

13. JAVA – Objektové programování

Třídy, metody (návrátové datové typy, argumenty, parametry), konstruktory, přetěžování metod a konstruktorů, klíčové slovo static a this, variabilní počet argumentů.

Třídy

- definuje data a metody
- slouží jako šablona pro vytváření instancí
- data mohou být buď:
 - proměnné primitivního datového typu nebo
 - odkazy na další objekty
- instancemi tříd jsou objekty

Samotná třída tedy nepředstavuje vlastní informace, jedná se pouze o **předlohu, data obsahují až objekty**.

Objektů jedné třídy přitom může existovat **libovolné množství** (samozřejmě také žádný).

Metody

Návratový typ metody může být **primitivní datový typ** (int, bool, float,...), **reference** (na objekt) nebo **void** (žádná hodnota není vrácena).

Za návratovým typem a názvem metody je v kulatých závorkách napsán seznam parametrů (a jejich typů) oddělených čárkami.

```
void nazevMetody(int prvniParametr, bool druhyParametr){...};
```

Argument je už samotná instance, dosazená do parametru.

Výsledek metody (návrátová hodnota) je uveden klíčovým slovem **return**.

Konstruktory

Speciálním typem metody je tzv. **konstruktor**. Tato metoda nemá jméno (jmenuje se stejně jako třída) a obsahuje **inicializaci** objektu.

Konstruktor je volán **automaticky** při vytváření objektu. Pokud při deklaraci konstruktor nepopíšete, konstruktor je automaticky **prázdný**.

Přetěžování metod a konstruktorů

Konstruktorů jednoho objektu, nebo **metod se stejným jménem** v jednom objektu **může být i více**, musí se však **lišit v parametrech**, aby Java poznala, **který konstruktor nebo metodu** chcete **zavolat**.

Této vlastnosti se říká **polymorfismus**.

Klíčové slovo **static** a **this**

Klíčové slovo **static** můžeme přidat před datový typ proměnné nebo metody.

```
static int statickaPromenna = 6;
```

Takto označené proměnné a metody jsou **přístupitelné i bez vytvoření instance**; za pomoci názvu třídy.

```
NazevTridy.statickaPromenna = 12;
```

Statická proměnná má "**stálou**" **hodnotu**, ve všech instancích. Pokud ji někde změníme, změní se ve všech ostatních instancích.

Statická metoda **nedokáže pracovat s ne-statickými** proměnnými. Může přistoupit ke **statickým proměnným** třídy, nebo použít **parametry** pro získání jiných hodnot.

Klíčové slovo **this** může být použito k přístupu k proměnným a metodám třídy. Například v konstruktoru, když se názvy parametrů shodují s názvy proměnných ve třídě.

```
class Třída{
    int x = 10;

    Třída(int x){
        this.x = x;
    }
}
```

Variabilní počet argumentů

Variabilního počtu argumentů (0 nebo více) lze dosáhnout napsáním **třech teček**(...) za datový typ parametru.

```
void nazevMetody(int ... variabilni)
```

Taková proměnná může být napsána jako poslední.

Takhle **NE**:

```
void nazevMetody(int ... variabilni, int dalsiPromenna)
```

Zdroj

<https://www.interval.cz/clanky/naucte-se-javu-tridy-a-objekty-1/>