

Opakovanie



- Rôzne druhy cyklov
 - `do-while` cyklus
 - `for` cyklus
 - príkazy `break` a `continue`
- Vnorené príkazy
- „Verifikácia“
 - Kontrolné podmienky
 - Invarianty a varianty cyklov

6.5 Ďalšie riadiace štruktúry



- `break`- a `continue` s návěstím v Java

- `switch`- aj v Java aj v C (Výber z viacerých možností)

Príkaz break s návěstím v Java

```
a:
while (bed1)
{
    while (bed2)
    { ...
        ... break;    // ukončí vnitorný cyklus
        ... break a; // ukončí vnitorný aj vonkajší
                      // cyklus
    }
    ... break;    // ukončí vonkajší cyklus
}
```

continue
analogicky!

Príkaz goto s návěstím v C

```
while (bed1)
{
    while (bed2)
    { ...
        ... break;    // ukončí vnútorný cyklus
        ... goto a;   // ukončí vnútorný aj vonkajší
                       // cyklus (skočí na návěstie a:)
    }
    ... break;       // ukončí vonkajší cyklus
}
a:
```

continue
analogicky!

Príkaz switch



```
// Vstup int cislo;
String den = "";

switch (cislo)
{ case 1: den = "Pondelok"; break;
  case 2: den = "Utorok"; break;
  case 3: den = "Streda"; break;
  case 4: den = "Stvrtok"; break;
  case 5: den = "Piatok"; break;
  case 6:
  case 7: den = "Vikend"; break;
  default: System.out.println("Nie je v tyzdni");
}
```

II. Základné elementy



7. Dátové štruktúry / Polia (Arrays)

V programovacích jazykoch:

Postupnosť viacerých elementov (alebo objektov) rovnakého typu.

Príkazy



- Parametre príkazového riadku: `args`
`args[0]`, `args[1]`, `args[2]`, ...
- Lineárna algebra
 - Vektor `vektor`
`vektor[0]`, `vektor[1]`, `vektor[2]`
 - Matica `matrix`
`matrix[i][j]`

Príklady (pokrač.)

□ Všeobecne: Indexované premenné:

x_i `x[i]`

```
double sum = 0.0;
for ( int i = 1; i <= n; i++)
{
    sum = sum + x[i];
}
```

$$\sum_{i=1}^n x_i$$

7.1 Polia: Deklarácia v Java

Príklady:

```
String[] args;           // Pole znakových reťazcov
String args[];          // to isté
double[] x, y;          // dve polia typu
                        // double (Vektory)
double[][] matrix;     // Jedno dvojdimenzionálne
                        // pole typu double (Matica)
int[][] k;              // Dvojdimenyionálne pole typu
                        // int
double[] vektor;       // dalsi Vektor
```

7.1 Polia: Deklarácia v Java

Príklady:

```
String[] args;          // Pole znakových reťazcov
String args[];         // to isté
double[] x, y;         // dve polia typu
                        // double (Vektory)
double[][] matrix;    // Jedno dvojdimenzionálne
                        // pole typu double (Matica)
int[][] k;            // Dvojdimenyonálne pole typu
                        // int
double[] vektor;      // dalsi Vektor
```

Princíp (Java)

Typ:
double

Pole typu
double

Meno poľa

```
double[] x, y;
```

Meno poľa

```
double[][] matrix;
```

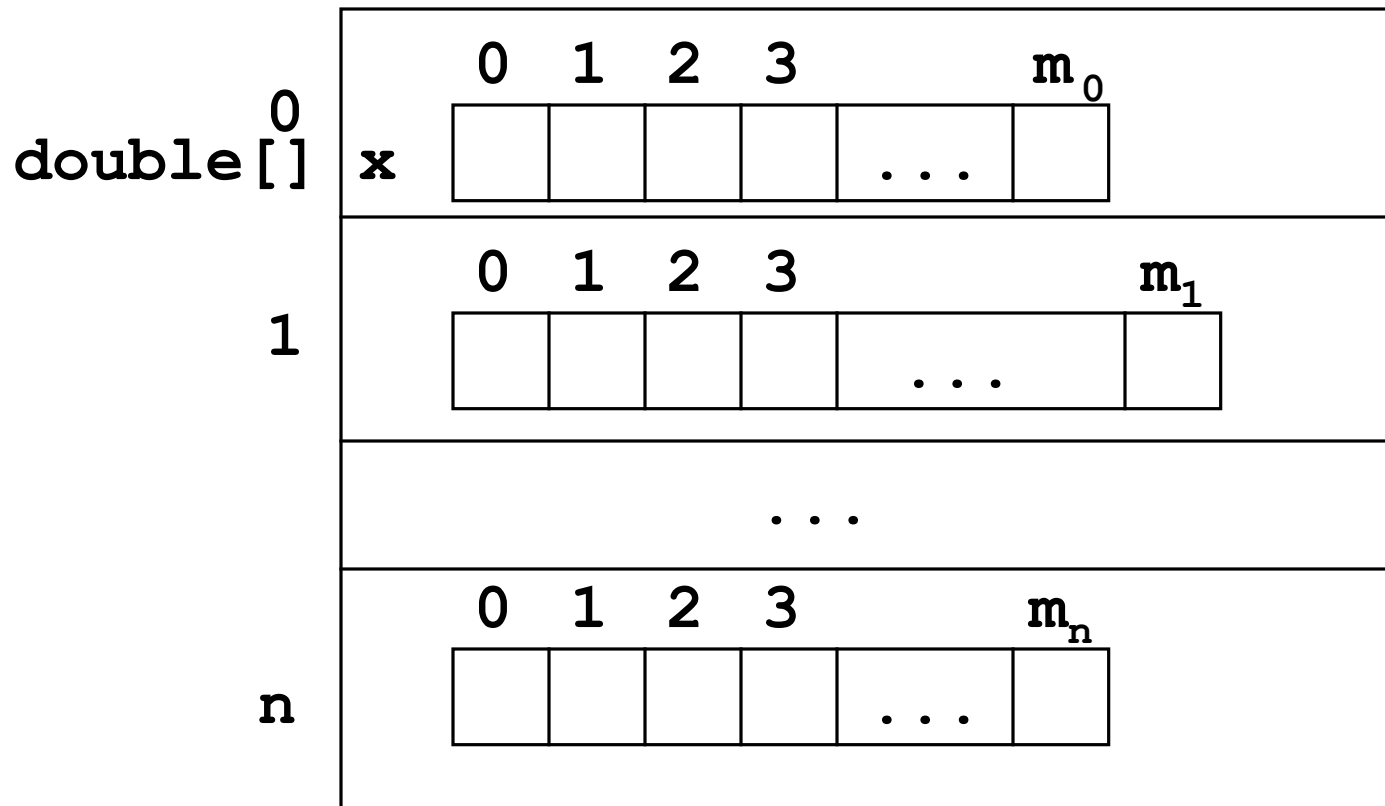
Typ:
double

Pole typu
double

Pole typu Pole
typu double

Viacdimenzionálne polia (Java)

`double[][] matrix`



7.2 Polia: Priradenie (Java)

Príklady:

```
String[] mesiace = new String[]  
    { "Jan", "Feb", "Mrc", "Apr", "Maj", "Jun",  
      "Jul", "Aug", "Sep", "Okt", "Nov", "Dec" };
```

	0	1	2	3		11
mesiace	"Jan"	"Feb"	"Mrc"	"Apr"	...	"Dec"

Príklady (Java) (pokrač.)

```
int[][] k = new int[][]  
    { { 1, 2, 3 }, {4, 5, 6}, {7, 8, 9, 0} };
```

k	0	0	1	2	
		1	2	3	
	1	0	1	2	
		4	5	6	
2	0	1	2	3	
		7	8	9	0

Príklady (Java) (pokrač.)



```
double[] x = new double[100];
```

Pole so 100 `double` prvkami, pričom každý prvok má počiatočnú hodnotu `0.0`; prvky sú inicializované implicitne

Pozor:

Prvky majú index od 0 po 99!

Príklady (Java) (pokrač.)

```
double[][] matrix = new double[50][100];
```

Dvojdimenziálne pole (Matica) s `double` prvkami; pole pozostáva z 50 –tich polí, každé po 100 prvkov typu `double`, pričom každý prvok má hodnotu `0.0`;

Pozor:

```
matrix[0][0], ..., matrix[49][99]
```


Príklady (Java) (pokrač.)

```
double[][] matrix;
```

```
...
```

```
matrix = new double[50][100];
```

```
...
```

```
matrix = new double[][]
```

```
    {{0.0, 0.7}, {0.25, 0.0}, {0.0} };
```

Premenným typu polí môže byť priradená hocikedy hodnota zodpovedajúceho typu, t.j. musí zodpovedať počet a typ dimenzií

Indexy v poliach začínajú 0!



- V Jave začína indexovanie prvkov poľa pevne a to indexom 0
- Prvky sú vždy oindexované zasebou idúcimi číslami

Implicitná inicializácia (Java)

- S pomocou `new` vytvoríme nové pole; ak hodnoty poľa nie sú pri inicializácii explicitne uvedené, ale je uvedený iba počet prvkov, nadobúdajú prvky nasledovné implicitné hodnoty
 - `0` (resp. `0.0`) pri všetkých číselných typoch
 - `false` pri type `boolean`
 - `'\u0000'` pri type `char`
 - `null` inak (bude neskôr vysvetlené)

7.3 Prístup k prvkom poľa

Prvok poľa sa správa presne ako premenná daného typu

- `matrix[0][1]` má typ `double`
- `matrix[0]` má typ `double[]`
- `vektor` má typ `double[]`
- `vektor[7]` má typ `double`

Prístup k prvkom poľa

- Čítanie hodnoty vo výrazoch:

```
double erg = matrix[3][4] * vektor[2];
```

- Priradenie hodnoty prvku poľa:

```
matrix[3][4] = vektor[2] + erg;
```

```
matrix[2] = vektor; //!!! Neskôr viac
```

Počítanie s indexom

- Index môže byť ľubovoľný výraz typu `int` (t.j. nielen konštanty ako doposiaľ)

Príklad:

```
double vysledok = 0.0;
for (int i = 0; i < vektor.length; i++)
{ vysledok = vysledok + vektor[i] * vektor[i]; }
System.out.println("l= " + Math.sqrt(vysledok));
```