

Uvod v programiranje

1. vaje

Borut Lužar

Fakulteta za informacijske študije Novo mesto

Novo mesto, 18. oktober 2016

Osnovni podatki

- ▶ Asistent: Borut Lužar
- ▶ E-naslov: borut.luzar@fis.unm.si
- ▶ Govorilna ura: po dogovoru (pišite e-mail)
- ▶ Pisarna: FIŠ, 1. nadstropje, na koncu hodnika - desno

- ▶ Dodatna pojasnitev snovi s predavanj
- ▶ Pisanje programov v javi
- ▶ Domače naloge
 - ▶ Na koncu vsakih vaj
 - ▶ Zagovorjate tri naključno izbrane
 - ▶ Zagovori bodo: 1.rok na 10. vajah, 2. rok na 15. vajah.
 - ▶ Popravni rok: pred zagovori seminarskih nalog
- ▶ Osnovna literatura - prosojnice s predavanj in vaj
- ▶ Dodatna literatura - glej prosojnice s predavanj

Kaj potrebujemo

- ▶ Namestitev **Java Development Kit**
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- ▶ Nastavitev sistemske poti do mape z datoteko **javac.exe**
Primer za Windows 7: Računalnik → Lastnosti → Dodatne nastavitve sistema → Spremenljivke okolja → 'PATH' → Uredi → Dodaj pot (običajno C:/Program Files/Java/jdk1.8.0/bin;)
- ▶ (priporočeno) Namestitev **TextPad**
<http://textpad.com/>
- ▶ (priporočeno) Svoj prenosni računalnik

Prvi program!

- ▶ Navada je, da prvi program, ki ga napišemo, izpiše "Hello, world!":
- ▶ V urejevalnik besedil (npr. TextPad) zapišemo naslednji program:

```
class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, world!");  
    }  
}
```

- ▶ Program shranimo v datoteko z imenom, ki opiše program in končnico **.java**, npr. **HelloWorld.java**

Zagon programa

- ▶ Datoteko **[ImeDatoteke].java** prevedemo s prevajalnikom **javac**
- ▶ Če se uspešno prevede, se shrani v datoteko **[ImeRazreda].class**
- ▶ Program poženemo s programom **java**
- ▶ Prevajanje in zagon programov bomo izvajali v ukazni vrstici (Windows gumb + R → cmd)
c:/JavaProgrami>javac HelloWorld.java
c:/JavaProgrami>java HelloWorld
Hello, world!

- ▶ Pri pisanju programa se nam pogosto zgodijo napake
- ▶ Tipične:
 - ▶ pozabljeno podpičje ';'
 - ▶ pozabljeni končni ali začetni oklepaji '{', '}', '(', ')'
 - ▶ napačno zapisano ime spremenljivke
- ▶ Prevajalnik programa z napako ne zna prevesti in izpiše napako
- ▶ Opis napake vedno pozorno preberemo!

Osnovni tipi in spremenljivke

- ▶ Naloga 1: Definiraj pet spremenljivk različnih tipov, določi jim poljubne vrednosti in na zaslon izpiši njihovo ime ter vrednost.

Osnovni tipi in spremenljivke

- Naloga 1: Definiraj pet spremenljivk različnih tipov, določi jim poljubne vrednosti in na zaslon izpiši njihovo ime ter vrednost.

```
class SpremenljivkeInTipi {  
    public static void main(String[] args) {  
  
        // Definicije spremenljivk in prireditve vrednosti  
        char crka = 'a';  
        String beseda = "Abeceda";  
        int naravnoStevilo = 5;  
        double realnoStevilo = -2.3;  
        boolean resnica = true;  
  
        // Izpis  
        System.out.println("crka = " + crka);  
        System.out.println("beseda = " + beseda);  
        System.out.println("naravnoStevilo = " + naravnoStevilo);  
        System.out.println("realnoStevilo = " + realnoStevilo);  
        System.out.println("resnica = " + resnica);  
    }  
}
```

- ▶ Programsko kodo opisujemo s komentarji
- ▶ Enovrstični: `//` Ena vrstica komentarja
- ▶ Večvrstični: `/*` Komentar
v treh
vrsticah `*/`
- ▶ Naloga 2: V programu iz naloge 1 opiši, kaj vsaka izmed spremenljivk pomeni.

- ▶ Naloga 3: Na spletu poišči šifro Unicode standarda za znake 'A', 'a', 'ž' in '\$' in jih izpiši.

- Naloga 3: Na spletu poišči šifro Unicode standarda za znake 'A', 'a', 'ž' in '\$' in jih izpiši.

```
class UbeznaZaporedja {  
  
    public static void main(String[] args) {  
  
        char znak_A = '\u0041';  
        char znak_a = '\u0061';  
        char znak_zh = '\u017E';  
        char znak_hash = '\u0023';  
  
        // Izpis  
        System.out.println("A = " + znak_A);  
        System.out.println("a = " + znak_a);  
        System.out.println("ž = " + znak_zh);  
        System.out.println("# = " + znak_hash);  
    }  
}
```

- ▶ Naloga 4: V eno spremenljivko zapiši svoj naslov v treh vrsticah.

- ▶ Naloga 4: V eno spremenljivko zapiši svoj naslov v treh vrsticah.

```
class DomaciNaslov {  
    public static void main(String[] args) {  
        String naslov = " Janez Novak\n Cesta na trg 12\n 1215 Pošta";  
  
        // Izpis  
        System.out.println(naslov);  
    }  
}
```

Operatorji

- ▶ Naloga 5: Definiraj dve celoštevilski spremenljivki x in y ter izpiši njun:
 - ▶ produkt;
 - ▶ vsoto;
 - ▶ razliko;
 - ▶ kvocient;
 - ▶ ostanek (za obe) pri deljenju s 3;

Operatorji

- ▶ Naloga 5: Definiraj dve celoštevilski spremenljivki x in y ter izpiši njun:
 - ▶ produkt;
 - ▶ vsoto;
 - ▶ razliko;
 - ▶ kvocient;
 - ▶ ostanek (za obe) pri deljenju s 3;

```
class Racunanje {  
  
    public static void main(String[] args) {  
  
        int x = 3, y = 4;  
  
        System.out.println(x + y);  
        System.out.println(x - y);  
        System.out.println(x * y);  
        System.out.println(x / y);  
        System.out.println(x % 3);  
        System.out.println(y % 3);  
    }  
}
```


- ▶ Naloga 6: Definiraj celi števili b in s , katerih vrednosti sta $b =$ število bratov, $s =$ število sester. Nato definiraj spremenljivko, katere vrednost bo stavek "**Imam b bratov in s sester.**", pri čemer se vrednosti b in s prebereta iz spremenljivk.

- ▶ Naloga 6: Definiraj celi števili b in s , katerih vrednosti sta b = število bratov, s = število sester. Nato definiraj spremenljivko, katere vrednost bo stavek "**Imam b bratov in s sester.**", pri čemer se vrednosti b in s prebereta iz spremenljivk.

```
class BratjeInSestre {  
  
    public static void main(String[] args) {  
  
        int b = 0;  
        int s = 0;  
  
        String izpis = "Imam " + b + " bratov in " + s + " sester.";   
        System.out.println(izpis);  
  
    }  
}
```

Logične operacije

- ▶ Naloga 7: Definiraj boolean spremenljivko ter izpiši njeno vrednost za naslednje trditve:

- ▶ $3 \leq 4$;
- ▶ $4! = 5$;
- ▶ $(3 + 2 < 6 \quad || \quad 1 > 2)$;
- ▶ $5 + 5 < 3 \quad \&\& \quad 6 > 2$.

Preveri pravilnost!

- ▶ Naloga 7: Definiraj boolean spremenljivko ter izpiši njeno vrednost za naslednje trditve:

- ▶ $3 \leq 4$;
- ▶ $4! = 5$;
- ▶ $(3 + 2 < 6 \quad || \quad 1 > 2)$;
- ▶ $5 + 5 < 3 \quad \&\& \quad 6 > 2$.

Preveri pravilnost!

```
class Logika {  
  
    public static void main(String[] args) {  
  
        boolean resnicaAliLaz;  
  
        resnicaAliLaz = (3 <= 4);  
        System.out.println(resnicaAliLaz);  
  
        resnicaAliLaz = (4 != 5);  
        System.out.println(resnicaAliLaz);  
  
        resnicaAliLaz = ((3 + 2 < 6 || 1 > 2));  
        System.out.println(resnicaAliLaz);  
  
        resnicaAliLaz = (5 + 5 < 3 && 6 > 2);  
        System.out.println(resnicaAliLaz);  
  
    }  
}
```

Elementarne funkcije

V javi so elementarne funkcije, npr. `sin`, `cos`, `sqrt`, na voljo v paketu **Math**.

- ▶ Naloga 8: Izračunaj ploščino kroga z danim polmerom r .

Elementarne funkcije

V javi so elementarne funkcije, npr. `sin`, `cos`, `sqrt`, na voljo v paketu **Math**.

- Naloga 8: Izračunaj ploščino kroga z danim polmerom r .

```
class PloscinaKroga {  
  
    public static void main(String[] args) {  
  
        double r = 1.0;  
        double ploscina = Math.PI * r * r;  
        System.out.println("Ploščina kroga je: " + ploscina);  
    }  
}
```

1. domača naloga

- ▶ Izpiši tabelo (malih) črk slovenske abecede v treh stolpcih. Pri tem za črke uporabi šifre Unicode standarda!

Namig: stolpce simuliramo z uporabo znaka `'\t'`, ki predstavlja tabulator.

Povezava do Unicode tabele: http://en.wikipedia.org/wiki/List_of_Unicode_characters

- ▶ Podobno kot v nalogi 8 izračunaj obseg kroga s polmerom enakim številu dni v mesecu, v katerem si se rodila/rodil.

Uvod v programiranje

2. vaje

Borut Lužar

Fakulteta za informacijske študije Novo mesto

Novo mesto, 20. oktober 2016

- ▶ Naloga 1: Definiraj dve realni števili x in y , določi jima poljubno vrednost in na zaslon izpiši stavek "**Povprečna vrednost števil x in y je enaka $\frac{x+y}{2}$.**", z dejanskima vrednostima obeh spremenljivk ter njunim povprečjem. Preveri ali so vrednosti pravilne!

- ▶ Naloga 1: Definiraj dve realni števili x in y , določi jima poljubno vrednost in na zaslon izpiši stavek "**Povprečna vrednost števil x in y je enaka $\frac{x+y}{2}$.**", z dejanskima vrednostima obeh spremenljivk ter njunim povprečjem. Preveri ali so vrednosti pravilne!

```
class Ponavljanje {  
  
    public static void main(String[] args) {  
  
        double x = 1.3;  
        double y = 6.7;  
  
        double povprecje = (x + y) / 2;  
  
        System.out.println("Povprečna vrednost spremenljivk " + x + " in " + y +  
            " je enaka " + povprecje + ".");  
    }  
}
```

- ▶ Zanke omogočajo ponavljanje enakih ukazov s spremenjenimi parametri
- ▶ Oblika:

```
for (začetek; pogoj; korak) {  
    ukaz;  
}
```

- Primer: Seštevanje prvih desetih naravnih števil.

```
class ForPrimer {  
  
    public static void main(String[] args) {  
  
        int vsota = 0;  
        for(int i = 1; i <= 10; i = i + 1)  
        {  
            vsota = vsota + i;  
        }  
        System.out.println("Vsota prvih desetih " +  
            "naravnih števil je " + vsota + ".");  
    }  
}
```

- ▶ Naloga 2: Z uporabo zanke **for** zmnoži prvih osem **sodih** naravnih števil. Nato izpiši **polovično vrednost** rezultata.

- Naloga 2: Z uporabo zanke **for** zmnoži prvih osem **sodih** naravnih števil. Nato izpiši **polovično vrednost** rezultata.

```
class MnozenjeSodih {  
  
    public static void main(String[] args) {  
  
        int zmnozek = 1;  
  
        for(int i = 2; i <= 16; i = i + 2)  
        {  
            zmnozek = zmnozek * i;  
        }  
        zmnozek = zmnozek / 2;  
        System.out.println("Polovična vrednost zmnožka je " + zmnozek + ".");  
    }  
}
```

Tabele

- ▶ Vsebujejo vrednosti istega tipa
- ▶ Deklariramo jih z oglatima oklepajema `[]`
- ▶ Za prirejanje vrednosti potrebujemo ukaz **new**
- ▶ Primer: `int[] tabela = new int[7];`
- ▶ Dolžino tabele dobimo s klicem lastnosti: `tabela.length`
- ▶ Direkten vnos elementov:
`int[] tabela = new int[]{1, 2, 3};`
- ▶ Do *i*-tega elementa v tabeli dostopamo z ukazom: `tabela[i]`
- ▶ POZOR! Prvi element ima index 0!

Dnevi v tednu

- ▶ Naloga 3: Definiraj tabelo *teden*, katere elementi bodo imena dni v tednu ter izpiši vse njene elemente **na dva načina!**

- Naloga 3: Definiraj tabelo *teden*, katere elementi bodo imena dni v tednu ter izpiši vse njene elemente **na dva načina!**

```
class Teden {  
  
    public static void main(String[] args) {  
  
        String[] teden = new String[]{"ponedeljek", "torek", "sreda",  
                                       "četrtek", "petek", "sobota", "nedelja"};  
  
        // Način 1  
        System.out.println(teden[0]);  
        System.out.println(teden[1]);  
        System.out.println(teden[2]);  
        System.out.println(teden[3]);  
        System.out.println(teden[4]);  
        System.out.println(teden[5]);  
        System.out.println(teden[6]);  
  
        // Način 2  
        for(int i = 0; i < teden.length; i++)  
        {  
            System.out.println(teden[i]);  
        }  
    }  
}
```

Prijatelji

- ▶ Naloga 4: Definirajte dve tabeli. V prvi naj bodo elementi imena petih prijateljev v drugi pa starosti teh prijateljev. Nato izpišite podatke v obliki: "**Prijatelj** *Toni* **je star 25 let.**".

- Naloga 4: Definirajte dve tabeli. V prvi naj bodo elementi imena petih prijateljev v drugi pa starosti teh prijateljev. Nato izpišite podatke v obliki: **"Prijatelj *Toni* je star 25 let."**.

```
class Prijatelji {  
  
    public static void main(String[] args) {  
  
        String[] prijatelji = new String[]{"Alenka", "Bine", "Cene", "David", "Eneja"};  
        int[] stevilo = new int[]{4,23,2,54,43};  
  
        for(int i = 0; i < prijatelji.length; i++)  
        {  
            System.out.println("Prijatelj " + prijatelji[i] +  
                " ima " + stevilo[i] + " prijateljev.");  
        }  
    }  
}
```

Vezana vloga

- ▶ Naloga 5: Na banko ste položili 1000 enot premoženja. Napišite program, ki izračuna vaše premoženje v naslednjih 12 mesecih, če se vloga vsak mesec poveča za 3 odstotke. Namig: Definirajte tabelo s 13 polji, nato polja napolnite po vrsti.

Vezana vloga

- Naloga 5: Na banko ste položili 1000 enot premoženja. Napišite program, ki izračuna vaše premoženje v naslednjih 12 mesecih, če se vloga vsak mesec poveča za 3 odstotke. Namig: Definirajte tabelo s 13 polji, nato polja napolnite po vrsti.

```
class VezanaVloga {  
  
    public static void main(String[] args) {  
  
        double start = 1000;  
        double[] zneski = new double[13];  
  
        zneski[0] = start;  
        for(int i = 1; i < zneski.length; i++)  
        {  
            zneski[i] = zneski[i-1] * 1.03;  
        }  
  
        System.out.println("Zneski po mesecih:");  
        for(int i = 0; i < zneski.length; i++)  
        {  
            System.out.println(i + ". mesec: " + zneski[i]);  
        }  
    }  
}
```

- ▶ Naloga 6: Definirajte tabelo predmetov, ki jih poslušate in tabelo pričakovanih ocen za te predmete. Izpišite vse v obliki: **Predmet: ocena**. Na koncu izpišite še povprečje ocen.

- Naloga 6: Definirajte tabelo predmetov, ki jih poslušate in tabelo pričakovanih ocen za te predmete. Izpišite vse v obliki: **Predmet: ocena**. Na koncu izpišite še povprečje ocen.

```
class Ocene {  
  
    public static void main(String[] args) {  
  
        String[] predmeti = new String[]{"Matematika", "Programiranje", "Nekaj sociološkega"};  
        int[] ocene = new int[]{10,10,6};  
  
        System.out.println("Spričevalo\n");  
        double vsota = 0;  
        for(int i = 0; i < predmeti.length; i++)  
        {  
            vsota = vsota + ocene[i];  
            System.out.println(predmeti[i] + ": " + ocene[i]);  
        }  
        double povprecje = vsota / ocene.length;  
        System.out.println("Povprečje: " + povprecje);  
    }  
}
```

- ▶ Naloga 7: Z avtom ste se v štirih dneh vozili na štiri različne kraje: v Ljubljano, v Celje, v Krško in na Lepše. Do Ljubljane ste prevozili 80 km, do Celja 101, do Krškega 30 in do Lepšega 2 km. Poraba vašega avta je 5.8 l/100km. Uporabite tabele in zanko for za izpis podatkov o potovanjih. Primer izpisa za 1. dan: **"1. dan: Odpeljal sem se v Ljubljana, prevozil 80 km in porabil 4.64 litrov goriva."** Na koncu izpišite še koliko kilometrov ste prevozili in koliko goriva ste porabili.

Poraba goriva

```
class PorabaGoriva {  
  
    public static void main(String[] args) {  
  
        String[] mesta = new String[]{"Ljubljana", "Celje", "Krško", "Lepše"};  
        double[] razdalja = new double[]{80, 101, 30, 2};  
        double poraba = 5.8 / 100; // 0.058 litra na kilometer  
  
        double kmSkupaj = 0, gorivoSkupaj = 0;  
  
        for(int i = 0; i < mesta.length; i++) {  
            double porabaDanes = razdalja[i] * poraba;  
            System.out.println((i+1) + ". dan: Odpeljal sem se v " + mesta[i] +  
                ", prevozil " + razdalja[i] + " km in porabil " + porabaDanes + " litrov goriva." );  
            kmSkupaj = kmSkupaj + razdalja[i];  
            gorivoSkupaj = gorivoSkupaj + porabaDanes;  
        }  
  
        System.out.println("Prevozil sem " + kmSkupaj +  
            " km in porabil " + gorivoSkupaj + " litrov goriva.");  
    }  
}
```

2. domača naloga

- ▶ Napišite program, ki za poljuben n izpiše prvih n Fibonaccijevih števil. Na koncu izpišite še njihovo vsoto. Več o Fibonaccijevih številih najdete na naslovu:
http://sl.wikipedia.org/wiki/Fibonaccijevo_število

Uvod v programiranje

3. vaje

Borut Lužar

Fakulteta za informacijske študije Novo mesto

Novo mesto, 25. oktober 2016

- Naloga 6 (z 2. vaj): Definirajte tabelo predmetov, ki jih poslušate in tabelo pričakovanih ocen za te predmete. Izpišite vse v obliki: **Predmet: ocena**. Na koncu izpišite še povprečje ocen.

```
class Ocene {  
  
    public static void main(String[] args) {  
  
        String[] predmeti = new String[]{"Matematika", "Programiranje", "Nekaj sociološkega"};  
        int[] ocene = new int[]{10,10,6};  
  
        System.out.println("Spričevalo\n");  
        double vsota = 0;  
        for(int i = 0; i < predmeti.length; i++)  
        {  
            vsota = vsota + ocene[i];  
            System.out.println(predmeti[i] + ": " + ocene[i]);  
        }  
        double povprecje = vsota / ocene.length;  
        System.out.println("Povprečje: " + povprecje);  
    }  
}
```

Pogoste napake - 1

- Spremenljivka nima deklariranega tipa

```
spremenljivka = 3;
```

```
error: cannot find symbol
```

Pogoste napake - 1

- Spremenljivka nima deklariranega tipa

```
spremenljivka = 3;
```

```
error: cannot find symbol
```

```
double spremenljivka = 3;
```

Pogoste napake - 1

- Spremenljivka nima deklariranega tipa

```
spremenljivka = 3;
```

```
error: cannot find symbol
```

```
double spremenljivka = 3;
```

- Tabela nima definirane dimenzije

```
double[] spremenljivka = new double[];
```

```
error: array dimension missing
```

Pogoste napake - 1

- ▶ Spremenljivka nima deklariranega tipa

```
spremenljivka = 3;
```

```
error: cannot find symbol
```

```
double spremenljivka = 3;
```

- ▶ Tabela nima definirane dimenzije

```
double[] spremenljivka = new double[];
```

```
error: array dimension missing
```

```
double[] spremenljivka = new double[10];
```


Pogoste napake - 1

- ▶ Spremenljivka nima deklariranega tipa

```
spremenljivka = 3;  
error:  cannot find symbol  
double spremenljivka = 3;
```

- ▶ Tabela nima definirane dimenzije

```
double[] spremenljivka = new double[];  
error:  array dimension missing  
double[] spremenljivka = new double[10];
```

- ▶ Pri definiciji tabele manjka ključna beseda new

```
double[] spremenljivka = double[3];  
error:  ']' expected
```

Pogoste napake - 1

- ▶ Spremenljivka nima deklariranega tipa

```
spremenljivka = 3;  
error:  cannot find symbol  
double spremenljivka = 3;
```

- ▶ Tabela nima definirane dimenzije

```
double[] spremenljivka = new double[];  
error:  array dimension missing  
double[] spremenljivka = new double[10];
```

- ▶ Pri definiciji tabele manjka ključna beseda new

```
double[] spremenljivka = double[3];  
error:  ']' expected  
double[] spremenljivka = new double[3];
```

Pogoste napake - 2

- Ponovna definicija spremenljivke z že uporabljenim imenom

```
int[] vsota = new int[5];
```

```
int vsota = 3;
```

```
error: variable vsota is already defined
```

Pogoste napake - 2

- Ponovna definicija spremenljivke z že uporabljenim imenom

```
int[] vsota = new int[5];
```

```
int vsota = 3;
```

```
error: variable vsota is already defined
```

```
int[] vsota = new int[5];
```

```
int drugoIme = 3;
```

Pogoste napake - 2

- ▶ Ponovna definicija spremenljivke z že uporabljenim imenom

```
int[] vsota = new int[5];
```

```
int vsota = 3;
```

error: variable vsota is already defined

```
int[] vsota = new int[5];
```

```
int drugoIme = 3;
```

- ▶ Izpis brez določitve polja

```
int[] spremenljivka = new int[3];
```

```
System.out.println("Izpis "+ spremenljivka);
```

Ni napake - izpis pa je "čuden"...

Pogoste napake - 2

- ▶ Ponovna definicija spremenljivke z že uporabljenim imenom

```
int[] vsota = new int[5];
```

```
int vsota = 3;
```

error: variable vsota is already defined

```
int[] vsota = new int[5];
```

```
int drugoIme = 3;
```

- ▶ Izpis brez določitve polja

```
int[] spremenljivka = new int[3];
```

```
System.out.println("Izpis "+ spremenljivka);
```

Ni napake - izpis pa je "čuden"...

```
int[] spremenljivka = new int[3];
```

```
System.out.println("Izpis "+ spremenljivka[0]);
```

Pogoste napake - 2

- ▶ Ponovna definicija spremenljivke z že uporabljenim imenom

```
int[] vsota = new int[5];
```

```
int vsota = 3;
```

error: variable vsota is already defined

```
int[] vsota = new int[5];
```

```
int drugoIme = 3;
```

- ▶ Izpis brez določitve polja

```
int[] spremenljivka = new int[3];
```

```
System.out.println("Izpis "+ spremenljivka);
```

Ni napake - izpis pa je "čuden"...

```
int[] spremenljivka = new int[3];
```

```
System.out.println("Izpis "+ spremenljivka[0]);
```

- ▶ Zanka for zaključena s ';'

```
for(int i=0; i<tabela.length; i++);
```

S ';' smo zaključili zanko for!

Pogoste napake - 2

- ▶ Ponovna definicija spremenljivke z že uporabljenim imenom

```
int[] vsota = new int[5];
```

```
int vsota = 3;
```

error: variable vsota is already defined

```
int[] vsota = new int[5];
```

```
int drugoIme = 3;
```

- ▶ Izpis brez določitve polja

```
int[] spremenljivka = new int[3];
```

```
System.out.println("Izpis "+ spremenljivka);
```

Ni napake - izpis pa je "čuden"...

```
int[] spremenljivka = new int[3];
```

```
System.out.println("Izpis "+ spremenljivka[0]);
```

- ▶ Zanka for zaključena s ';'

```
for(int i=0; i<tabela.length; i++);
```

S ';' smo zaključili zanko for!

```
for(int i=0; i<tabela.length; i++){
```


- Deklaracija spremenljivke, ki teče, manjka
`for(i=0; i<tabela.length; i++)`
`error: cannot find symbol`

- Deklaracija spremenljivke, ki teče, manjka
`for(i=0; i<tabela.length; i++)`
`error: cannot find symbol`
`for(int i=0; i<tabela.length; i++)`

- Deklaracija spremenljivke, ki teče, manjka

```
for(i=0; i<tabela.length; i++)
```

error: cannot find symbol

```
for(int i=0; i<tabela.length; i++)
```

- Nepravilna uporaba operatorjev

```
for(int i=0; i<tabela.length; i = i++)
```

Zapletemo se v neskončno zanko

- Deklaracija spremenljivke, ki teče, manjka

```
for(i=0; i<tabela.length; i++)
```

error: cannot find symbol

```
for(int i=0; i<tabela.length; i++)
```

- Nepravilna uporaba operatorjev

```
for(int i=0; i<tabela.length; i = i++)
```

Zapletemo se v neskončno zanko

```
for(int i=0; i<tabela.length; i++)
```

- ▶ Deklaracija spremenljivke, ki teče, manjka

```
for(i=0; i<tabela.length; i++)  
error: cannot find symbol  
for(int i=0; i<tabela.length; i++)
```

- ▶ Nepravilna uporaba operatorjev

```
for(int i=0; i<tabela.length; i = i++)  
Zapletemo se v neskončno zanko  
for(int i=0; i<tabela.length; i++)
```

- ▶ Klic polja, ki ni v tabeli

```
int[] spremenljivka = new int[3];  
System.out.println("Izpis "+ spremenljivka[3]);  
java.lang.ArrayIndexOutOfBoundsException
```

- ▶ Deklaracija spremenljivke, ki teče, manjka

```
for(i=0; i<tabela.length; i++)  
error: cannot find symbol  
for(int i=0; i<tabela.length; i++)
```

- ▶ Nepravilna uporaba operatorjev

```
for(int i=0; i<tabela.length; i = i++)  
Zapletemo se v neskončno zanko  
for(int i=0; i<tabela.length; i++)
```

- ▶ Klic polja, ki ni v tabeli

```
int[] spremenljivka = new int[3];  
System.out.println("Izpis "+ spremenljivka[3]);  
java.lang.ArrayIndexOutOfBoundsException  
int[] spremenljivka = new int[3];  
System.out.println("Izpis "+ spremenljivka[2]);
```

- Pri lepljenju nizov manjka '+'

```
String beseda = "Število "+ x "je veliko.";
```

error: ')' expected

- Pri lepljenju nizov manjka '+'

```
String beseda = "Število "+ x "je veliko.";
```

error: ')' expected

```
String beseda = "Število "+ x + "je veliko.";
```


- Pri lepljenju nizov manjka '+'

```
String beseda = "Število "+ x "je veliko.";
```

error: ')' expected

```
String beseda = "Število "+ x + "je veliko.";
```

- Lepljenje nizov in seštevanje celih števil

```
String beseda = "Vsota je "+ x + y;
```

Izpišeta se x in y in ne njuna vsota

- Pri lepljenju nizov manjka '+'

```
String beseda = "Število "+ x "je veliko.";
```

error: ')' expected

```
String beseda = "Število "+ x + "je veliko.";
```

- Lepljenje nizov in seštevanje celih števil

```
String beseda = "Vsota je "+ x + y;
```

Izpišeta se x in y in ne njuna vsota

```
String beseda = "Vsota je "+ (x + y);
```

Večdimenzionalne tabele

- ▶ Vsak element osnovne tabele je spet tabela
- ▶ Pod-tabele imajo isti tip
- ▶ Vsaka pod-tabela ima lahko drugo dolžino
- ▶ Primer:

```
class DvojnaTabela {  
  
    public static void main(String[] args) {  
  
        int[] [] matrika = new int[3] [];  
        matrika[0] = new int[1];  
        matrika[1] = new int[2];  
        matrika[2] = new int[1];  
  
        matrika[0][0] = 9;  
        matrika[1][0] = 8;  
        matrika[1][1] = 7;  
        matrika[2][0] = 6;  
  
        System.out.println(matrika[0][0]);  
        System.out.println(matrika[1][0]);  
        System.out.println(matrika[1][1]);  
        System.out.println(matrika[2][0]);  
    }  
}
```

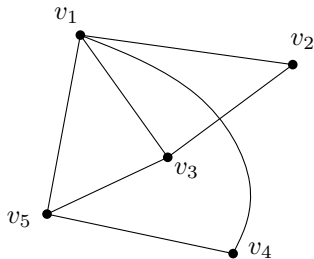
Večdimenzionalne tabele

- ▶ Vsak element osnovne tabele je spet tabela
- ▶ Pod-tabele imajo isti tip
- ▶ Vsaka pod-tabela ima lahko drugo dolžino
- ▶ Primer:

```
class DvojnaTabela {  
    public static void main(String[] args) {  
        int[] [] matrika = new int[] []{{9}, {8, 7}, {6}};  
  
        System.out.println(matrika[0][0]);  
        System.out.println(matrika[1][0]);  
        System.out.println(matrika[1][1]);  
        System.out.println(matrika[2][0]);  
    }  
}
```

Matrika sosednosti

- Naloga 1: S pomočjo dvojnih tabel izpišite matriko sosednosti spodnjega grafa (v matrični obliki).



Matrika sosednosti

```
class MatrikaSosednosti {  
  
    public static void main(String[] args) {  
  
        int[][] matSosed = new int[][]{  
            {0,1,1,1,1},  
            {1,0,1,0,0},  
            {1,1,0,0,1},  
            {1,0,0,0,1},  
            {1,0,1,1,0}  
        };  
  
        System.out.println("Matrika sosednosti grafa je: ");  
        for(int i=0; i<matSosed.length; i++)  
        {  
            for(int j=0; j<matSosed[i].length; j++)  
            {  
                System.out.print(" " + matSosed[i][j]);  
            }  
            System.out.println("");  
        }  
    }  
}
```

Branje iz ukazne vrstice

- Pri zagonu programa lahko podamo dodatne parametre/argumente:

```
java BeriParametre 3 2 1
```

```
class BeriParametre {
```

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

```
        for(int i=0; i<args.length; i++)
```

```
        {
```

```
            System.out.println(args[i]);
```

```
        }
```

```
    }
```

```
}
```

- Ločimo jih s presledkom

- ▶ Argumenti so tipa `String`, v druge tipe jih prevedemo na naslednji način:

1. cela števila:

```
int n = Integer.parseInt(args[0]);
```

ali

```
int n = (Integer.valueOf(args[0])).intValue();
```

2. realna števila:

```
double n = (Double.parseDouble(args[0]));
```

3. znaki:

```
char znak = args[0].charAt(0);
```


- ▶ Naloga 2: Napišite program, ki kot argument prejme naravno število in izpiše njegovo fakulteto.

- ▶ Naloga 2: Napišite program, ki kot argument prejme naravno število in izpiše njegovo fakulteto.

```
class Fakulteta {  
  
    public static void main(String[] args) {  
  
        int n = Integer.parseInt(args[0]);  
        int fakulteta = 1;  
        for(int i=1; i <= n; i++)  
        {  
            fakulteta = fakulteta * i;  
        }  
        System.out.println(n + "! = " + fakulteta);  
    }  
}
```

Krmilni stavki - Pogojni

- ▶ Stavek `if` omogoča, da se koda izvede le pri nekem pogoju.
- ▶ V stavku `else` zapišemo kodo, ki se izvede, če pogoj ni izpolnjen
- ▶ Stavek `else if` uporabimo, ko ima pogoj različne možne vrednosti

```
class IfElse {  
    public static void main(String[] args) {  
  
        int n = Integer.parseInt(args[0]);  
  
        if(n % 2 == 0) {  
            System.out.println(n + " je sodo število!");  
        }  
        else {  
            System.out.println(n + " je liho število!");  
        }  
    }  
}
```

- ▶ Naloga 3: Napišite program, ki kot argument dobi zunanjo temperaturo in glede na njo izpiše, katero izmed oblačil naj oblečem: bundo, jakno, majico z dolgimi rokavi, majico s kratkimi rokavi.

- Naloga 3: Napišite program, ki kot argument dobi zunanjo temperaturo in glede na njo izpiše, katero izmed oblačil naj oblečem: bundo, jakno, majico z dolgimi rokavi, majico s kratkimi rokavi.

```
class Vreme {  
  
    public static void main(String[] args) {  
  
        int n = Integer.parseInt(args[0]);  
  
        if(n < 1)  
        {  
            System.out.println("Obleci bundo!");  
        }  
        else if(n < 10)  
        {  
            System.out.println("Obleci jakno!");  
        }  
        else if(n < 15)  
        {  
            System.out.println("Obleci majico z dolgimi rokavi!");  
        }  
        else  
        {  
            System.out.println("Obleci majico s kratkimi rokavi!");  
        }  
    }  
}
```


- ▶ Stavek `switch` uporabimo podobno kot `if - else if - else`
- ▶ Hitrejši
- ▶ Za prekinitev uporabimo vejitveni stavek `break`
- ▶ Privzeta akcija se izvede v stavku `default`
- ▶ Za vrednost pogoja se sprejmejo le tipi `byte`, `short`, `int` in `char`

Krmilni stavki - Pogojni

```
► class DanVTednu {  
    public static void main(String[] args) {  
        int n = (Integer.valueOf(args[0])).intValue();  
        switch(n)  
        {  
            case 1:  
                System.out.println("Ponedeljek");  
                break;  
            case 2:  
                System.out.println("Torek");  
                break;  
            case 3:  
                System.out.println("Sreda");  
                break;  
            case 4:  
                System.out.println("Četrtek");  
                break;  
            case 5:  
                System.out.println("Petek");  
                break;  
            case 6:  
                System.out.println("Sobota");  
                break;  
            case 7:  
                System.out.println("Nedelja");  
                break;  
            default:  
                System.out.println("Tako dolgega tedna pa nimamo.");  
        }  
    }  
}
```


- ▶ Naloga 5: Z uporabo `switch` stavka za dani argument *zacetnica* tipa `char` izpišite neko ime, ki se začne s črko *zacetnica* oziroma izpišite: "Imena s takšno začetnico nimamo v programu". Vnesite vsaj pet imen.

Koda je v priloženem paketu datotek s programi.

- ▶ Zanke: for, while, do - while
- ▶ Zanko for že poznamo
- ▶ While: (pogoj se preverja na začetku)
`while (pogoj) { stavek }`
- ▶ Do - While: (pogoj se preverja na koncu)
`do {
 stavek
} while (pogoj);`
- ▶ Torej - zanka do - while se vedno izvede vsaj enkrat!

- ▶ Naloga 6: Seštejte števila od 1 do danega argumenta n na tri načine.

Koda je v priloženem paketu datotek s programi.

- ▶ Naloga 7: Poiščite vse delitelje števila podanega kot argument programa.

Koda je v priloženem paketu datotek s programi.

3. domača naloga

- ▶ Napišite program, ki za dani argument n izpiše vsa praštevila manjša od n . Pozor: 1 ni praštevilo!

Uvod v programiranje

4. vaje

Borut Lužar

Fakulteta za informacijske študije Novo mesto

Novo mesto, 27. oktober 2016

- ▶ Spremenljivke
- ▶ Tabele
- ▶ Branje argumentov iz ukazne vrstice
- ▶ Krmilni stavki
 - ▶ Pogojni
 - ▶ Zanke
 - ▶ Vejitveni

Ponovitev in opombe

- ▶ Kaj je *blok*?
- ▶ Pazite na zamike stavkov znotraj blokov!
- ▶ Preverjajte tipične napake!
- ▶ Ko kaj ni jasno, vprašajte!
- ▶ Če po premisleku ne razumete, kaj kakšen delček kode naredi, vprašajte!
- ▶ Kaj so zanke? Koliko jih poznamo?
- ▶ Kako nam pomagajo pogojni stavki?

Ponovitev - naloga 7(3)

- ▶ Naloga 7 (s 3. vaj): Poiščite vse delitelje števila podanega kot argument programa.

Ponovitev - naloga 7(3)

```
class Delitelji {  
    public static void main(String[] args)  
    {  
        int n = Integer.parseInt(args[0]);  
  
        System.out.println("Delitelji števila " + n + ":");  
        int stevec = 1;  
        while (stevec <= n)  
        {  
            if(n % stevec == 0)  
                System.out.println(stevec);  
            stevec++;  
        }  
    }  
}
```

Zanka for nad množico

- ▶ Za izbiro elementov tabele lahko uporabimo drugo sintakso:

```
for (spremenljivka : tabela) {  
    stavek;  
}
```

- ▶ Elementi tabele morajo biti istega tipa kot *spremenljivka*
- ▶ Primer:

```
int[] tabela = new int[]{1,2,3,4,5};  
for(int i : tabela)  
{  
    System.out.println("Element je " + i);  
}
```

- ▶ **Naloga 1:** Kot argument programa preberite najmanj deset celih števil (vsako predstavlja nov argument programa) in jih zapišite v tabelo. Nato uporabite zanko `for` nad množico ter izpišite število sodih števil in število števil, ki delijo število 100.

- ▶ **Naloga 2:** Napišite program, ki kot argumente dobi vsaj 10 naravnih števil in jih zapiše v tabelo A. Nato definirajte tabelo tabel B (dvodimenzionalno tabelo), ki ima toliko pod-tabel kot je argumentov, vsaka izmed pod-tabel pa vsebuje dve celi števili - na prvih mestih so števila, podana kot argumenti, urejena po velikosti, drugo število v pod-tabeli pa predstavlja indeks števila na prvem mestu v tabeli A.

Primer: $A = [1, 5, 4, 2]$

$B = [[5, 1], [4, 2], [2, 3], [1, 0]]$

- ▶ Uporabi se za prekinitev zank oziroma stavka `switch`
- ▶ Izvajanje se nadaljuje po koncu bloka zanke oziroma stavka
- ▶ Dva tipa:
 - ▶ neoznačeni (prekine trenutni blok)
 - ▶ označeni (prekine imenovani blok)

Neoznačeni break

```
class NeoznaceniBreak {  
    public static void main(String[] args) {  
        int[] tabela1 = new int[]{1,2,3,4,5};  
        int[] tabela2 = new int[]{9,8,7,6,5};  
  
        for(int i : tabela1) {  
            for(int j : tabela2) {  
                if(j < 9) {  
                    System.out.println(i + ", " + j);  
                    break;  
                }  
            }  
        }  
    }  
}
```

Označeni break

```
class OznaceniBreak {  
    public static void main(String[] args) {  
        int[] tabela1 = new int[]{1,2,3,4,5};  
        int[] tabela2 = new int[]{9,8,7,6,5};  
  
        zunanja:  
        for(int i : tabela1) {  
            for(int j : tabela2) {  
                if(j < 9) {  
                    System.out.println(i + ", " + j);  
                    break zunanja;  
                }  
            }  
        }  
    }  
}
```


Za predah - Tri funkcije nad nizi (`String`-i)

- ▶ Število znakov v spremenljivki `niz` tipa `String` dobimo s klicem funkcije `length()`:
`int dolzina = niz.length();`
- ▶ Znak na mestu `i` (začenši z 0) v spremenljivki `niz` tipa `String` dobimo s klicem funkcije `charAt(i)`:
`char znak = niz.charAt(i);`
- ▶ Besedo v besedilu (ali podniz v nizu) dobimo s klicem funkcije `substring(zacetniIndeks, dolzina)`, kjer je *zacetniIndex* indeks prvega znaka podniza, *dolzina* pa dolžina podniza.

Primer:

```
String prveTriCrke = besedilo.substring(0,3);
```

- ▶ **Naloga 3:** Napišite program, ki kot argumente prejme vsaj pet besed. Za vsako izmed njih izpišite njihovo dolžino, zadnji znak ter pod-besedo do prve pojavitve črke 'a' oziroma 'A'. Če se v kakšni besedi pojavi črka 'z', prekinite izvajanje programa.

- ▶ Uporabi se za prekinitev trenutnega koraka zank
- ▶ Izvajanje se nadaljuje z novim korakom, če spremenljivka izpolnjuje pogoj
- ▶ Dva tipa:
 - ▶ neoznačeni (preskoči v nov korak izvajanja v trenutnem bloku)
 - ▶ označeni (preskoči v nov korak izvajanja v imenovanem bloku)

Neoznačeni `continue`

```
class NeoznaceniContinue {  
    public static void main(String[] args) {  
        String niz = "raca reže ribi rep";  
        int stRjev = 0;  
        for (int i = 0; i < niz.length(); i++) {  
            if (niz.charAt(i) != 'r')  
                continue;  
            stRjev++;  
        }  
        System.out.println("Našel " + stRjev +  
            " r-je v nizu.");    }  
}
```

Označeni `continue`

```
class OznačeniBreak {  
    public static void main(String[] args) {  
  
        String[] tabelaNizov = new String[]{"riba", "res", "reže", "raci", "rep"};  
        int besedZRji = 0;  
  
        zunanja:  
        for(int i=0; i<tabelaNizov.length; i++)  
        {  
            for (int j = 0; j < tabelaNizov[i].length(); j++) {  
                if (tabelaNizov[i].charAt(j) == 'r')  
                {  
                    besedZRji++;  
                    continue zunanja;  
                }  
            }  
        }  
        System.out.println("Našel " + besedZRji + " besed z \'r\'-ji.");  
    }  
}
```

- ▶ **Naloga 4:** Napišite program, ki v danem besedilu prešteje vse črke 'a' do prve pojavitve črke 'z'. Upoštevajte velike in male črke in uporabite ključni besedi `break` in `continue`!

- ▶ **Primer besedila:**

Ob zmagi klerikalnega političnega tabora se vaško učiteljstvo odpove svojim liberalnim nazorom. Samo učitelj Jerman, v predvolilnem boju indiferenten, se ne pusti podrediti in pride tako v konflikt z materjo in župnikom. Mati, ki jo ljubi, zahteva od njega, naj ne zataji boga, lastno prepričanje pa ga postavlja na stran socialdemokrata, kovača Kalandra. Na agitacijskem večeru ga ljudstvo skoraj linča in župnik ga sklene premestiti na oddaljeno Goličavo. Na predvečer odhoda se prideta od Jermana poslovit vanj zaljubljena Lojzka in nasprotni, pa vendar simpatični Hvastja.

4. domača naloga

- ▶ Napišite program, ki za dano besedilo izpiše vse različne znake, ki se v besedilu pojavijo ter mesta (indekse), na katerih se zadnjikrat pojavijo. Izpiše naj tudi, koliko različnih znakov je v besedilu. Velikih in malih črk naj ne ločuje! Besedilo skopirajte s spleta, vsebuje pa naj vsaj trideset besed.
- ▶ **Opomba:** Črko, zapisano v spremenljivki *crka* tipa `char`, spremenimo v malo (ne glede ali je velika ali mala) s funkcijo `Character.toLowerCase(crka)`.

Uvod v programiranje

5. vaje

Borut Lužar

Fakulteta za informacijske študije Novo mesto

Novo mesto, 22. november 2016

Še dve funkciji nad nizi (`String`-i)

- ▶ Znake v spremenljivki `niz` tipa `String` spremenimo v male oziroma velike s klicem funkcij `toLowerCase()` oziroma `toUpperCase()`:

```
String maliNiz = niz.toLowerCase();
```

```
String velikiNiz = niz.toUpperCase();
```

- ▶ Spremenljivko `crka` tipa `char` spremenimo v malo oziroma veliko s pomočjo statične funkcije razreda `Character`:

```
char malaCrka = Character.toLowerCase(crka);
```

```
char velikaCrka = Character.toUpperCase(crka);
```

Diagram poteka

- ▶ Grafični prikaz algoritma
- ▶ Simboli so naslednji:

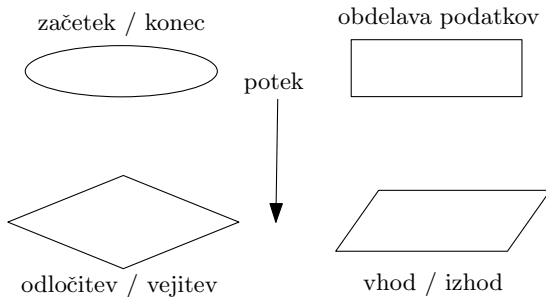
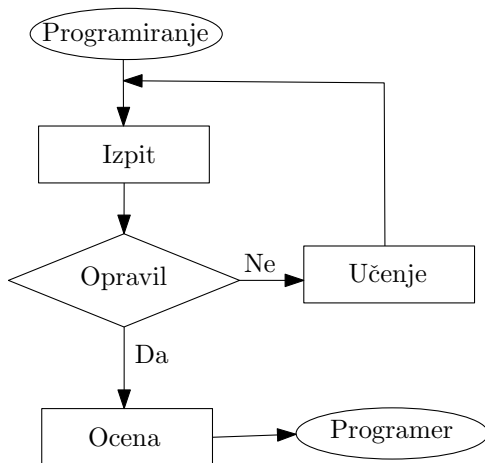


Diagram poteka - primer



Seštej števila od 0 do 11

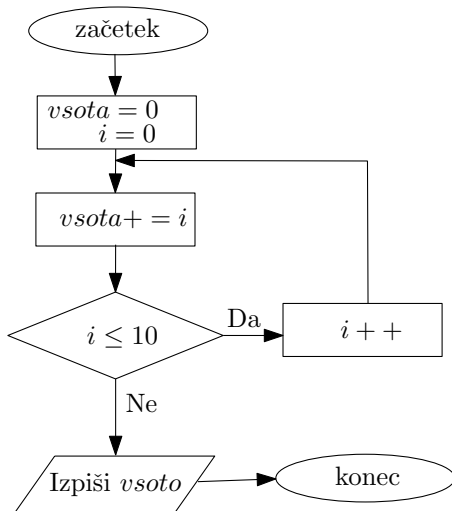


Diagram poteka - naloga

- ▶ **Naloga 1:** Izberite si eno izmed nalog s prejšnjih vaj in narišite diagram poteka zanjo.

- **Naloga 4:** Na bankomatu lahko dvignemo največ 500 evrov v bankovcih po 10 ali 50 evrov. Narišite diagram poteka za postopek dviga določenega zneska. V primeru, ko je znesek previsok, naj bo odgovor: "Znesek je previsok!", če znesek ni deljiv z 10, izpišemo "Znesek ni ustrezen!", sicer pa se izpišeta števili izplačanih bankovcev po 10 in po 50 evrov. Bankomat od uporabnika zahteva vpis zneska, dokler ta ni izplačan. Pozor: števili bankovcev morate izračunati, število bankovcev po 50 evrov pa mora biti največje možno!

Na koncu s pomočjo diagrama poteka napišite še program.

- ▶ Samostojen del programa
- ▶ Dobi množico parametrov (lahko prazno)
- ▶ Vrne vrednost (lahko nič)

- ▶ Sintaksa:

```
vrstaMetode tipRezultata imeMetode(parametri) {  
    teloMetode  
}
```

- ▶ Vrsta metode: npr. `public`, `static`,...
- ▶ Tip rezultata: npr. `int`, `void`,...
- ▶ Vrednost vrnemo z uporabo ukaza `return`;
- ▶ Primer: metoda `main`

Število znakov v nizu

- ▶ **Naloga 2:** Napišite metodo, ki kot parameter prejme niz in znak ter vrne število pojavitev znaka v nizu. Nato metodo kličite v metodi `main`, parametra pa naj bosta argumenta programa.

- ▶ Izjema se zgodi ob nepričakovanem dogodku
- ▶ Npr. program pričakuje celo število, dobi pa črko
- ▶ Izjeme lahko "ujamemo" s stavkom `try-catch`
- ▶ Sintaksa:

```
try {  
    ukaziKiJihPreverjamo  
}  
catch (TipIzjeme imeIzjeme) {  
    ukaziPoUjetjuIzjeme  
}
```

- ▶ Metoda, ki bi lahko vrnila izjemo mora v deklaraciji vsebovati `throws` ali pa imeti "nevarno" kodo v `try-catch`

Izjeme - primer 1

```
public class Izjema1 {  
    public static void main(String[] args) {  
  
        try {  
            int stevilo = (Integer.valueOf(args[0])).intValue();  
            System.out.println("Argument je " + stevilo + ".");  
        }  
        catch(IndexOutOfBoundsException ind) {  
            System.out.println("Pozabil si vnesti argument!");  
        }  
        catch(NumberFormatException form) {  
            System.out.println("Argument ni število!");  
        }  
    }  
}
```

Izjeme - primer 2

```
public class Izjema2 {  
    public static void main(String[] args) throws Exception {  
  
        int stevilo = (Integer.valueOf(args[0])).intValue();  
        System.out.println("Argument je " + stevilo + ".");  
    }  
}
```

Branje s tipkovnice

- ▶ Vnos parametrov med izvajanjem programa
- ▶ Potrebujemo ukaz `import java.io.*;`
- ▶ Prestrezali bomo 'napake' - `throws Exception`
- ▶ Tipkovnica je 'vhodni tok', odpremo ga z ukazom
`InputStreamReader vt = new
InputStreamReader(System.in);`
- ▶ Vrstico preberemo z ukazoma:
`BufferedReader vhod = new BufferedReader(vt);
String vrstica = vhod.readLine();`

Branje s tipkovnice - primer

```
import java.io.*;

public class BranjeSTipkovnice {
    public static void main(String[] args) throws Exception {

        InputStreamReader vt = new InputStreamReader(System.in);
        BufferedReader vhod = new BufferedReader(vt);

        System.out.print("Vnesi poljuben niz: ");
        String niz = vhod.readLine();

        System.out.print("Vnesi celo stevilo: ");
        int celo = Integer.parseInt(vhod.readLine());

        System.out.print("Vnesi realno stevilo: ");
        double realno = Double.parseDouble(vhod.readLine());

        System.out.print("Vnesi znak: ");
        char znak = vhod.readLine().charAt(0);

        System.out.println("\nVnesel si niz '" + niz + "', celo stevilo '" + celo +
            "', realno stevilo '" + realno + "' in znak '" + znak + "'.");
    }
}
```

Število znakov v nizu - 2

- ▶ **Naloga 3:** Spremenite program iz naloge 2 tako, da boste niz in znak prebrali potem, ko bo program že zagnan.

Branje iz datotek

- ▶ Podatki so običajno zapisani v datotekah
- ▶ Poznati moramo pot do datoteke (oziroma samo ime, če je datoteka v isti mapi kot program)
- ▶ Pred branjem jo moramo odpreti:
`FileReader d = new FileReader("imeDat.txt");`
- ▶ Celotno vrstico preberemo z ukazom
`BufferedReader dat = new BufferedReader(d);`
`String vrstica = dat.readLine();`
- ▶ Na koncu moramo datoteko zapreti:
`dat.close();`

Branje iz datotek - primer

```
import java.io.*;

public class PreberiDatoteko {
    public static void main(String[] args) throws Exception {
        String ime = args[0];
        FileReader fr = new FileReader(ime);
        BufferedReader dat = new BufferedReader(fr);

        while(dat.ready()) {
            System.out.println(dat.readLine());
        }
        dat.close();
    }
}
```


Število samoglasnikov

- ▶ **Naloga 4:** V datoteko *Besedilo.txt* shranite nekaj odstavkov poljubnega besedila. Nato napišite program, ki bo z uporabo metode iz naloge 2 preštel vse samoglasnike v besedilu.

Pisanje v datoteke

- ▶ Podatki so običajno zapisani v datotekah
- ▶ Tudi pred pisanjem moramo datoteko odpreti:
`FileWriter d = new FileWriter("imeDat.txt");`
- ▶ Uporabimo razred `PrintWriter` za zapisovanje
`PrintWriter dat = new PrintWriter(d);`
`dat.println("Poljuben niz znakov.");`
- ▶ Na koncu moramo datoteko zapreti:
`dat.close();`

Pisanje v datoteke - primer

```
import java.io.*;

public class ZapisiVDatoteko {
    public static void main(String[] args) throws Exception {
        String ime = "NovaDatoteka.txt";

        FileWriter fw = new FileWriter(ime);
        PrintWriter dat = new PrintWriter(fw);

        dat.println("Prva vrstica v datoteki.");
        dat.println();
        dat.println("Zadnja vrstica v datoteki.");
        dat.close();
    }
}
```

- ▶ **Naloga 5:** Iz datoteke *Besedilo.txt* zamenjajte vse znake 'e' z znakom 'a' in jih zapišite v datoteko *BesediloBrezE.txt*.

5. domača naloga

- ▶ Napišite metodo `obrniNiz`, ki kot parameter prejme niz, vrne pa njegovo obrnjeno vrednost. Primer `obrniNiz("žirafa") = "afariž"`. Nato iz tipkovnice preberite ime datoteke *A*, v kateri imate besedilo ter ime datoteke *B*, v katero boste zapisovali nove podatke. Na koncu zapišite vsako izmed vrstic v datoteki *A* v datoteko *B* z njeno obrnjeno vrednostjo.

Uvod v programiranje

6. vaje

Borut Lužar

Fakulteta za informacijske študije Novo mesto

Novo mesto, 30. november 2016

Objektno orientirano programiranje (OOP)

- ▶ Eden izmed pristopov k programiranju
- ▶ Pri funkcijskem programiranju so enote funkcije, pri objektnem pa razredi (`class`)
- ▶ Modularnost in večkratna uporabnost rešitve

- ▶ Osnovne enote pri izvajanju programov
- ▶ Imajo definirane lastnosti - stanja in funkcionalnosti - metode, s katerimi spreminjajo stanja oziroma objekt
- ▶ Stanja se shranjujejo v **polja**, to so spremenljivke definirane v razredu, vendar izven vseh metod

- ▶ Definicija oziroma načrt objekta je shranjen v razredu (**class**)
- ▶ Posamezen objekt je primerek (**instance**) razreda
- ▶ En razred **človek** lahko vsebuje večino opisov lastnosti in funkcionalnosti človeka, primerkov pa je nekaj milijard...
- ▶ Primer: razred **Clovek** ima npr. stanja teža, starost, barva las ter funkcionalnost shujšaj.

- **Naloga 1:** V razred `Clovek` dodajte polje, ki določa višino človeka. V neprazen konstruktor dodajte določitev te lastnosti. Ustvarite še metodo `zrasti`, ki bo kot parameter prejela količino, za katero se naj spremeni višina danega objekta. Popravite tudi izpis.

- ▶ Vrednosti lastnosti objektov bomo določali z metodami tipa `setLastnost` in pridobivali z metodami tipa `getLastnost`
- ▶ Primer je v datoteki `Avto.java` v paketu nalog!

- ▶ **Naloga 2:** V razred `Avto` dodajte polje, ki določa maksimalno hitrost in dodajte konstruktor, ki jo prejme, v ostalih nastavite privzeto vrednost. Nato popravite metodo `pospesi` tako, da bo pri prekoračenju maksimalne hitrosti hitrost postavila na maksimalno.

- **Naloga 3:** V razred `Avto` dodajte še metodo tipa `get`, ki vrne maksimalno hitrost, popravite izpis in napišite metodo tipa `set`, ki bo maksimalno hitrost avta spremenila.

Vrste spremenljivk

- ▶ **Spremenljivke razreda:**
 - ▶ polja, ki imajo še lastnost `static`
 - ▶ samo ena za razred
 - ▶ če so tudi `final`, se ne morejo spremeniti (so `konstante`)
- ▶ **Spremenljivke objekta:**
 - ▶ polja, ki nimajo lastnosti `static`
 - ▶ vsak objekt ima svojo kopijo
- ▶ **Lokalne spremenljivke:**
 - ▶ spremenljivke definirane znotraj metode
- ▶ **Parametri:**
 - ▶ spremenljivke podane metodi

Naloga 4: Recimo, da imamo tri košare s krompirjem. Krompir je lahko bel ali rdeč. Definirajte razred `Kosara`, ki ima lastnosti `stBelihKrompirjev` in `stRdecihKrompirjev`, dva konstruktorja - praznega in konstruktor, ki kot parameter dobi vrednosti obeh polj, ter metodo `prestaviKrompir`, ki kot parameter dobi celo število `barva` tipa `byte` ter nek objekt `druga` tipa `Kosara`. Če je `barva` enaka 0, se iz košare `druga` prestavi en rdeč krompir v košaro, katere metodo smo klicali, sicer se prestavi bel krompir.

Naloga 5: Ustvarite tri košare, v prvi imejte 5 belih in 0 rdečih krompirjev, v drugi 3 bele in 12 rdečih, v tretji pa 4 bele in 10 rdečih. S pomočjo metode `prestaviKrompir` preložite vse rdeče krompirje iz druge in tretje košare v prvo, nato pa vse bele iz prve in druge v tretjo. Izdelajte še metodo `izpis`, ki izpiše koliko krompirjev je v košari ter jo pokličite z vsako izmed košar.

6. domača naloga

- Za razred `Avto` si izmislite še dve lastnosti avta, zanju naredite polji, dodajte ju v konstruktor, določite privzeti vrednosti in napišite metodi za upravljanje z njima. Popravite tudi izpis.

Uvod v programiranje

7. vaje

Borut Lužar

Fakulteta za informacijske študije Novo mesto

Novo mesto, 7. december 2016

- ▶ Navodila za izdelavo seminarske naloge najdete v spletni učilnici v prvem okvirju.
- ▶ Do naslednjega tedna razmislite ali želite rešiti kakšen svoj problem in se z mano dogovorite za govorilno uro

Glavni elementi razreda so:

- ▶ polja
- ▶ konstruktorji
- ▶ metode (get, set, ostale)

Vrste spremenljivk

- ▶ **Spremenljivke razreda:**
 - ▶ polja, ki imajo še lastnost `static`
 - ▶ samo ena za razred
 - ▶ če so tudi `final`, se ne morejo spremeniti (so `konstante`)
- ▶ **Spremenljivke objekta:**
 - ▶ polja, ki nimajo lastnosti `static`
 - ▶ vsak objekt ima svojo kopijo
- ▶ **Lokalne spremenljivke:**
 - ▶ spremenljivke definirane znotraj metode
- ▶ **Parametri:**
 - ▶ spremenljivke podane metodi

- ▶ Podobnim razredom lahko priredimo istega prednika - nadrazred
- ▶ Kabriolet, limuzina in dostavnik imajo vsi prednika avto
- ▶ V predniku shranimo metode, ki so skupne vsem potomcem
- ▶ Za določitev prednika nekemu razredu uporabimo ključno besedo `extends`
- ▶ Primer: `Avto` in `Kabriolet`

this & super

- ▶ "Znotraj" objekta se nanj sklicujemo z besedo `this`
- ▶ Za sklic na nadrazred uporabimo `super`
- ▶ Primeri v datoteki `Kabriolet.java`

Naloga 1: Razredu **Avto** poleg razreda **Kabriolet** definirajte še dva podrazreda s pripadajočimi lastnostmi in izpisi.

Naloga 2: Ustvarite nadrazred za košaro, koš, sod in čeber. Recite mu *Posoda* in mu določite dve polji, ki sta skupni vsem štirih potomcem. Napišite še vse metode tipov *get* in *set*. Nato popravite izpis pri košari, da bo tidve lastnosti izpisal.

- ▶ Navadno pri shranjevanju objektov v tabelo ne poznamo njihovega števila, zato moramo v nekaterih primerih tabelo povečati
- ▶ Seznami (`ArrayList<>`) so podatkovni tipi, ki imajo podatke tudi urejene po vrsti, vendar se samodejno povečujejo
- ▶ Osnovne funkcije:
 - ▶ `size` - število elementov v seznamu
 - ▶ `add` - dodajanje elementa
 - ▶ `get` - pridobivanje elementa z indeksa
 - ▶ `remove` - odstranjevanje elementa z indeksa
- ▶ Primer: `Seznami.java`
- ▶ Pozor! Med oklepajema `< in >` zapišemo tip elementov seznama. Tip ne sme biti osnovni podatkovni tip. Na primer, namesto `int` moramo uporabiti `Integer`.

Naloga 3: Ustvarite razred `NakupovalniSeznam`, ki kot polje vsebuje seznam `seznam` s stvarmi, ki jih morate kupiti. Napišite prazen konstruktor, ki seznamu določi začetno vrednost (prazen seznam). Nato napišite metodi `dodaj` in `odstrani`, ki znata stvari na seznam dodati oziroma jih izbrisati. V metodi `main` na koncu ustvarite objekt `NakupovalniSeznam`, katerega seznam vsebuje vsaj 5 stvari.

Oglejte si naloge z izpita in jih poskusite rešite kar največ. Izpit najdete v paketu s primeri.

7. domača naloga

- ▶ Ustvarite razred `Zival`, ki ima lastnosti `starost`, `ime`, `mesojeda`. Ustvarite še tri podrazrede (tri vrste živali), ki dedujejo razred `Zival`. Vsaj ena vrsta živali naj ima prvo črko enako prvi črki vašega imena. Podrazredom določite dve polji in metode tipa `get` in `set` zanju. V razredu `Zival` ustvarite metodo `main` in v njej ustvarite pet živali (treh različnih vrst) ter izpišite njihove lastnosti.

Uvod v programiranje (VS) – izpit

Novo mesto, 19. junij 2015

Ime in priimek: _____

Navodila: Čas pisanja izpita je 60 min, uporabljate lahko vso literaturo in zapiske, telefoni in računalniki pa niso dovoljeni.

1. naloga (15%): Kdaj uporabimo stavek return? Kaj se izpiše na zaslonu ob klicu spodnje metode?

```
private static void returnStavek() {  
    System.out.println("Pozdravljeni v metodi");  
    int x = 5;  
    while (x < 9) {  
        x += 2;  
        System.out.println("Trenutni x = " + x);  
        if(x == 8) {  
            return;  
        }  
        else if (x == 9) {  
            System.out.println("Nasvidenje!");  
            return;  
        }  
    }  
}
```

2. naloga (10%): Kakšna je razlika med zankama while in do-while? Zapišite kodo v Javi za množenje prvih 10 naravnih števil s pomočjo zanke do-while.

3. naloga (25%): V kazinoju mečemo igralno kocko, dokler ne vržemo treh šestic zapored. Štejemo število metov. Na koncu se nam število metov izpiše na zaslonu. Narišite diagram poteka za postopek opisan zgoraj.

4. naloga (30%): V jeziku Java napišite dva razreda: razred *Oddaja*, ki bo predstavljal *radijsko oddajo* in razred *DnevniProgram*, ki bo predstavljal *dnevni program* radia. Razred *Oddaja* naj vsebuje:

- polja za shranjevanje *imena voditelja*, *dolžino oddaje*, *uro pričetka* ter *ali je oddaja otroška*,
- konstruktor, ki bo omogočal poljubno določitev začetne vrednosti vseh polj,
- metode za pridobivanje vrednosti vseh polj (torej metode tipa *get*),
- metodo tipa *set*, za nastavitev vrednosti polja z imenom voditelja,
- metodo, ki izpiše vse podatke o oddaji.

Razred *DnevniProgram* naj vsebuje:

- seznam oddaj (kot polje),
- konstruktor, ki omogoča določitev seznama oddaj,
- metodo, ki preveri ali je oddaja otroška,
- metodo, ki izpiše vse oddaje.

5. naloga (20%): V Javi napišite metodo *kvadrat(n)*, ki za dan parameter *n* na zaslon izriše kvadrat, torej znak '*' na vseh štirih robovih, sicer pa znak ' '. Ob klicu

kvadrat(5);

bi se tako na zaslon izpisalo:

```
*****
*      *
*      *
*      *
*      *
*****
```

Uvod v programiranje

8. vaje

Borut Lužar

Fakulteta za informacijske študije Novo mesto

Novo mesto, 8. december 2016

- ▶ Kateri so glavni elementi razreda?
- ▶ Kakšna je razlika med konstruktorjem in metodo?
- ▶ Kaj naredi konstruktor?
- ▶ Katere vrste spremenljivk poznamo?
- ▶ Kaj je nadrazred? Povejte primer.
- ▶ Kaj je `this` in kaj `super`?

- ▶ Vsi razredi so potomci razreda `Object`
- ▶ Tam so definirane nekatere splošne metode, npr. `clone`, `equals` ter `toString`, ki je namenjena izpisu lastnosti objekta
- ▶ Če želimo izpis prilagoditi našim potrebam, moramo metodo “povoziti” in jo napisati ponovno v našem razredu (glej npr. razred `Avto`)

- ▶ Oglejmo si primer programa za zasedenost parkirišča [Parkirisce.java](#)
- ▶ Razredu [Avtomobil](#) smo dodali tabelo potnikov kot lastnost.

Naloga 1: V razredu `Avto` ustvarite metodo `vnesiAvto`, ki vrne objekt tipa `Avto`. Razredu `Parkirisce` nato dodajte konstruktor, ki kot parameter dobi število avtomobilov in od uporabnika zahteva vnos takega števila avtomobilov.

Uporabniški vmesnik

- ▶ Uporabniku omogoča interakcijo s programom
- ▶ Pripravili bomo metode za vpis podatkov, potrebnih za kreiranje objektov
- ▶ Primer `UporabniskiVmesnik`

Naloga 2: V razredu `Parkirisce` povozite metodo `toString`, da bo izpisovala avtomobile, ki so trenutno na parkirišču. Nato v program `UporabniskiVmesnik` dodajte ukaz `izpis`, ki izpiše avtomobile na parkirišču.

Oglejte si naloge z izpita in jih poskusite rešite kar največ. Izpit najdete v paketu s primeri.

8. domača naloga

- ▶ Najprej razredu `Avto` dodajte polje za shranjevanje številke parkirnega prostora. Nato v program `UporabniskiVmesnik` dodajte funkcionalnost za odstranitev avta s parkirišča. Avto odstranite tako, da ga odstranite s seznama `avtomobili` glede na njegovo številko parkirnega prostora.

Uvod v programiranje (VS) – izpit

Novo mesto, 24. avgust 2015

Ime in priimek: _____

Navodila: Čas pisanja izpita je 60 min, uporabljate lahko vso literaturo in zapiske, telefoni in računalniki pa niso dovoljeni.

1. naloga (15%): Opišite uporabo stavka `continue`. Kaj se izpiše na zaslonu ob klicu spodnje metode?

```
private static void continueStavek() {
    System.out.println("Lepo_pozdravljeni_v_metodi");
    int x = 10;
    while (x > 1) {
        x += 2;
        System.out.println("Trenutni_x_=" + x);
        if(x % 4 == 0) {
            System.out.println("Spreminjam!");
            x = x - 6;
            continue;
        }
        System.out.println("Nadaljujem!");
    }
}
```

2. naloga (10%): Kakšna je razlika med spremenljivko (poljem) razreda in spremenljivko (poljem) objekta? Napišite ustrezni deklaraciji za:

- spremenljivko razreda za shranjevanje vrednosti temperature kot realnega števila in
- spremenljivko objekta za shranjevanje dvo-dimenzionalne tabele števil.

3. naloga (25%): V nagradni igri moramo čim hitreje uganiti število, ki si ga je zamislil računalnik. Ob vsakem ugibu računalnik pove, če smo uganili oziroma, če je število manjše ali večje od našega. Ko število uganemo, se na zaslon izpiše število poskusov. Narišite diagram poteka za postopek opisan zgoraj.

4. naloga (30%): V jeziku Java napišite dva razreda: razred `Napitek` in razred `Ponudba`, ki predstavlja ponudbo napitkov gostinskega lokala. Razred `Napitek` naj vsebuje:

- polja za shranjevanje *imena napitka*, *količino*, *ceno* ter *ali je napitek alkoholni*,
- konstruktor, ki bo omogočal poljubno določitev začetne vrednosti vseh polj,
- metodo tipa *set*, za nastavitev vrednosti polja s ceno,
- metode za pridobivanje vrednosti vseh polj (torej metode tipa *get*),
- metodo, ki izpiše vse podatke o napitku.

Razred `Ponudba` naj vsebuje:

- seznam napitkov (kot polje),
- konstruktor, ki omogoča določitev seznama napitkov,
- metodo, ki izpiše vse alkoholne napitke.

V razred `Ponudba` dodajte še metodo `main` in v njej ustvarite eno ponudbo z vsaj dvema napitkoma.

5. naloga (20%): V Javi napišite metodo `pescenaUra(n)`, ki za dano liho naravno število n na zaslon izriše peščeno uro, torej znak `'*'` na zgornjem in spodnjem robu ter obeh diagonalah, sicer pa znak `' '`. Ob klicu

`pescenaUra(7);`

bi se tako na zaslon izpisalo:

```
*****
*   *
* *
*
* *
*   *
*   *
*****
```

Uvod v programiranje

9. vaje

Borut Lužar

Fakulteta za informacijske študije Novo mesto

Novo mesto, 14. december 2016

Podajanje parametrov

- ▶ Razlika, če podamo za parameter metode osnovni podatkovni tipi ali sklicni.
- ▶ Vrednost osnovnega podatkovnega tipa ostane po klicu metode enaka.
- ▶ Vrednost sklicnega se spreminja, če mu ne priredimo novega primerka (glej primer [PodajanjeRazredi.java](#)).

Primerjanje nizov

- ▶ Za primerjanje nizov (`String`-ov) uporabljamo metodo `equals` razreda `Object` in ne operatorja `==`!
- ▶ Prva preveri enakost vsebine dveh nizov, medtem ko operator `==` preverja ali sta referenci enaki (glej primer [EnakostNizov.java](#))
- ▶ Opomba: dve spremenljivki tipa `String` bo prevajalnik razumel kot enaki z operatorjem `==` tudi, če sta definirani s konstantno vrednostjo.

Shranjevanje in branje podatkov

- ▶ Da si program zapomni, katere objekte smo dodajali, jih moramo shraniti
- ▶ Za vsak objekt pripravimo metodo za zapis v datoteko in branje zapisa iz datoteke
- ▶ Primer: [UporabniskiVmesnik.java](#) in odvisni razredi

Naloga 1: Popravite funkcionalnost za vnos avtomobila tako, da bo ob poskusu vnosa avtomobila z registracijo, ki jo že ima nek avto na parkirišču, zavrnila vnos.

Naloga 2: Podobno kot v nalogi 1, popravite še vnose pri branju iz datoteke.

Naloga 3: Dodajte funkcionalnost, ki izpiše vse avtomobile na parkirišču, ki imajo vsaj k potnikov, kjer je k število, ki ga poda uporabnik.

Naloga 4: Popravite dodajanje potnikov v avto tako, da bo v množici potnikov natanko en šofer!

9. domača naloga

- ▶ Ustvarite razred `Avtobus` z lastnostmi `registracija`, `velikost` (dvonadstropni, navadni, mini) in `sofer`. Razredu `Parkirisce` dodajte lastnost `avtobusi` (seznam avtobusov) in zanje dopišite enake funkcionalnosti kot za avtomobile. Shranjevanje avtomobilov in avtobusov na parkirišču naj bo v eno datoteko.

Uvod v programiranje

10. vaje

Borut Lužar

Fakulteta za informacijske študije Novo mesto

Novo mesto, 23. december 2016

Naloga 1

V času pisanja izpita se odvija svetovno nogometno prvenstvo. V jeziku Java napišite dva razreda: razred, ki bo predstavljal *reprezentanco* in razred, ki bo predstavljal *igralca* te reprezentance. Razred `Igralec` naj vsebuje:

- ▶ polja za shranjevanje *imena*, *priimka*, *številke na dresu* ter *ali je igralec v začetni enajsterici*,
- ▶ konstruktor, ki bo omogočal poljubno določitev začetne vrednosti vseh polj,
- ▶ metode za pridobivanje vrednosti vseh polj (torej metode tipa *get*),
- ▶ metodo tipa *set*, za nastavitve vrednosti polja `vZacetniEnajsterici`,
- ▶ metodo, ki izpiše podatke posameznega igralca.

Razred `Reprezentanca` naj vsebuje:

- ▶ *seznam igralcev* (kot polje),
- ▶ konstruktor, ki omogoča določitev seznama igralcev,
- ▶ metodo, ki izpiše začetno enajsterico.

Naloga 2

V jeziku Java napišite dva razreda: razred *Recept*, ki bo predstavljal *recept* in razred *Kuharica*, ki bo predstavljal *knjigo receptov*. Razred *Recept* naj vsebuje:

- ▶ polja za shranjevanje *imena jedi*, *sestavin*, *časa priprave*, *navodil za pripravo* ter *ali je recept za pripravo sladice*,
- ▶ konstruktor, ki bo omogočal poljubno določitev začetnih vrednosti vseh polj,
- ▶ metode za pridobivanje vrednosti vseh polj (torej metode tipa *get*),
- ▶ metodo, ki izpiše recept.

Razred *Kuharica* naj vsebuje:

- ▶ polje za shranjevanje *seznama receptov*,
- ▶ konstruktor, ki omogoča določitev seznama receptov,
- ▶ metodo, ki vrne seznam vseh sladice,
- ▶ metodo, ki izpiše vse recepte, ter
- ▶ metodo *main*, v kateri ustvarite kuharico z vsaj enim receptom.

Naloga 3

V jeziku Java napišite dva razreda: razred *Pisarna*, ki bo predstavljal *pisarno* in razred *PoslovnaStavba*, ki bo predstavljal *poslovno stavbo*, v kateri so pisarne. Razred *Pisarna* naj vsebuje:

- ▶ polja za shranjevanje *delovne površine* pisarne, *seznama opreme*, in *števila delovnih prostorov*,
- ▶ konstruktor, ki bo omogočal poljubno določitev začetnih vrednosti vseh polj,
- ▶ metode za pridobivanje vrednosti vseh polj (torej metode tipa *get*),
- ▶ metodo, ki izpiše vse lastnosti pisarne.

Razred *PoslovnaStavba* naj vsebuje:

- ▶ polje za shranjevanje *seznama pisarn*,
- ▶ konstruktor, ki omogoča določitev seznama pisarn,
- ▶ metodo, ki izpiše lastnosti vseh pisarn, ter
- ▶ metodo *main*, v kateri ustvarite poslovno stavbo z vsaj dvema pisarnama.

Naloga 4

V Javi napišite metodo `kvadrat(n)`, ki za dan parameter n na zaslon izriše kvadrat, torej znak `*` na vseh štirih robovih, sicer pa znak `' '`. Ob klicu

`kvadrat(5);`

bi se tako na zaslon izpisalo:

```
*****  
*      *  
*      *  
*      *  
*      *  
*****
```


Naloga 5

Ugotovite, kaj izpiše spodnji program ob klicu **java naloga 9**.

```
public class naloga
{
    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        skrivnost(n);
    }

    public static void skrivnost(int n) {
        for(int i=0; i<n; i++) {
            for(int j=0; j<n; j++) {
                String znak = " ";
                if (j==0 || j==n-1 || j==i || i==n-j-1
                    || (j > n/2 && j > i && i > n-j-1)
                    || (j < n/2 && j < i && i < n-j-1))
                    znak = "*";
                System.out.print(znak);
            }
            System.out.println();
        }
    }
}
```

Igramo priljubljeno igro *vislice*. Nasprotnik si zamisli besedo, mi pa ugibamo črke, ki se pojavijo v besedi. Če smo uganili črko, ki se pojavi v besedi, se le-ta zapiše v tabelo znakov, ki določajo besedo, sicer izgubimo eno "življenje". Ko izgubimo šest življenj se igra konča z izpisom "Izgubili ste!", če pa prej uganemo vse črke besede, se igra konča z izpisom "Uganili ste!". Narišite diagram poteka za postopek opisan zgoraj.

Prijetne praznike in srečno 2017!

Uvod v programiranje

11. vaje

Borut Lužar

Fakulteta za informacijske študije Novo mesto

Novo mesto, 6. januar 2017

Grafični uporabniški vmesnik (GUI)

- ▶ Okna, menuji, gumbi, oznake,...
- ▶ Uporabljali bomo knjižnico `javax.swing`
- ▶ Primer: `PozdravljenSwing.java`

Grafični uporabniški vmesnik (GUI)

- ▶ Vsak uporabniški element je objekt
- ▶ Poznamo vidne in nevidne elemente
- ▶ Vidni: okno, gumb, slika, menu,...
- ▶ Nevidni: dogodki (events), razporejevalci (layout managers),...

- ▶ `JFrame`: samostojno okno, ki vsebuje ostale elemente
- ▶ `JButton`: gumb, ko ga pritisnemo se zgodi dogodek, na katerega program lahko reagira
- ▶ `JLabel`: prikazuje besedilo ali sliko, npr. ime okna
- ▶ `JTextArea`: področje za prikazovanje in urejanje besedila
- ▶ `JFileChooser`: element za izbiranje datotek
- ▶ `JPanel`: področje, ki vsebuje druge elemente

- ▶ **ActionEvent**: objekt, ki se ustvari, ko pride do dogodka, npr. pritisk na gumb
- ▶ **MouseEvent**: objekt, ki se ustvari, ko premaknemo miško ali pritisnemo na gumb
- ▶ **BorderLayout**: objekt, ki skrbi za razporeditev elementov znotraj okna (preprosta verzija)
- ▶ **GridBagLayout**: zmogljivejši razporejevalnik elementov v oknu
- ▶ Še več razporejevalnikov najdete [tukaj](#)

- ▶ Oglejmo si datoteki `NaseOkno.java` in `Sporocilo.java`

- ▶ **Naloga 1:** V oknu `NaseOkno` sta gumba lepo razporejena na objekt `JPanel1`. Takšen objekt ustvarite tudi v razredu `Sporocilo` in gumb `Zapri` premaknite nanj.

- ▶ **Naloga 2:** Razredu `NaseOkno` dodajte tretji gumb `Seštej`. Ob kliku nanj naj se odpre novo okno tipa `Sporocilo`. Sporočilo naj bo vsota števil od 1 do n , pri čemer n zapišete v besedilno polje okna `NaseOkno`. Uporabite `try/catch` za lovljenje, če uporabnik ne napiše celega števila.

Datoteke:

- ▶ `GUV.java` (grafični uporabniški vmesnik)
- ▶ `Parkirisce.java`
- ▶ `Avto.java`
- ▶ `Potnik.java`
- ▶ `GlavnoOkno.java`
- ▶ `VnesiAvtoOkno.java`

- ▶ **Naloga 3:** Primeru **Parkirisce** dodajte možnost dodajanja potnikov avtu s pomočjo uporabniškega vmesnika.

Namig: Ustvarite novo okno, podobno oknu **VnesiAvtoOkno**, ki se odpre, ko kliknete na nov gumb **Dodaj potnika**.

Opomba: Preverite, da se po spremembah prevedejo vse datoteke programa!

11. domača naloga

- ▶ V razred **UporabniskiVmesnik** dodajte funkcionalnosti (gumba) za shranjevanje in branje avtomobilov iz datotek. Uporabite funkcije, ki so že narejene, ustvariti morate le dve dodatni okni za določanje imen datotek, kamor se podatki zapisujejo oziroma, od koder se berejo.

Uvod v programiranje

12. vaje

Borut Lužar

Fakulteta za informacijske študije Novo mesto

Novo mesto, 10. januar 2017

- ▶ Element uporabljamo za izbiranje datotek
- ▶ Bolj podrobno elementa ne bomo obravnavali
- ▶ Natančen opis in primere uporabe najdete [tukaj](#)
- ▶ Enostaven primer je v datoteki [IzbiraDatoteke.java](#)

Seznami različnih tipov objektov

- ▶ Različne objekte lahko shranimo v skupni seznam
- ▶ Tip elementov seznama mora biti skupni nadrazred vseh elementov
- ▶ Za preverjanje tipa objekta uporabimo ključno besedo `instanceof`
- ▶ Primer: [SeznamiObjektov.java](#)

Seznami različnih tipov objektov

- ▶ Ob klicu metode `toString` na objektu iz seznama, se kliče metoda njegovega razreda
- ▶ Primer: sklop programov `Parkirisce`

Serializacija objektov iz podrazredov

- **Naloga 1:** V razreda `Kabrio` in `Limuzina` dodajte metode za branje in zapisovanje objektov teh tipov v datoteko.

- ▶ Javanske programe lahko zapakiramo v `jar` datoteko (Java ARchive)
- ▶ Poleg naših razredov vsebuje tudi 'navodila': datoteko `META-INF/MANIFEST.MF`, kjer lahko povemo kateri razred naj se izvede, če hočemo da je naša datoteka izvedljiva z ukazom `java -jar jardatoteka.jar`
- ▶ Vsebina datoteke `MANIFEST.MF` je npr.:
`Main-Class: UporabniskiVmesnik`

Naredimo `jar` datoteko

- ▶ Ustvarimo datoteko `manifest.txt` in vanjo zapišemo ime glavnega razreda (npr. `Main-Class: UporabniskiVmesnik`) ter dodamo prazno vrstico (enter!)
- ▶ V ukazni vrstici izvedemo ukaz
`jar cmf manifest.txt mojProgram.jar *.class`
- ▶ Na koncu smo povedali, naj se datoteke s končnico `class` zapakirajo v `jar`

Pripravite svoj paket

- ▶ **Naloga 2:** Izdelajte `jar` datoteko iz datotek, ki jih potrebujete za program `Parkirisce`.

- ▶ Rešujemo izpit : [Izpit-2013-03-vs](#)

12. domača naloga

- ▶ Rešite nalogi 4 in 5 z VS izpita 4. septembra 2013 (Izpit-2013-03-vs)

Ime in priimek: _____

Navodila: Čas pisanja izpita je 60 min, uporabljate lahko vso literaturo in zapiske, telefoni in računalniki pa niso dovoljeni.

1. naloga (10%): Na računalniku najdemo datoteko z imenom `Neznan.java`. Začetek njene vsebine zgleda takole:

```
public class Neznan {  
  
    private static String uporabnik;  
  
    public static void main(String[] args) {  
        uporabnik = args[0];  
        System.out.println("Dobrodosel_" + uporabnik + "!");  
        ...  
    }  
}
```

Kaj lahko na osnovi tega sklepamo o najdeni datoteki in njeni vsebini?

2. naloga (20%): Kaj izpiše spodnji program? Odgovor primerno utemeljite!

```
public class Neznan2 {  
    private static final int d = 5;  
    public static void main(String[] args) {  
        int[][] a = new int[d][d];  
        int[][] b = new int[d][d];  
        for (int i = 0; i < a.length; i++) {  
            for (int j = 0; j < a[i].length; j++) {  
                a[i][j] = (i == j) ? i+1 : 0;  
            }  
        }  
        for (int i = 0; i < a.length; i++) {  
            for (int j = 0; j < a[i].length; j++) {  
                b[i][j] = a[a.length-i-1][j];  
            }  
        }  
        for (int i = 0; i < b.length; i++) {  
            for (int j = 0; j < b[i].length; j++) {  
                System.out.print(b[i][j]);  
                if (j < b[i].length - 1) {  
                    System.out.print(",");  
                } else {  
                    System.out.println();  
                }  
            }  
        }  
    }  
}
```

3. naloga (20%): Narišite diagram poteka za algoritem igre, kjer mora igralec uganiti (vnaprej določeno) skrivno (celo) število. Po vsakemu poskusu algoritem igralcu pove ali je njegovo število večje, manjše ali enako skrivnemu številu. Igra se konča, ko igralec pravilno ugaane skrivno število, ali ko izkoristi največje število dovoljenih poskusov N .

4. naloga (25%): V jeziku Java napišite dva razreda: razred, ki bo predstavljal *mesečno evi-denco meteoroloških meritev*, in razred, ki bo predstavljal *meteorološke meritve enega dneva*. Razred `Dan` naj vsebuje:

- polji (ali polja) za shranjevanje *temperature* in *zračnega tlaka zjutraj, sredi dneva in zvečer*,
- konstruktor, ki bo omogočal poljubno določitev začetne vrednosti vseh polj,
- metode za dostop do vseh polj (t.i. *get* in *set* metode).

Razred `Mesec` naj vsebuje:

- polje (tabelo) za shranjevanje *vseh dni v mesecu*,
- metodo, ki bo vrnila število dni v mesecu, ko je temperatura dosegla $30\text{ }^{\circ}\text{C}$ ali več.

5. naloga (25%): Napišite metodo v Javi, ki bo izpisala tabelo poštevanke za množenje pozitivnih celih števil manjših ali enakih podanemu številu `int n`. Klic metode `postevanka(4)` bi tako na zaslonu izpisal:

```

    | | 1 | 2 | 3 | 4 |
=====
1 | | 1 | 2 | 3 | 4 |
-----
2 | | 2 | 4 | 6 | 8 |
-----
3 | | 3 | 6 | 9 | 12 |
-----
4 | | 4 | 8 | 12 | 16 |
-----

```

Uvod v programiranje

13. vaje

Borut Lužar

Fakulteta za informacijske študije Novo mesto

Novo mesto, 12. januar 2016

- ▶ Na prosojnicah imate naloge, s pomočjo katerih bomo še enkrat pregledali snov, ki smo se jo na letošnjih vajah naučili.
- ▶ Pri reševanju si pomagajte z letošnjimi prosojnicami s predavanj in vaj ter s svojimi zapiski.
- ▶ Vsak naj poskusi rešiti kar največ nalog sam.
- ▶ Če naloge ne znate rešiti, se lotite naslednje.
- ▶ Pri nalogah, pri katerih morate napisati program, preverite, če so rezultati pravilni.

Ploščina trikotnika

Naloga 1: Napišite metodo `ploscinaTrikotnika`, ki kot parametre prejme dolžine vseh treh stranic trikotnika in vrne njegovo ploščino.

Osnovni podatkovni tipi

Naloga 2: Naštejte vseh osem osnovnih podatkovnih tipov v javi. Za vsakega izmed njih definirajte spremenljivko, ji določite vrednost in jo izpišite.

Naloga 3: Definirajte tabelo realnih števil, katere dimenzijo pridobite kot argument ob zagonu programa. Uporabnika za vsako izmed njenih polj vprašajte, katero vrednost želi vnesti. Na koncu izračunajte polovico povprečne vrednosti števil v tabeli in jo izpišite.

Naloga 4: Napišite diagram poteka za nalogo 3.

Naloga 5: Napišite program, ki uporabnika vpraša po imenu datoteke, nato pa izpiše število vrstic v datoteki ter število vseh znakov.

Naloga 6: Napišite razred `Lik`, ki ima lastnosti `ploscina` in `obseg` ter napišite metode tipa `get` in `set`. Povežite tudi metodo `toString`. Ustvarite še podrazreda `Trikotnik` in `Kvadrat`. Vsakemu dodajte potrebne lastnosti in konstruktor, da boste lahko ustvarili dva objekta teh tipov, za katera boste znali izračunati ploščino.

Naloga 7: Za nalogo 3 pripravite `jar` paket in pokličite program.

Naloga 8: Uporabite funkcijo `Math.random` za generiranje naključnih števil (med 0 in 1). Zgenerirajte 100 števil in vsako, ki je večje od 0.48, zapišite v seznam `velika`. Nato izračunajte povprečje števil v seznamu.

Naloga 9: Ustvarite dvodimenzionalno $n \times n$ tabelo celih števil, pri čemer n poda uporabnik. V vsako polje zapišite vsoto $(i + i * j)$, kjer je i številka vrstice, j pa številka stolpca. Napišite metodo `transponiraj`, ki kot parameter dobi tabelo in vrne njeno transponiranko. Obe matriki tudi izpišite!

12. domača naloga

- ▶ Rešite naloge, ki jih niste uspeli rešiti na vajah.