

Téma 3: Objekty

Třída je vzor, podle kterého se objekty vytváří. Definuje jejich vlastnosti a schopnosti. Objekt, který se vytvoří podle třídy, se nazývá instance. **Instance** mají stejné rozhraní jako třída, podle které se vytváří, ale navzájem se liší svými daty (atributy). Komunikace mezi objekty probíhá pomocí předávání zpráv, díky čemuž je syntaxe přehledná. Zpráva obvykle vypadá takto: **příjemce.jménoMetody(parametry)**.

Mějme **například** třídu Student a od ní si vytvořme instance Chlubna a Veselý. Obě jistě ty samé metody a atributy, jako třída (např. jméno a věk), ale hodnoty v nich se liší (první instance má v atributu jméno hodnotu „Chlubna“ a věk 18, druhá „Veselý“ a věk 7).

Pro **deklaraci třídy** používáme klíčové slovo `class`, před kterým je specifikátor přístupu, a za kterým následuje název třídy. Samotné tělo třídy je uzavřeno ve složených závorkách.

```
public [final / abstract] class jmeno {  
    proměnné třídy;  
    proměnné instancí;  
    vnořené třídy;  
    konstruktory;  
    metody;  
}
```

Klíčové slovo final označuje třídu, od níž již nelze definovat potomky. **Klíčovým slovem abstract** označujeme třídu, od které nelze vytvářet žádné instance a která se používá jako předek pro další třídy (lze zde nadefinovat, co vše mají tyto třídy společné). Jedině abstraktní třída může obsahovat abstraktní metody. Je možná pouze kombinace `public abstract`.

Název třídy by měl začínat velkým písmenem. Název by měl být výstižný a popisující. Pokud se rozhodneme použít specifikátor přístupu `public`, tak jméno souboru `*.java` musí být totožné s názvem třídy (daný soubor může obsahovat pouze jednu veřejnou

Konstruktor je speciální funkce, která se jmenuje stejně jako třída, ve které se nachází. Při deklaraci nepoužíváme návratový typ ani klíčové slovo `void`. Slouží k inicializaci objektu a jeho obsahu a je metodou, která se zavolá automaticky při vytváření každého objektu. Dokonce, i když nespecifikujeme konstruktor tak ho daná třída obsahuje – Java do každé třídy, která neobsahuje žádný konstruktor, přidává implicitní konstruktor, který nemá žádné parametry.

Garbage collector je vlastně program, který běží paralelně s naší aplikací, v samostatném vlákně. Občas se spustí a podívá se, na které objekty již v paměti nevedou žádné reference. Ty potom odstraní.

```
public class Student {  
    protected String jmeno;  
    protected int vek;  
  
    public Student(String jmeno, int vek) {  
        this.jmeno = jmeno;  
        this.vek = vek;  
    }  
}
```

O garbage collection můžeme sice také zažádat pomocí volání `System.gc()`, ale není zaručeno, že nám virtuální stroj vyhoví a skutečně kolekci zavolá, případně že zničí i náš objekt (garbage collection je velmi složitý proces). Z tohoto důvodu se volání kolekce vesměs necháváme na virtuálním stroji, aby se sám rozhodl, kdy je nejvhodnější okamžik.

Podobně jako existuje konstruktor, tak v Javě existuje **metoda `finalize()`**, kterou mají všechny objekty, a jež je volána jako poslední pomazání před zničením objektu. Tuto metodu můžeme ve svých objektech překrýt (znovu deklarovat a implementovat), ale její použití je vhodné pouze pro velmi specifický kód, jelikož nevíme, kdy přesně dojde k jejímu zavolání.