

## 03. Objekty (třída, instance, konstruktor, garbage collector)

### OOP

Object-oriented programming – objektově orientované programování.

### Třída

Obecný předpis objektu (šablona pro instance). Obsahuje formální proměnné a metody pro instance → Určuje, jak bude objekt vypadat a jak se bude chovat.

Skupina s podobnými vlastnostmi, která nese informace o instancích. Data mohou být buď proměnné primitivního datového typu, nebo odkazy na další objekty.

Program obsahuje vždy alespoň jednu třídu označenou klíčovým slovem **class** a ohraničenou složenými závorkami { }.

### Instance

Konkrétní objekt (instance; klon třídy), který je vytvořen podle vzoru třídy se skutečnými hodnotami. Termíny objekt a instance se často zaměňují.

### Konstruktor

- Složí k inicializaci proměnných objektu.
- Žádný návratový typ se neuvádí, i když konstruktor vždy vrátí vytvořený objekt.
- Metoda, která vrací referenci na objekt, může mít libovolný počet parametrů různých typů.
- Má vždy stejné jméno, jako je jméno třídy.
- Každá třída, kde není uveden konstruktor, má svůj vlastní bezparametrický (init()). Pokud je napsán jiný, tento překrývá.

Pomocí tříd se vytvářejí jejich instance, které mají stejné metody, ale jiné vlastnosti.

### Vytvoření objektu

Třída Auto může mít vlastnosti barva, motor a metodu pohyb(). Instancí této třídy může být MojeAuto s vlastnostmi: barva – Červená, motor – 2.0 TDI 125kW.

Zavolá se konstruktor s 2 parametry:

```
Auto MojeAuto = new Auto("Červená", "2.0 TDI 125kW");
```

Připravená proměnná MojeAuto nyní v sobě uchovává referenci na skutečný objekt, který je pomocí klíčového slova **new** vytvořen v paměti, s parametry uvedenými v závorce.

```
Scanner sc = new Scanner(System.in);
```

## Datové složky

Třídy i objekty si uchovávají své informace v datových složkách (proměnné a konstanty). Datové složky tříd a instancí se zapisují v těle třídy, nikoliv v těle metod, pokud by se deklarovali v těle metody, jednalo by se o lokální proměnné. Datové složky třídy jsou jedinečné pro třídu a pro její objekty, ovšem datové složky objektů jsou jedinečné pro každý objekt zvlášť.

Datové složky třídy jsou statické datové složky, a jak je z názvu zřejmé jsou uvedeny klíčovým slovem `static`. Tyto složky patří třídě a jsou jedinečné a nezávislé na objektech.

Datové složky objektů (instancí) jsou bez klíčového slova `static`. Při vytvoření každého objektu se pro něj vytvoří zvláštní datové složky, které uchovávají stav objektu.

V oop platí: co není nutné, aby bylo vidět, vidět být nemusí → *private*. Přístup k nim je potom řešen pomocí **getrů** a **setrů**.

## Garbage collector (čistič paměti)

Je součástí Java Virtual Machine a stará se o odstraňování nepotřebných objektů (nepotřebný objekt pozná podle toho, že na něj už neukazuje žádná reference). Lze ho vyvolat pomocí příkazu:

```
System.gc();
```

## Finalizér

Metoda objektu, která se spustí, když je objekt rušen garbage collectorem.