

OSNOVE PROGRAMIRANJA I

1. letnik VSP Računalništvo in informatika
Študijsko leto 2002/2003

1. Praštevila (28.-30. oktober 2002)

Napišite program, ki poišče in izpiše vsa praštevila iz določenega območja ter izračuna njihovo vsoto. Meje območja naj bodo določene s konstantama `spMeja` in `zgMeja` v programski kodi.

Primer: Če je spodnja meja območja 10 in zgornja 30 naj bo izpis naslednji:

```
Praštevila med 10 in 30 : 11 13 17 19 23 29
Vsota praštevil : 112
```

2. Matematične funkcije (4.-8. november 2002)

Napišite program za izračun vrednosti funkcij $\cos x$ in $(1+x)^{1/4}$. Ustrezni vrsti za ti funkciji sta:

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!} \pm \dots \quad |x| < \infty$$
$$(1+x)^{1/4} = 1 + \frac{1}{4}x - \frac{1 \cdot 3}{4 \cdot 8}x^2 + \frac{1 \cdot 3 \cdot 7}{4 \cdot 8 \cdot 12}x^3 - \frac{1 \cdot 3 \cdot 7 \cdot 11}{4 \cdot 8 \cdot 12 \cdot 16}x^4 + \dots \quad |x| \leq 1$$

Delovanje matematičnih funkcij preverite s tabelaričnim izpisom. Pri $\cos x$ izpišite izračunane vrednosti funkcije na intervalu 0..360 s korakom 30 stopinj, pri $(1+x)^{1/4}$ pa na intervalu -1..1 s korakom 0.1, povsod na 4 decimalke natančno.

3. Razred Krog (11.-15. november 2002)

Sprogramirajte razred `Krog` z atributi `polmer`, `obseg` in `ploš` in ter metodami `nastaviPolmer()`, `izracunajObseg()`, `izracunajPloscino()`, `izpisiPolmer()`, `izpisiObseg` in `izpisiPloscino()`. Za generiranje krogov sprogramirajte dva konstruktorja: prvi naj bo brez parametrov, drugi pa naj ima kot parameter polmer kroga. Ob klicu prvega konstruktorja se generira krog s polmerom 1, vrednosti obsega in ploščine pa ostaneta nedefinirani. Ob klicu drugega konstruktorja naj se pravilno nastavijo vrednosti vseh atributov: polmera, obsega in ploščine.

Za testiranje razreda `krog` napišite razred `KrogTest`. Z uporabo prvega konstruktorja definirajte dva kroga, z uporabo drugega pa še dva. Izpišite polmere, obsege in površine za vse kroge. Nato spremenite polmera prvih krogov in ponovno izpišite vse vrednosti.

4. Volitve (18.-22. november 2002)

Napišite program, ki bo omogočal spremljanje predsedniških volitev. Zakon določa, da je Slovenija razdeljena na 8 volilnih enot, vsaka enota pa še na 11 volilnih okrajev. Denimo, da je v vsakem volilnem okraju natanko 18200 volilcev. Če na volitvah za predsednika države kandidirajo 5 kandidatov, lahko podatke o glasovih za posameznega kandidata po volilnih okrajih zapišemo v trodimenzionalno tabelo `rezVolitev`, ki je deklarirana na naslednji način:

```
static int rezVolitev[][][] = new int[8][11][5];
```

Program naj sestavljata naslednji metodi:

- a) Metoda `napolni` s pomočjo generatorja naključnih števil napolni tabelo `rezVolitev` tako, da bo vsota glasov vsakega volilnega okraja največ 18200.
- b) Metoda `statistika` za celotno državo izračuna in izpiše:
 - število volilnih upravičencev, število oddanih glasov ter odstotek volilne udeležbe
 - zmagovalca oziroma kandidata z največ glasovi, število glasov za zmagovalca in odstotek glasov zmagovalca.

5. Stiskanje besed (25.-29. novembra 2002)

Napišite program, ki omogoča stiskanje in razširjanje besed. S stiskanjem zapišemo besede z veliko ponavljajočimi zaporednimi črkami z manj znaki. Besede stisnemo na naslednji način:

```
AAA                zapišemo kot  A3
ABBCD              zapišemo kot  AB2CD
AAAAEEEEEEEEUUU   zapišemo kot  A4E9U
```

Program naj bo omejen na besede, ki so sestavljene samo iz črk (a-z in A-Z). Kot je razvidno iz primera stisnjena beseda ni nikoli daljša od originala, saj število zaporednih pojavitev črke dodamo samo v primeru, ko se črka dejansko ponavlja. Zaradi enostavnosti upoštevajte, da je število pojavitev vedno enomestno število (torej največ 9).

Program naj vsebuje tako metodo za stiskanje kot metodo za razširjanje besed. Zaradi preverjanja pravilnosti naj program omogoča, da izhodiščno besedo preberemo s standardnega vhoda, jo s prvo metodo stisnemo in tako izpišemo. Nato stisnjeno besedo z drugo metodo znova razširimo in razširjeno izpišemo.

6. Ocene izpitov (2. - 6. december 2002)

V 1. letnik fakultete je bilo lani vpisano `ST_STUD` študentov, katerih podatki (vpisna številka, priimek, ime in ocene osmih opravljenih izpitov) so shranjeni v tabeli `studenti`. Pri tem veljajo naslednje deklaracije:

```
class Student
{
    private int vpisnaStevilka;
    private String ime;
    private String priimek;
    private int[] ocene = new int[8];
}

class TestStudent
{
    final int ST_STUD=10;
    Student[] studenti = new Student[ST_STUD];
}
```

Napišite program, ki bo omogočal vnos vseh potrebnih podatkov v tabelo `studenti` in izpisal urejen seznam študentov glede na doseženo povprečno oceno (na prvem mestu naj bo študent

z najvišjo povprečno oceno, na zadnjem pa tisti z najslabšo). Upoštevajte, da so vsi študentje že opravljali vseh osem izpitov, ocene pa so z intervala med 5 in 10.

7. Anagrami (9. – 13. december 2002)

Napišite program, ki prebere dva niza, in ugotovi, ali sta anagrama. (Dva niza sta anagrama, če sta sestavljena iz istih črk, npr. DESTILARNA in NADREALIST ali ZORA in ROZA). Nalogo rešite na dva načina:

- tako, da za predstavitev nizov uporabite tabelo znakov
- tako, da za predstavitev nizov uporabite razred `StringBuffer`

V obeh primerih morate najprej sortirati znake v obeh nizih. Če sta po sortiranju oba niza enaka, sta anagrama.

8. Razdalje med točkami (16. - 20. december 2002)

Napišite program, ki bo prebral koordinate petih točk v tridimenzionalnem prostoru ter izračunal in izpisal njihove medsebojne razdalje. Predpostavite, da je vsaka točka predstavljena kot objekt razreda `Točka`:

```
class Točka
{
    // atributi
    private double x;           // koordinata x
    private double y;           // koordinata y
    private double z;           // koordinata z

    // morebitni konstruktor in metode deklarirajte sami
    ...
}
```

9. Živali (6. – 10. januar 2003)

Zgled o živalih, ki smo ga naredili na predavanjih, dopolnite tako, da bo ustrezal naslednjim zahtevam:

- Za vsako žival želimo hraniti še atribut starost (v letih).
- Za vsakega psa potrebujemo podatek o pasmi (tipa `String`) in datumu zadnjega cepljenja (datum lahko predstavite s tremi samostojnimi atributi `dan`, `mesec` in `leto`, lahko pa uporabite razred `Date` iz paketa `java.util`). Poleg tega želimo za pse z rodovnikom hraniti podatke o starših (ime očeta in matere).
- Za vsako kravo želimo vedeti, koliko litrov mleka je dala v zadnjih 12 mesecih (podatek potrebujemo za vsak mesec posebej).
- Za vsako kačo nas zanima, ali je strupena (atribut tipa `boolean`) in koliko je dolga (dolžina v cm).

Napišite program, ki bo omogočal:

- vnos podatkov za navedene živali v tabelo `z`, ki je deklarirana kot `Zival[] z=new Zival[10];`
- izpis vseh podatkov v tabeli.

10. Grafični prikaz povprečnih mesečnih temperatur (13.-17. januar)

Napišite program, ki prebere podatke o povprečni mesečni temperaturi za vse mesece od januarja do decembra in jih shrani v tabelo 12 realnih števil. Prebrane podatke prikažite v obliki stolpičnega diagrama, v katerem je povprečna temperatura za vsak mesec prikazana kot pravokotnik ustrezne višine.

Prikaz naj vsebuje ustrezno legendo: okrajšana imena mesecev na osi x in vrednosti za temperaturo na osi y.

A*. Volitve II (18.-22. november 2002)

Dopolnite četrto nalogo z metodo `StatistikaEnot`. Ta za vsako volilno enoto posebej izračuna in izpiše:

- število volilnih upravičencev, število udeležencev volitev ter odstotek volilne udeležbe
- vseh pet kandidatov po vrsti od tistega z največ do tistega z najmanj glasovi
- ob vsakem kandidatu naj izpiše število glasov in odstotek glasov tega kandidata v enoti.

Napotek: Pri izračunu si pomagajte s pomožnimi tabelami.

Primer izpisa za tretjo volilno enoto:

```
Volilna enota: 3
Upravi encev : 200200
Udeležencev: 102342
Volilna udeležba: 51.12 %
```

```
1. Kandidat 4 : 34512 glasov, 33.72 %
2. Kandidat 1 : 23120 glasov, 22.59 %
3. Kandidat 2 : 19312 glasov, 18.87 %
4. Kandidat 5 : 16444 glasov, 16.07 %
5. Kandidat 3 : 8954 glasov, 8.75 %
```

B*. Splošna rešitev za sortiranje tabel objektov (6. – 10. januar 2003)

Koncept dedovanja omogoča, da napišemo metodo za sortiranje tabel, ki vsebujejo objekte poljubnega tipa (tj. z isto metodo lahko sortiramo katerokoli tabelo objektov). V ta namen moramo definirati abstraktni razred `Element`, ki vsebuje abstraktno metodo `manjsi`:

```
public abstract class Element
{
    public abstract boolean manjsi(Element b);
}
```

Objekti, ki jih sortiramo, morajo pripadati podrazredu, ki je izpeljan iz razreda `Element`. V podrazredu je treba deklarirati attribute objektov (odvisno od problema, ki ga želimo sprogramirati) in redefinirati metodo `manjsi`.

Metodo za sortiranje je treba napisati tako, da sortira tabelo, ki jo sestavljajo objekti tipa `Element`, in za primerjanje dveh elementov uporablja metodo `manjsi`. Metoda mora biti deklarirana v posebnem razredu (npr. `SortiranjeObjektov`), od koder jo lahko pokličejo vsi programi, ki jo potrebujejo.

Sprogramirajte algoritem za sortiranje z navadnim izbiranjem, ki smo ga spoznali na predavanjih, tako, da bo ustrezal zgoraj opisanim zahtevam in ga preizkusite na dveh primerih:

- na primeru tabele delavcev, ki je bil predstavljen na predavanjih (prosojnice 121-127; datoteki `Delavec4.java` in `Delavec4Glavni.java`),
- na primeru tabele študentov iz 7. naloge.

C*. Hanojski stolpiči (13. – 17. januar 2003)

Podane so tri palice in n različno velikih obročev, ki jih lahko zlagamo enega na drugega in tako tvorimo stolpiče. Na začetku je vseh n obročev zloženih po velikosti na prvi palici, mi pa jih moramo prestaviti na drugo palico upoštevajoč naslednja pravila:

- na vsakem koraku lahko prestavimo samo en obroč;
- večjega obroča ne smemo prestaviti na manjšega;
- tretjo palico lahko uporabimo kot pomožno odlagališče.

Napišite program za grafični prikaz zaporedja korakov, ki so potrebni, da prestavimo obroče s prve palice na drugo.

Namig: Za risanje obročev uporabite metodo `drawRect` ali `fillRect`. Algoritem za prestavljanje obročev je rekurziven:

- najprej prestavimo $n - 1$ obročev s prve palice na pomožno palico;
- nato prestavimo največji (edini preostali) obroč s prve palice na drugo;
- nazadnje prestavimo $n - 1$ obročev s pomožne palice na drugo palico.

PRASTEVILA

```
public class Prastevila
{
    public static void main(String[] args)
    {
        long vsota=0;
        short stDelj=0;
        int spMeja=1, zgMeja=88;
        System.out.print("Prastevila med intervalom "+spMeja+" in "+zgMeja+" :");
        while (spMeja<=zgMeja)
        {
            for(int i=1;i<=spMeja;i++)
                if (spMeja%i==0) stDelj++;
            if (stDelj<=2)
            {
                vsota=vsota+spMeja;
                System.out.print(spMeja+" ");
            }
            stDelj=0;
            spMeja++;
        }
        System.out.print("\nVsota teh prastevil je "+vsota);
    }
}
```

MATEMATIKA

```

import java.lang.Math;

public class Matematika
{
    public static double cosinus(double z)
    {
        double x1=1, imenovalec=1, stevec=1, a=0;
        z=z*Math.PI/180;
        while(imenovalec/stevec>0.00001)
        {
            imenovalec=imеноvalec*z*z;
            stevec=stevec*(a+1)*(a+2);
            a=a+2;
            x1=x1-imenovalec/stevec;
            imenovalec=imеноvalec*z*z;
            stevec=stevec*(a+1)*(a+2);
            a=a+2;
            x1=x1+imеноvalec/stevec;
        }
        x1=x1*10000;
        x1=Math.round(x1);
        x1=x1/10000;
        return x1;
    }
    public static double enacba(double x)
    {
        double a=3, b=32, c=3, d=8, ex=x*x;
        double x2=1+0.25*x-a/b*x*x;
        while(a/b*ex>0.00001)
        {
            c=c+4;
            d=d+4;
            a=a*c;
            b=b*d;
            ex=ex*x;
            x2=x2+a/b*ex;
            c=c+4;
            d=d+4;
            a=a*c;
            b=b*d;
            ex=ex*x;
            x2=x2-a/b*ex;
        }
        x2=x2*10000;
        x2=Math.round(x2);
        x2=x2/10000;
        return x2;
    }
    public static void main(String[] args)
    {

```

```

for(int i=0;i<=360;i+=30)
    System.out.println("cos("+i+") = "+cosinus(i));
for(int j=1;j<=21;j++)
{
    double dj;
    dj=j;
    dj=(dj-11)/10;
    System.out.println("Rezultat druge enacbe za x "+dj+" :"+enacba(dj));
}
}
}

```

KROG

```
import java.lang.Math;
```

```

public class Krog
{
    double polmer=0, obseg=0, ploscina=0;
    public void nastaviPolmer(double r)
    {
        polmer=r;
    }
    public void izracunajObseg()
    {
        obseg=2*Math.PI*polmer;
    }
    public void izracunajPloscino()
    {
        ploscina=Math.PI*polmer*polmer;
    }
    public String izpisiPolmer()
    {
        double a;
        a=polmer*100;
        a=Math.round(a);
        a=a/100;
        return "Polmer je: "+a;
    }
    public String izpisiObseg()
    {
        double a;
        a=obseg*100;
        a=Math.round(a);
        a=a/100;
        return "Obseg je: "+a;
    }
    public String izpisiPloscino()
    {
        double a;
        a=ploscina*100;
    }
}

```

```

a=Math.round(a);
a=a/100;
return "Ploscina je: "+a;
}
public Krog()
{
nastaviPolmer(1);
}
public Krog(double pol)
{
nastaviPolmer(pol);
izracunajObseg();
izracunajPloscino();
}
}

```

```

public class KrogTest
{
public static void main(String[] args)
{
Krog krog1=new Krog(),krog2=new Krog(),krog3=new Krog(5), krog4=new Krog(4.6);
System.out.println("Krog 1:\t"+krog1.izpisiPolmer()+"\t"+krog1.izpisiObseg()+"\t"
+krog1.izpisiPloscino());
System.out.println("Krog 2:\t"+krog2.izpisiPolmer()+"\t"+krog2.izpisiObseg()+"\t"
+krog2.izpisiPloscino());
System.out.println("Krog 3:\t"+krog3.izpisiPolmer()+"\t"+krog3.izpisiObseg()+"\t"
+krog3.izpisiPloscino());
System.out.println("Krog 4:\t"+krog4.izpisiPolmer()+"\t"+krog4.izpisiObseg()+"\t"
+krog4.izpisiPloscino());
krog1.nastaviPolmer(3);
krog2.nastaviPolmer(1.13);
krog1.izracunajObseg();
krog1.izracunajPloscino();
krog2.izracunajObseg();
krog2.izracunajPloscino();
System.out.println("\n\nKrog 1:\t"+krog1.izpisiPolmer()+"\t"+krog1.izpisiObseg()+"\t"
+krog1.izpisiPloscino());
System.out.println("Krog 2:\t"+krog2.izpisiPolmer()+"\t"+krog2.izpisiObseg()+"\t"
+krog2.izpisiPloscino());
System.out.println("Krog 3:\t"+krog3.izpisiPolmer()+"\t"+krog3.izpisiObseg()+"\t"
+krog3.izpisiPloscino());
System.out.println("Krog 4:\t"+krog4.izpisiPolmer()+"\t"+krog4.izpisiObseg()+"\t"
+krog4.izpisiPloscino());
System.runFinalizersOnExit(true);
}
}

```

VOLITVE + A DODATNA

```

import java.lang.Math;
import java.util.Random;

```



```

public class Volitve
{
    static int rezVolitev[][][] = new int[8][11][5];
    public static void napolni()
    {
        boolean k[] = new boolean[5];
        for(int i=0;i<5;i++)
            k[i]=false;
        Random naklj = new Random();
        int kan;
        for(int i=0;i<8;i++)
            for(int j=0;j<11;j++)
            {
                int vsota=18201;
                for(int m=0;m<5;m++)
                    k[m]=false;
                while ((k[0]&& k[1]&& k[2]&& k[3]&& k[4])!=true)
                {
                    kan=naklj.nextInt(5);
                    if (k[kan]==false)
                    {
                        rezVolitev[i][j][kan]=naklj.nextInt(vsota);
                        k[kan]=true;
                        vsota=vsota-rezVolitev[i][j][kan];
                    }
                }
            }
    }
    public static void statistika()
    {
        long skupajGlasov=0;
        int glasovZmagovalec=0, stUpravivencev=1601600,stKandidata=0;
        double odstotekGlasov,odstotekZmagovalec;
        int ka[] = new int[5];
        for(int i=0;i<8;i++)
            for(int j=0;j<11;j++)
                for(int k=0;k<5;k++)
                {
                    skupajGlasov=skupajGlasov+rezVolitev[i][j][k];
                    ka[k]=ka[k]+rezVolitev[i][j][k];
                }
        odstotekGlasov=(double)skupajGlasov/stUpravivencev*10000;
        odstotekGlasov=Math.round(odstotekGlasov);
        odstotekGlasov=odstotekGlasov/100;
        System.out.println("St. Volilnih Upravivencev: "+stUpravivencev+"St. Oddanih Glasov: "
            +skupajGlasov+"\nOdstotek volilne Udelezbe: "+odstotekGlasov+"\n\n");
        for(int i=0;i<5;i++)
            if (ka[i]>glasovZmagovalec)
            {
                glasovZmagovalec=ka[i];
            }
    }
}

```

```

    stKandidata=i+1;
}
odstotekZmagovalec=(double)glasovZmagovalec/stUpravicecev*10000;
odstotekZmagovalec=Math.round(odstotekZmagovalec);
odstotekZmagovalec=odstotekZmagovalec/100;
System.out.println("ZMAGOVALEC: Kandidat "+stKandidata+"\nSt. Glasov: "
    +glasovZmagovalec+" (" +odstotekZmagovalec+"%"));
}
public static void statistikaEnot()
{
System.out.println("VOLITVE II-----VOLITVE II-----VOLITVE II");
int pomozna[] = new int[5];
int pomozna2[] = new int[5];
int udelezenci=0,pom;
double odstotekUdelezbeVolilneEnote=0,odstotekPosamezniKandidat=0;
for(int i=0;i<8;i++)
{
for(int j=0;j<11;j++)
for(int k=0;k<5;k++)
{
pomozna[k]+=rezVolitev[i][j][k];
udelezenci=udelezenci+rezVolitev[i][j][k];
}
System.out.println("Volilna Enota "+(i+1));
System.out.println("St. Upravicecev: 200200");
System.out.println("St. Udelezencev: "+udelezenci);
odstotekUdelezbeVolilneEnote=((double)udelezenci/200200)*10000;
odstotekUdelezbeVolilneEnote=Math.round(odstotekUdelezbeVolilneEnote);
odstotekUdelezbeVolilneEnote=odstotekUdelezbeVolilneEnote/100;
System.out.println("Volilna Udelezba: "+odstotekUdelezbeVolilneEnote+"\n");
udelezenci=0;
for(int x=0;x<5;x++)
pomozna2[x]=pomozna[x];
for(int l=0;l<=5;l++)
for(int m=1;m<5;m++)
if (pomozna[m]>pomozna[m-1])
{
pom=pomozna[m];
pomozna[m]=pomozna[m-1];
pomozna[m-1]=pom;
}
for(int n=0;n<5;n++)
{
odstotekPosamezniKandidat=((double)pomozna[n]/200200)*10000;
odstotekPosamezniKandidat=Math.round(odstotekPosamezniKandidat);
odstotekPosamezniKandidat=odstotekPosamezniKandidat/100;
System.out.print("Kandidat st. ");
for(int p=0;p<5;p++)
if (pomozna2[p]==pomozna[n])
System.out.print(p+1);
}
}
}

```

```

        System.out.println(" "+pomozna[n]+" glasov, " +odstotekPosamezniKandidat+" %");
    }
    for(int r=0;r<5;r++)
        pomozna[r]=0;
    }
}
public static void main(String args[])
{
    napolni();
    statistika();
    statistikaEnot();
}
}

```

ZHNJAR

```
import java.lang.Character;
```

```

public class Zhnjar
{
    public static String skrci(String razsirjena)
    {
        int stIstihZnakov=0;
        String nov="";
        for(int i=0;i<(razsirjena.length()-1);i++)
            if (razsirjena.charAt(i)!=razsirjena.charAt(i+1))
            {
                nov+=razsirjena.charAt(i);
                if (stIstihZnakov>0)
                {
                    nov+=String.valueOf(stIstihZnakov+1);
                    stIstihZnakov=0;
                }
            }
            else stIstihZnakov++;
        nov+=razsirjena.charAt(razsirjena.length()-1);
        if (stIstihZnakov>0) nov+=String.valueOf(stIstihZnakov+1);
        return nov;
    }
    public static String razsiri(String skrcena)
    {
        String nov2="";
        for(int j=0;j<skrcena.length();j++)
            if (Character.isDigit(skrcena.charAt(j))==true)
                for(int k=1;k<Character.getNumericValue(skrcena.charAt(j));k++)
                    nov2+=skrcena.charAt(j-1);
            else nov2+=skrcena.charAt(j);
        return nov2;
    }
}
public static void main (String args[])
{

```

```

String bes;
System.out.print("Vnesi besedo : ");
bes = BranjePodatkov.preberiString();
bes=skrci(bes);
System.out.println("Stisnjena beseda: "+bes);
bes=razsiri(bes);
System.out.println("Razsirjena beseda: "+bes+" "+bes.length());
}
}

```

STUDENT

```
import java.lang.Math;
```

```

public class Student
{
private int vpisnaStevilka;
private String ime;
private String priimek;
private int[] ocene = new int[8];
public void VpSt (int st)
{
    vpisnaStevilka=st;
}
public void Ime (String im)
{
    ime=im;
}
public void Priimek (String pr)
{
    priimek=pr;
}
public void Ocene()
{
    for (int i=0;i<8;i++)
    {
        System.out.print("\nVnesi oceno "+(i+1)+" :");
        ocene[i]=BranjePodatkov.preberiInt();
    }
}
public float Pov()
{
    float p=0;
    for (int j=0;j<8;j++)
        p+=ocene[j];
    p/=8;
    p*=10;
    p=Math.round(p);
    p/=10;
    return p;
}
}

```

```

public void Izpisi()
{
    System.out.println("Vpisna Stevilka Studenta: "+vpisnaStevilka);
    System.out.println("Ime Studenta: "+ime);
    System.out.println("Priimek Studenta: "+priimek);
    System.out.println("Povprečna Ocena Studenta: "+Pov());
    System.out.println("_____");
}
public Student()
{
    System.out.print("Vnesi vpisno stevilko studenta:");
    VpSt(BranjePodatkov.preberiInt());
    System.out.print("\nVnesi ime studenta:");
    Ime(BranjePodatkov.preberiString());
    System.out.print("\nVnesi priimek studenta:");
    Priimek(BranjePodatkov.preberiString());
    Ocene();
}
}
}


---


public class TestStudent
{
    public static void sortiraj (Student[] st, int SS)
    {
        Student sm;
        float pmin;
        int index;
        for (int x=0;x<SS;x++)
        {
            pmin=0;
            for (int y=x;y<SS;y++)
            {
                index=x;
                if (st[y].Pov()>pmin)
                {
                    pmin=st[y].Pov();
                    index=y;
                }
                sm=st[x];
                st[x]=st[index];
                st[index]=sm;
            }
        }
    }
}
public static void main(String args[])
{
    final int ST_STUD=10;
    Student[] studenti = new Student[ST_STUD];
    for (int i=0;i<ST_STUD;i++)
        studenti[i] = new Student();
    sortiraj(studenti, ST_STUD);
}
}

```

```

for (int j=0;j<ST_STUD;j++)
    studenti[j].Izpidi();
}
}

```

ANAGRAM

```

public class Anagram1
{
    public static String sort(String s)
    {
        s=s.toUpperCase();
        int index;
        char znak,pom;
        for (int i=0;i<s.length()-1;i++)
        {
            index=i;
            znak='Z';
            for (int j=i;j<s.length();j++)
                if (s.charAt(j)<znak)
                {
                    index=j;
                    znak=s.charAt(j);
                }
            pom=s.charAt(i);
            s=s.substring(0,i)+znak+s.substring(i+1,s.length());
            s=s.substring(0,index)+pom+s.substring(index+1,s.length());
        }
        return s;
    }
    public static void main(String Args[])
    {
        System.out.print("Vnesi prvo besedo: ");
        String s1=BranjePodatkov.preberiString();
        System.out.print("Vnesi drugo besedo: ");
        String s2=BranjePodatkov.preberiString();
        if (sort(s1).equals(sort(s2))==true)
            System.out.print("Besedi sta anagrama");
        else
            System.out.print("Besedi nista anagrama");
    }
}

```

```

public class Anagram2
{
    public static StringBuffer sort(StringBuffer s)
    {
        int index;
        char znak,pom;
        for (int i=0;i<s.length()-1;i++)
        {
            index=i;

```

```

znak='Z';
for (int j=i;j<s.length();j++)
if (s.charAt(j)<znak)
{
index=j;
znak=s.charAt(j);
}
pom=s.charAt(i);
s.setCharAt(i,znak);
s.setCharAt(index,pom);
}
return s;
}
public static void main(String Args[])
{
System.out.print("Vnesi prvo besedo: ");
StringBuffer s1=new StringBuffer((BranjePodatkov.preberiString()).toUpperCase());
System.out.print("Vnesi drugo besedo: ");
StringBuffer s2=new StringBuffer((BranjePodatkov.preberiString()).toUpperCase());
s1=sort(s1);
s2=sort(s2);
if ((s1.toString()).equals(s2.toString())==true)
System.out.print("Besedi sta anagrama");
else
System.out.print("Besedi nista anagrama");
}
}

```

```

public class Anagram3
{
public static char[] sort(char[] c)
{
int index;
char znak,pom;
for (int i=0;i<c.length-1;i++)
{
index=i;
znak='Z';
for (int j=i;j<c.length;j++)
if (c[j]<znak)
{
index=j;
znak=c[j];
}
pom=c[i];
c[i]=c[index];
c[index]=pom;
}
return c;
}
public static void main(String Args[])

```

```

{
System.out.print("Vnesi prvo besedo: ");
String s=(BranjePodatkov.preberiString()).toUpperCase();
char[] c1=s.toCharArray();
System.out.print("Vnesi drugo besedo: ");
s=(BranjePodatkov.preberiString()).toUpperCase();
char[] c2=s.toCharArray();
sort(c1);
String p1=new String(c1);
sort(c2);
String p2= new String(c2);
if (p1.equals(p2)==true)
    System.out.print("Besedi sta anagrama");
else
    System.out.print("Besedi nista anagrama");
}
}

```

TOCKA

```

public class Tocka
{
private double x;
private double y;
private double z;
public double X()
{
return x;
}
public double Y()
{
return y;
}
public double Z()
{
return z;
}
public void XJe(double a)
{
x=a;
}
public void YJe(double b)
{
y=b;
}
public void ZJe(double c)
{
z=c;
}
public Tocka(double d,double e,double f)
{

```



```

XJe(d);
YJe(e);
ZJe(f);
}
}

```

```

public class TockaTest
{
public static void main(String Args[])
{
Tocka[] t=new Tocka[5];
for (int i=0;i<5;i++)
{
System.out.println("Vnesi Kordinate "+(i+1)+" . Tocke:!!!");
t[i]=new
Tocka(BranjePodatkov.preberiDouble(),BranjePodatkov.preberiDouble(),BranjePodatkov.pre
beriDouble());
}
double rez;
int j;
for (j=1;j<5;j++)
{
rez=Math.sqrt((t[0].X()-t[j].X())*(t[0].X()-t[j].X())+(t[0].Y()-t[j].Y())*(t[0].Y()-t[j].Y))+
(t[0].Z()-t[j].Z())*(t[0].Z()-t[j].Z()));
rez=rez*100;
rez=Math.round(rez);
rez=rez/100;
System.out.println("Razdalja med tocko 1 in "+(j+1)+" je: "+rez);
}
for (j=2;j<5;j++)
{
rez=Math.sqrt((t[1].X()-t[j].X())*(t[1].X()-t[j].X())+(t[1].Y()-t[j].Y())*(t[1].Y()-t[j].Y))+
(t[1].Z()-t[j].Z())*(t[1].Z()-t[j].Z()));
rez=rez*100;
rez=Math.round(rez);
rez=rez/100;
System.out.println("Razdalja med tocko 2 in "+(j+1)+" je: "+rez);
}
for (j=3;j<5;j++)
{
rez=Math.sqrt((t[2].X()-t[j].X())*(t[2].X()-t[j].X())+(t[2].Y()-t[j].Y())*(t[2].Y()-t[j].Y))+
(t[2].Z()-t[j].Z())*(t[2].Z()-t[j].Z()));
rez=rez*100;
rez=Math.round(rez);
rez=rez/100;
System.out.println("Razdalja med tocko 3 in "+(j+1)+" je: "+rez);
}
rez=Math.sqrt((t[3].X()-t[4].X())*(t[3].X()-t[4].X())+(t[3].Y()-t[4].Y())*(t[3].Y()-t[4].Y))+
(t[3].Z()-t[4].Z())*(t[3].Z()-t[4].Z()));
rez=rez*100;
rez=Math.round(rez);
}
}

```

```

    rez=rez/100;
    System.out.println("Razdalja med tocko 4 in 5 je: "+rez);
}
}

```

ZIVALI

```

public abstract class Zival
{
    private String ime;
    private int starost;
    Zival(String imeZivali,int strZiv)
    {
        ime=imeZivali;
        starost=strZiv;
    }
    public void izpisiVse()
    {
        System.out.println("Ime: "+ime);
        System.out.println("Starost: "+starost);
    }
    public abstract void oglasanje();
}

```

```

import java.util.Date;

```

```

public class Pes extends Zival
{
    private String pasma;
    private Date datCep;
    private String mati;
    private String oce;
    Pes(String imePsa,int strPsa,String pasPsa,Date datum)
    {
        super(imePsa,strPsa);
        pasma=pasPsa;
        datCep=datum;
        mati="";
        oce="";
    }
    Pes(String imePsa,int strPsa,String pasPsa,Date datum,String iMa,String iOce)
    {
        super(imePsa,strPsa);
        pasma=pasPsa;
        datCep=datum;
        mati=iMa;
        oce=iOce;
    }
    public void izpisiVse()
    {
        super.izpisiVse();
        System.out.println("Pasma: "+pasma);
    }
}

```

```

    System.out.println("Datum Zadnjega Cepljenja: "+datCep.getDate()+". "+
(datCep.getMonth()+1)+". "+
    +datCep.getYear());
    if (mati!="")
        System.out.println("Ime matere: "+mati+" Ime oceta: "+oce);
    }
    public void oglasanje()
    {
        System.out.println("Hov,hov");
    }
}

```

```

public class Kaca extends Zival
{
    private boolean strupena;
    private int dolzina;
    Kaca(String imeKace,int stKace,boolean strup,int dolzinaK)
    {
        super(imeKace,stKace);
        strupena=strup;
        dolzina=dolzinaK;
    }
    public void izpisiVse()
    {
        super.izpisiVse();
        System.out.println("Strupenost: "+strupena+" Dolzina: "+dolzina);
    }
    public void oglasanje()
    {
        System.out.println("Ssssss");
    }
}

```

```

public class Krava extends Zival
{
    private int[] mleko=new int[12];
    Krava(String imeKrave,int stKrave,int[] mlek)
    {
        super(imeKrave,stKrave);
        mleko=mlek;
    }
    public void izpisiVse()
    {
        super.izpisiVse();
        System.out.print("Mleko: ");
        for(int i=0;i<12;i++)
            System.out.print(mleko[i]+"l ");
        System.out.println();
    }
    public void oglasanje()
    {
        System.out.println("Muuuu");
    }
}

```

```

}
}
import java.util.Date;

class ZivaliGlavni
{
public static void main(String[] args)
{
    Zival[] z=new Zival[3];
    int odg,odg2,odg3;
    String imeZivali;
    int starost,dolzina;
    String pasma,mati,oce;
    Date cepljenje;
    int i,k;
    int[] mleko=new int[12];
    boolean strup;
    for (i=0; i<z.length; ++i)
    {
        System.out.print("Katero zival zelis vnesti (1 - pes, 2 - krava, 3 - kaca:");
        odg=BranjePodatkov.preberiInt();
        System.out.print("Ime: ");
        imeZivali=BranjePodatkov.preberiString();
        System.out.print("Starost: ");
        starost=BranjePodatkov.preberiInt();
        switch (odg)
        {
            case 1:
                System.out.print("Ali je pes z rodovnikom 1- Da, 2 -Ne: ");
                odg2=BranjePodatkov.preberiInt();
                System.out.print("Pasma: ");
                pasma=BranjePodatkov.preberiString();
                System.out.println("Vnesi datum zadnjega cepljenja: (leto,mesec,dan)");
                cepljenje=new Date(BranjePodatkov.preberiInt(),(BranjePodatkov.preberiInt()-
1),BranjePodatkov.preberiInt());
                if (odg2==2) z[i]=new Pes(imeZivali,starost,pasma,cepljenje);
                if (odg2==1)
                {
                    System.out.print("Ime matere: ");
                    mati=BranjePodatkov.preberiString();
                    System.out.print("Ime oceta: ");
                    oce=BranjePodatkov.preberiString();
                    z[i]=new Pes(imeZivali,starost,pasma,cepljenje,mati,oce);
                }
                break;
            case 2:
                System.out.println("Litrov mleka: ");
                for(k=0;k<12;k++)
                {
                    System.out.print((k+1)+". mesec: ");

```

```

    mleko[k]=BranjePodatkov.preberiInt();
}
z[i]=new Krava(imeZivali,starost,mleko);
break;
case 3:
System.out.print("Ali je Kača strupena: 1 - DA, 0 - Ne: ");
odg3=BranjePodatkov.preberiInt();
if (odg3==0) strup=false;
    else strup=true;
System.out.print("Dolzina(cm): ");
dolzina=BranjePodatkov.preberiInt();
z[i]=new Kaca(imeZivali,starost,strup,dolzina);
break;
}
}
for (i=0; i<z.length; ++i)
{
z[i].izpisiVse();
System.out.print("Oglasanje: ");
z[i].oglasanje();
System.out.println("_____");
}
}
}

```

TEMPERATURA

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class Panel1 extends JPanel
{
public int[] tmp=new int[12];
Panel1(int[] temp)
{
super();
tmp=temp;
}
public void paintComponent(Graphics g)
{
super.paintComponent(g);
Font f=new Font("SansSerif",Font.BOLD,16);
g.setFont(f);
int a=50;
int b=200;
g.drawString(" 0",20,200);
g.drawString("- 5",20,165);
g.drawString("-10",20,130);
g.drawString("-15",20,95);
g.drawString("-20",20,60);
g.drawString("-25",20,25);
}
}

```

```

g.drawString(" 5",20,235);
g.drawString(" 10",20,270);
g.drawString(" 15",20,305);
g.drawString(" 20",20,340);
g.drawString(" 25",20,375);
g.drawString(" 30",20,410);
g.drawString(" 35",20,445);
g.drawString(" 40",20,480);
g.drawString(" 45",20,515);
g.drawString("JAN",50,15);
g.drawString("FEB",105,15);
g.drawString("MAR",160,15);
g.drawString("APR",215,15);
g.drawString("MAJ",270,15);
g.drawString("JUN",325,15);
g.drawString("JUL",380,15);
g.drawString("AVG",435,15);
g.drawString("SEP",490,15);
g.drawString("OKT",545,15);
g.drawString("NOV",600,15);
g.drawString("DEC",655,15);
for(int i=0;i<12;i++)
{
if (tmp[i]<0) b=b+tmp[i]*7;
g.fillRoundRect(a,b,50,(Math.abs(tmp[i])*7),10,10);
if (tmp[i]<0) b=b-tmp[i]*7;
a=a+55;
}
}
}
class Poslusalec extends WindowAdapter
{
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
}
class Okno extends JFrame
{
public Okno(int[] temp)
{
WindowListener p=new Poslusalec();
addWindowListener(p);
setTitle("TABELA LETNIH TEMPERATUR");
setSize(750,550);
Container vsebina=getContentPane();
JPanel panel1=new Panel1(temp);
vsebina.add(panel1);
}
}
}

```

```

public class Temperatura
{
    public static void main(String[] args)
    {
        int[] temp=new int[12];
        System.out.println("VNESI TEMPERATURE!!!!");
        for(int i=0;i<12;i++)
        {
            System.out.print("Vnesi temperaturo za "+(i+1)+" mesec: ");
            temp[i]=BranjePodatkov.preberiInt();
        }
        JFrame okno=new Okno(temp);
        okno.show();
    }
}

```

B DODATNA

```

public class Sortiraj
{
    public static void sort(Element[] tab)
    {
        int j, minIndx;
        Element min;
        Element pom;
        for(int i=0; i<tab.length-1; i++)
        {
            min=tab[i];
            minIndx=i;
            for(j=i+1; j<tab.length; j++)
            if(tab[j].manjsi(min))
            {
                min=tab[j];
                minIndx=j;
            }
            pom=tab[i];
            tab[i]=min;
            tab[minIndx]=pom;
        }
    }
}

```

```

public abstract class Element
{
    public abstract boolean manjsi(Element b);
}

```

```

public class Delavec4 extends Element
{
    private int matStev;
    private String priimek;
    private String ime;
}

```

```

private double[] od;
Delavec4()
{
    System.out.print("Maticna stevilka:");
    matStev=BranjePodatkov.preberiInt();
    System.out.print("Priimek:");
    priimek=BranjePodatkov.preberiString();
    System.out.print("Ime:");
    ime=BranjePodatkov.preberiString();
    od=new double[12];
    for(int i=0; i<od.length; ++i)
    {
        System.out.print("Osebni dohodek za "+Integer.toString(i+1)+". mesec:");
        od[i]=BranjePodatkov.preberiDouble();
    }
}
public void vpisiMatSt(int st)
{
    matStev=st;
}
public void vpisiPriimek(String p)
{
    priimek=p;
}
public void vpisiIme(String i)
{
    ime=i;
}
public void vpisiOd(double[] osDoh)
{
    for(int i=0; i<od.length; ++i)
        od[i]=osDoh[i];
}
public int vrniMatSt()
{
    return matStev;
}
public String vrniPriimek()
{
    return priimek;
}
public String vrniIme()
{
    return ime;
}
public double vrniOd(int i)
{
    return od[i];
}
public double vrniVsotoOd()

```



```

{
double vsota=0;
for (int i=0; i<od.length; ++i)
vsota+=od[i];
return vsota;
}
public void izpisiVse()
{
System.out.println("Maticna stevilka: "+matStev);
System.out.println("Priimek in ime: "+priimek+' '+ime);
for (int i=0; i<od.length; ++i)
System.out.println("Osebni dohodek za "+Integer.toString(i+1)+" mesec:"+od[i]);
}
public boolean manjsi(Element b)
{
if(this.vrniVsotoOd() < (((Delavec4)b).vrniVsotoOd())) return true;
else return false;
}
}

```

```

public class Delavec4Glavni
{
static final int ST_DEL=5;
public static void main(String[] args)
{
Delavec4[] td=new Delavec4[ST_DEL];
for (int i=0; i<ST_DEL; ++i)
td[i]=new Delavec4();
System.out.println();
System.out.println("Pred sortiranjem:");
izpisi(td);
Sortiraj.sort(td);
System.out.println();
System.out.println("Po sortiranju:");
izpisi(td);
}
static void izpisi(Delavec4[] td)
{
for (int i=0; i<ST_DEL; ++i)
{
System.out.print(td[i].vrniMatSt());
System.out.print(" "+td[i].vrniPriimek()+" "+td[i].vrniIme());
System.out.println(" "+td[i].vrniVsotoOd());
}
}
}

```

```

import java.lang.Math;

```

```

public class Student2 extends Element
{
private int vpisnaStevilka;

```

```

private String ime;
private String priimek;
private int[] ocene = new int[8];
public void VpSt (int st)
{
    vpisnaStevilka=st;
}
public void Ime (String im)
{
    ime=im;
}
public void Priimek (String pr)
{
    priimek=pr;
}
public void Ocene()
{
    for (int i=0;i<8;i++)
    {
        System.out.print("Vnesi oceno "+(i+1)+" :");
        ocene[i]=BranjePodatkov.preberiInt();
    }
}
public float Pov()
{
    float p=0;
    for (int j=0;j<8;j++)
        p+=ocene[j];
    p/=8;
    p*=10;
    p=Math.round(p);
    p/=10;
    return p;
}
public void Izpisi()
{
    System.out.println("Vpisna Stevilka Studenta: "+vpisnaStevilka);
    System.out.println("Ime Studenta: "+ime);
    System.out.println("Priimek Studenta: "+priimek);
    System.out.println("Povprečna Ocena Studenta: "+Pov());
    System.out.println("_____");
}
public boolean manjsi(Element b)
{
    if(this.Pov() < ((Student2)b).Pov()) return true;
    else return false;
}
public Student2()
{
    System.out.print("Vnesi vpisno stevilko studenta:");
}

```

```

VpSt(BranjePodatkov.preberiInt());
System.out.print("Vnesi ime studenta:");
Ime(BranjePodatkov.preberiString());
System.out.print("Vnesi priimek studenta:");
Priimek(BranjePodatkov.preberiString());
Ocene();
}
}

```

```

public class TestStudent2
{
public static void main(String args[])
{
final int ST_STUD=5;
Student2[] studenti = new Student2[ST_STUD];
for (int i=0;i<ST_STUD;i++)
studenti[i] = new Student2();
Sortiraj.sort(studenti);
for (int j=0;j<ST_STUD;j++)
studenti[j].Izpis();
}
}

```

HANOI

```

/**
 * class Hanoi - Naloga C*<p>
 *
 * Podane so tri palice in n različno velikih obročev, ki jih lahko zlagamo
 * enega na drugega in tako tvorimo stolpiče. Na začetku je vseh n obročev
 * zloženih po velikosti na prvi palici, mi pa jih moramo prestaviti na drugo
 * palico upoštevajoč naslednja pravila:
 * · na vsakem koraku lahko prestavimo samo en obroč;
 * · večjega obroča ne smemo prestaviti na manjšega;
 * · tretjo palico lahko uporabimo kot pomožno odlagališče.
 *
 * Napišite program za grafični prikaz zaporedja korakov, ki so potrebni, da
 * prestavimo obroč s prve palice na drugo.
 *
 * @version: 1.0 (2003-01-05)
 * @author: roman@impresija.com
 */

```

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

/**
 * panel kamor risemo
 */
class Panel1 extends JPanel{
    Board bd;
}

```

```

boolean solved=false;

Panel1(Board b){ //constructor
    bd=b;
}

public void paintComponent(Graphics g){
    super.paintComponent(g);
    //System.out.println("--");
    g.setColor(Color.white);
    g.fillRect(0,0,400,200);
    drawHanoi(g);
    g.setColor(Color.black);
    if(solved){
        g.drawString("done", 200,165);
    }
}

void drawHanoi(Graphics g){
    int width=0, disc=0 ;
    // draw discs
    for (int p=0; p<=2; p++) { //for stack 1,2,3...
        for (int d=0; d<=bd.getStackTop(p); d++) {
            disc = bd.getDisc(d,p);
            if (disc!=0) {
                width = (disc*10);
                drawDisc( (p*100+100)-((int)(width/2)), 150-((d+1)*10),width,g );
            }
        }
    }
}

// draw individual disc
void drawDisc(int x,int y,int width, Graphics g) {
    Color c = g.getColor();
    g.setColor(Color.blue);
    g.fillRect(x,y,width,10);
    g.setColor(c);
}

/**
 * Dejanska rekurzivna metoda,
 * ki resi matematicni del naloge
 */
public void solveHanoi(int discs,int source,int aux,int target) {
    if(solved) return; // nothing to solve
    if (discs==0) return; // end recursion if no discs left
    solveHanoi(discs-1,source,target,aux); // recursive call #1
    bd.moveDisc(source,target); // move disc
    repaint(); // redraw Panel
}

```

```

        try { Thread.sleep(1000);}           // delay
        catch (InterruptedException e){}
        //System.out.println(++ "");
        solveHanoi(discs-1,aux,source,target); // recursive call #2
    }

    public void rrun(){ //own run method
        solveHanoi(bd.getDiscs(),0,1,2); //solveHanoi(#ofDiscs,stack1,stack2,stack3)
        solved=true;
        repaint();
    };
}

class Poslusalec extends WindowAdapter{
    public void windowClosing(WindowEvent e){
        System.exit(0);
    }
}

class Prikaz extends JFrame{
    Panel1 pan1;
    boolean solved=false;
    Board bd;

    public Prikaz(Board b){
        int width=400;
        int height=200;
        bd=b;

        // deklaracija poslušalca
        WindowListener p=new Poslusalec();
        addWindowListener(p);

        setTitle("Hanojski stolpici");
        setSize(width,height);
        setResizable(false);

        Container vsebina=getContentPane();

        pan1=new Panel1(bd);
        vsebina.add(pan1);
    }
    public void rrun(){
        pan1.rrun();
    }
}

/**
 * class Board služi za
 * organizacijo obročev(discs)

```

```

* na palicah(stacks)
*/
class Board {
    private int stack[][], stackTop[]=new int[3], discs;

    Board(int discs) { //constructor
        this.discs = discs;
        stack = new int[discs][3];
        // put all the disks on the first stack
        for (int i=0; i<discs; i++) stack[i][0] = discs-i;
        stackTop[0] = discs-1 ;
        for (int i=1; i<3; i++) stackTop[i] = -1;
    }

    void setDisc(int d,int p) { stack[++stackTop[p]][p] = d ; }
    int getDisc(int d,int p) { return stack[d][p] ; }
    int getDiscs() { return discs ; }
    int getTopDisc(int p) { return stack[stackTop[p]--][p] ; }
    int getStackTop(int p) { return stackTop[p] ; }

    void moveDisc(int p1,int p2) {
        setDisc(getTopDisc(p1),p2) ;
    }
}

/**
 * glavni class naloge
 * klice ostale
 */
public class Hanoi{
    public static void main(String[] args){
        int discs=3; //stevilo obrocev;
        Board bd = new Board(discs);

        Prikaz frejm=new Prikaz(bd);
        frejm.show();

        frejm.rrun();
    }
}

```

Pregled programov

Ime datoteke	Opis
Prvi.java	Prvi program v javi
DemoSpr.java	Prikaz deklaracije spremenljivk, preprostih izračunov in

	izpisa
DemoPrireditveni.java	Prikaz prirejanja vrednosti s poudarkom na pretvorbi numeričnih tipov
DemoSwitch.java	Prikaz stavka switch
Vsota1.java	Izračun vsote prvih 200 naravnih števil s stavkom do ... while
Vsota2.java	Izračun vsote prvih 200 naravnih števil s stavkom while
Vsota3.java	Izračun vsote prvih 200 naravnih števil s stavkom for
Lovec.java	Izpis možnih potez lovca na šahovnici Primer preprostega programa, ki vsebuje samo metodo main()
NaslovFRI.java	Izpis naslova fakultete Primer metode brez argumentov
ObrestiProcedura.java	Izračun obresti v obliki procedure Primer metode z dvema argumentoma
ObrestiFunkcija.java	Izračun obresti v obliki funkcije Primer metode, ki vrača vrednost
Fibonacci.java	Izračun n-tega Fibonaccijevega števila Primer rekurzivne metode in ustrezna iterativna rešitev
PerfektnaStevila.java	Iskanje perfektnih števil z intervala od 1 do 1000 Primer programa, ki poleg metode main() vsebuje še eno metodo
MatFunkcije.java Tabela.java	Izračun kotne funkcije sinus in izpis tabele z vrednostmi za kote od 0 do 360 stopinj s korakom 30 Primer metode, ki temelji na rekurenčni relaciji Uporaba metode iz drugega razreda
BranjePodatkov.java TestBranja.java	Program za branje celih števil (tipa int), realnih števil (tipa double) in nizov
Delavec.java TestDelavec.java	Deklaracija in test razreda Delavec brez lastnega konstruktorja
Delavec1.java TestDelavec1.java	Deklaracija in test razreda Delavec z lastnim konstruktorjem
DemoOverload.java	Prikaz večkratnega definiranja metod (metoda za izračun obresti)
Delavec2.java TestDelavec2.java	Deklaracija in test razreda Delavec z večkratno definiranim konstruktorjem
DemoCharacter.java	Prikaz delovanja metod iz razreda Character
Sort.java	Sortiranje tabele celih števil
Demo3D.java	Prikaz branja in izpisa tridimenzionalne tabele
MnozenjeMatrik.java	Program za množenje matrik Primer programa za delo z dvodimenzionalnimi tabelami
TabelaDelavcev.java Delavec2.java	Branje in izpis podatkov o delavcih Primer programa za delo s tabelo objektov
Pregovor.java	Ugibanje črk, ki sestavljajo nek pregovor Prikaz metod za delo z nizi tipa String
DemoStringBuffer1.java	Prikaz konstruktorjev za nize tipa StringBuffer; ilustracija pojmov kapaciteta in dolžina niza
DemoStringBuffer2.java	Prikaz metod za delo z nizi tipa StringBuffer
DemoArgMain.java	Prikaz argumentov metode main()

Delavec4.java Delavec4Glavni.java	Izračun vrstnega reda delavcev glede na zaslužek v preteklem letu. Primer nekoliko zahtevnejšega programa za obdelavo tabele objektov. Eden izmed atributov objekta je zopet tabela.
Student.java IzredniStudent.java DemoDedovanje.java	Prikaz dedovanja. Razred IzredniStudent podeduje attribute in metode razreda Student.
Student1.java IzredniStudent1.java DemoRedefinicija.java	Prikaz redefinicije metod ob dedovanju. Razred IzredniStudent1 redefinira metodo izpisTipa().
Student2.java IzredniStudent2.java DemoSuper.java	Prikaz klica metode nadrazreda s pomočjo rezervirane besede super. Podrazred IzredniStudent2 redefinira metodo izpisiVse() in pri tem uporabi istoimensko metodo nadrazreda.
Student3.java IzredniStudent3.java DemoKonstruktor.java	Prikaz dejstva, da se ob kreiranju objekta, ki pripada podrazredu, res kličeta dva konstruktorja.
Student4.java IzredniStudent4.java DemoKonstruktor1.java	Prikaz inicializacije atributov: konstruktor nadrazreda poskrbi za inicializacijo atributov nadrazreda, konstruktor podrazreda pa za inicializacijo atributov, ki so deklarirani v podrazredu.
Student5.java IzredniStudent5.java DemoKonstruktor2.java	Prikaz konstruktorja nadrazreda z argumenti. Konstruktor podrazreda kliče konstruktor nadrazreda s super in ustreznimi dejanskimi parametri.
Krogi.java	Izračun števila krogov, ki so v kvadratu, sekajo kvadrat oziroma so v celoti izven kvadrata. Preprost prikaz zanke in razvejitve.
UrejenaTabela.java	Vstavljanje števil v urejeno tabelo.
Razdalje.java	Izračun razdalj med točkami v dvodimenzionalnem prostoru. Primer programa za delo z dvodimenzionalno tabelo.
Razdalje1.java	Isto kot Razdalje.java, le da so vse tabele deklarirane kot globalne spremenljivke.
Palindrom.java	Ugotavljanje, ali je niz palindrom. Primer preprostega programa, ki uporablja tip String. Vhodni niz je podan kot argument metode main().
Anagrami.java	Ugotavljanje, ali sta niza anagrama. Primer uporabe razreda StringBuffer.
Krogla.java TestKrogla.java	Deklaracija in test razreda Krogla. Primer spremenljivk in metod razreda ter spremenljivk in metod objekta
Zival.java Pes.java Krava.java Kaca.java DemoAbstraktni.java	Prikaz abstraktnega razreda in abstraktne metode. Abstraktni razred Zival deklarira abstraktno metodo oglasanje, ki je redefinirana v razredih Pes, Krava in Kaca.
Zival.java Pes.java Krava.java	Prikaz dinamičnega povezovanja metod. Spremenljivka tipa Zival lahko vsebuje naslov objekta tipa Pes, Krava ali Kaca. Izvede se metoda oglasanje() iz

Kaca.java DemoAbstraktni1.java	tistega podrazreda, katerega objekt je trenutno shranjen v spremenljivki z.
Zival.java Pes.java Krava.java Kaca.java DemoHeterogeneTabele.java	Prikaz tabele, ki vsebuje objekte različnih podrazredov.
Zival1.java Pes1.java Krava1.java DemoEquals.java	Prikaz redefinicije metode equals() v razredu Zival. Redefinirano metodo avtomatsko podedujeta podrazreda Pes in Krava.
OsnovnoOkno.java	Kreiranje okna tipa JFrame. Širina in višina okna sta enaki 0.
TestOkna.java	Kreiranje okna z razširitvijo tipa JFrame (ob zapiranju se program ne konča).
TestOkna1.java	Prekinitev izvajanja programa ob zapiranju okna. Deklaracija poslušalca.
TestOkna2.java	Odpiranje okna na podlagi podatkov o dejanski velikosti ekrana, izraženi s številom pik. Odpre okno sredi ekrana; širina in višina okna sta enaki polovici širine oziroma višine ekrana.
TestOkna3.java	Dodajanje panela.

```
public class Prvi
{
    public static void main(String[] args)
    {
        System.out.println("Nas prvi program");
    }
}
```

```
public class DemoSpr
{
    public static void main(String[] args)
    {
        // deklaracije spremenljivk
        float netoZasluzek;
        byte stOtrok=2;
        short ureNaDelu=172, cenaUre=1200;
        short otroskiDodatek=6500;
        float prispevki=23.5F; // prispevki v odstotkih

        // izracun zasluzka
        netoZasluzek = ureNaDelu*cenaUre*(1-prispevki/100)+stOtrok*otroskiDodatek;

        // izpis vrednosti
        System.out.println("Delavec je delal "+ureNaDelu+" ur.");
        System.out.print("Na uro je dobil ");
    }
}
```

```
System.out.print(cenaUre);
System.out.println(" SIT.");
System.out.println("Njegov bruto zaslupek je "+ureNaDelu*cenaUre+" SIT.");
System.out.println("Njegov neto zaslupek je "+netoZasluzek+" SIT.");
}
}
```

```
public class DemoPrireditveni
{
    public static void main(String[] args)
    {
        int a=10;
        short b=5;
        int rez1;
        float rez2;
        short rez3;
        byte rez4, rez5;

        rez1=a*b;
        rez2=a*b;
        rez3=(short)(a*b);
        rez4=(byte)(a*b);

        rez5=(byte)(rez4*b);

        System.out.println(rez1);
        System.out.println(rez2);
        System.out.println(rez3);
        System.out.println(rez4);
        System.out.println(rez5);
    }
}
```

```
class DemoSwitch
{
    public static void main(String[] args)
    {
        for(int ocena=1; ocena<=10; ++ocena)
            switch (ocena) // vsebuje oceno od 1 do 10
            {
                case 6:
                    System.out.println("zadostno");
                    break;
                case 7:
                    System.out.println("dobro");
                    break;
                case 8:
                    System.out.println("prav dobro");
                    break;
                case 9:
                    System.out.println("prav dobro");
                    break;
            }
    }
}
```

```

        case 10:
            System.out.println("odlicno");
            break;
        default:
            System.out.println("nezadostno");
    }
}
}

```

```

public class Vsota1
{
    public static void main(String[] args)
    {
        int s=0; // zacetna vrednost vsote
        int i=1; // prvi clen, zacetna vrednost Ÿtevca
        do
        {
            s=s+i; // priŸtevanje clena
            i=i+1; // naslednji clen, povecanje Ÿtevca
        } while (i<=200);
        System.out.println("Vsota je "+s);
    }
}

```

```

public class Vsota2
{
    public static void main(String[] args)
    {
        int s=0; // zacetna vrednost vsote
        int i=1; // prvi clen, zacetna vrednost Ÿtevca
        while (i<=200)
        {
            s=s+i; // priŸtevanje clena
            i=i+1; // naslednji clen, povecanje Ÿtevca
        }
        System.out.println("Vsota je "+s);
    }
}

```

```

public class Vsota3
{
    public static void main(String[] args)
    {
        int s=0;
        for (int i=1; i<=200; i=i+1)
        {
            s=s+i; // priŸtevanje clena
        }
        System.out.println("Vsota je "+s);
    }
}

```

```

public class Lovec
{

```

```

public static void main(String[] args)
{
    int zacVrstica=3, zacKolona=2;
    int v, k;
    for (v=1; v<=8; ++v)
    {
        for (k=1; k<=8; ++k)
            if ((v-k==zacVrstica-zacKolona) || (v+k==zacVrstica+zacKolona))
                System.out.print("*");
            else if ((v+k)%2==0)
                System.out.print("B");
            else
                System.out.print("C");
        System.out.println();
    }
}

```

```

class NaslovFRI
{
    public static void main(String[] args)
    {
        naslov();
    }

    public static void naslov()
    {
        System.out.print("Fakulteta za racunalnistvo");
        System.out.println(" in informatiko");
        System.out.println();
        System.out.println("Trzaska 25");
        System.out.println("1000 Ljubljana");
    }
}

```

```

class ObrestiProcedura
{
    public static void main(String[] args)
    {
        double g=100000, o=5.;
        poEnemLetu(g,o);
        poEnemLetu(20000,8);
    }

    public static void poEnemLetu(double glavnica, double obrMera)
    {
        double novoStanje;
        novoStanje=glavnica+glavnica*obrMera/100;
        System.out.println("Po enem letu dobimo "+novoStanje);
    }
}

```

```

class ObrestiFunkcija

```

```

{
public static void main(String[] args)
{
    double g=100000, o=5., ns;
    ns=poEnemLetu(g,o);
    System.out.println("Stanje po enem letu:"+ns);
    System.out.println("Stanje po enem letu:"+poEnemLetu(20000,8));
}

public static double poEnemLetu
(double glavnica, double obrMera)
{
    double novoStanje;
    novoStanje=glavnica+glavnica*obrMera/100;
    return novoStanje; // namesto izpisa vrne vrednost
}

}
}

public class Fibonacci
{
public static void main(String[] args)
{
    int n=6;
    System.out.println(n+". Fibonaccijevo stevilo (rekurzivno) je "+fib(n));
    System.out.println(n+". Fibonaccijevo stevilo (iterativno) je "+fibIte(n));
}

public static int fib(int n)
{
    if (n<=1)
        return n;
    else
        return fib(n-1)+fib(n-2);
}

public static int fibIte(int n)
{
    int x=1,y=0;           // zadnje in predzadnje Fibonaccijevo Ÿtevililo
    int z;
    for (int i=2; i<=n; ++i)
    {
        z=x;               // zacasno shranimo zadnje Fibonaccijevo Ÿtevililo
        x=x+y;             // naslednje Fibonaccijevo Ÿtevililo
        y=z;               // prejŸnje Fibonaccijevo Ÿtevililo
    }
    return x;
}

}

public class PerfektnaStevila
{
public static void main(String[] args)

```

```

{
    int stevilo;
    for (stevilo=1; stevilo<=1000; ++stevilo)
        if (perfektno(stevilo))
            System.out.println(stevilo);
}

public static boolean perfektno(int st)
{
    int delitelj, vsota=0;
    for (delitelj=1; delitelj<=st/2; ++delitelj)
        if (st % delitelj == 0) vsota+=delitelj;
    if (vsota==st)
        return true;
    else
        return false;
}
}

```

```

public class MatFunkcije
{
    static double eps=0.00001;

    public static double sinus(double x)
    {
        double vsota, clen;
        int k;
        clen=x;
        k=1;
        vsota=clen;
        while (Math.abs(clen)/Math.abs(vsota)>eps)
        {
            k+=2;           // k=k+2;
            clen=-clen*x*x/(k*(k-1));
            vsota+=clen;    // vsota=vsota+clen;
        }
        return vsota;
    }
}

```

```

public class Tabela
{
    public static void main(String[] args)
    {
        int kot;
        System.out.println("Kot      Sinus");
        System.out.println("-----");
        for (kot=0; kot<=360; kot+=30)
        {
            System.out.println(kot+"      "+MatFunkcije.sinus(Math.PI/180*kot));
        }
    }
}

```

```

}
import java.util.*;
import java.io.*;

public class BranjePodatkov {

    private static boolean uspeh = true;

    public static int preberiInt() {
        int result = 0;

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st;

        do {
            try {
                st = new StringTokenizer(br.readLine());
                result = Integer.parseInt(st.nextToken());
                uspeh = true;
            }
            catch(NumberFormatException nfe) {
                System.out.print("Napacen format stevila ! Ponovitev vnosa : ");
                uspeh = false;
            }
            catch(IOException ioe) {
                System.out.print("IO napaka pri branju stevila ! Ponovitev vnosa : ");
                uspeh = false;
            }
            catch(NoSuchElementException nsee) {
                System.out.print("Prazen vnos ! Ponovitev vnosa : ");
                uspeh = false;
            }
            catch(NullPointerException npe) {
                System.out.print("Prekinittev ! ");
                uspeh = false;
            }
        }
        while (! BranjePodatkov.uspeh);

        return result;
    }

    public static double preberiDouble() {
        double result = 0;

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st;

```

```

do {
    try {
        st = new StringTokenizer(br.readLine());
        result = new Double(st.nextToken()).doubleValue();
        uspeh = true;
    }

    catch(NumberFormatException nfe) {
        System.out.print("Napacen format stevila ! Ponovitev vnosa : ");
        uspeh = false;
    }
    catch(IOException ioe) {
        System.out.print("IO napaka pri branju stevila ! Ponovitev vnosa : ");
        uspeh = false;
    }
    catch(NoSuchElementException nsee) {
        System.out.print("Prazen vnos ! Ponovitev vnosa : ");
        uspeh = false;
    }
    catch(NullPointerException npe) {
        System.out.println("Prekinitev !");
        uspeh = false;
    }
}
while (! BranjePodatkov.uspeh);

return result;
}

public static String preberiString() {
    String result = null;

    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

    do {
        try {
            result = br.readLine();
            uspeh = true;
        }
        catch(IOException ioe) {
            System.out.print("IO napaka pri branju stringa ! Ponovitev vnosa : ");
            uspeh = false;
        }
    }
    while (! BranjePodatkov.uspeh);

    return result;
}
}

```

```

public class TestBranja

```



```

{
public static void main(String args[])
{
    int st;
    String bes;
    double re;

    System.out.print("Celo stevilo: ");
    st = BranjePodatkov.preberiInt();
    System.out.println("Prebrana vrednost: "+ st);
    System.out.println();

    System.out.print("Niz: ");
    bes = BranjePodatkov.preberiString();
    System.out.println("Prebrana vrednost: "+ bes);
    System.out.println();

    System.out.print("Realno stevilo: ");
    re = BranjePodatkov.preberiDouble();
    System.out.println("Prebrana vrednost: "+ re);
    System.out.println();
}
}

```

```

public class Delavec
{
    // atributi

    private int matStev;
    private String priimek;
    private String ime;
    private int stUr;

    // metode

    public void vpisiMatSt(int st)
    {
        matStev=st;
    }

    public void vpisiPriimek(String p)
    {
        priimek=p;
    }

    public void vpisiIme(String i)
    {
        ime=i;
    }
}

```

```

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

public int vrniMatSt()
{
    return matStev;
}

public String vrniPriimek()
{
    return priimek;
}

public String vrniIme()
{
    return ime;
}

public int vrniStUr()
{
    return stUr;
}

public int izracunajBrutoOD(int cenaUre)
{
    return stUr*cenaUre;
}

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

```

```

public class TestDelavec
{
    public static void main(String[] args)
    {
        // kreiranje objekta
        Delavec d=new Delavec();

        // vpis vrednosti atributov
        d.vpisiMatSt(234);
        d.vpisiPriimek("Novak");
        d.vpisiIme("Janez");
        d.vpisiStUr(182);
    }
}

```

```

// izpis podatkov s pomocjo posameznih metod
System.out.println("Maticna stevilka: "+d.vrniMatSt());
System.out.println("Priimek in ime: "+d.vrniPriimek()+" "+d.vrniIme());
System.out.println("Stevilo ur: "+d.vrniStUr());

// izpis bruto osebnega dohodka
System.out.println("Bruto OD: "+d.izracunajBrutoOD(1500));

// izpis podatkov z metodo izpisiVse
d.izpisiVse();
}
}

```

```

public class Delavec1
{

// atributi

private int matStev;
private String priimek;
private String ime;
private int stUr;

// konstruktor

Delavec1(int ms, String p, String i)
{
    matStev=ms;
    priimek=p;
    ime=i;
}

// metode

public void vpisiMatSt(int st)
{
    matStev=st;
}

public void vpisiPriimek(String p)
{
    priimek=p;
}

public void vpisiIme(String i)
{
    ime=i;
}

public void vpisiStUr(int u)

```

```

    {
        stUr=u;
    }

    public int vrniMatSt()
    {
        return matStev;
    }

    public String vrniPriimek()
    {
        return priimek;
    }

    public String vrniIme()
    {
        return ime;
    }

    public int vrniStUr()
    {
        return stUr;
    }

    public int izracunajBrutoOD(int cenaUre)
    {
        return stUr*cenaUre;
    }

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

```

```

public class TestDelavec1
{
    public static void main(String[] args)
    {

        // kreiranje objekta z lastnim konstruktorjem
        Delavec1 d=new Delavec1(234,"Novak","Janez");

        // vpis stevila ur
        d.vpisiStUr(182);

        // izpis podatkov s pomocjo posameznih metod
        System.out.println("Maticna stevilka: "+d.vrniMatSt());
        System.out.println("Priimek in ime: "+d.vrniPriimek()+" "+d.vrniIme());
    }
}

```

```
System.out.println("Stevilo ur: "+d.vrniStUr());

// izpis bruto osebnega dohodka
System.out.println("Bruto OD: "+d.izracunajBrutoOD(1500));

// izpis podatkov z metodo izpisiVse
d.izpisiVse();
}
}
```

```
public class DemoOverload
{
    public static void main(String[] args)
    {
        System.out.println(obresti(1000.,0.08));
        System.out.println(obresti(1000.,8));
    }

    public static double obresti(double g, double om)
    {
        return g*om;
    }

    public static double obresti(double g, int om)
    {
        return g*om/100;
    }
}
```

```
public class Delavec2
{

// zgled z vec konstruktorji

// atributi

private int matStev;
private String priimek;
private String ime;
private int stUr;

// konstruktorji

Delavec2()
{
    matStev=9999;
}

Delavec2(int ms)
{
    matStev=ms;
}
```

```
Delavec2(int ms, String p, String i)
{
    matStev=ms;
    priimek=p;
    ime=i;
}
```

```
// metode
```

```
public void vpisiMatSt(int st)
{
    matStev=st;
}
```

```
public void vpisiPriimek(String p)
{
    priimek=p;
}
```

```
public void vpisiIme(String i)
{
    ime=i;
}
```

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

```
public int vrniMatSt()
{
    return matStev;
}
```

```
public String vrniPriimek()
{
    return priimek;
}
```

```
public String vrniIme()
{
    return ime;
}
```

```
public int vrniStUr()
{
    return stUr;
}
```

```

public int izracunajBrutoOD(int cenaUre)
{
    return stUr*cenaUre;
}

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

public class TestDelavec2
{
    public static void main(String[] args)
    {

        // kreiranje objekta z različnimi konstruktorji
        Delavec2 d1=new Delavec2();
        Delavec2 d2=new Delavec2(5555);
        Delavec2 d3=new Delavec2(234,"Novak","Janez");

        // izpis podatkov
        d1.izpisiVse();
        d2.izpisiVse();
        d3.izpisiVse();
    }
}
}
}
}

public class DemoCharacter
{
    public static void main(String[] args) throws Exception
    {
        char znak;

        do
        {
            System.out.println("Vtipkaj znak: ");
            znak=(char)System.in.read();
            System.in.read(); System.in.read(); // tipka Enter

            if (Character.isLetterOrDigit(znak))
            {
                System.out.println(znak+" je crka ali stevilka.");
                if (Character.isLetter(znak))
                {
                    if (Character.isUpperCase(znak))
                        System.out.println(znak+" je velika crka.");
                    else
                        System.out.println(znak+" je mala crka.");
                }
            }
        }
    }
}
}
}
}
}

```

```

    }
    else
        System.out.println(znak+" je stevilka.");
    }
    else if (Character.isWhitespace(znak))
        System.out.println(znak+" je \"whitespace\".");
    else
        System.out.println(znak+" je locilo ali poseben znak.");
    } while (znak!='#');
}
}

```

```

public class Sort
{
    public static void main(String[] args)
    {
        int[] a= {24,60,55,33,98,6,43,15};
        int indeks;
        sortiraj(a);
        for (indeks=0; indeks<a.length; ++indeks)
            System.out.println(a[indeks]);
    }

    public static void sortiraj(int[] a)
    {
        int i,j,iMin,vMin;
        for (i=0; i<=a.length-2; ++i)
        {
            iMin=i; vMin=a[i];
            for (j=i+1; j<=a.length-1; ++j)
                if (a[j]<vMin)
                {
                    iMin=j; vMin=a[j];
                }
            a[iMin]=a[i]; a[i]=vMin;
        }
    }
}

```

```

public class Demo3D
{
    public static void main(String[] args)
    {
        int[][][] a=new int[3][4][2];
        for(int i=0; i<a.length; ++i)
            for (int j=0; j<a[i].length; ++j)
                for (int k=0; k<a[i][j].length; ++k)
                    a[i][j][k]=BranjePodatkov.preberiInt();

        for(int i=0; i<a.length; ++i)
            for (int j=0; j<a[i].length; ++j)
                for (int k=0; k<a[i][j].length; ++k)

```



```

        System.out.println(i+", "+j+", "+k+" "+a[i][j][k]);
    }
}

```

```

public class MnozenjeMatrik
{
    public static void main(String[] args)
    {
        int[][] a={{2,6,5,5},{3,8,6,3},{1,5,0,2}};
        int[][] b={{3,2},{2,4},{1,5},{2,2}};
        int[][] c=new int[a.length][b[0].length];
        mnozi(a,b,c);
        izpisi(c);
    }

    public static void mnozi(int[][] a, int[][] b, int c[][])
    {
        int i,j,k;
        int vb=b.length; // stevilo vrstic matrike b (za pristevanje posameznih produktov)
        for (i=0; i<=c.length-1; ++i)
            for (j=0; j<=c[0].length-1; ++j)
                {
                    c[i][j]=0;
                    for (k=0; k<=vb-1; ++k)
                        c[i][j]+=a[i][k]*b[k][j];
                }
    }

    public static void izpisi(int[][] c)
    {
        int i,j;
        for (i=0; i<=c.length-1; ++i)
            {
                for (j=0; j<=c[0].length-1; ++j)
                    System.out.print(c[i][j]+" ");
                System.out.println();
            }
    }
}

```

```

public class TabelaDelavcev
{
    public static void main(String[] args)
    {
        Delavec2[] td=new Delavec2[5];
        int matSt,stUr;
        String priimek,ime;

        // v zanki beremo podatke za posamezne delavce in kreiramo objekte
        for (int i=0; i<=4; ++i)
            {
                System.out.print("Maticna stevilka:");
            }
    }
}

```

```

matSt=BranjePodatkov.preberiInt();
System.out.print("Priimek:");
priimek=BranjePodatkov.preberiString();
System.out.print("Ime:");
ime=BranjePodatkov.preberiString();
System.out.print("Stevilo ur:");
stUr=BranjePodatkov.preberiInt();
td[i]=new Delavec2(matSt,priimek,ime);
td[i].vpisiStUr(stUr);
}

for (int i=0; i<=4; ++i)
{
    System.out.print(td[i].vrniMatSt()+" ");
    System.out.print(td[i].vrniPriimek()+" ");
    System.out.print(td[i].vrniIme()+" ");
    System.out.println(td[i].vrniStUr());
}
}
}
}
}

public class Pregovor
{
    public static void main(String[] args) throws Exception
    {
        String iskaniNiz="Rana ura, zlata ura.";
        String prikazaniNiz="***** ***, ***** ***.";
        char crka; // vtipkana crka
        int poz; // pozicija vtipkane crke v iskaniNiz
        System.out.println(prikazaniNiz);
        while (prikazaniNiz.indexOf('*') != -1)
        {
            System.out.print("Vtipkaj crko:");
            crka=(char)System.in.read();
            System.in.read(); System.in.read(); // Enter
            poz=iskaniNiz.indexOf(crka);
            if (poz == -1)
                System.out.println("Te crke ni, ugibaj ponovno!");
            else
            {
                do
                {
                    prikazaniNiz=prikazaniNiz.substring(0,poz)+crka+
                        prikazaniNiz.substring(poz+1,prikazaniNiz.length());
                    poz=iskaniNiz.indexOf(crka,poz+1);
                } while (poz != -1);
                System.out.println(prikazaniNiz);
            }
        }
    }
}
}
}
}
}

```

```

public class DemoStringBuffer1
{
    public static void main(String[] args)
    {
        // uporaba razlicnih konstruktorjev
        System.out.println("Konstruktor brez parametrov");
        StringBuffer niz1=new StringBuffer();
        System.out.println(""+niz1+"");
        System.out.println("Dolzina  :"+niz1.length());
        System.out.println("Kapaciteta:"+niz1.capacity());

        System.out.println("Konstruktor s parametrom tipa String");
        StringBuffer niz2=new StringBuffer("Dobro jutro");
        System.out.println(""+niz2+"");
        System.out.println("Dolzina  :"+niz2.length());
        System.out.println("Kapaciteta:"+niz2.capacity());

        System.out.println("Konstruktor s parametrom kapaciteta");
        StringBuffer niz3=new StringBuffer(50);
        System.out.println(""+niz3+"");
        System.out.println("Dolzina  :"+niz3.length());
        System.out.println("Kapaciteta:"+niz3.capacity());

        // spreminjanje dolzine

        System.out.println("Dolzino niza postavimo na 20");
        niz2.setLength(20);
        System.out.println(""+niz2+"");
        System.out.println("Dolzina  :"+niz2.length());
        System.out.println("Kapaciteta:"+niz2.capacity());

        System.out.println("Dolzino niza postavimo na 30");
        niz2.setLength(30);
        System.out.println(""+niz2+"");
        System.out.println("Dolzina  :"+niz2.length());
        System.out.println("Kapaciteta:"+niz2.capacity());

        System.out.println("Dolzino niza postavimo na 10");
        niz2.setLength(10);
        System.out.println(""+niz2+"");
        System.out.println("Dolzina  :"+niz2.length());
        System.out.println("Kapaciteta:"+niz2.capacity());
    }
}

```

```

public class DemoStringBuffer2
{
    public static void main(String[] args)
    {
        String podniz;
        char zn;
    }
}

```

```

int poz;

// zacetni niz
StringBuffer niz=new StringBuffer("Dobro jutro");
izpis(niz);

// dodajanje niza na koncu
niz.append(" vam zelim");
izpis(niz);

// dodajanje znaka na koncu
niz.append('.');
izpis(niz);

// vrivanje niza
niz.insert(11," in prijeten danasnji dan");
izpis(niz);

// brisanje podniza
niz.delete(0,14);
izpis(niz);

// brisanje znaka
niz.deleteCharAt(0);
izpis(niz);

// vpis znaka (povozi prejÜnji znak)
niz.setCharAt(0,'P');
izpis(niz);

// zamenjava podniza z drugim nizom
niz.replace(0,6,"Uspes");
izpis(niz);

// branje znaka
zn=niz.charAt(4);
System.out.println("Znak na poziciji 4:"+zn);
izpis(niz);

// branje podniza
podniz=niz.substring(4,11);
System.out.println("Znaki od 4. do 10. mesta:"+podniz);
izpis(niz);

// iskanje prve pojavitve niza "dan"
poz=niz.indexOf("dan");
System.out.println(poz);

// iskanje naslednje pojavitve niza "dan"
poz=niz.indexOf("dan",poz+1);

```

```

    System.out.println(poz);
}

public static void izpis(StringBuffer niz)
{
    System.out.println("*"+niz+"*");
    System.out.println("Dolzina  :"+niz.length());
    System.out.println("Kapaciteta:"+niz.capacity());
}
}

```

```

public class DemoArgMain
{
    public static void main(String[] args)
    {
        System.out.println(args.length);
        for (int i=0; i<args.length; ++i)
            System.out.println(args[i]);
    }
}

```

```

public class Delavec4
{

    // atributi

    private int matStev;
    private String priimek;
    private String ime;
    private double[] od;

    // konstruktor

    Delavec4()
    {
        System.out.print("Maticna stevilka:");
        matStev=BranjePodatkov.preberiInt();
        System.out.print("Priimek:");
        priimek=BranjePodatkov.preberiString();
        System.out.print("Ime:");
        ime=BranjePodatkov.preberiString();
        od=new double[12];
        for(int i=0; i<od.length; ++i)
        {
            System.out.print("Osebni dohodek za "+Integer.toString(i+1)+". mesec:");
            od[i]=BranjePodatkov.preberiDouble();
        }
    }

    // metode

```

```

public void vpisiMatSt(int st)
{
    matStev=st;
}

public void vpisiPriimek(String p)
{
    priimek=p;
}

public void vpisiIme(String i)
{
    ime=i;
}

public void vpisiOd(double[] osDoh)
{
    for(int i=0; i<od.length; ++i)
    {
        od[i]=osDoh[i];
    }
}

public int vrniMatSt()
{
    return matStev;
}

public String vrniPriimek()
{
    return priimek;
}

public String vrniIme()
{
    return ime;
}

public double vrniOd(int i)
{
    return od[i];
}

public double vrniVsotoOd()
{
    double vsota=0;
    for (int i=0; i<od.length; ++i)
        vsota+=od[i];
    return vsota;
}

```

```

public void izpisiVse()
{
    System.out.println("Maticna stevilka: "+matStev);
    System.out.println("Priimek in ime: "+priimek+' '+ime);
    for (int i=0; i<od.length; ++i)
        System.out.println("Osebni dohodek za "+Integer.toString(i+1)+" mesec:"+od[i]);
}
}

```

```

public class Delavec4Glavni
{
    static final int ST_DEL=5;
    public static void main(String[] args)
    {
        Delavec4[] td=new Delavec4[ST_DEL];
        for (int i=0; i<ST_DEL; ++i)
            td[i]=new Delavec4();
        System.out.println();
        System.out.println("Pred sortiranjem:");
        izpisi(td);
        Sortiraj.sort(td);
        System.out.println();
        System.out.println("Po sortiranju:");
        izpisi(td);
    }
    static void izpisi(Delavec4[] td)
    {
        for (int i=0; i<ST_DEL; ++i)
        {
            System.out.print(td[i].vrniMatSt());
            System.out.print(" "+td[i].vrniPriimek()+" "+td[i].vrniIme());
            System.out.println(" "+td[i].vrniVsotoOd());
        }
    }
}

```

```

public class Student
{
    private int vpisSt;
    private String priimek;
    private String ime;

    public void vpisiVpisSt(int st)
    {
        vpisSt=st;
    }

    public void vpisiPriimek(String p)
    {
        priimek=p;
    }
}

```

```
public void vpisiIme(String i)
{
    ime=i;
}
```

```
public int vrniVpisSt()
{
    return vpisSt;
}
```

```
public String vrniPriimek()
{
    return priimek;
}
```

```
public String vrniIme()
{
    return ime;
}
```

```
}
```

```
public class IzredniStudent extends Student
```

```
{
```

```
    private int znesekSolnine;
```

```
public void vpisiZnesekSolnine(int solnina)
{
    znesekSolnine=solnina;
}
```

```
public int vrniZnesekSolnine()
{
    return znesekSolnine;
}
```

```
}
```

```
public class DemoDedovanje
```

```
{
```

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

```
    {
```

```
        Student s=new Student();
```

```
        IzredniStudent is=new IzredniStudent();
```

```
        s.vpisiVpisSt(63020888);
```

```
        s.vpisiPriimek("Novak");
```

```
        s.vpisiIme("Janez");
```

```
        System.out.println("Student");
```

```
        System.out.println(s.vrniVpisSt()+" "+s.vrniPriimek()+" "+s.vrniIme());
```

```
        is.vpisiVpisSt(63020999);
```

```
        is.vpisiPriimek("Bogataj");
```



```
is.vpisiIme("Franc");
is.vpisiZnesekSolnine(250000);
System.out.println("IzredniStudent");
System.out.println(is.vrniVpisSt()+" "+is.vrniPriimek()+" "+is.vrniIme());
System.out.println("je placal "+is.vrniZnesekSolnine()+" SIT solnine.");
}
}
```

```
public class Student1
{
    private int vpisSt;
    private String priimek;
    private String ime;

    public void vpisiVpisSt(int st)
    {
        vpisSt=st;
    }

    public void vpisiPriimek(String p)
    {
        priimek=p;
    }

    public void vpisiIme(String i)
    {
        ime=i;
    }

    public int vrniVpisSt()
    {
        return vpisSt;
    }

    public String vrniPriimek()
    {
        return priimek;
    }

    public String vrniIme()
    {
        return ime;
    }

    public void izpisTipa()
    {
        System.out.println("Student");
    }
}
```

```
public class IzredniStudent1 extends Student1
```

```

{
private int znesekSolnine;

public void izpisTipa()
{
System.out.println("IzredniStudent");
}

public void vpisiZnesekSolnine(int solnina)
{
znesekSolnine=solnina;
}

public int vrniZnesekSolnine()
{
return znesekSolnine;
}
}

```

```

public class DemoRedefinicija
{
public static void main(String[] args)
{
Student1 s=new Student1();
IzredniStudent1 is=new IzredniStudent1();

s.vpisiVpisSt(63020888);
s.vpisiPriimek("Novak");
s.vpisiIme("Janez");
s.izpisTipa();
System.out.println(s.vrniVpisSt()+" "+s.vrniPriimek()+" "+s.vrniIme());

is.vpisiVpisSt(63020999);
is.vpisiPriimek("Bogataj");
is.vpisiIme("Franc");
is.vpisiZnesekSolnine(250000);
is.izpisTipa();
System.out.println(is.vrniVpisSt()+" "+is.vrniPriimek()+" "+is.vrniIme());
System.out.println("je placal "+is.vrniZnesekSolnine()+" SIT solnine.");
}
}

```

```

public class Student2
{
private int vpisSt;
private String priimek;
private String ime;

public void vpisiVpisSt(int st)
{
vpisSt=st;
}
}

```

```

public void vpisiPriimek(String p)
{
    priimek=p;
}

public void vpisiIme(String i)
{
    ime=i;
}

public int vrniVpisSt()
{
    return vpisSt;
}

public String vrniPriimek()
{
    return priimek;
}

public String vrniIme()
{
    return ime;
}

public void izpisTipa()
{
    System.out.println("Student");
}

public void izpisiVse()
{
    System.out.println(vpisSt);
    System.out.println(priimek+" "+ime);
}
}

```

```

public class IzredniStudent2 extends Student2
{
    private int znesekSolnine;

    public void izpisTipa()
    {
        System.out.println("IzredniStudent");
    }

    public void izpisiVse()
    {
        super.izpisiVse();
        System.out.println(znesekSolnine);
    }
}

```

```

}

public void vpisiZnesekSolnine(int solnina)
{
    znesekSolnine=solnina;
}

public int vrniZnesekSolnine()
{
    return znesekSolnine;
}
}

```

```

public class DemoSuper
{
    public static void main(String[] args)
    {
        Student2 s=new Student2();
        IzredniStudent2 is=new IzredniStudent2();

        s.vpisiVpisSt(63020888);
        s.vpisiPriimek("Novak");
        s.vpisiIme("Janez");
        s.izpisTipa();
        s.izpisiVse();

        is.vpisiVpisSt(63020999);
        is.vpisiPriimek("Bogataj");
        is.vpisiIme("Franc");
        is.vpisiZnesekSolnine(250000);
        is.izpisTipa();
        is.izpisiVse();
    }
}

```

```

public class Student3

// enak kot Student, le da je dodan konstruktor
// konstruktor vsebuje samo izpis sporočila kot dokaz, da se v resnici izvede

{
    private int vpisSt;
    private String priimek;
    private String ime;

    Student3()
    {
        System.out.println("V konstruktorju Student3");
    }
}

```

```

public void vpisiVpisSt(int st)
{
    vpisSt=st;
}

public void vpisiPriimek(String p)
{
    priimek=p;
}

public void vpisiIme(String i)
{
    ime=i;
}

public int vrniVpisSt()
{
    return vpisSt;
}

public String vrniPriimek()
{
    return priimek;
}

public String vrniIme()
{
    return ime;
}
}

```

```

public class IzredniStudent3 extends Student3

```

```

// enak kot IzredniStudent, le da vsebuje konstruktor
// konstruktor vsebuje samo izpis sporočila kot dokaz, da se v resnici izvede

```

```

{
    private int znesekSolnine;

    IzredniStudent3()
    {
        System.out.println("V konstruktorju IzredniStudent");
    }

    public void vpisiZnesekSolnine(int solnina)
    {
        znesekSolnine=solnina;
    }

    public int vrniZnesekSolnine()
    {

```

```

    return znesekSolnine;
}
}

```

```

public class DemoKonstruktor

// spremenjena je samo inicializacija obeh objektov
// preostala koda je enaka kot v DemoDedovanje.java

{
public static void main(String[] args)
{
    Student3 s=new Student3();

    s.vpisiVpisSt(63020888);
    s.vpisiPriimek("Novak");
    s.vpisiIme("Janez");
    System.out.println("Student");
    System.out.println(s.vrniVpisSt()+" "+s.vrniPriimek()+" "+s.vrniIme());

    IzredniStudent3 is=new IzredniStudent3();
    is.vpisiVpisSt(63020999);
    is.vpisiPriimek("Bogataj");
    is.vpisiIme("Franc");
    is.vpisiZnesekSolnine(250000);
    System.out.println("IzredniStudent");
    System.out.println(is.vrniVpisSt()+" "+is.vrniPriimek()+" "+is.vrniIme());
    System.out.println("je placal "+is.vrniZnesekSolnine()+" SIT solnine.");
}
}

```

```

public class Student4

// enak kot Student3, le da konstruktor inicializira vrednosti atributov

{
private int vpisSt;
private String priimek;
private String ime;

Student4()
{
    System.out.println("V konstruktorju Student4");
    vpisSt=63020000;
    priimek="xxxxx";
    ime="yyyyy";
}

public void vpisiVpisSt(int st)
{
    vpisSt=st;
}
}

```

```
public void vpisiPriimek(String p)
{
    priimek=p;
}
```

```
public void vpisiIme(String i)
{
    ime=i;
}
```

```
public int vrniVpisSt()
{
    return vpisSt;
}
```

```
public String vrniPriimek()
{
    return priimek;
}
```

```
public String vrniIme()
{
    return ime;
}
```

```
}
```

```
public class IzredniStudent4 extends Student4
```

```
// enak kot IzredniStudent3, le da konstruktor inicializira atribut podrazreda
```

```
{
    private int znesekSolnine;
```

```
IzredniStudent4()
{
    System.out.println("V konstruktorju IzredniStudent4");
    znesekSolnine=250000;
}
```

```
public void vpisiZnesekSolnine(int solnina)
{
    znesekSolnine=solnina;
}
```

```
public int vrniZnesekSolnine()
{
    return znesekSolnine;
}
```

```
}
```

```
public class DemoKonstruktor1
```

```

{
public static void main(String[] args)
{
    Student4 s=new Student4();
    System.out.println("Student");
    System.out.println(s.vrniVpisSt()+" "+s.vrniPriimek()+" "+s.vrniIme());

    IzredniStudent4 is=new IzredniStudent4();
    System.out.println("IzredniStudent");
    System.out.println(is.vrniVpisSt()+" "+is.vrniPriimek()+" "+is.vrniIme());
    System.out.println("je placal "+is.vrniZnesekSolnine()+" SIT solnine.");
}
}

```

```

public class Student5

```

```

// enak kot Student4, le da konstruktor zahteva argumente

```

```

{
private int vpisSt;
private String priimek;
private String ime;

Student5(int vs, String p, String i)
{
    vpisSt=vs;
    priimek=p;
    ime=i;
}

public void vpisiVpisSt(int st)
{
    vpisSt=st;
}

public void vpisiPriimek(String p)
{
    priimek=p;
}

public void vpisiIme(String i)
{
    ime=i;
}

public int vrniVpisSt()
{
    return vpisSt;
}

```



```
public String vrniPriimek()
{
    return priimek;
}
```

```
public String vrniIme()
{
    return ime;
}
}
```

```
public class IzredniStudent5 extends Student5
```

```
// enak kot IzredniStudent4, le da konstruktor poskrbi za klic
// konstruktorja nadrazreda z argumenti
```

```
{
    private int znesekSolnine;

    IzredniStudent5(int vs,String p,String i,int zn)
    {
        super(vs,p,i);
        znesekSolnine=zn;
    }
}
```

```
public void vpisiZnesekSolnine(int solnina)
{
    znesekSolnine=solnina;
}
```

```
public int vrniZnesekSolnine()
{
    return znesekSolnine;
}
}
```

```
public class DemoKonstruktor2
```

```
{
    public static void main(String[] args)
    {
        Student5 s=new Student5(63020888,"Novak","Janez");
        System.out.println("Student");
        System.out.println(s.vrniVpisSt()+" "+s.vrniPriimek()+" "+s.vrniIme());

        IzredniStudent5 is=new IzredniStudent5(63020999,"Bogataj","Franc",250000);
        System.out.println("IzredniStudent");
        System.out.println(is.vrniVpisSt()+" "+is.vrniPriimek()+" "+is.vrniIme());
        System.out.println("je placal "+is.vrniZnesekSolnine()+" SIT solnine.");
    }
}
```

```
public class Krogi
```

```

{
static final int N=5; // Űtevílo krogov
static final int A=10; // stranica kvadrata

public static void main(String[] args)
{
double premer;
int v=0,izven=0,seka=0;
for (int i=1;i<=N;++i)
{
System.out.print("Vtipkaj premer:");
premer=BranjePodatkov.preberiDouble();
if (premer<A)
++v;
else if (premer>A*Math.sqrt(2))
++izven;
else
++seka;
}
System.out.println("V kvadratu je "+v+" krogov.");
System.out.println("Kvadrat seka "+seka+" krogov.");
System.out.println("Izven kvadrata je "+izven+" krogov.");
}
}

```

```

public class UrejenaTabela
{
public static void main(String[] args)
{
int[] t=new int[10];
int stevílo;

for (int i=0; i<t.length; ++i)
{
// preberi stevílo
System.out.print("Vpisi stevílo:");
stevílo=BranjePodatkov.preberiInt();

// poisci mesto, kamor naj se vstavi
int j=0;
while ((j<i) && (stevílo>t[j]))
++j;

// pripravi prostor
for (int k=i-1; k>=j; --k)
t[k+1]=t[k];

// vpisi stevílo
t[j]=stevílo;
}
}

```

```

// izpisi tabelo
for (int i=0; i<t.length; ++i)
    System.out.print(" "+t[i]);
}
}

```

```

public class Razdalje
{
    static final int ST_TOCK=5;

    public static void main(String[] args)
    {
        double[] x=new double[ST_TOCK];
        double[] y=new double[ST_TOCK];
        double[][] r=new double[ST_TOCK][ST_TOCK];

        PreberiKoordinate(x,y);
        IzracunajRazdalje(x,y,r);
        IzpisiRazdalje(r);
    }

    public static void PreberiKoordinate(double[] x,double[] y)
    {
        for (int i=0; i<ST_TOCK; ++i)
        {
            System.out.print("Koordinata x "+(i+1)+" . tocke:");
            x[i]=BranjePodatkov.preberiDouble();
            System.out.print("Koordinata y "+(i+1)+" . tocke:");
            y[i]=BranjePodatkov.preberiDouble();
        }
    }

    public static void IzracunajRazdalje(double[] x,double[] y,double[][] r)
    {
        double dx,dy;
        for (int i=0; i<ST_TOCK; ++i)
            for (int j=0; j<=i; ++j)
                if (i==j)
                    r[i][j]=0;
                else
                {
                    dx=x[i]-x[j];
                    dy=y[i]-y[j];
                    r[i][j]=Math.sqrt(dx*dx+dy*dy);
                    r[j][i]=r[i][j];
                }
    }

    public static void IzpisiRazdalje(double[][] r)
    {
        for (int i=0; i<ST_TOCK; ++i)

```

```

    {
        for (int j=0; j<ST_TOCK; ++j)
            System.out.print(r[i][j]);
        System.out.println();
    }
}
}
}
}
}

public class Razdalje1
{
    // globalne spremenljivke
    static final int ST_TOCK=5;
    static double[] x=new double[ST_TOCK];
    static double[] y=new double[ST_TOCK];
    static double[][] r=new double[ST_TOCK][ST_TOCK];

    public static void main(String[] args)
    {
        PreberiKoordinate();
        IzracunajRazdalje();
        IzpisiRazdalje();
    }

    public static void PreberiKoordinate()
    {
        for (int i=0; i<ST_TOCK; ++i)
        {
            System.out.print("Koordinata x "+(i+1)+" . tocke:");
            x[i]=BranjePodatkov.preberiDouble();
            System.out.print("Koordinata y "+(i+1)+" . tocke:");
            y[i]=BranjePodatkov.preberiDouble();
        }
    }

    public static void IzracunajRazdalje()
    {
        double dx,dy;
        for (int i=0; i<ST_TOCK; ++i)
            for (int j=0; j<=i; ++j)
                if (i==j)
                    r[i][j]=0;
                else
                {
                    dx=x[i]-x[j];
                    dy=y[i]-y[j];
                    r[i][j]=Math.sqrt(dx*dx+dy*dy);
                    r[j][i]=r[i][j];
                }
    }

    public static void IzpisiRazdalje()

```

```

{
  for (int i=0; i<ST_TOCK; ++i)
  {
    for (int j=0; j<ST_TOCK; ++j)
      System.out.print(r[i][j]);
    System.out.println();
  }
}
}

```

```

public class Palindrom
{
  public static void main(String[] args)
  {
    if (args.length!=1)
      System.out.println("Program zahteva en argument tipa String");
    else if (jePalindrom(args[0]))
      System.out.println(args[0]+" je palindrom.");
    else
      System.out.println(args[0]+" ni palindrom.");
  }

  public static boolean jePalindrom(String niz)
  {
    int i=0, j=niz.length()-1;
    while ((i<j) && (niz.charAt(i)==niz.charAt(j)))
    {
      ++i;
      --j;
    }
    return i>=j;
  }
}

```

```

public class Anagrami
{
  public static void main(String[] args)
  {
    // branje vhodnih podatkov
    String niz1=BranjePodatkov.preberiString();
    String niz2=BranjePodatkov.preberiString();

    // generiranje objektov tipa StringBuffer
    StringBuffer n1=new StringBuffer(niz1);
    StringBuffer n2=new StringBuffer(niz2);

    // sortiranje obeh nizov
    sortiraj(n1);
    System.out.println("Po sortiranju");
    System.out.println(n1);
    sortiraj(n2);
    System.out.println(n2);
  }
}

```



```

// metoda objekta (vezana na posamezen objekt)
public double volumen()
{
    return 4.0/3*PI*polmer*polmer*polmer;
}
}

```

```

public class TestKrogla
{
    public static void main(String[] args)
    {
        Krogla k1=new Krogla(1,0,0,0);
        System.out.println("Stevilo krogel: "+Krogla.vrniSteviloKrogel());
        System.out.println("Volumen krogel: "+k1.volumen());

        Krogla k2=new Krogla(2,5,3,1);
        System.out.println("Stevilo krogel: "+Krogla.vrniSteviloKrogel());
        System.out.println("Volumen krogel: "+k2.volumen());
    }
}

```

```

public abstract class Zival1
{
    // atribut
    private String ime;

    // konstruktor
    Zival1(String imeZivali)
    {
        ime=imeZivali;
    }

    // metode
    public String vrniIme()
    {
        return ime;
    }

    public abstract void oglasanje(); // abstraktna metoda

    public boolean equals(Zival1 z2) // redefinicija metode equals()
    {
        return vrniIme().equals(z2.vrniIme());
    }
}

```

```

public class Pes1 extends Zival1
{
    // konstruktor
    Pes1(String imePsa)
    {

```

```
    super(imePsa);
}

// redefinicija abstraktne metode
public void oglasanje()
{
    System.out.println("Hov,hov");
}
}

```

```
public class Krava1 extends Zival1
{

    // konstruktor
    Krava1(String imeKrave)
    {
        super(imeKrave);
    }

    // redefinicija abstraktne metode
    public void oglasanje()
    {
        System.out.println("Muuuu");
    }
}

```
