

# Téma 4: Atributy tříd

**Třída** je obecný popis vlastností a činností objektů. Například budeme chtít pracovat se studenty, vytvoříme třídu Student, která bude obecně popisovat všechny studenty, jejich vlastnosti (např. jméno, příjmení, datum narození, adresu, absolvované kurzy, známky atd.) a činnosti (vytvoření nového studenta, zapsání předmětu, změna adresy atd.). Třída (class) je obecný popis vlastností a činností objektů. Například budeme chtít pracovat se studenty, vytvoříme třídu Student, která bude obecně popisovat všechny studenty, jejich vlastnosti (např. jméno, příjmení, datum narození, adresu, absolvované kurzy, známky atd.) a činnosti (vytvoření nového studenta, zapsání předmětu, změna adresy atd.).

**Statické proměnné** (proměnné třídy) na rozdíl od nich existují pouze jedenkrát pro celou třídu, a to i v případě, že neexistují žádné její instance. Nepopisují vlastnost objektů třídy, ale třídy jako takové.

Příkladem pro třídu Osoba může být počet osob uchovávaných v programu nebo jejich seznam.

```
public static int prom = 5;
```

**Statické metody** (metody třídy) mohou přistupovat pouze ke statickým proměnným a lze je volat i v případě, že neexistuje žádný objekt dané třídy.

```
Public static void main(String[] args){
    ..
}
```

V paměti je proměnná pouze jednou.

**Instanční proměnná** je proměnná jako jakákoliv jiná. Znamená, že každý objekt dané třídy bude mít určité vlastnosti (instanční proměnné). Můžeme k nim přistupovat pomocí metod v rámci dané třídy nebo zpřístupnění metodou get.

```
public class Osoba {
    private int vek;
    private String jmeno;
    public Osoba (int vek, String jmeno){
        ...
    }
}
```

**Instanční metoda** se vztahuje se ke konkrétní instanci (a jejím datům), volání přímo na daném objektu, nikoliv na třídě.

```
public String predstavSe() {
    return "Ja jsem: " + this.jmeno + " a muj vek je: " + this.vek;
}
```

**Modifikátory přístupu** jsou public, private, protected a package friendly. Public je přístup z libovolné třídy, private pouze z dané třídy, protected z kterékoliv třídy anebo z potomka a package friendly pouze z kterékoliv třídy z balíku.