

06. Metody tříd (deklarace, přetěžování)

Metoda; Podprogram

Část kódu konající nějakou specializovanou funkci, která je zpravidla použita vícekrát. Metody jsou umístěny mimo hlavní program, mohou být spouštěny z `main()` nebo z jakékoliv jiné metody. Díky tomu, že se na tuto operaci vytvoří metoda, nemusí se pokaždé opakovat tentýž kus kódu, ale stačí pouze zavolat tuto metodu.

Metody v programu vystupují svojí signaturou (jméno a seznam parametrů), dle signatury se v programu metody rozeznávají. V třídě nesmí být dvě metody se stejnou signaturou.

Skládání programu

Všechny netriviální programy v Javě se skládají z tříd, třídy se dále skládají z metod a datových složek (proměnných). Podprogramům se v Javě říká metody, v jazyce C/PHP jsou to zase funkce.

Statické metody

Statické metody jsou metody, které patří třídě. Pokud se má statická metoda zavolat, nemusí se vytvářet instance:

```
Modifikátor návratový_typ jmeno(seznam_parametrů) {}
```

Deklarace metod

Modifikátor přístupu: Existují 3 modifikátory přístupu (`Public`, `Private`, `Protected`). Modifikátory přístupu jsou jedna z hlavních vlastností samotného objektově orientovaného programování.

Návratový typ: typ hodnoty, kterou vrátí podprogram volajícímu. Metody mohou vracet všechny datové typy a navíc i typ `void` (prázdnost) → metoda nic nevrací. Taková metoda se například využívá k výpisům na obrazovku.

Jméno: libovolný název metody. Název metody by měl začínat malým písmenem a vystihovat chování metody. Špatné pojmenování má za následek zneprůhlednění kódu. Dále se doporučuje metody pojmenovávat slovesem, které lépe zdůrazní, co daná metoda provádí.

Seznam parametrů: seznam hodnot, které metoda dostane od svého volajícího.

Metoda může převzít libovolný počet parametrů, dokonce to nemusí být ani určeno, stejně tak, ale nemusí převzít žádné parametry, pak se uvedou prázdné kulaté závorky.

Parametry metody: deklarují se v kulaté závorce za jménem metody, uvede se nejdříve typ proměnné a poté její jméno. Pokud metoda přebírá více hodnot, mezi proměnnými se udělá čárka.

Klíčové slovo `return` zařizuje vrácení hodnoty, metoda vždy může vracet jen jednu proměnnou a to stejného datového typu jako je uveden v deklaraci metody. Pokud metoda nic nevrací, `return` se neuvádí. Pokud ovšem metodu návratového typu `void` a chcete ji ukončit předčasně lze použít **`return`**;

```
public class SoucetMetodou {  
    /** metoda převezme dvě hodnoty a vrátí jejich součet */  
    public static int secti(int a, int b) {  
        return a + b;  
    }  
    /**  
    public static int sectiNeomez(int... a){  
        int vysl = 0;  
        for(int b : a) {  
            vysl += b;  
        }  
        return vysl;  
    }  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 5;  
        /** volání metody, metoda vrátí celé číslo */  
        System.out.println(secti(a, b));  
        System.out.println(sectiNeomez(5, 6, 8, 12));  
    }  
}  
  
public class VypisMetodou {  
    /** metoda nic nepřevzme, a vypíše text na obrazovku */  
    public static void pis() {  
        System.out.println("Tento text vypíše metoda," + " může se  
        volat kolikrát chcete");  
    }  
    public static void main(String[] args) {  
        /** volání metody, metoda nic nevrací */  
        pis();  
    }  
}
```

Přetěžování metody

Pokud je zapotřebí sčítat dvě čísla, tak mohou být buď celá, nebo reálná pak je vhodné využít přetěžování metod.

- Přetížená metoda má stejné jméno, ovšem různý seznam parametrů.
- Pokud se program spustí, vyhledá se metoda, která svým seznamem parametrů odpovídá a ta se zavolá.
- Nedoporučuje se používat přetěžování metod, ovšem pokud potřebujete zavolat metodu pro různé parametry je to nejlepší možný způsob.
- Nejvíce přetěžovaná metoda je **toString()**.

```
public class PretizeneMetody {  
    /** metoda převezme dvě celá čísla a vypíše součet */  
    public static void secti(int a, int b) {  
        System.out.println("Toto vypsala metoda celých čísel\n" +  
            "součet je: " + (a + b));  
    }  
  
    /** metoda převezme dvě reálná čísla a vypíše součet */  
    public static void secti(double a, double b) {  
        System.out.println("Toto vypsala metoda reálných čísel\n" +  
            "součet je: " + (a + b));  
    }  
  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 5;  
        double c = 10.5;  
        double d = 20.5;  
        /** volání metod */  
        secti(a, b);  
        secti(c, d);  
    }  
}
```