

Fakulteta za Računalništvo in Informatiko

Programiranje I

(II. DEL)

Staš Repše

Kazalo (II. DEL)

1. Osnovne stvari

Glava programa, tip podatkov (številke, znaki, logične vrednosti)	6. stran
Predstavitev celih števil, realnih števil, operatorji, boolean (,&&,!), ubežna sekvenca, branje podatkov (z razredom BranjePodatkov (int), (double), (String))	7. stran
Prioriteta operatorjev (od najvišje do najnižje), IF stavek, SWITCH stavek	8. stran

I. FOR, IF,...

DemoSwitch.java	Prikaz stavka switch.	9. stran
Vsota.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	10. stran
ObrestiProcedura.java	Izračun obresti v obliki procedure	
ObrestiFunkcija.java	Izračun obresti v obliki funkcije	
Fibonacci.java	Izračun n-tega Fibonaccijevega števila; Primer rekurzivne metode in ustrezna iterativna rešitev	11. stran
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 kode 0 do 360 stopinj s korakom 30; Primer metode, ki temelji na rekurenčni relaciji; Uporaba metode iz drugega razreda	12. stran
Izracunaj.java	Program, ki števila od 1 do 100 pomnoži z 100!	
Delavec2.java, TestDelavec2.java	Dekleracija in test razreda Delavec z večkratno definiranim konstruktorjem	13. stran
DemoOverload	Prikaz večkratnega definiranja metod (metoda za izračun obresti	14. stran

II. Tabele 2-3 dim.,...

DemoCharacter.java	Prikaz delovanja metod iz razreda Character.	15. stran
SORTIRANJE		16. stran
Sort.java	Soritane tabele celih števil	
Sort2.java	Soritane tabele realnih števil	
TABELE		17. stran
Demo3D.java	Prikaz branja in izpisa tridimenzionalne tabele.	
MnozenjeMatrik.java	Program za množenje matrix; Primer programa za delo z dvodimenzionalnimi tabelami.	
TabelaDelavcev.java, Delavec2.java	Branje in izpis podatkov o delavcih; Primer programa za delo s tabelo objektov.	18. stran

III. Objekti in tabele...

DemoStringBuffer1.java	Prikaz konstruktorjev za nize tipa StringBuffer; ilustracija pojmov kapaciteta in dožina niza.	19. stran
DemoStringBuffer2.java	Prikaz metod za delo z niti tipa StringBuffer	20. stran
DemoArgMain.java	Prikaz argumentov metode main()	21-22. stran
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.	

IV. Dedovanje, objekti...

Student.java, IzredniStudent.java, DemoDedovanje.java	Prikaz dedovanja. Razred IzredniStudent podeduje attribute in metode razreda Student.	23. stran
Student.java, IzredniStudent1.java DemoRedefinicija.java	Prikaz redefinicije metodo ob dedovanju. Razred IzredniStudent1 redefinira metodo izpisTipa().	
DEDOVANJE		24. stran
Student2.java, IzredniStudent2.java, DemoSuper.java	Prikaz klika metode nadrazreda s pomočjo rezervirane besede super. Podrazred IzredniStudent2 redefinira metodo izpisiVse() in pri tem uporabi istoimensko metodo nadrazreda.	
Student5.java, IzredniStudent5.java, DemoKonstruktor2.java	Prikaz konstruktorja nadrazreda z argumenti. Konstruktor podrazreda kliče konstruktor nadrazreda s super in ustreznimi dejanskimi parametri.	25. stran
Krogi.java	Izračun števila krogov, ki so v kradratu, sekajo kvadrat oziroma so v celoti izven kvadrata. Preprost prikaz znake in razvejitve. UrejenaTabela.java – Vstavljanje števil v urejeno tabelo.	26. stran
Razdalje1.java	Isto kot Razdalje.java, le da so vse tabele deklarirane kot globalne spremenljivke.	27. stran
Palindrom.java	Ugotavljanje, ali je niz palindrom. Primer preprostega programa, ki uporablja tip String. Vhodni niz je podan kot argument metode main().	28. stran
Anagrami.java	Ugotavljanje, ali sta niza anagrama. Primer uporabe razreda StringBuffer.	
Krogla.java, TestKrogla.java	Dekleracija in test razreda Krogla. Primer spremenljivk in metod razreda ter spremenljivk in metod objekta.	29. stran
Zival.java, Pes.java, Krava.java, Kaca.java, DemoHeterogeneTabele.java	Prikaz tabele, ki vsebuje objekte različnih podrazredov.	30. stran
Zival1.java, Pes1.java, Krava1.java, DemoEquals.java	Prikaz redefinicije metode equals() v razredu Zival. Redefiniramo metodo avtomatsko podedujeta podrazreda Pes in Krava.	31. stran

V. Grafika

Omogoči uporabo paketov (paketi za grafiko)	32. stran
Oblika pisave	
Barva (barve - setColor(barva))	
Ozadja	
Risalna barva	
Risanje polnih likov	
Črke	
5. naloga iz izpita – Pet koncentričnih (enak center) krogov	33-34. stran
Uporaba vseh možnih objektov (s komentarji)	35. stran
Prikaz s sliko vseh možnih objektov (vse je vznačeno)	36. stran

VI. Aplet

Prikaz vseh metod (eno izmed njih je potrebno deklarirati)	37. stran
Prikaz vnosa besedila	
Prikaz metod za urejanje	
HTML KODA	
APPLET (Applet.java) (I. del) – s poslušalcem	
APPLET (Applet.java) (II. del)	38. stran
Prikaz s slikami izris v Applet Viewer-u	
Razred MATH. Matematične konstante...	39. stran
Metode za delo z nizi (replace(), length(), indexOf(), toString()) Šumniki (UNICODE)	40. stran
Dostopna določila (public, private, protected, brez dostopnega določila)	
Razred StringBuffer (append(), insert(), delete(), indexOf(),...) Druge metode razreda object (getClass(), wait(),...)	41. stran
Tip dogodka, vmesnik, odzivne metode (WindowEvent, MouseEvent)	42. stran
Komponente (izvori dogodkov), Metode za dodajanje poslušalcev (JButton, JTextField,...)	

1. Začetna črka naj bo velika (brez presledkov ali _) »npr. **PriProgram.java**«
2. Glavni program more vedno vsebovati **CLASS** in **MAIN!!!**

GLAVA PROGRAMA

```
public class PrviProgram
{
    public static void main (String[] args)
    {
        System.out.println("Testing!");
    }
}
```

3. <razred>.<objekt>.<metoda>
4. {} – začetek/konec bloka
5. Številke se začnejo od 0 naprej...

TIPI podatkov

Številke:

- byte, short, int, long (**cela števila**; npr. 2)
- float, double (**realna števila**; npr. 2.6)

S PIKO!!!

Znaki:

- char (**ena črka; enojni narekovaj**; npr. 'A')
- String (**z veliko črko se piše; z dvojnimi narekovaji**; npr. "Blablabla")

Logične vrednosti:

- boolean (true (1)/false(2))

1. primer:

```
public class PrviProgram
{
    public static void main (String[] args)
    {
        double prvoStevilo = 15.42;
        double drugoStevilo = 5.540;

        int n = (int)(prvoStevilo/drugoStevilo); // 2 (byte, short, int, long)
// ali double = (prvoStevilo/drugoStevilo); // 2.783393501805054

        System.out.println(n);
    }
}
```

2. primer:

```
public class PrviProgram
{
    public static void main (String[] args)
    {
        char crka = 'A';
        String beseda = "Blablabla";

        System.out.println(crka); // A
        System.out.println(beseda); // Blablabla
    }
}
```

Predstavitev CELIH ŠTEVIL (byte, short, int, long)

tip	bit	min	max
byte	8	-128	127
short	16	-32.768	32.767
int	32	-2.147.483.648	2.147.483.647
long	64	-9.223.372.036.854.775.808	9.223.372.036.854.775.807

Predstavitev REALNIH ŠTEVIL (float, double)

tip	bit	min	max	št. točnih decimalnih mest
float	32	$-3,4 * 10^{38}$	$3,4 * 10^{38}$	6 do 7
double	64	$-1,7 * 10^{308}$	$1,7 * 10^{308}$	14 do 15

OPERATORJI

operator	opis	primer
+	seštevanje	32+3, rezultat je 35
-	odštevanje	32-3, rezultat je 29
*	množenje	32*3, rezultat je 96
/	deljenje	32/3, rezultat je 10
%	ostanek	32%3, rezultat je 2

operator	opis	primeru za true	primer za false
<	manjše kot	5 < 7	7 < 5
>	večje kot	7 > 5	5 > 7
<=	manjše ali enako	5 <= 5	7 <= 5
>=	večje ali enako	7 >= 3	3 >= 7
==	enako	5 == 5	5 == 6
!=	ni enako	8 != 6	5 != 5

BOOLEAN

&& - in
|| - ali
! - negacija

UBEŽNA SEKVENCA

ubežna sekvenca	opis	ubežna sekvenca	opis
\t	tab	\r	izpiše vse, kar je pred njim
\n	nova vrsta	\"	dvojni narekovaj
\f	Φ	\'	enojni narekovaj

BranjePodatkov:

Celo število:

```
st = BranjePodatkov.preberiInt();
```

Realno število:

```
re = BranjePodatkov.preberiDouble();
```

Niz:

```
bes = BranjePodatkov.preberiString();
```

PRIORITETA OPERATORJEV

prioriteta	operatorji	simboli
najvišja	množenje, deljenje	* / %
	seštevanje, odštevanje	+ -
	primerjanje	> < >= <=
	enakost	== !=
	logični IN	&&
	logični ALI	
	pogojni	?:
najnižja	prirejanje	=

Operacije se izvajajo od LEVE proti DESNE v skladu s spodaj navedeno prioriteto operatorjev! Vrstni red lahko spremenimo s pomočjo oklepajev!

IF stavek

```
if (2==2)
{
    // <-- ta stavek se bo izvršil!!!
}
else if (2>4)
{
}
else
{
}
```

SWITCH stavek

```
int ocena = 8;

switch (ocena)
{
    case 10:
        System.out.println("Odlično");
        break;

    case 9:
    case 8:
        System.out.println("Prav dobro");
        break;

    case 7:
        System.out.println("Dobro");
        break;

    case 6:
        System.out.println("Zadostno");
        break;

    default:
        System.out.println("Nezadostno");
}
```


I. FOR, IF,...

DemoSwitch.java - Prikaz stavka switch.

```
for(int ocena=1; ocena<=10; ++ocena)
switch (ocena) // vsebuje oceno od 1 do 10
{
    case 6:
        System.out.println("zadosno");
        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("nezadosno");
}
}
```

IZPIS

```
nezadosno
nezadosno
nezadosno
nezadosno
zadosno
dobro
prav dobro
prav dobro
odlicno
```

Vsota.java - Izračun vsote prvih 200 naravnih števil s stavkom do ... while.;

Vsota2.java - Izračun vsote prvih 200 naravnih števil s stavkom while.

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

Vsota3.java - Izračun vsote prvih 200 naravnih števil s stavkom for.

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

IZPIS

```
Vsota je 20100
```

Lovec.java - Izpis možnih potez lovca na šahovnici.

```
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();
        }
    }
}
```

IZPIS

```
BCB*BCBC
*B*BCBCB
B*BCBCBC
*B*BCBCB
BCB*BCBC
CBCB*BCB
BCBCB*BC
CBCBCB*B
```

ObrestiProcedura.java - Izračun obresti v obliki procedure.

```
class ObrestiProcedura
{
    public static void main(String[] args)
    {
        double g=100000;
        double o=5;

        poEnemLetu(g,o);
        poEnemLetu(20000,8);
        poEnemLetu(40000,16);
        poEnemLetu(60000,24);
    }

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

IZPIS

```
Po enem letu dobimo 105000.0
Po enem letu dobimo 21600.0
Po enem letu dobimo 46400.0
Po enem letu dobimo 74400.0
```

ObrestiFunkcija.java – Izračun obresti v obliki funkcije.

```
class ObrestiFunkcija
{
    public static void main(String[] args)
    {
        double g=100000;
        double o=5;
        double ns;

        ns=poEnemLetu(g,o);
        System.out.println("Novo stanje: "+ns);
        System.out.println("Novo stanje: "+poEnemLetu(20000,8));
        System.out.println("Novo stanje: "+poEnemLetu(40000,16));
        System.out.println("Novo stanje: "+poEnemLetu(60000,24));
    }

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

IZPIS

```
Novo stanje: 105000.0
Novo stanje: 21600.0
Novo stanje: 46400.0
Novo stanje: 74400.0
```

Fibonacci.java - Izračun n-tega Fibonaccijevega števila; Primer rekurzivne metode in ustrežna iterativna rešitev.

```
public class Fibonacci
{
    public static void main(String[] args)
    {
        int n=6;
        System.out.println(n+". Fibonaccijevo število (rekurzivno) je "+fib(n));
        System.out.println(n+". Fibonaccijevo število (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 število
        int z;
        for (int i=2; i<=n; ++i)
        {
            z=x; // zacasno shranimo zadnje Fibonaccijevo število
            x=x+y; // naslednje Fibonaccijevo število
            y=z; // prejšnje Fibonaccijevo število
        }
        return x;
    }
}
```

IZPIS

6. Fibonaccijevo število (rekurzivno) je 8
6. Fibonaccijevo število (iterativno) je 8

PerfektnaStevila.java - Iskanje perfektnih števil z intervala od 1 do 1000; Primer programa, ki poleg metode main() vsebuje še eno metodo.

```
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;
    }
}
```

IZPIS

6
28
496

MatFunkcije.java, Tabela.java - Izračun kotne funkcije sinus in izpis tabele z vrednostmi za kode 0 do 360 stopinj s korakom 30; Primer metode, ki temelji na rekurenčni relaciji; Uporaba metode iz drugega razreda.

```
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;
    }
}
```

IZPIS

Kot	Sinus
0	0.0
30	0.4999...
60	0.8666...
90	0.9999...
120	0.8660...
150	0.4999...
180	2.3489...
210	-0.5000...
240	-0.8660...
270	-0.9999...
300	-0.8660...
330	-0.4999...
360	-4.8039...

```
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));
        }
    }
}
```

Izracunaj.java - Program, ki števila od 1 do 100 pomnoži z 100.

```
public class Izracun
{
    public static void main (String[] main)
    {
        for (int stevilo=1; stevilo<=100; stevilo++)
        {
            System.out.println(stevilo+" "+racunaj.metoda(stevilo));
        }
    }
}

class racunaj
{
    public static int metoda(int x)
    {
        int vsota;
        vsota = (x * 100);
        return vsota;
    }
}
```

int > double
(vse tri intje spremenimo
v double in šele takrat
dobimo realno število!

IZPIS

1	10
2	200
3	300
4	400
5	500
.	.
.	.
.	.
100	100000

Delavec2.java, TestDelavec2.java - Dekleracija in test razreda Delavec z večkratno definiranimi konstruktorjem.

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

public String vrniPriimek()
{
return priimek;
}

...
...
...
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);
}
}
```

IZPIS

```
Maticna stevilka: 234
Priimek in ime: Novak Janez
Stevilo ur: 182
Bruto OD: 273000
Maticna stevilka: 234
Priimek in ime: Nova Janez
Stevilo ur: 182
```

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

// kreiranje objekta z razlicnimi 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();
}
}
```

DemoOverload - Prikaz večkratnega definiranja metod (metoda za izračun obresti).

<pre>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; } }</pre>	<table border="1"><tr><td>IZPIS</td></tr><tr><td>80.0</td></tr><tr><td>80.0</td></tr></table> <p>Ko je 2x double izvede prvo metodo. Ko je 1x double, 1x int pa druga metodo!</p>	IZPIS	80.0	80.0
IZPIS				
80.0				
80.0				

II. Tabele 2-3 dim.,...

DemoCharacter.java - Prikaz delovanja metod iz razreda Character.

```
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!='#');
    }
}
```

IZPIS

```
6
28
496
```

SORTIRANJE

Sort.java - Soritanje tabele celih števil.

```
public class Sort
{
    public static void main(String[] args)
    {
        int[] a= {24,60,55,33,98,6,43,15};
        sortiraj(a);

        for (int indeks=0; indeks<a.length; ++indeks)
            System.out.println(a[indeks]);
    }
}
```

IZPIS

6
15
24
33
43
55
60
98

```
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;
    }
}
}
```

IZPIS

6.65
15.0
24.87
33.678
43.0
55.876
60.67
98.45

Sort2.java - Soritanje tabele realnih števil.

```
public class Sort
{
    public static void main(String[] args)
    {
        double[] a= {24.87,60.67,55.876,33.678,98.45,6.65,43.0,15.0};
        sortiraj(a);

        for (int indeks=0; indeks<a.length; ++indeks)
        {
            System.out.println(a[indeks]);
        }
    }
}
```

```
public static void sortiraj(double[] a)
{
    int iMin;
    double vMin;

    for (int i=0; i<=a.length-2; ++i)
    {
        iMin=i;
        vMin=a[i];

        for (int j=i+1; j<=a.length-1; ++j)
        {
            if (a[j]<vMin)
            {
                iMin=j; vMin=a[j];
            }
        }
        a[iMin]=a[i]; a[i]=vMin;
    }
}
}
```


TABELE

Demo3D.java – Prikaz branja in izpisa tridimenzionalne tabele.

```
public class Demo3D
{
    public static void main(String[] args)
    {
        int[][][] a=new int[3][4][2];           // deklaracija tabele

        // vpisovanje podatkov v tabelo
        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();

        // izpis tabele po vrstnem redu vnašanja
        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]);
    }
}
```

IZPIS

```
0,0,0 1
0,0,1 2
0,1,0 3
0,1,1 4
0,2,0 5
0,2,1 6
0,3,0 7
0,3,1 8
1,0,0 9
1,0,1 10
1,1,0 2
1,1,1 2
1,2,0 8
1,2,1 8
1,3,0 8
1,3,1 9
2,0,0 9
2,0,1 24
2,1,0 25
2,1,1 26
2,2,0 27
2,2,1 28
2,3,0 29
2.3.1 30
```

MnozenjeMatrik.java – Program za množenje matrix

Primer programa za delo z dvodimenzionalnimi tabelami.

```
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},{2,5},{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();
        }
    }
}
```

IZPIS

```
38 63
43 74
17 26
```

TabelaDelavcev.java, Delavec2.java - Branje in izpis podatkov o delavcih; Primer programa za delo s tabelo objektov.

```
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());
        }
    }
}
```

IZPIS

1 Nova Janez 45
 2 Krneki Blabla 65
 3 Gree Rgr 543
 4 Gre Grt 4
 5 Hfg Fhghf 54

odisno od vnosa podatkov

Public class Delavec2 je že zgoraj napisan!!!

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

public String vrniPriimek()
{
    return priimek;
}

...

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);
}
}
```

III. Objekti in tabele...

DemoStringBuffer1.java - Prikaz konstruktorjev za nize tipa StringBuffer; ilustracija pojmov kapaciteta in dožina niza.

```
public class DemoStringBuffer1
{
    public static void main(String[] args)
    {
        // uporaba razlicnih konstruktorjev
        System.out.println();
        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();
        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();
        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();
        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();
        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();
        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());
    }
}
```

IZPIS

Konstruktor brez parametrov

Dolzina : 0

Kapaciteta: 16

Konstruktor s parametrom tipa

String

Dobro jutro

Dolzina : 11

Kapaciteta: 27

Konstruktor s parametrom

kapaciteta

**

Dolzina : 0

Kapaciteta: 50

Dolzino niza postavimo na 20

*Dobro jutro *

Dolzina : 20

Kapaciteta: 27

Dolzino niza postavimo na 30

*Dobro jutro *

Dolzina : 30

Kapaciteta: 56

Dolzino niza postavimo na 10

Dobro jutro

Dolzina : 10

Kapaciteta: 56

DemoStringBuffer2.java - Prikaz metod za delo z niti tipa StringBuffer.

```
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 prejšnji 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());
        System.out.println();
    }
}
```

IZPIS

```
*Dobro jutro*
Dolzina : 11
Kapaciteta: 27

*Dobro jutro vam zelim*
Dolzina : 21
Kapaciteta: 27

*Dobro jutro vam zelim.*
Dolzina : 22
Kapaciteta: 27

*Dobro jutro in prijeten danasnji
dan vam zelim.*
Dolzina : 47
Kapaciteta: 56

* prijeten danasnji dan vam zelim.*
Dolzina : 33
Kapaciteta: 56

*prijeten danasnji dan vam zelim.*
Dolzina : 32
Kapaciteta: 56

*Prijeten danasnji dan vam zelim.*
Dolzina : 32
Kapaciteta: 56

*Uspesen danasnji dan vam zelim.*
Dolzina : 31
Kapaciteta: 56

*Uspesen danasnji dan vam zelim.*
Dolzina : 31
Kapaciteta: 56

Znak na poziciji 4: s
*Uspesen danasnji dan am zelim.*
Dolzina : 31
Kapaciteta: 56

Znaki od 4. do 10. mesta: sen dan
*Uspesen danasnji dan vam zelim.*
Dolzina : 31
Kapaciteta: 56

8
17
```

DemoArgMain.java - Prikaz argumentov metode main().

```
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]);
    }
}
```

```
IZPIS
-----
java DemoArgMain
0

java DemoArgMain 24
1
24

java DemoArgMain 24 35 36 60
4      // število argumentov
24     // prvi argument
35     // ...
36     // ...
60     // zadnji argument
```

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.

```
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;
}

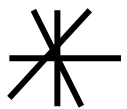
public String vrniPriimek()
{
    return priimek;
}

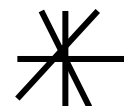
public String vrniIme()
{
    return ime;
}

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)
    {
        // kreiranje tabele objektov
        Delavec4[] td=new Delavec4[ST_DEL];
        for (int i=0; i<ST_DEL; ++i)
            td[i]=new Delavec4();

        // izpis pred sortiranjem
        System.out.println();
        System.out.println("Pred sortiranjem:");
        izpisi(td);

        // urejanje tabele
        sortiraj(td);

        // izpis po sortiranju
        System.out.println();
        System.out.println("Po sortiranju:");
        izpisi(td);
    }

    static void izpisi(Delavec4[] td)
    {
        for (int i=0; i<ST_DEL; ++i)
        {
            // izpis podatkov s pomocjo posameznih metod
            System.out.print(td[i].vrniMatSt());
            System.out.print(" "+td[i].vrniPriimek()+" "+td[i].vrniIme());
            System.out.println(" "+td[i].vrniVsotoOd());
        }
    }

    static void sortiraj(Delavec4[] a)
    {
        int i,j,iMin;
        Delavec4 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].vrniVsotoOd()>vMin.vrniVsotoOd())
                {
                    iMin=j; vMin=a[j];
                }
            a[iMin]=a[i]; a[i]=vMin;
        }
    }
}
```

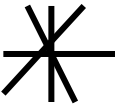
IZPIS

Odvisno, kako vnašaš podatke!

Matična št. Priimek Ime Dohodek
npr.
12 Novak Janez 78.0
1,2,3,4,5,6,7,8,9,10,11,12

Sortirano
od največjega 1000
do najmanjšega 100

IV. Dedovanje, objekti...



Student.java, IzredniStudent.java, DemoDedovanje.java - Prikaz dedovanja. Razred IzredniStudent podeduje attribute in metode razreda Student.

```
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;
    }
}
```

IZPIS

```
Student
63020888 Novak Janez
IzredniStudent
63020999 Bogataj Franc
je placal 250000 SIT solnine.
```

```
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.");
    }
}
```

Student.java, IzredniStudent1.java DemoRedefinicija.java - Prikaz redefinicije metodo ob dedovanju. Razred IzredniStudent1 redefinira metodo izpisTipa().

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

```
public class IzredniStudent1 extends Student1
{
    private int znesekSolnine;
```

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

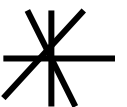
```
public void vpisiZnesekSolnine(int solnina)
{
    znesekSolnine=solnina;
}
```

```
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.");
    }
}
```



DEDOVANJE

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.

```
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 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();
    }
}
```

IZPIS

```
Student
63020888
Novak Janez
IzredniStudent
53020999
Bogataj Franc
250000
```


Student5.java, IzredniStudent5.java, DemoKonstruktor2.java

Prikaz konstruktorja nadrazreda z argumenti. Konstruktor podrazreda kliče konstruktor nadrazreda s super in ustreznimi dejanskimi parametri.

public class Student5

```
// enak kot Student4, le da konstruktor  
zahteva argumente
```

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

```
// konstruktor
```

```
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 void vpisiZnesekSolnine(int solnina)
```

```
{  
    znesekSolnine=solnina;  
}
```

```
public int vrniZnesekSolnine()
```

```
{  
    return znesekSolnine;  
}
```

```
}
```

IZPIS

```
Student  
63020888 Novak Janez  
IzredniStudent  
63020999 Bogataj Franc  
je placal 250000 SIT solnine.
```

```
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.");  
    }  
}
```

Krogi.java - Izračun števila krogov, ki so v kvadratu, sekajo kvadrat oziroma so v celoti izven kvadrata. Preprost prikaz znake in razvejitve.

```
public class Krogi
{
    static final int N=5; // število 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.");
    }
}
```

IZPIS

1. primer

Vtipaj premer: 2
Vtipaj premer: 2
Vtipaj premer: 2
Vtipaj premer: 2
Vtipaj premer: 2
V kvadratu je 5 krogov.
Kvadrat seka 0 krogov.
Izven kvadrata je 0 krogov.

2. primer

Vtipaj premer: 54
Vtipaj premer: 53
Vtipaj premer: 534
Vtipaj premer: 65
Vtipaj premer: 635
V kvadratu je 0 krogov.
Kvadrat seka 0 krogov.
Izven kvadrata je 5 krogov.

UrejenaTabela.java – Vstavljanje števil v urejeno tabelo.

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

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

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

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

            // vpisi stevilo
            t[j]=stevilo;
        }

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

IZPIS

Vpisi stevilo: 1
Vpisi stevilo: 2
Vpisi stevilo: 4
Vpisi stevilo: 5
Vpisi stevilo: 8
Vpisi stevilo: 9
Vpisi stevilo: 10
Vpisi stevilo: 7
Vpisi stevilo: 6
Vpisi stevilo: 3

1 2 3 4 5 6 7 8 9 10

Razdalje1.java – Isto kot Razdalje.java, le da so vse tabele deklarirane kot globalne spremenljivke.

```
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)
    {
        // v primeru lokalne definicije treh tabel

        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();
        }
    }
}
```

IZPIS

```
Koordinata x 1. tocke: 1
Koordinata y 1. tocke: 2
Koordinata x 2. tocke: 3
Koordinata y 2. tocke: 4
Koordinata x 3. tocke: 5
Koordinata y 3. tocke: 6
Koordinata x 4. tocke: 7
Koordinata y 4. tocke: 8
Koordinata x 5. tocke: 90
Koordinata y 6. tocke: 6
0.02.828427...
2.828427124...
5.656854249...
8.485281374...
89.08984229...
```

Palindrom.java – Ugotavljanje, ali je niz palindrom. Primer preprostega programa, ki uporablja tip String. Vhodni niz je podan kot argument metode main().

```
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;
    }
}
```

IZPIS

```
java Palindrom
Program zahteva en argument tipe String
```

```
java Palindrom aba
aba je palindrom
```

Anagrami.java - Ugotavljanje, ali sta niza anagrama. Primer uporabe razreda StringBuffer.

```
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);

        // testiranje enakosti in izpis rezultata
        if (n1.toString().equals(n2.toString()))
            System.out.println("Niza "+niz1+" in "+niz2+" sta anagrama.");
        else
            System.out.println("Niza "+niz1+" in "+niz2+" nista anagrama.");
    }
}
```

IZPIS

```
1. primer
blablabla
tralala..
Po sortiranju
aaabbblll
..aaallrt
Niza blablabla in tralala.. nista anagrama.
```

```
2. primer
krneki
ikenrk
Po sortiranju
eikknr
eikknr
Niza krneki in ikenrk sta anagrama.
```

```
public static void sortiraj(StringBuffer a)
{
    int i,j,iMin;
    char vMin;
    for (i=0; i<=a.length()-2; ++i)
    {
        iMin=i; vMin=a.charAt(i);
        for (j=i+1; j<=a.length()-1; ++j)
            if (a.charAt(j)<vMin)
            {
                iMin=j; vMin=a.charAt(j);
            }
        a.setCharAt(iMin,a.charAt(i)); a.setCharAt(i,vMin);
    }
}
```

Krogla.java, TestKrogla.java - Dekleracija in test razreda Krogla. Primer spremenljivk in metod razreda ter spremenljivk in metod objekta.

```
public class Krogla
{
    // spremenljivki razreda
    private static final double PI=3.14159;
    private static int steviloKrogel=0;

    // spremenljivke objektov
    private double polmer;
    private double xSredisca,ySredisca,zSredisca;    // koordinate središca

    // konstruktor
    Krogla(double r, double x, double y, double z)
    {
        polmer=r;
        xSredisca=x;
        ySredisca=y;
        zSredisca=z;
        ++steviloKrogel;    // povecevanje števca objektov
    }

    // metoda razreda (skupna vsem objektom)
    static int vrniSteviloKrogel()
    {
        return steviloKrogel;
    }

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

IZPIS

```
Stevilo krogel: 1
Volumen krogle: 4.1887866666666666
Stevilo krogel: 2
Volumen krogle: 33.510293333333333
```

```
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 krogle: "+k1.volumen());

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

Zival.java, Pes.java, Krava.java, Kaca.java, DemoHeterogeneTabele.java - Prikaz tabele, ki vsebuje objekte različnih podrazredov.

```
public abstract class Zival
{
    // atribut
    private String ime;

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

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

    public abstract void oglasanje(); // abstraktna metoda
}
```

IZPIS

```
fifi se oglasa Hov,hov
milka se oglasa Muuuu
kobra se oglasa Ssssss
rex se oglasa Hov,hov
krava se oglasa Muuuu
kaca se oglasa Ssssss
pes se oglasa Hov,hov
muuu se oglasa
Muuuu
```

```
public class Pes extends Zival
{
    // konstruktor
    Pes(String imePsa)
    {
        super(imePsa);
    }
    // redefinicija abstraktne metode
    public void oglasanje()
    {
        System.out.println("Hov,hov");
    }
}
```

```
public class Krava extends Zival
{
    // konstruktor
    Krava(String imeKrave)
    {
        super(imeKrave);
    }
    // redefinicija abstraktne metode
    public void oglasanje()
    {
        System.out.println("Muuuu");
    }
}
```

```
public class Kaca extends Zival
{
    // konstruktor
    Kaca(String imeKace)
    {
        super(imeKace);
    }
    // redefinicija abstraktne metode
    public void oglasanje()
    {
        System.out.println("Ssssss");
    }
}
```

```
class DemoHeterogeneTabele
{
    public static void main(String[] args)
    {
        Zival[] z=new Zival[10];
        int odg;
        String imeZivali;
        int i;
    }
}
```

```
// vpis podatkov
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("Vtipkaj ime zivali: ");
    imeZivali=BranjePodatkov.preberiString();
    switch (odg)
    {
        case 1:
            z[i]=new Pes(imeZivali);
            break;
        case 2:
            z[i]=new Krava(imeZivali);
            break;
        case 3:
            z[i]=new Kaca(imeZivali);
            break;
    }
}
```

```
// izpis vsebine tabele
for (i=0; i<z.length; ++i)
{
    System.out.print(z[i].vrniIme()+" se oglasa ");
    z[i].oglasanje();
}
}
```

Zival1.java, Pes1.java, Krava1.java, DemoEquals.java - Prikaz redefinicije metode equals() v razredu Zival. Redefiniramo metodo avtomatsko podedujeta podrazreda Pes in Krava.

```
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

    // redefinicija metode equals()
    public boolean equals(Zival1 z2)
    {
        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");
    }
}
```

```
class DemoEquals
{
    public static void main(String[] args)
    {
        Pes1 pes1=new Pes1("Fifi");
        Pes1 pes2=new Pes1("Fifi");
        Krava1 krava1=new Krava1("Liska");
        Krava1 krava2=new Krava1("Fifi");

        System.out.println(pes1.equals(pes2));    // true
        System.out.println(krava1.equals(krava2));    // false
        System.out.println(pes1.equals(krava2));    // true
    }
}
```

IZPIS

```
true
false
true
```

V. GRAFIKA

Omogoči uporabo paketov

```
import java.awt.*;           // awt
import java.awt.event.*;    // dogodke znotraj awt paketa
import javax.swing.*;       // Swing
```

Oblika pisave

Font.PLAIN (normalno), Font.BOLD (poudarjeno), Font.ITALIC (postrani),
Font.BOLD+Font.ITALIC (kombinacija)

BARVA - setColor(barva);

Barve		
black	green	red
blue	lightGray	white
cyan	magenta	yellow
darkGray	orange	
gray	pink	

Primer: g.setColor(Color.blue);

Color(int rdeca, int zelena, int modra)

Ozadje – nastavi barvo podlage (metodo je potrebno poklicati preden prikažemo okno)
setBackground(barva);

Risalna barva – nastavi barvo s katero rišemo
setForeground(barva);

Risanje polnjenih likov

Imena metod so enaka kot za risanje likov, le predpona draw se spremeni s fill .	
pravokotnik	fillRect(x,y,sirina,visina);
pravokotnik z zaobljenimi oglišči	fillRoundRect(x,y,sirina,visina,rH,rV);
Pravokotnik, ki daje vtis gumba	fill3DRect(x,y,sirina,visina,dvig);
elipsa, krog	fillOval(x,y,sirina,visina):
mnogokotnik	fillPolygon(p);
mnogokotnik	fillPolygon(x,y,n);
lok	fillArc(x,y,sirina,visina,zackot,koncnikot)

Črte

črta	drawLine(x1,y1,x2,y2);
črte v tem primeru z desetimi kordinati x,y!	public static int[] x = {2,4,6,8,10,12,14,16,18,20}; public static int[] y = {3,5,7,9,11,13,15,17,19,21}; drawPolyline(x,y,10);

5. naloga iz izpita – Pet koncentričnih (enak center) krogov

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

```
class MojKomponent extends JComponent
{
```

```
    // tabela z barvami
    private final static Color[] barva = { Color.blue, Color.red, Color.green,
    Color.yellow, Color.orange };;
```

```
    // tabela z radiji
    private final static int[] radij = new int[] { 20, 30, 40, 50, 60 };
```

```
    // kordinate - centra
    private final static int CX = 100, CY = 100;
```

```
    public MojKomponent()
    {
        setBackground(Color.white);
        setPreferredSize( new Dimension(200,200) );
    }
```

```
    public void paintComponent(Graphics g)
    {
        g.setColor( getBackground() );           // barva ozadja

        g.fillRect( 0, 0, getWidth(), getHeight() );

        for(int i=0; i<radij.length; i++) {
            g.setColor( barva[i] );
            g.drawOval(
                CX - radij[i], // x
                CY - radij[i], // y
                2 * radij[i], // dolžina
                2 * radij[i] // višina
            );
        }
    }
```

```
    }
}

public class Naloga5 extends JFrame
{
    public Naloga5()
    {
        setTitle( getClass().getName() );
        setResizable(false);
        setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );

        MojKomponent mc = new MojKomponent();

        Container c = getContentPane();
        c.setLayout( new BorderLayout() );
        c.add( mc );

        pack();           // določa velikost okna
    }

    public static void main(String[] args)
    {
        JFrame f = new Naloga5(); // tukaj tudi prebere ime okna
        f.setVisible(true);       // omogoča vidljivost okna
    }
}
```



```

public void paintComponent(Graphics g)
{
    // barva odzadja
    g.setColor( getBackground() );
    g.fillRect( 0, 0, getWidth(), getHeight() );

    for(int i=0; i<radij.length; i++) {
        g.setColor( barva[i] );
        g.drawOval(
            50,           // x
            50,           // y
            2 * radij[i], // dolžina
            2 * radij[i] // višina
        );
    }
}

```

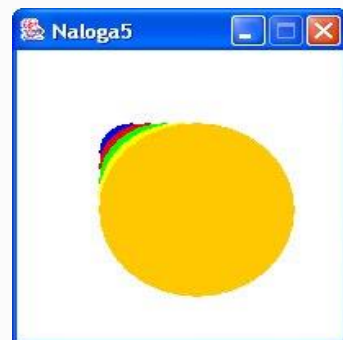


```

public void paintComponent(Graphics g)
{
    // barva odzadja
    g.setColor( getBackground() );
    g.fillRect( 0, 0, getWidth(), getHeight() );

    for(int i=0; i<radij.length; i++) {
        g.setColor( barva[i] );
        g.fillOval(
            50,           // x
            50,           // y
            2 * radij[i], // dolžina
            2 * radij[i] // višina
        );
    }
}

```



class Panel extends JPanel **DRAW = RISANJE; FILL = POLNO!!!**

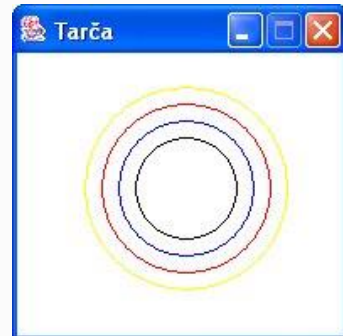
```

{
    private final static int[] radij = new int[] { 20, 30, 40, 50, 60 };
    private static Color[] barve = {Color.white, Color.black, Color.blue, Color.red, Color.yellow};

    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);
        setBackground(Color.white);

        for (int i=0; i<radij.length; i++)
        {
            g.setColor(barve[i]);
            g.drawOval(100-radij[i],80-radij[i],2*radij[i],2*radij[i]);
        }
    }
}

```



class Panel extends JPanel **VRSTNI RED V TABELI JE ZELO POMEMBEN!!!**

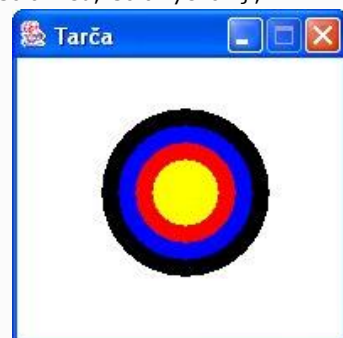
```

{
    private final static int[] radij = new int[] { 60, 50, 40, 30, 20 };
    private static Color[] barve = {Color.white, Color.black, Color.blue, Color.red, Color.yellow};

    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);
        setBackground(Color.white);

        for (int i=0; i<radij.length; i++)
        {
            g.setColor(barve[i]);
            g.fillOval(100-radij[i],80-radij[i],2*radij[i],2*radij[i]);
        }
    }
}

```



```

class Panel extends JPanel
{
    public static int[] x = {2,4,6,8,10,12,14,16,18,20};
    public static int[] y = {3,5,7,9,11,13,15,17,19,21};

    public void paintComponent(Graphics g)
    {
        super.paintComponent(g); // vzamemo g spremenljivko (graphics)
        setBackground(Color.white);

        // uporabimo g, ker je spremenljivka, ki se nanaša na paintComponent
        // črte
        // črta
        g.drawLine(10,10,100,80); // (x, y, x2, y2); x,y - začetna pozicija, x2,y2 - končna pozicija

        // pisava
        // pisava (niz znakov)
        g.drawString("Niz besed (String)",40,20); // x,y kordinatami (npr. 40,20)

        // pisava (nova oblika, velikost,...)
        Font f=new Font("SansSerif",Font.PLAIN+Font.BOLD,15);

        g.setFont(f);
        g.drawString("Drugi niz!",40,120); // x,y kordinatami (npr. 40,20)

        // vrne dolžino niza
        FontMetrics fm=g.getFontMetrics(f);
        int dolzina=fm.stringWidth("gfd");
        g.drawString("fd",50,50);

        // risanje lokov
        g.drawArc(120,120,40,60,30,300);

        // pacman
        g.drawArc(180,180,80,80,0,300);
        g.setColor(Color.yellow);
        g.fillArc(181,181,79,79,0,300);

        // mnogokotnik
        g.setColor(Color.blue);

        Polygon p=new Polygon();
        p.addPoint(300,300); // prva kordinata (x,y)
        p.addPoint(300,600); // druga kordinata (x,y)
        p.addPoint(800,500); // tretja kordinata (x,y)
        // p.addPoint(x,y); // ...
        g.drawPolygon(p);

        g.setColor(Color.black);

        // risanje, črte z mnogo x,y kordinatov! 10 - je število vseh kordinat (x,y) skupaj!
        g.drawPolyline(x,y,10);

        // kvadrat
        g.drawRect(250,250,40,40);

        // pravokotnik
        g.drawRect(250,200,80,60);
        // drawRect(x,y,sirina,visina);

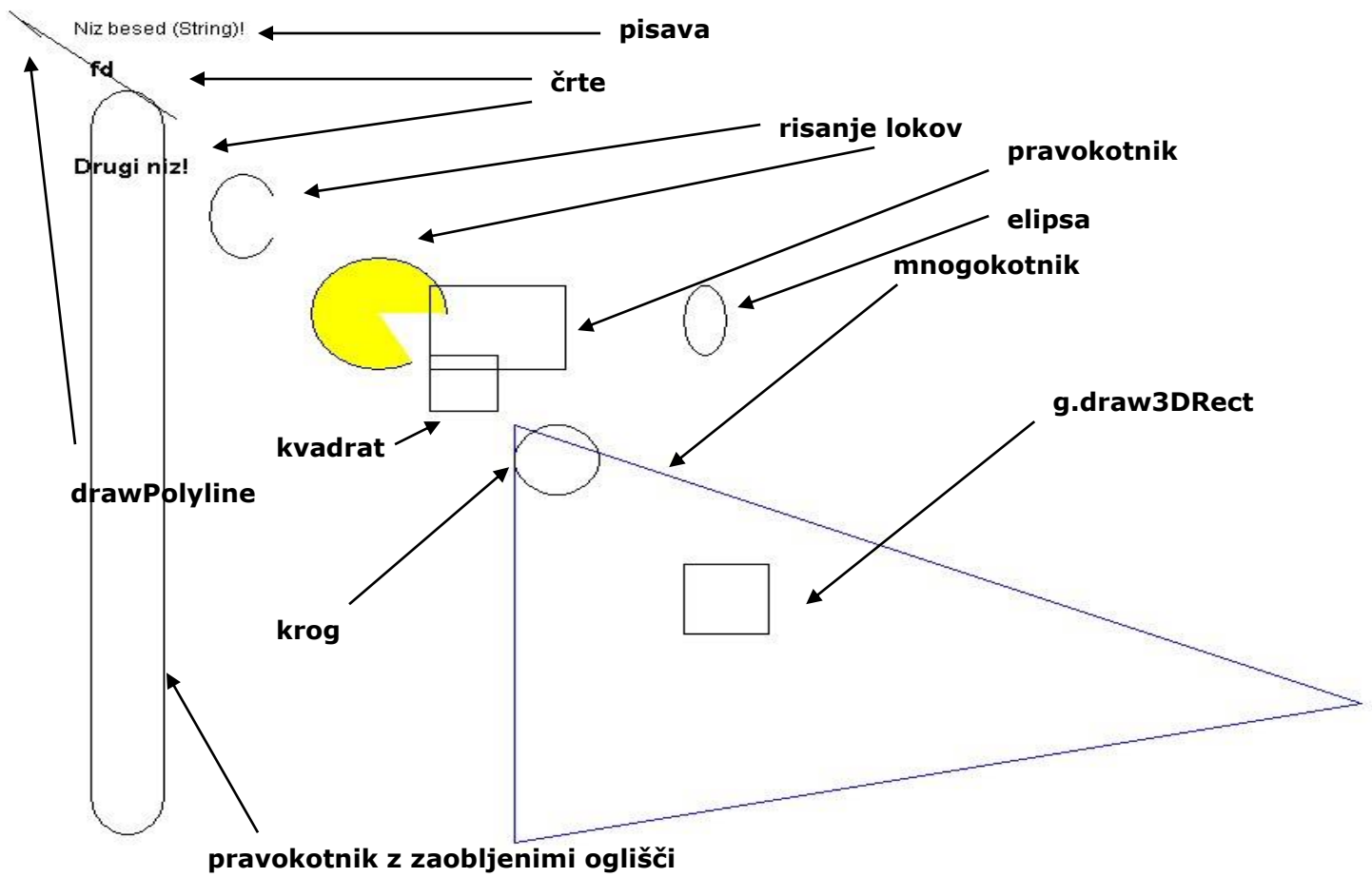
        // pravokotnik z zaobljenimi oglišči
        g.drawRoundRect(50,60,43,534,543,53);
        // drawRoundRect(x,y,sirina,visina,rH,rV); rH in rV - določate horizontalni in vertikalni polmer loka

        // pravokotnik
        g.draw3DRect(400,400,50,50,true);
        // g.draw3DRect(x,y,sirina,visina,dvig); dvig = true/false; dvignjen/vgreznjen na okno

        // krog
        g.drawOval(300,300,50,50);
        // g.drawOval(x,y,sirina,visina);

        // elipsa
        g.drawOval(400,200,25,50);
    }
}

```



VI. APLET

Vsaj eno izmed spodnjih metod je potrebno deklarirati!!!

```
public void init()           // izvede, ko se applet prvič naloži
public void start()          // se izvede takoj za metodo init()
public void stop()           // se izvede, ko uporabnik zapusti applet
public void destroy()        // se izvede, ko uporabnik zapre brskalnik
```

```
l.setText("blablaba");
l.getText()
```

```
GridLayout           // zaporedi v celice
GridBagLayout        // omogoča dodajanje komponent v točno določeno celico
CardLayout           // komponente se nalagajo ena na drugo
BoxLayout             // vse komponente razporedi v eno vrsto ali v en stolpec
```

HTML KODA (applet.html)

```
<html>
<applet code="Applet.class" width="450" height="200">
</applet>
</html>
```

APPLET (Applet.java)

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

public class Applet extends JApplet implements ActionListener
{
    // deklaracija in inicializacija spremenljivk
    JLabel labela=new JLabel("Pozdravljen(a)!");
    JLabel vprasanje=new JLabel("Kako ti je ime?");
    JButton gumb=new JButton("Preberi");
    JTextField vnosnoPolje=new JTextField(10);

    Font pisava1 = new Font("TimesRoman",Font.ITALIC,24);
    Font pisava2 = new Font("Helvetica",Font.BOLD,20);

    FlowLayout flow=new FlowLayout();

    public void init()
    {
        // elemente dodamo na polje!
        Container con = getContentPane();
        con.setBackground(Color.white);

        con.setLayout(flow);

        labela.setFont(pisava1);
        con.add(labela);

        labela.setFont(pisava2);
        con.add(vprasanje);
        con.add(vnosnoPolje);
        con.add(gumb);

        vnosnoPolje.requestFocus();           // kurzor v polju!

        // poslušalec
        gumb.addActionListener(this);
        vnosnoPolje.addActionListener(this);
    }
}
```

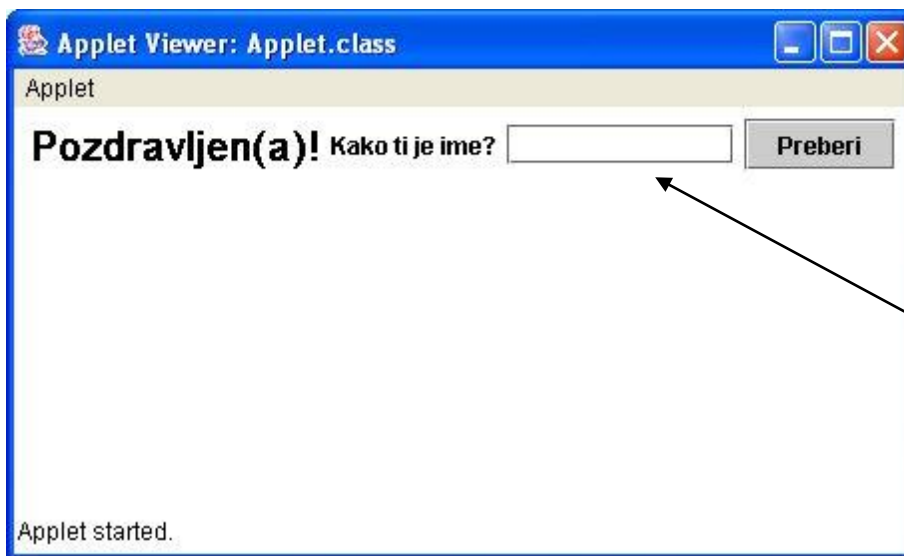
```

public void actionPerformed(ActionEvent d)
{
    Object izvor=d.getSource();

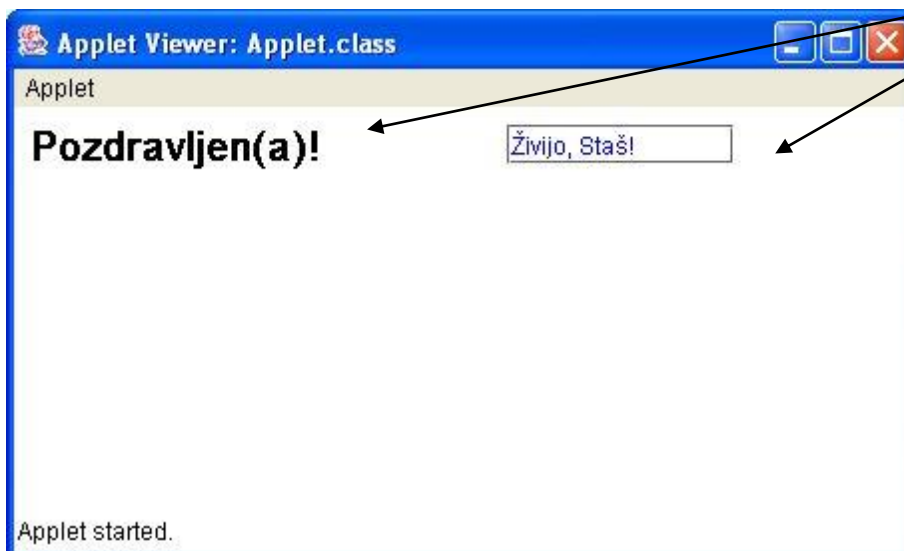
    if ((izvor==gumb) || (izvor==vnosnoPolje))
    // gumb = v primeru pritiska na gumb
    // vnosnoPolje = v primeru pritiska na tipko ENTER
    {
        String ime=vnosnoPolje.getText();
        remove(vprasanje);
        remove(gumb);

        vnosnoPolje.setForeground(Color.blue);
        vnosnoPolje.setText("Živijo, "+ime+"!");
        repaint();
    }
}
}

```



Vnosno polje!



Ko se pritisne gumb »Preberi« ali pritisne tipko »Enter« se izbrišeta tudi dva elementa (vprasanje) in (gumb)!

Math.	
static int double, float, long	abs (int a) Vrne absolutna vrednost (npr. 4).
static double	acos (double a) Vrne inverzno funkcijo kosinusa (in the range of 0.0 through π).
static double	asin (double a) Vrne inverzno funkcijo sinusa (in the range of $-\pi/2$ through $\pi/2$).
static double	atan (double a) Vrne inverzno funkcijo tangensa (in the range of $-\pi/2$ through $\pi/2$).
static double	ceil (double a) Vrne najmanjše število (double), ki pa ni manjše od argumenta a in je enako enmu matematičnu integerju.
static double	cos (double a) Vrne cosinus argumenta a.
static double	exp (double a) EkspONENTNA funkcija.
static double	floor (double a) Vrne največje število (double), ki pa ni večje od argumenta a in je enako enmu matematičnu integerju.
static double	log (double a) Logaritemska funkcija.
static int double, float, long	max (int a, int b) Vrne argument, ki je večji.
static int double, float, long	min (int a, int b) Vrne argument, ki je manjši.
static double	pow (double a, double b) Vrne prvo vrednost agrumenta, ki je na potenco (npr. $\text{Math.pow}(4,2) \Rightarrow 4^2 = 16.0$).
static double	random () Vrne naključno vrednost (med 0.0 in 1.0).
static double	rint (double a) Vrne število, ki je najbližje argumentu in je (spet) »en« amtematičen integer.
static long	round (double a) Zaokroži na najbližje celo število.
static int	round (float a) Vrne celo število (zanemari manjšo napako) (npr. Diagram Krožnih Izsekov).
static double	sin (double a) Vrne sinus argumenta.
static double	sqrt (double a) Vrne kvadratni koren argumenta.
static double	tan (double a) Vrne tangens argumenta.
static double	toDegrees (double angrad) Pretvori radiane v stopinje.
static double	toRadians (double angdeg) Pretvori stopinje v radiane.

a = argument !!!

double a = 4.0

double b = 2.0

npr. $\text{Math.pow}(a,b) \Rightarrow 16.0$

Konstante: PI, E... $\text{Math.<ime konstante>}$

Metode za delo z nizi	
replace()	zamenja vsa nastopanja znaka zn1 z znakom zn2
length()	vrne dolžino niza (število znakov)
delete()	izbriše določen znak, črko
indexOf(zn)	vrne pozicijo znaka zn znotraj nekega niza (indeksi tečejo od 0 dalje); če niz ne vsebuje znaka zn, vrne -1
indexOf(zn,poz)	išče znak zn od pozicije poz dalje
toString()	je implicitno prisotna pri konkatenciji
charAt(n)	vrne znak, ki se nahaja na n-tem mestu v nizu
startsWith(niz1)	vrne true, če se niz začne s podnizom niz1
endsWith(niz)	Vrne true, če se niz konča s podnizom niz1
	pretvori v tiskane črke
substring(zac,kon)	Vrne podniz, ki se prične na mestu zac in konča na mestu kon-1
toString(arg)	Pretvori v niz argument arg, ki je lahko kateregakoli osnovnega tipa (ni metoda razreda String) - implicitno prisotna pri konkatenciji
Metode	Opis
isUpperCase()	Preveri ali je znak velika črka (A - true/a - false)
toUpperCase()	Pretvori malo črko v veliko (a → A)
isLowerCase()	Preveri, ali je znak mala črka (a - true/ A - false)
toLowerCase()	Pretvori veliko črko v malo (A → a)
isDigit()	Preveri, ali je znak številka ('0' - '9'; true/false)
isLetter()	Preveri, ali je znak črka ('a' - 'z'; true/false)
isLetterOrDigit()	Preveri, ali je znak črka ali številka ('A' - '0'; true/false)
isWhiteSpace()	Preveri, ali je znak presledek, tab, newline, carriage, return ali form feed (\n, \t,...)

Šumniki	
š	\u00e7
Š	\u0107
č	\u017a
Č	\u00ac
ž	\u00a7
Ž	\u00a6

Dostopna določila	Dovoljen dostop
public	Iz kateregakoli razreda ne glede na paket.
private	Samo znotraj razreda.
protected	Iz kateregakoli razreda v istem paketu in iz kateregakoli podrazreda ne glede na paket.
brez dostopnega določila	Iz kateregakoli razreda v istem paketu

Razred StringBuffer	
Spreminjanje vsebine niza	
append(niz2)	Omogoča dodajanje na koncu niza (doda na konec niz2 tipa String)
insert(poz,niz2)	Omogoča vrivanje sredi niza (vrine niz2 na pozicijo poz)
delete(zac,kon)	Briše znake med pozicijama zac in kon-1
deleteCharAt(poz)	Briše samo en znak na poziciji poz
setCharAt(poz,zn)	Vpiše znak zn na pozicijo poz
replace(zac,kon,niz2)	Briše znake med pozicijama zac in kon-1 ter jih nadomesti z nizom niz2
Branje podatkov iz niza	
charAt(poz)	Vrne znak, ki se nahaja na poziciji poz
substring(zac,kon)	Vrne podniz, ki vsebuje znake na pozicijah od zac do kon-1
Iskanje podatkov v nizu	
indexOf(niz2)	Vrne pozicijo, kjer se prične niz2
indexOf(niz2,poz)	Išče niz2 od pozicije poz dalje
Primerjava nizov	
niz1.equals(niz2)	Primerja niz1 z niz2
equalsIgnoreCase()	Deluje na podoben način, kot equals() samo ne razlikuje med velikimi in malimi črkami
compareTo()	Vrne celo število (če sta oba niza enaka vrne 0) Če je biz, iz katerega je bila metoda poklicana, večji, vrne pozitivno vrednost v nasprotnem primeru pa negativno
Kapaciteta, dolžina niza	
capacity()	Vrne trenutno kapaciteto
length()	Vrne trenutno dolžino niza
setLength()	Omogoča nastavljanje dolžine niza

Druge metode razreda object	
getClass()	Vrne objekt tipa Class, ki vsebuje ime razreda
hashCode()	Izračuna has kodo (za shranjevanje objektov v razpršenih tabelah)
notify()	Pri delu z nitmi (ang. threads)
notifyAll()	Pri delu z nitmi (ang. threads)
wait()	Pri delu z nitmi (ang. threads)
finalize()	Metoda, ki se izvede ob uničenju objekta

Tip dogodka	Vmesnik	Odzivne metode
ActionEvent	ActionListener	actionPerformed(ActionEvent e)
ItemEvent	ItemListener	itemStateChanged(ItemEvent e)
TextEvent	TextListener	textValueChanged(ActionEvent e)
AdjustmentEvent	AdjustmentListener	adjustmentValueChanged(AdjustmentEvent e)
ContainerEvent	ContainerListener	componentAdded(ContainerEvent e) componentRemoved(ContainerEvent e)
ComponentEvent	ComponentListener	componentMoved(ComponentEvent e) componentHidden(ComponentEvent e) componentResized(ComponentEvent e) componentShown(ComponentEvent e)
FocusEvent	FocusListener	focusGained(FocusEvent e) focusLost(FocusEvent e)
MouseEvent	MouseListener MouseMotionListener	mousePressed(MouseEvent e) mouseReleased(MouseEvent e) mouseEntered(MouseEvent e) mouseExited(MouseEvent e) mouseClicked(MouseEvent e) mouseDragged(MouseEvent e) mouseMoved(MouseEvent e) mouseMoved(MouseEvent e)
KeyEvent	KeyListener	keyPressed(KeyEvent e) keyTyped(KeyEvent e) keyReleased(KeyEvent e)
WindowEvent	WindowListener	windowActivated(WindowEvent e) windowClosing(WindowEvent e) windowClosed(WindowEvent e) windowDeactivated(WindowEvent e) windowDeiconified(WindowEvent e) windowIconified(WindowEvent e) windowOpened(WindowEvent e)

Vmesnik **WindowListener** predpisuje 7 metod, ki ustrezajo posameznih dogodkom tipa WindowEvent!

V razredu **WindowAdapter** se te metode implementirane tako, da ob dogodku ne izvršijo nobene akcije!

Komponente (izvori dogodkov)	Metode za dodajanje poslušalcev
JButton, JCheckBox, JComboBox, JToolBar, JTextField, JRadioButton	addActionListener()
JScrollBar	addAdjustmentListener()
Vse Swing komponente	addFocusListener(), addKeyListener(), addMouseListener(), addMouseMotionListener()
JButton, JCheckBox, JComboBox, JRadioButton	addItemListener()
JWindow, JFrame	addWindowListener()
JSlider	addChangeListener()