

Objektovo orientované programovanie

Učiteľ:
Ing. Jozef Wagner PhD.

Učebnica:
<https://oop.wagjo.com/>

OPG

Teória 8

1. Proces vytvárania objektu
2. Inicializácia objektu
3. Konštruktory
4. Statická inicializácia triedy

Proces vytvárania objektu

Nový objekt sa vytvára pomocou operátora **new**

Vytváranie objektu sa skladá z dvoch častí:

1. Alokácia pamäti pre objekt
2. Inicializácia objektu

Príklad:

```
Rat potkan = new Rat();
```

Inicializácia objektu

Vykoná sa po alokácii pamäti

Cieľom je nastaviť atribúty objektu na správne hodnoty.

3 časti inicializácie, vykonávajú sa v poradí

1. Priama inicializácia atribútov
2. Inicializačný blok
3. Konštruktor

Ak nejaký atribút neinicializujeme, inicializuje sa automaticky na `0`, `null` alebo `false`

Priama inicializácia atribútov

```
public class Rat {  
    private int health = 100; // priama inicializácia
```

```
    public int getHealth() {  
        return health;  
    }  
}
```

```
// použitie  
Rat potkan = new Rat();  
assert potkan.getHealth() == 100 : "Mlady potkan ma mat  
zdravie 100";
```

Inicializačný blok

```
public class Rat {  
    private int health;  
    private double damage;  
  
    // inštančný inicializátor  
    {  
        health = 100;  
        damage = 0.5;  
    }  
  
    public int getHealth() {  
        return health;  
    }  
}
```

Konštruktor

```
public class Rat {  
    private int health;  
    private double damage;  
  
    // konštruktor  
    public Rat() {  
        health = 100;  
        damage = 0.5;  
    }  
  
    public int getHealth() {  
        return health;  
    }  
}
```

Konštruktor

Konštruktor je špeciálna metóda, ktorá **nemá návratový typ a má rovnaký názov ako jeho trieda**.

- *defaultný konštruktor* - nemá žiadne argumenty
- *kopírovací konštruktor* - má jeden argument, typu svojej triedy
- *parametrizovaný konštruktor* - má jeden alebo viac argumentov

Ak nemáme napísaný žiaden konštruktor, Java automaticky vygeneruje defaultný konštruktor.

Konštruktor

```
public Rat() { // defaultný konštruktor
    health = 100;
    damage = 0.5;
}
```

```
public Rat(Rat other) { // kopírovací konštruktor
    health = other.health;
    damage = other.damage;
}
```

```
public Rat(double damage) { // parametrizovaný konštruktor
    health = 100;
    this.damage = damage;
}
```

Modifikátor prístupu konštruktora

Konštruktor môže byť privátny, vtedy voláme továrenské metódy

```
private Rat(double damage) {  
    health = 100;  
    this.damage = damage;  
}
```

```
// továrenské metódy  
public static Rat newRat() {  
    return new Rat(0.5);  
}
```

```
public static Rat newEliteRat() {  
    return new Rat(2);  
}
```

Statická inicializácia triedy

Statické atribúty sa inicializujú iba jeden krát a to pri prvom použití triedy.

Statická inicializácia má 2 kroky:

- Priama inicializácia statických atribútov
- Statický inicializačný blok

```
public class Rat {  
    public static final double NORMAL_DAMAGE;  
    public static final double ELITE_DAMAGE;  
  
    static { // Statický inicializačný blok  
        NORMAL_DAMAGE = 0.5;  
        ELITE_DAMAGE = 2;  
    }  
}
```