

# Objektovo orientované programovanie

Učiteľ:  
**Ing. Jozef Wagner PhD.**

Učebnica:  
<https://oop.wagjo.com/>

# OPG

## Teória 7

1. Statické a inštančné metódy
2. Getter a Setter metódy
3. Preťaženie metód
4. Továrenské metódy

# Statické metódy

Nepatria objektu ale triede ako takej

Volajú sa cez názov triedy, napríklad `Arrays.sort()`

Využitie: utility, factory metódy a vstupný bod programu

Príklady:

```
Math.random()
```

```
Math.max()
```

```
Arrays.fill()
```

```
Integer.toString()
```

```
Objects.requireNonNull()
```

# Inštančné metódy

Pracujú s objektom, volanie `obj.metoda()`

Majú prístup k všetkým atribútom svojej triedy

Majú v sebe definovanú špeciálnu premennú `this`, pomocou ktorej prístupujeme k objektu

```
public int obvod() {  
    return 2 * (this.a + this.b);  
}
```

```
Obdlznik o = new Obdlznik(10, 15);  
int obvodObdlznika = o.obvod();
```

# Špeciálne inštančné metódy

`toString()` - sa zavolá, keď sa objekt prevádza na reťazec

```
public String toString() {  
    return String.format("%dx%d", a, b);  
}
```

`equals()` - porovnanie dvoch objektov podľa hodnoty

```
public boolean equals(Obdlznik other) {  
    return this.a == other.a && this.b == other.b;  
}
```

# Preťaženie metód

Môžeme mať viacero metód s rovnakým menom

Preťažené metódy sa **musia vzájomne líšiť počtom argumentov alebo typmi argumentov**

Definícia viacerých metód v rovnakým názvom sa nazýva **preťaženie**, anglicky **overloading**

Na návratový typ sa neprihliada

# Preťaženie metód

```
class Kalkulacka {  
    public static int add(int a, int b) {  
        return a + b;  
    }  
    public static double add(double a, double b) {  
        return a + b;  
    }  
    public static int add(int a, int b, int c) {  
        return a + b + c;  
    }  
}
```

# Getter a Setter metódy

**Getter metódy** slúžia na zistenie hodnoty nejakého atribútu v objekte

**Setter metódy** slúžia na zmenu hodnoty v objekte

```
public class Obdlznik {  
    private int a;  
    private int b;  
  
    public void setA(int a) {  
        if (a <= 0) {  
            throw new IllegalArgumentException("...");  
        }  
        this.a = a;  
    }  
}
```



# Továrenské metódy

Anglicky factory methods

Sú to statické metódy, ktoré **vracajú inštanciu svojej triedy**

Továrenské metódy oddeľujú proces vytvárania objektov od kódu, ktorý ich používa

```
public static Obdlznik createNaSirku(int a, int b) {  
    if (a > b)  
        new Obdlznik(a, b);  
    else  
        new Obdlznik(b, a);  
}
```

# Továrenské metody

Príklady továrenských metód v knižniciach Java:

- `LocalDate.now()` vytvorí nový objekt s aktuálnym dátumom
- `URI.create()` vytvorí URI identifikátor
- `Integer.valueOf()` vytvorí číslo z reťazca znakov