

PB162 Programování v jazyce Java

6. cvičenie

2. a 3. úloha

- 2. úloha
 - bude opravená čoskoro – snád' v priebehu víkendu :-)
- 3. úloha
 - deadline sa posúva o týždeň

dnešné cvičenie

- písomka na konvencie
- primitívne a objektové typy
- cyklus for
- pole
 - zásobník implementovaný poľom

písomka na konvencie

primitívne a objektové typy

- primitívne typy

- `int, long; float, double; boolean`

- do metód predávané **hodnotou**

- objektové typy

- „vstavane“ a užívateľské triedy

- `String`

- `Integer, Long, Float, Double, Boolean`

- do metód predávané **odkazom**

- v metóde sa môže zmeniť obsah meniteľných objektov

cyklus for

- pevný počet opakovaní
- nejbežnější a nejprehlednější cyklus

```
int factorial = 1;  
for (int i = 2; i <= n; i++) {  
    factorial *= i;  
}
```

pole

- lineárna homogénna dátová štruktúra
 - ? lineárna
 - ? homogénna
- pravdepodobne najjednoduchšia
- programátorov pohľad – „očíslované“ premenné

pole

- pamäťový pohľad – súvislý blok pamäte
 - adresa prvku = báza + $n \cdot$ veľkosť prvku
- predpoklad
 - všetky prvky majú rovnakú veľkosť (homogénnosť)
- dôsledok
 - polia číslujeme od nuly

pole – syntax

- deklarácia

```
int [] elements;
```

- inicializácia

```
elements = new int [20];
```

- optimálne

```
int [] elements = new int [20];
```

```
int [] elements = new int [size];
```

- prístup k prvku

```
elements[4] = 5;
```

```
int x = elements[4];
```

prechádzanie poľa

- cyklus for

```
int sum = 0;
for (int i = 0; i < elements.length; i++) {
    sum += elements[i];
}
```

- alebo elegantnejšie – for-each

```
int sum = 0;
for (int i : elements) {
    sum += elements[i];
}
```

veľkosť poľa

- veľkosť prvku
 - primitívny typ (`int`, `double`, `boolean`...)
 - veľkosť prvku = veľkosť typu
 - objektový typ
 - v poli len odkaz na haldu (pár bytov)

- ak veľkosť poľa nevyhovuje

```
elements =
```

```
Arrays.copyOf(elements, newLength) ;
```

- lineárna zložitosť => zväčšovať aspoň na dvojnásobok

príklad zásobník

- **interface** `Stack`
 - homogénna dátová štruktúra
 - princíp **last in – first out**
 - operácie `push`, `pop`, `isEmpty`, (`peek`)
- **implementácia** `ArrayStack`
 - zásobník reprezentovaný poľom
 - ukazovateľ vrcholu zásobníka