PI/512.0	1.0000000000078748
PI/1024.0	1.0000000000004923

14. Napisati program za tabeliranje vrednosti funkcije $F(x, y)$ za zadatih n realnih vrednosti argumenata x_i . Zavisno od odnosa parametara A i B , definisan je drugi argument funkcije:

$$F(x, y) = \begin{cases} F(x_i, x_i + A + B) & \text{ako je } A < B \\ F(x_i, x_i + A^2) & \text{ako je } A = B \\ F(x_i, x_i + A - B) & \text{ako je } A > B \end{cases}$$

$$F(x, y) = \sin(x + y) + e^{x-y} - \sqrt{|x^2 - y^2 - 1|} + 3y^3 - 2.3x^2.$$

Na ulazu učitati niz $X = \{x_i\}_n$ i parametre A i B .

Rešenje:

```
import java.text.*;
import java.io.*;
public class Funkcija
{
    public double fun(double x, double y)
    {
return Math.sin(x + y)+Math.exp(x - y) - Math.sqrt(Math.abs(x*x-y*y-1))+3*Math.pow(y,3.0)-
2.3*x*x;
    }

    public static void main(String[] args) throws Exception
    {
        Funkcija f = new Funkcija ();
        BufferedReader ulaz = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Unesite broj elemenata niza X:");

        int n = Integer.parseInt(ulaz.readLine());
        double[] x = new double[100];

        System.out.println("\n Unesite elemente niza X:");
        for(int i = 1; i <= n; i ++)
        {
            System.out.print("x1["+i+"]=");
            x[i] = Double.parseDouble(ulaz.readLine());
        }

        System.out.println("\n Unesite vrednost za A:");
        double a = Double.parseDouble(ulaz.readLine());
        System.out.println("\n Unesite vrednost za B:");
        double b = Double.parseDouble(ulaz.readLine());

        double y;
        System.out.println("i+t"x[i]+t"f");
        DecimalFormat df=new DecimalFormat("###,###0.000");

        for(int i = 1; i <= n; i ++)
        {
            if(a < b)
                y = f.fun(x[i], x[i] + a +b);
            else if(a == b)
                y = f.fun(x[i], x[i] +a * a);
            else
                y = f.fun(x[i], x[i] + a - b);

            System.out.println(i+t"+df.format(x[i])+t"+df.format(y));
        }
    }
}
```


7. Zadaci

Izlaz:

Unesite broj elemenata niza X:

5

Unesite elemente niza X:

x1[1]=1.2

x1[2]=-3.4

x1[3]=5.8

x1[4]=10.23

x1[5]=-0.34

Unesite vrednost za A:

-3.4

Unesite vrednost za B:

3.5

i	x[i]	f
1	1.200	3.664
2	-3.400	-134.473
3	5.800	537.435
4	10.230	3066.354
5	-0.340	-0.921

15. Napisati program za iterativno rešavanje sistema jednačina

$$\begin{aligned}4x^2 - 3xy + 2x - 1 &= 0 \\ x - 2y^2 + 3 &= 0\end{aligned}$$

koristeći sledeće formule:

$$x_{n+1} = \frac{1}{2} \sqrt{x_n(3y_n - 1) + 1} \quad y_{n+1} = \sqrt{\frac{x_n + 3}{2}}$$

Iterativni proces završiti kada bude istovremeno ispunjen uslov

$$|x_{n+1} - x_n| \leq \varepsilon \text{ i } |y_{n+1} - y_n| \leq \varepsilon.$$

Na ulazu je data vrednost ε i početne vrednosti x_0 i y_0 . Na izlazu štampati vrednosti rešenja sistema.

Rešenje:

```
import java.text.*;
import java.io.*;
```

```
public class Zadatak15
{
    public static void main(String[] args) throws Exception
    {
        BufferedReader ulaz = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Unesite vrednost za Epsilon:");
        double eps = Double.parseDouble(ulaz.readLine());
        System.out.println("\n Unesite vrednost za X0:");
        double x0 = Double.parseDouble(ulaz.readLine());
        System.out.println("\n Unesite vrednost za Y0:");
        double y0 = Double.parseDouble(ulaz.readLine());

        double x1 = 0.5 * Math.sqrt(x0 * (3 * y0 - 1) + 1);
        double y1 = Math.sqrt((x0 + 3)/2);

        while(Math.abs(x1 - x0) > eps && Math.abs(y1 - y0) > eps)
        {
            x0 = x1;
            x1 = 0.5 * Math.sqrt(x0 * (3 * y0 - 1) + 1);
            y0 = y1;
            y1 = Math.sqrt((x0 + 3)/2);
        }

        System.out.println("x1 = " + x1 + "\t" + "y1 = " + y1);
    }
}
```

Izlaz:

Unesite vrednost za Epsilon:
1.0E-6

Unesite vrednost za X0:
1.2

Unesite vrednost za Y0:
3.4

x1 = 1.0549073815904801 y1 = 1.423887359597348

16. Napisati program za izračunavanje verižnog razlomka

$$f(x) = \sqrt{a} + \frac{x_1 - x_2}{2\sqrt{a} + \frac{x_2 - x_3}{3\sqrt{a} + \frac{x_3 - x_4}{\vdots}}}$$
$$\frac{(n-1)\sqrt{a} + \frac{x_{n-1} - x_n}{n\sqrt{a}}}{n\sqrt{a}}$$

Na ulazu učitati vrednost za a i niz $X = \{x_i\}_n$.

7. Zadaci

Rešenje:

```
import java.text.*;
import java.io.*;

public class Zadatak16
{
    public static void main(String[] args) throws Exception
    {
        BufferedReader ulaz = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Unesite broj elemenata niza X:");
        int n = Integer.parseInt(ulaz.readLine());

        double[] x = new double[100];

        System.out.println("\n Unesite unesite elemente niza X:");
        for(int i = 1; i <= n; i ++)
        {
            System.out.print("x["+i+"]=");
            x[i] = Double.parseDouble(ulaz.readLine());
        }

        System.out.println("\n Unesite vrednost za A:");
        double a = Double.parseDouble(ulaz.readLine());

        double ver = n * Math.sqrt(a);
        for(int i = n-1; i >= 1; i --)
            ver=i * Math.sqrt(a)+(x[i]-x[i+1])/ver;

        System.out.print("Vrednost veriznog razlomka je"+ver);
    }
}
```

Izlaz:

Unesite broj elemenata niza X:

5

Unesite unesite elemente niza X:

x[1]=4.5

x[2]=6.7

x[3]=-2.3

x[4]=45.6

x[5]=2.56

Unesite vrednost za A:

34.5

Vrednost veriznog razlomka je 29.368350311176826

17. Napisati program za nalaženje NZD brojeva p i q.

Rešenje:

```
/**
 * Zadatak sastavio
 * Dr.-Ing. Karl-Heinz Lehner, Katedra za inženjersku informatiku, Ruhr-Univerzitet Bochum
 */
public class NZD
{
    int metod1(int p, int q)
    {
        int r;

        if(p>q)
        {
            r=p%q;
            while(r!=0)
            {
                p=q;
                q=r;
                r=p%q;
            }
            return q;
        }
        else
        {
            r=q%p;
            while(r!=0)
            {
                q=p;
                p=r;
                r=q%p;
            }
            return p;
        }
    }

    int metod2(int p, int q)
    {
        int r;

        r=p%q;
        while(r!=0)
        {
            p=q;
            q=r;
            r=p%q;
        }
    }
}
```

7. Zadaci

```
    }
    return q;
}

int metod3(int p, int q)
{
    int r;

    while(p!=q)
    {
        if(p>q)
        {
            p=p-q;
        }

        else
        {
            q=q-p;
        }
    }

    return p;
}

public static void main(String[] args)
{
    NZD nzd=new NZD();

    int s;
    s=nzd.metod1(36, 16);
    System.out.println("Rezultati metoda 1");
    System.out.println(s);
    s=nzd.metod1(16, 36);
    System.out.println(s);

    s=nzd.metod2(36, 16);
    System.out.println("Rezultati metoda 2");
    System.out.println(s);
    s=nzd.metod2(16, 36);
    System.out.println(s);

    s=nzd.metod3(36, 16);
    System.out.println("Rezultati metoda 3");
    System.out.println(s);
    s=nzd.metod3(16, 36);
    System.out.println(s);
}
}
```

Izlaz:

Rezultati metoda 1
4
4
Rezultati metoda 2
4
4
Rezultati metoda 3
4
4

18. Napisati program za nalaženje NZD brojeva p i q u obliku grafičkog korisničkog interfejsa.

Rešenje:

```
/**  
 * Zadatak sastavio  
 * Dr.-Ing. Karl-Heinz Lehner, Katedra za inženjersku informatiku, Ruhr-Univerzitet Bochum  
 */  
import java.awt.*;  
import java.awt.event.*;  
  
public class GuiNZD extends Frame implements ActionListener  
{  
    Label pLabel = new Label("p:", Label.RIGHT);  
    TextField pTextField= new TextField("10", 6);  
  
    Label qLabel = new Label("q:", Label.RIGHT);  
    TextField qTextField= new TextField("15", 6);  
  
    Label nzdLabel = new Label("NZD:", Label.RIGHT);  
    TextField nzdTextField= new TextField(6);  
  
    Button okButton=new Button("OK");  
    Button krajButton=new Button("Kraj!");  
  
    public static void main(String[] args)  
    {  
        GuiNZD guiNZD=new GuiNZD();  
        guiNZD.setVisible(true);  
    }  
  
    GuiNZD()  
    {  
        setLayout(new GridLayout (4, 2));  
        add(pLabel);
```

7. Zadaci

```
        add(pTextField);
        add(qLabel);
        add(qTextField);
        add(okButton);
        okButton.addActionListener(this);
        add(krajButton);
        krajButton.addActionListener(this);
        add(nzdLabel);
        add(nzdTextField);

        pack();
    }

    public void actionPerformed(ActionEvent e)
    {
        System.out.println("button label:"+ e.getActionCommand());
        if(e.getSource()==okButton)
        {
            int p=Integer.parseInt(pTextField.getText());
            int q=Integer.parseInt(qTextField.getText());
            int nzd=NZD.method2(p,q);
            System.out.println("NZD="+nzd);
            nzdTextField.setText(Integer.toString(nzd));
        }
        else if(e.getSource()==krajButton)
        {
            System.out.println("kraj");
            System.exit(0);
        }
        else
        {
            System.out.println("nepoznato"+e.getActionCommand());
        }
    }
}

class NZD
{
    static int method2(int p, int q)
    {
        int r;

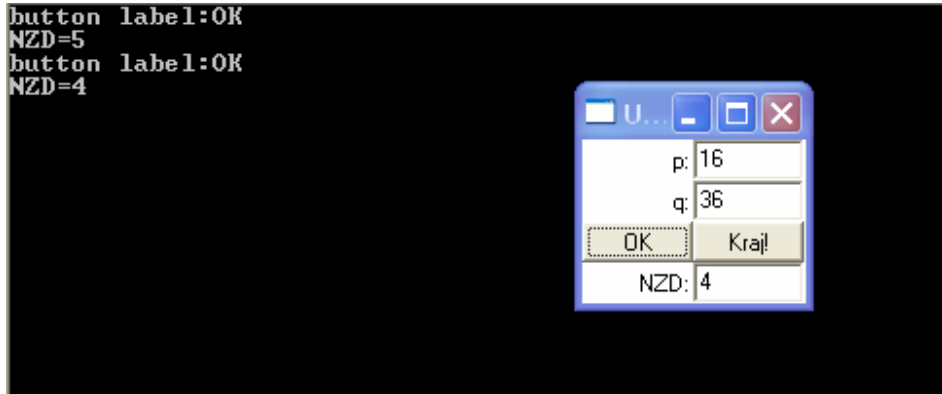
        r=p%q;
        while(r!=0)
        {
            p=q;
            q=r;
            r=p%q;
        }
    }
}
```

```

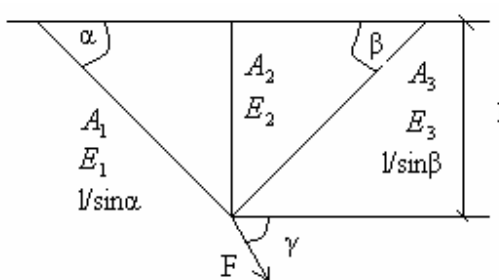
    }
    return q;
}
}

```

Izlaz:



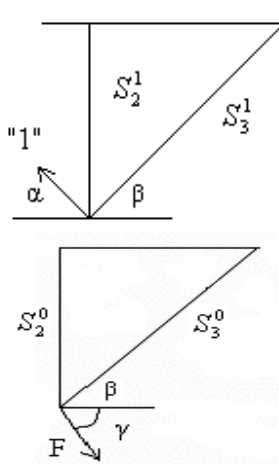
19. Za dati (slika 7.1.) statički neodređen sistem sračunati sile, napon i deformaciju u štapovima, kao i izduženje stapova. Dati su sledeći ulazni podaci:



$\alpha = 30^\circ, \beta = 30^\circ, \gamma = 90^\circ$
 $l = 5\text{m}, F = 100\text{kN}$
 $E_1 = E_2 = E_3 = 210 \cdot 10^6 \text{ kPa}$
 $A_1 = A_2 = A_3 = 0.01\text{m}^2$

Slika 7.1. Statički sistem

Rešenje:



$$1 \cdot \cos \alpha - S_3^1 \cdot \cos \beta = 0$$

$$1 \cdot \sin \alpha + S_2^1 + S_3^1 \cdot \sin \beta = 0$$

$$S_3^1 = \frac{\cos \alpha}{\cos \beta}$$

$$S_2^1 = -(\sin \alpha + \cos \alpha \cdot \text{tg} \beta)$$

$$S_3^0 \cdot \cos \beta + F \cdot \cos \gamma = 0$$

$$S_2^0 + S_3^0 \cdot \sin \beta - F \cdot \sin \gamma = 0$$

7. Zadaci

$$S_3^0 = -F \cdot \frac{\cos \gamma}{\cos \beta}$$

$$S_2^0 = F(\sin \gamma + \cos \gamma \cdot \operatorname{tg} \beta)$$

Dužine:

$$l_1 = \frac{l}{\sin \alpha}, l_2 = l, l_3 = \frac{l}{\sin \beta}$$

$$\delta_{11} = \frac{l_1}{E_1 A_1} + \frac{(S_2^1)^2 l_2}{E_2 A_2} + \frac{(S_3^1)^2 l_3}{E_3 A_3}$$

$$\delta_{10} = \frac{S_2^1 S_2^0 l_2}{E_2 A_2} + \frac{S_3^1 S_3^0 l_3}{E_3 A_3}$$

$$x_1 \delta_{11} + \delta_{10} = 0 \Rightarrow x_1 = -\frac{\delta_{10}}{\delta_{11}}$$

Sile u štapovima:

$$S_1 = x_1$$

$$S_2 = x_1 \cdot S_2^1 + S_2^0$$

$$S_3 = x_1 \cdot S_3^1 + S_3^0$$

Napon u štapovima:

$$\sigma_1 = \frac{S_1}{A_1}, \sigma_2 = \frac{S_2}{A_2}, \sigma_3 = \frac{S_3}{A_3}$$

Deformacija u štapovima:

$$\varepsilon_1 = \frac{\sigma_1}{A_1}, \varepsilon_2 = \frac{\sigma_2}{A_2}, \varepsilon_3 = \frac{\sigma_3}{A_3}$$

Izduženje štapova:

$$\Delta l_1 = \varepsilon_1 l_1, \Delta l_2 = \varepsilon_2 l_2, \Delta l_3 = \varepsilon_3 l_3$$

/**

* Zadatak sastavio

* Dipl.Ing. Zoran Stankovic, M.Sc., Bochum, 2004.

*/

import java.awt.*;

public class Resetka extends Frame

{

double l; // duzina

double alfa, beta, gama; // uglovi

```
double F; // sila
double A1, A2, A3; // površina poprečnog preseka
double E1, E2, E3; // modul elastičnosti
```

```
// ulazne veličine
```

```
TextField lField = new TextField("");
TextField aField = new TextField("");
TextField bField = new TextField("");
TextField gField = new TextField("");
TextField fField = new TextField("");
TextField a1Field = new TextField("");
TextField a2Field = new TextField("");
TextField a3Field = new TextField("");
TextField e1Field = new TextField("");
TextField e2Field = new TextField("");
TextField e3Field = new TextField("");
```

```
// izlazne veličine
```

```
TextField s1Field = new TextField("");
TextField sigma1Field = new TextField("");
TextField eps1Field = new TextField("");
TextField dl1Field = new TextField("");
TextField s2Field = new TextField("");
TextField sigma2Field = new TextField("");
TextField eps2Field = new TextField("");
TextField dl2Field = new TextField("");
TextField s3Field = new TextField("");
TextField sigma3Field = new TextField("");
TextField eps3Field = new TextField("");
TextField dl3Field = new TextField("");
```

```
Button okButton = new Button("Izračunaj");
```

```
public Resetka()
{
    super("Resetka");

    setLayout(new GridLayout(18, 5));

    add(new Label ("Unošenje podataka"));
    add(new Label (""));
    add(new Label (""));
    add(new Label (""));
    add(new Label (""));
    add(new Label ("Dužina (m)"));
    add(lField);
}
```

7. Zadaci

```
add(new Label (""));
add(new Label (""));
add(new Label (""));

add(new Label ("Ugao alfa (o)"));
add(aField);
add(new Label (""));
add(new Label (""));
add(new Label (""));

add(new Label ("Ugao beta (o)"));
add(bField);
add(new Label (""));
add(new Label (""));
add(new Label (""));

add(new Label ("Ugao gama (o)"));
add(gField);
add(new Label (""));
add(new Label (""));
add(new Label (""));

add(new Label ("Sila (kN)"));
add(fField);
add(new Label (""));
add(new Label (""));
add(new Label (""));

add(new Label ("Površina pop. preseka A1 (m^2)"));
add(a1Field);
add(new Label (""));
add(new Label (""));
add(new Label (""));

add(new Label ("Površina pop. preseka A2 (m^2)"));
add(a2Field);
add(new Label (""));
add(new Label (""));
add(new Label (""));

add(new Label ("Površina pop. preseka A3 (m^2)"));
add(a3Field);
add(new Label (""));
add(new Label (""));
add(new Label (""));

add(new Label ("Modul elastičnosti E1 (kPa)"));
add(e1Field);
```

```
add(new Label (""));
add(new Label (""));
add(new Label (""));

add(new Label ("Modul elastičnosti E2 (kPa)"));
add(e2Field);
add(new Label (""));
add(new Label (""));
add(new Label (""));

add(new Label ("Modul elastičnosti E3 (kPa)"));
add(e3Field);
add(new Label (""));
add(new Label (""));
add(new Label (""));

add (okButton);
add(new Label (""));
add(new Label (""));
add(new Label (""));
add(new Label (""));

add(new Label ("Rezultati"));
add(new Label (""));
add(new Label (""));
add(new Label (""));
add(new Label (""));

add(new Label ("Štap"));
add(new Label ("Sila"));
add(new Label ("Napon"));
add(new Label ("Deformacija"));
add(new Label ("Izduženje"));

add(new Label ("1"));
add(s1Field);
add(sigma1Field);
add(eps1Field);
add(dl1Field);

add(new Label ("2"));
add(s2Field);
add(sigma2Field);
add(eps2Field);
add(dl2Field);

add(new Label ("3"));
add(s3Field);
```

7. Zadaci

```
add(sigma3Field);
add(eps3Field);
add(dl3Field);

pack();
show();
}

public boolean handleEvent( Event e )
{
    if ( e.id == Event.WINDOW_DESTROY)
    {
        System.exit(0);
    }

    else if ( e.target == okButton && e.id == Event.ACTION_EVENT)
    {
        l = toDouble (lField);
        F = toDouble (fField);
        alfa = toDouble(aField)/180*Math.PI;
        beta = toDouble(bField)/180*Math.PI;
        gama = toDouble(gField)/180*Math.PI;
        A1 = toDouble (a1Field);
        A2 = toDouble (a2Field);
        A3 = toDouble (a3Field);
        E1 = toDouble (e1Field);
        E2 = toDouble (e2Field);
        E3 = toDouble (e3Field);

        double s31 = Math.cos(alfa)/ Math.cos(beta) ;
        double s21 = -(Math.sin(alfa)+ Math.cos(alfa)* Math.tan(beta)) ;
        double s30 = -F*(Math.cos(gama)/ Math.cos(beta)) ;
        double s20 = F*(Math.sin(gama)+ Math.cos(gama)* Math.tan(beta)) ;
        double l1 = l/ Math.sin(alfa);
        double l2 = l;
        double l3 = l/ Math.sin(beta);
        double d11 = l1/(E1*A1)+s21*s21*l2/(E2*A2)+s31*s31*l3/(E3*A3);
        double d10 = s21*s20*l2/(E2*A2)+s31*s30*l3/(E3*A3);
        double x1 = -d10/d11;

        // sile u stapovima

        double s1 = x1;
        double s2 = x1*s21+s20;
        double s3 = x1*s31+s30;

        // napon u stapovima
```

```
double sigma1 = s1/A1;
double sigma2 = s2/A2;
double sigma3 = s3/A3;

// deformacija u stapovima

double eps1 = sigma1/A1;
double eps2 = sigma2/A2;
double eps3 = sigma3/A3;

// izduzenje stapova

double dl1 = eps1*l1;
double dl2 = eps2*l2;
double dl3 = eps3*l3;

s1 = roundTo(s1,0) ;
s2 = roundTo(s2,0) ;
s3 = roundTo(s3,0) ;
sigma1 = roundTo(sigma1,0) ;
sigma2 = roundTo(sigma2,0) ;
sigma3 = roundTo(sigma3,0) ;
eps1 = roundTo(eps1,0) ;
eps2 = roundTo(eps2,0) ;
eps3 = roundTo(eps3,0) ;
dl1 = roundTo(dl1,0) ;
dl2 = roundTo(dl2,0) ;
dl3 = roundTo(dl3,0) ;

s1Field.setText(toString(s1));
s2Field.setText(toString(s2));
s3Field.setText(toString(s3));
sigma1Field.setText(toString(sigma1));
sigma2Field.setText(toString(sigma2));
sigma3Field.setText(toString(sigma3));
eps1Field.setText(toString(eps1));
eps2Field.setText(toString(eps2));
eps3Field.setText(toString(eps3));
dl1Field.setText(toString(dl1));
dl2Field.setText(toString(dl2));
dl3Field.setText(toString(dl3));
}

return super.handleEvent(e);
}

double roundTo (double d, int i)
{
```

7. Zadaci

```
        return ((double)Math.round(d*(int)Math.pow(10,i)));
    }

    double toDouble (TextField s)
    {
        return Double.valueOf (s.getText().trim()).doubleValue();
    }

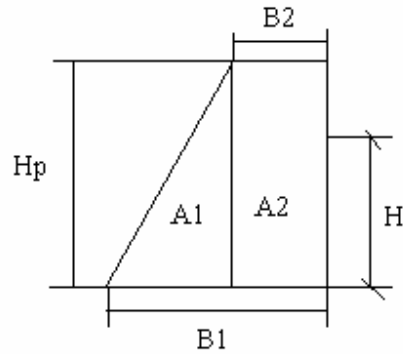
    String toString (double d)
    {
        return new Double(d).toString();
    }

    public static void main(String[] args)
    {
        new Resetka();
    }
}
```

Izlaz:

Štap	Sila	Napon	Deformacija	Izduženje
1	20.0	2000.0	200000.0	2000000.0
2	80.0	8000.0	800000.0	4000000.0
3	20.0	2000.0	200000.0	2000000.0

20. Za prelivni prag zadate geometrije (slika 7.2.) sračunati napone u tlu ispod temelja. Za gama betona uzeti $\gamma_{bet}=26,5 \text{ kN/m}^3$, a za gama vode $\gamma_w=10,0 \text{ kN/m}^3$.



Slika 7.2. Prelivni prag

Rešenje:

Na osnovu slike 7.2. dobijaju se sledeći izrazi:

$$A = \frac{B_1 + B_2}{2} H_p$$

$$A_1 = \frac{B_1 - B_2}{2} H_p$$

$$A_2 = B_2 H_p$$

Na osnovu slike 7.3. dobijaju se sledeći izrazi:

$$w_1 = A_1 \gamma_{bet}, e_1 = B_2 + \frac{B_1 - B_2}{3}$$

$$w_2 = A_2 \gamma_{bet}, e_2 = \frac{B_2}{2}$$

$$p_1 = H \gamma_w, e_p = \frac{H}{3}$$

$$M_c = p_1 e_p + w_1 \left(\frac{B_1}{2} - e_1 \right) + w_2 \left(e_2 - \frac{B_1}{2} \right)$$

Otporni moment temelja: $W = \frac{B_1^2}{6}$

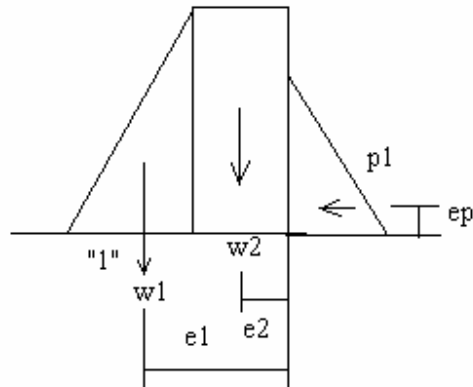
Suma vertikalnih sila: $V = w_1 + w_2$

Srednji napon: $\sigma_m = \frac{V}{A}$

Naponi u tlu ispod praga: $\sigma_1 = \sigma_m + \frac{M_c}{W}$

$$\sigma_0 = \sigma_m - \frac{M_c}{W}$$

7. Zadaci



Slika 7.3.

```
import java.awt.*;
/**
 * Program: Stabilnost prelivnog praga / naponi u tlu
 * Dipl. ing. Natasa Katic, Bochum, jul 2004.
 * Dipl. ing. Milan Gocic, Nis, jul 2004.
 */
public class PrelivniPrag extends Frame
{
    double Hp; // visina prelivnog praga
    double H; // dubina vode iza praga
    double B1; // sirina praga u temelju
    double B2; // sirina praga u kruni
    TextField HpField = new TextField("");
    TextField HField = new TextField("");
    TextField B1Field = new TextField("");
    TextField B2Field = new TextField("");

    TextField s1Field = new TextField("");
    TextField s0Field = new TextField("");

    Button okButton = new Button("Izračunaj");

    public PrelivniPrag()
    {
        super("PrelivniPrag");

        setLayout(new GridLayout(9,2));

        add(new Label ("Unos podataka:"));
        add(new Label (""));
        add(new Label ("Visina prelivnog praga [m]"));
        add(HpField);
        add(new Label ("Dubina vode [m]"));
        add(HField);
    }
}
```

```
add(new Label ("Širina u temelju [m]"));
add(B1Field);
add(new Label ("Širina u kruni [m]"));
add(B2Field);

add(okButton);
add(new Label (""));
add(new Label ("Rezultati:"));
add(new Label (""));

add(new Label ("Napon s1 [kN/m^3]"));
add(s1Field);
add(new Label ("Napon s0 [kN/m^3]"));
add(s0Field);

pack();
show();
}

public boolean handleEvent( Event e )
{
    if ( e.id == Event.WINDOW_DESTROY )
    {
        System.exit(0);
    }

    else if ( e.target == okButton && e.id == Event.ACTION_EVENT )
    {
        Hp = toDouble (HpField);
        H = toDouble (HField);
        B1 = toDouble(B1Field);
        B2 = toDouble(B2Field);

        double Gb=26.5;
        double Gw=10.0;
        double A=(B1+B2)*Hp/2;
        double A1=(B1-B2)*Hp/2;
        double A2=B2*Hp;
        double V1=A1*Gb;
        double V2=A2*Gb;
        double e1=B2+(B1-B2)/3;
        double e2=B2/2;
        double P1=(H*H/2)*Gw;
        double ep=H/3;
        double Mc=P1*ep+V1*(B1/2-e1)+V2*(e2-B1/2);
        double W=B1*B1/6; // otporni moment temelja
        double V=V1+V2; // suma vertikalnih sila
        double Sm=V/A; // srednji napon u tlu
    }
}
```

7. Zadaci

```
double s1=Sm+Mc/W;
double s0=Sm-Mc/W;

if (s0 < 0.0)
{
    s1Field.setText("Zatezanje ");
    s0Field.setText("u tlu");
}

else
{
    s1 = roundTo(s1,1) ;
    s0 = roundTo(s0,1) ;

    s1Field.setText(toString(s1));
    s0Field.setText(toString(s0));
}
}

return super.handleEvent(e);
}

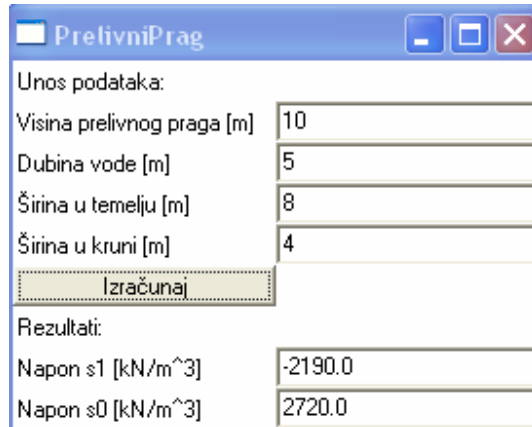
double roundTo (double d, int i)
{
    return ((double)Math.round(d*(int)Math.pow(10,i)));
}

double toDouble (TextField s)
{
    return Double.valueOf (s.getText().trim()).doubleValue();
}

String toString (double d)
{
    return new Double(d).toString();
}

public static void main(String[] args)
{
    new PrelivniPrag();
}
}
```

Izlaz:



7.2. Zadaci za vežbu

1. Sa ulaza se unose elementi matrice tenzora napona $(S) = \begin{Bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{Bmatrix}$, kao i početna

aproksimativna vrednost iteracije σ_0 . Napisati program koji će izračunavati vrednosti glavnih napona iterativno Newtonovom tangentsnom metodom, kao rešenje sekularne jednačine $\sigma_i^3 - g_1\sigma_i^2 + g_2\sigma_i - g_3 = 0$.

Invarijante napona računati po formulama:

$$g_1 = \sigma_x + \sigma_y + \sigma_z$$

$$g_2 = \begin{vmatrix} \sigma_y & \tau_{yz} \\ \tau_{zy} & \sigma_z \end{vmatrix} + \begin{vmatrix} \sigma_x & \tau_{xz} \\ \tau_{zx} & \sigma_z \end{vmatrix} + \begin{vmatrix} \sigma_x & \tau_{xy} \\ \tau_{yx} & \sigma_y \end{vmatrix}$$

$$g_3 = \begin{vmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{vmatrix}$$

Iterativni postupak prekinuti kada bude zadovoljen uslov da razlika između dve uzastopne iterativne vrednosti bude manja od zadate tačnosti ε .

2. Napisati program koji će za n proizvoljno zadatih tačaka koordinatama (x, y) u Dekartovom koordinatnom sistemu, računati površinu parcele prema formuli $P = \frac{1}{2} \sum_{i=1}^n y_i (x_{i-1} - x_{i+1})$.

Napomena: Nizove koordinata dodefinisati po formuli

7. Zadaci

$$x_0 = x_n; x_{n+1} = x_1$$

$$y_0 = y_n; y_{n+1} = y_1$$

3. Dat je niz brojeva sa vrednostima između 0 i 999 u formatu I3. Prebrojati koliko puta se pojavljuje svaka cifra dekadnog brojnog sistema u nizu brojeva kao značajna cifra, kao i na kome mestu (stotica, desetica i jedinica). Na izlazu štampati listu sledećeg oblika:

Cifra	Kao stotica	Kao desetica	Kao jedinica
0	XXXX	XXXX	XXXX
⋮	⋮	⋮	⋮
9	XXXX	XXXX	XXXX

4. Na osnovu izraza

$$\text{Cel} = \frac{\text{Far} - 32}{1.80},$$

gde je

Cel – temperatura u skali Celzija,

Far - temperatura u skali Farenhajta,

napisati metod za prevođenje temperature iz skale Farenhajta u skalu Celzija.

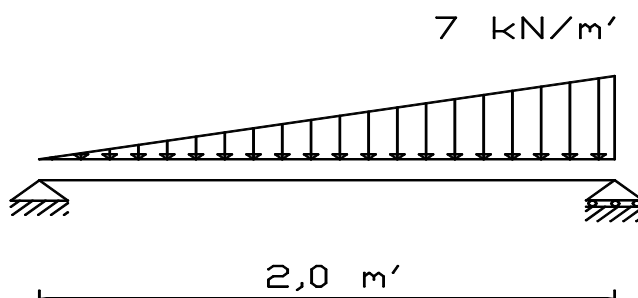
Na ulazu uneti temperature za mesec maj i u glavnom programu na osnovu metode prevesti temperature i izračunati prosečnu temperaturu za mesec maj. Na izlazu štampati redni broj, temperature (Farenhajt i Celzijus) i prosečnu temperaturu. Izlazna lista je oblika:

R.BR.	FARENHAJT	CELZIJUS
xx	xxxx.xxx	xxxx.xxx
⋮	⋮	⋮

PROSECNA TEMPERATURA ZA MESEC MAJ: xxx.xxx

5. Za dati nosač i opterećenje prema slici 7.4, data je jednačina momentne linije

$$M(x) = \frac{x}{3} - 7 \frac{x^3}{12} \left(\frac{\text{kN}}{\text{m}'} \right).$$



Slika 7.4

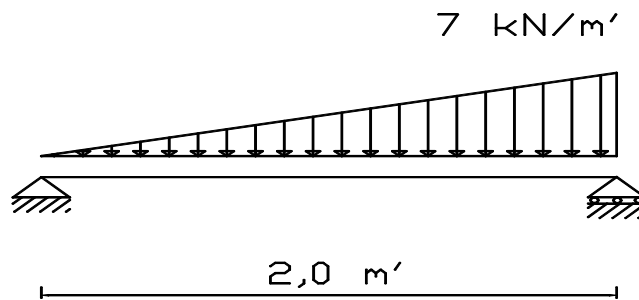
Odrediti:

- Mesto i veličinu maksimalnog momenta savijanja, numeričkim postupkom, primenom Newton-ove metode, iz uslova diferencijalne zavisnosti $\frac{\partial M(x)}{\partial x} = T(x)$ i uslova da se maksimalni momenat nalazi na mestu gde je transverzalna sila jednaka nuli. Za prvu aproksimativnu vrednost uzeti sredinu raspona. Iterativni postupak prekinuti kada bude zadovoljen uslov da je apsolutna vrednost razlike dve uzastopne nule manja od $\varepsilon \leq 10^{-5}$.
- Tabelirati vrednosti momenata i transverzalnih sila sa korakom $\Delta x = 0.1$ m na celoj dužini grede.

Štampati:

- Mesto maksimalnog momenta $l_{max} = x.xx$,
- Vrednost maksimalnog momenta $M_{max} = xxx.xx$,
- Odstojanje x , kao i vrednosti sila M i T u tabelarnom obliku.

6. Za nosač sa slike 7.5 numeričkim postupkom odrediti:



Slika 7.5

- Veličinu fiktivne sile kao rezultantu fiktivnog opterećenja po formuli $\phi = \int_0^l M(x) dx$,
- Položaj rezultante fiktivnog opterećenja iz uslova da je $\phi \cdot r = \int_0^l x \cdot M(x) dx$, gde je ϕ - rezultanta fiktivnog opterećenja, a r odstojanje rezultante od početka grede.

Postupak integracije sprovesti trapeznom formulom za broj koraka $n = 150$.

7. Napisati metod BERNULI(k, n, p, q) za izračunavanje Bernulijeve formule binomne raspodele verovatnoće:

$$P_{k,n}(A) = \frac{n!}{k!(n-k)!} p^k q^{n-k},$$

gde je: p – verovatnoća pojave događaja A ,
 q – verovatnoća da se ne pojavi događaj A ,
 n – broj nezavisnih ispitivanja događaja A ,

7. Zadaci

k – broj ponavljanja događaja A .

U glavnom programu koristeći metod BERNULI rešiti sledeći problem: Ako pri gađanju nekog objekta iz vatrenog oružja verovatnoća da se pogodi cilj iznosi $\frac{1}{3}$, i ako se ispali 9 hitaca, kolika je verovatnoća da se cilj pogodi tačno 3 puta?

8. Izračunati

$$S = \sum_{k=1}^{+\infty} \frac{x^k \left[P \sin\left(\frac{\pi}{4} + x\right) + B \cos\left(\frac{\pi}{4} - x\right) \right]}{k(k+1)},$$

za zadate vrednosti x , P i B . Sumiranje prekinuti kada je apsolutna vrednost poslednje dodatog člana manja od 10^{-3} . Izlazna lista neka izgleda:

K	S
xxx	xxxx.xxxx
⋮	⋮

9. Napisati program za tabeliranje vrednosti funkcije $f(x) = x^4 - 10x^3 + 35x^2 - 50x + 24$ koristeći Hornerovu šemu za izračunavanje vrednosti funkcije. Funkciju tabelirati za vrednost $x = 0.0$ (0.000001) 5.0. Na izlazu štampati tabelu izgleda:

TABELA FUNKCIJE F(X)	
X	F(X)
x.xxxxxxxx	xxx.xxxxxxxx
⋮	⋮

10. Napisati program za dimenzionisanje nosača, pravougaonog poprečnog preseka širine b i visine h , koristeći sledeće izraze:

$$h = \sqrt{\frac{6 M}{b \sigma_{\text{doz}}}}, \quad M = \max_{1 \leq i \leq n} \{ |M_i| \},$$

gde je M_i poznati moment savijanja u i -tom preseku nosača.

Na ulazu učitati b , σ_{doz} , n , M_i , ($i = 1, \dots, n$). Na izlazu štampati listu

Za moment $M = \text{xxxx.xxxx}$ potrebno je $H = \text{xx.xxx}$ m.

11. Tebelirati funkciju

$$f(x) = 0.3(\ln \sqrt{x^2 + y^2} + x^2 + y^2) + 2.6(\sin a_1 + \sin a_2) + 2.2(\cos b_1 + \cos b_2),$$

gde su x i y koreni jednačine

$$1.8p^2 + 0.2zp - 4.15z^2 = 0$$

za $z = 1(3)40$. a_1 i a_2 su koreni jednačine

$$p^2 - 5.5p - 3.25 = 0,$$

b_1 i b_2 su koreni jednačine

$$0.6p^2 + 1.3p + 0.1 = 0.$$

Sve kvadratne jednačine rešavati posebnom metodom. Na izlazu štampati tabelu sledećeg izgleda

TABELA FUNKCIJE F(Z)

Z	F(Z)
1	xxx.xxxxx
4	xxx.xxxxx
⋮	⋮

12. Napisati program za tabeliranje funkcije $y=f(x)$ zadate u obliku

$$f(x) = \sqrt{1+x} + \frac{\cos 2x}{1+\sqrt{x}}$$

za niz vrednosti promenljive x u jednakim intervalima. Ulazni podaci su početna i krajnja vrednost promenljive x i broj intervala. Štampati vrednosti x i y u tabeli sa odgovarajućim zaglavljem.

13. Napisati metod pod imenom SILA za izračunavanje sile privlačenja između dve materijalne tačke m_1 i m_2 na rastojanju r, datu formulom

$$F = \gamma \frac{m_1 m_2}{r^2},$$

ako je $\gamma = 6.67 \cdot 10^{-8}$. Za mase i rastojanja date na ulazu ($m_1 = 3 \cdot 10^6, m_2 = 3.7 \cdot 10^6, r = 10^4$) izlazna lista neka izgleda:

Za $m_1 = \pm x.xxxxxE \pm yy, m_2 = m_1 = \pm x.xxxxxE \pm yy$ i odstojanje $r = \pm x.xxxxxE \pm yy$, sila je $F = \pm x.xxxxxE \pm yy$.

7. Zadaci

14. Napisati program koji će za dati polinom n-tog stepena

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$$

izračunati površinu A u granicama $[x_p, x_k]$ kao i koordinatu x_t težišta te površine ako je

$$A = \int_{x_p}^{x_k} P(x) dx$$

$$S = \int_{x_p}^{x_k} xP(x) dx$$

$$x_t = \frac{S}{A}.$$

Ulazne podatke $n, (a_i, i = 0, \dots, n), x_p, x_k$ učitati u proizvoljnom formatu. Vrednosti A i S odrediti funkcijom koristeći Simpsonovo pravilo za izračunavanje integrala. Vrednost polinoma izračunavati funkcijom koristeći Hornerovu šemu. Na izlazu štampati vrednosti A, S i x_t u sledećem obliku:

- površina A $\pm x.xxxE \pm xx$
- statički moment S $\pm x.xxxE \pm xx$
- koordinata x_t $\pm xxxxxx.xxx$

15. Napisati program za izračunavanje položaja težišta površine ograničene proizvoljnom krivom $y = f(x)$, apscisom i pravama paralelnim y-osi, $x_1 = a$ i $x_2 = b$. Položaj težišta date površine sračunava se prema:

$$x_c = \frac{\int_a^b x f(x) dx}{\int_a^b f(x) dx} \qquad y_c = \frac{\frac{1}{2} \int_a^b [f(x)]^2 dx}{\int_a^b f(x) dx}$$

Vrednosti određenog integrala izračunati trapeznom metodom TRAP(a, b, n, F, I, S), gde su:

a i b – granice integrala,

n – broj podela integracije,

F – naziv funkcije sa podintegralnom funkcijom,

I – vrednost funkcije – I=1, 2, 3 definiše funkcije $f(x), xf(x), [f(x)]^2$, respektivno,

S – vrednost integrala.

Za primer uzeti odsečak parabole $y = c \cdot x^2$ u intervalu $[a, b]$. Učitati a, b, c, n i na izlazu štampati koordinate težišta.

16. Napisati program za tabelarno prikazivanje vrednosti horizontalnog pritiska na tlo, na bočnim površinama masivnog dubokog temelja, primenom sledećeg izraza

$$\sigma_z = m \cdot z \cdot (z_0 - z) \cdot \operatorname{tg} \omega$$

Na ulazu su zadate vrednosti $m, z_0, \omega, z = z_p (\Delta z) z_k$. Izlazna lista neka izgleda:

	z	σ_z
	xx.xxxxx	xx.xxxxx
	⋮	⋮
$\sigma_{\max} = \text{xxx.xxx}$	(za $z = \text{xxx.xxxxx}$)	

Na kraju dati vrednost najvećeg napona pritiska σ_{\max} , kao i dubinu z na kojoj se javlja ovaj napon.

17. Napisati program koji će odštampati n redova Paskalovog trougla.

```
0:1
1:1 1
2:1 2 1
3:1 3 3 1
4:1 4 6 4 1
5:1 5 10 10 5 1
6:1 6 15 20 15 6 1
⋮
```

Elementi se određuju kao zbir elemenata iz prethodnog reda i to elementa iznad njega i prvog elementa do njega sa leve strane. Prvi i poslednji elemenat u jednom redu imaju vrednost 1.

18. Napisati program kojim se ispituje vitkost štapova prostorne rešetke ako važi sledeće

$$\lambda_k = \frac{l_k}{i_{\min}}$$

gde je λ_k vitkost štapa, l_k slobodna dužina izvijanja koja je u ovom slučaju jednaka dužini štapa l . Vrednost i_{\min} je zadata izrazom

$$i_{\min} = \sqrt{\frac{I_{\min}}{F}}, \text{ gde je } I_{\min} = \frac{1}{12} b^3 h, F = bh \text{ (} b < h \text{)}.$$

Na ulazu učitati broj štapa m , dužinu štapa l , dimenzije poprečnog preseka b i h . Na izlazu štampati broj štapa m čija je vitkost $\lambda_k > 50$.

7. Zadaci

19. Za nosač su dati statički uticaji (M i N) u presecima 1 do 10, kao i geometrijske karakteristike svakog preseka (F, I, W_o i W_u). Uticaji su dati matricom $STAT_{10 \times 2}$, pri čemu su momenti dati u prvoj koloni, u kNm, sa znakom + za zatezanje donje ivice preseka (sa indeksom u), a normalne sile su date u kN, sa znakom + za pritisak. Geometrijske karakteristike date su matricom $GEOM_{10 \times 4}$ i to: F u m^2 , I u m^4 , W_o (otporni moment gornje ivice preseka) u m^3 , W_u (otporni moment donje ivice preseka) u m^3 .

Napisati program kojim će se:

- izračunati napon (u MPa) na gornjoj i donjoj ivici svakog preseka (sa znakom + za napone pritiska)
$$\sigma_{o,u} = \frac{N}{F} \pm \frac{M}{W_{q,u}}$$
- proveriti da li se sračunati napon nalazi u dozvoljenim granicama $\sigma_{\min.dop.}$ (najmanji dopušteni napon zatezanja) i $\sigma_{\max.dop.}$ (najveći dopušteni napon pritiska) i štampati upozorenje za nedozvoljena naponska stanja.
- izdvojiti presek sa maksimalnim naprezanjima i štampati ga posebno.

Napomena: Matrice STAT i GEOM, kao i vrednosti za dopuštene napone uneti u proizvoljnom formatu.

20. Napisati funkciju za nalaženje jednačine regresione prave $y(x)$ na osnovu rezultata merenja datih pomoću dva niza x_i i y_i od n članova. Prava ima oblik $y(x) = R_y x + b$.

Vrednosti R_y i b su date sledećim izrazima:

$$R_y = \frac{\sum_{i=1}^n [(x_i - x_0)(y_i - y_0)]}{\sum_{i=1}^n (x_i - x_0)^2}, \quad \text{gde je} \quad \begin{aligned} x_0 &= \frac{1}{n} \sum_{i=1}^n x_i, \\ y_0 &= \frac{1}{n} \sum_{i=1}^n y_i. \end{aligned}$$
$$b = y_0 - R_y x_0,$$

