

Jazyk Java 1

Seminář 3

Daniel Bazala



Katedra informatiky
Univerzita Palackého v Olomouci

Úvod do OOP

Úvod do OOP

- programovací paradigma, které vychází z podobnosti programovaného problému s reálným světem
- důraz na rozdělení problému do menších znovupoužitelných částí (objektů)
- podobné objekty sdílí strukturu, ale liší se ve vlastnostech (struktura a metody definovány třídou objektu)
- třída jako šablona pro vznik objektů (objekt je instancí své třídy)
- abstrakce od vnitřního fungování objektů (objekt jako černá skříňka)
- polymorfismus jako nástroj pro specifické chování objektů (univerzální použitelnost metod, ale konkrétní odlišná implementace u různých tříd)

Úvod do OOP, příklady

- program pro automobilku
 - rozdělení problému do tříd - třída auto, motor, kola, ...
 - kompozice objektů (objekt auto obsahuje jiný objekt, motor)
 - konfigurace různých modelů aut (např. jiný motor)
 - principy nicméně zůstávají stejné (auto slouží k ježdění, motor má určitý výkon)
- informační systém (např. STAG)
 - třída student, vyučující, předmět, místnost, ...
 - určité třídy mají stejné vlastnosti (student i vyučující má věk, bydliště, jméno, příjmení, ...)
 - dědičnost (student i vyučující jsou lidé - rodičovská třída člověk/osoba)
 - abstrakce (metoda věk mi vrátí stáří v letech, ale uvnitř probíhá výpočet z data narození)

OOP v Javě

OOP v Javě

- každá veřejná třída ve samostatném souboru
- třída obsahuje konstruktor (i více konstruktorů), implicitní konstruktor prázdný
- klíčové slovo `this` - přístup objektu k sobě samému

```
■ public class Student {  
    String jmeno;  
    String prijmeni;  
    String stav = "neznámý";  
  
    Student(String jmeno, String prijmeni, String stav) {  
        this.jmeno = jmeno;  
        this.prijmeni = prijmeni;  
        this.stav = stav;  
    }  
  
    Student(String jmeno, String prijmeni) {  
        this.jmeno = jmeno;  
        this.prijmeni = prijmeni;  
    }  
}
```

OOP v Javě

- vytváření objektů jako instancí své třídy

```
Student student1 = new Student("Prokop", "Dveře");  
Student student2 = new Student("Tomáš", "Putna", "svobodný");
```

- přístup k atributům

```
System.out.println(student1.jmeno);  
student1.prijmeni = "Okno";
```

- definice metod

```
public class Student {  
    deklarace atributů...  
    konstruktory...  
    metody...  
  
    void print() {  
        System.out.println(jmeno + "□" + prijmeni);  
    }  
}
```

- volání metod

```
student1.print();
```

- speciální hodnota `null` - ukazuje na prázdné místo v paměti

Řetězce

- referenční, objektový typ

- v Javě neměnné

```
■ String slovo1 = "slovo";  
  String slovo2 = "slovo";  
  String slovo3 = new String("slovo");  
  System.out.println(slovo1 == slovo2); // true  
  System.out.println(slovo1 == slovo3); // false  
  System.out.println(slovo1.equals(slovo3)); // true
```

```
■ String jedna = "1";  
  String slovo1 = "slovo1";  
  String slovo2 = "slovo" + jedna;  
  System.out.println(slovo1 == slovo2); // false  
  System.out.println(slovo1.equals(slovo2)); // true
```

- metoda `length` - délka řetězce

```
String palindrome = "Dot_saw_I_was_Tod";  
int len = palindrome.length() // len <- 17
```

- metoda `charAt(int i)` - znak na i-té pozici

```
String slovo1 = "slovo1";  
System.out.println(slovo1.charAt(1)) // 1
```

- pro efektivní zřetězení třída `StringBuilder`

```
StringBuilder sb = new StringBuilder();  
for (int i = 0; i < 100; i++) {  
    sb.append("i=");  
    sb.append(i);  
    sb.append(System.lineSeparator());  
}  
System.out.println(sb);
```

- Více: <https://docs.oracle.com/javase/tutorial/java/data/strings.html>

Pole

Pole

- referenční, objektový typ
- konstantní velikost, okamžitý přístup k prvkům
- definice

```
int[] cisla = {1, 2, 3};
String[] slova = new String[3];
slova[0] = "slovo1";
System.out.println(slova[0]); // slovo1
System.out.println(slova[1]); // null

int[] cisla = {1, 2, 3};
int[] cisla2 = {1, 2, 3};
System.out.println(cisla); // [I@5a01ccaa
System.out.println(cisla == cisla2); // false
```

- třída `java.util.Arrays`

```
import java.util.Arrays;
int[] cisla = {1, 2, 3};
int[] cisla2 = {1, 2, 3};
System.out.println(Arrays.toString(cisla)); // [1, 2, 3]
System.out.println(Arrays.equals(cisla, cisla2)); // true
```

- Více:

<https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/util/Arrays.html>

Úkol

Úkol 3

- Rejstřík osob
- Třídy `Country(String name, String phoneCode)`, `Person(int id, String firstname, String lastname, int age, String phone, Country country)`, `Index(Person[] persons, Country[] countries)`
- Metody vyhledávání v rejstříku, tisknutí na výstup, tel. číslo s předvolbou
- <http://marcus.weibly3d.net/ukol3>