

Programozási nyelvek Java

Kozsik Tamás előadása alapján

Készítette: Nagy Krisztián

4. előadás

Tömbök

```
void reverseCopy(int[] from, int[] to){
    for(int i = 0; i < from.length; ++i){
        to[from.length-i-1] = from[i];
    }
}
```

/** dokumentációs megjegyzés */ - javadoc kipróbálható

/** Copies the elements of array <code>from</code> into
<code>to</code> in reverse order */

--> reverseCopy metódus definíciója

```
id[] t = null;
t.length - futási hiba! → kivétel (exception): NullPointerException

int[] r = new int[0]; helyes
```

/** Precondition: <code> from != null & to != null &
from.length == to.length <code> & <code> from != to <code> */

```
t = new int[6];
reverseCopy(t,t);
```

Ezért nem szabad, hogy a from és a to tömb ugyan azon tömb legyen.

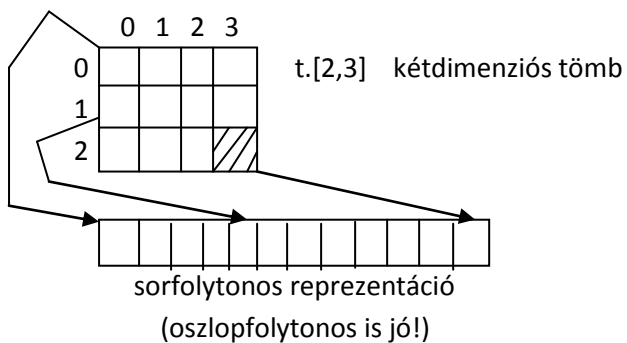
utasítás végrehajtása lépésenként:

1 2 3 4 5 6

1 2 3 4 5 1

1 2 3 4 2 1

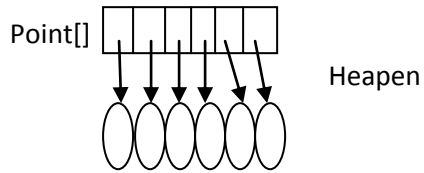
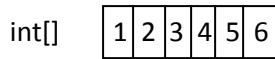
1 2 3 3 2 1



(i,j) i+1. sor j+1. elem

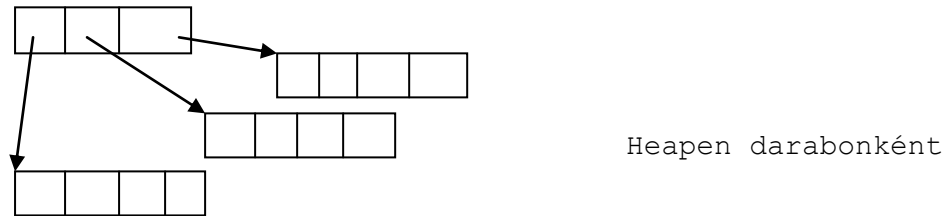
$t + i * m * I + j * I \sim t[in + j]$ $n \times m$ I méret

N dimenziós tömbben $O(N)$ azaz $k_1n + k_2$ aritmetikai művelet kell (kevés)



```
Point[] ps = new Point[5];  
for(int i = 0; i < ps.length; ++i){  
    ps[i] = new Point();  
}
```

`int[][] = new int[3][4]` - tömbök tömbje
`m[i][j]` - hivatkozás az $i+1$. sor $j+1$. elemére egy mátrixban

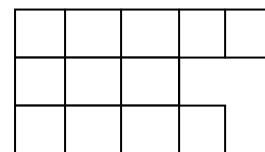


```
int[][] k = new int[3][];  
↓  


|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|--|--|--|--|


```

```
k[0] = new int[5];  
k[1] = new int[3];  
k[2] = new int[4];
```



Objektum létrehozása, osztálydefiníció

Eddig használt:

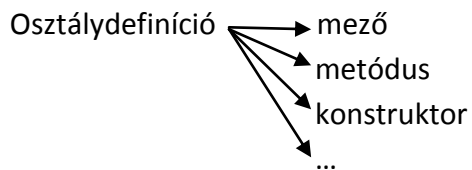
```
Point p = new Point();
```



```
Point p = new Point(3,2);
```



```
class Point{  
    int x,y; —————> mintha int x = 0,y = 0; autómátikusan 0-ra inicializálódik  
  
    Point(int x, int y) {—————> konstruktor  
        this.x = x;  
        this.y = y;  
    }  
    void move(int dx, int dy) {  
        x += dx;  
        y += dy;  
    }  
}
```



Konvenció:

- metódus / mező kisbetűvel kezdődik
- osztálynév/ konstruktor nagybetűvel

```
void Point(int x, int y) {  
    this.x = x;  
    this.y = y;  
}
```

Legális, de ebben az esetben metódus - `new Point(3,2)` nem működne

Konstruktor: objektum létrehozása során, inicializáló

idióma: mezőnév és konstruktorparaméter megegyezik (this)

x név: lehet több jelentése is

- mezőnév
- formális paraméter neve

Hatóköri (scope) „szabályok”

Deklaráció: Amikor nevet adunk egy entitásnak (metódus, osztály, csomag, változó)

Amire figyelni kell:

A közelebbi deklaráció elfedi a távolabbit (formális paraméter elfedte a mezőt például)

Ekkor minősítéssel esetleg elérhető.

Például: `this.x` - minősített név

`{int x = 2; {int x = 1; } }` – fordítási hiba (más nyelvben lehet megengedett, de Javában nem)

Hatóköri deklarációt közvetlenül tartalmazó blokk, deklarációtól a blokk végéig beleértve a beágyazódó blokkok.

Láthatóság: hatókör - elfedés