Řetězce

Table of Contents

`String`

`String`

- Řetězce, tj. posloupnosti znaků jsou v Javě reprezentovány objekty vestavěné třídy `java.lang.String`.
- Má to řadu dobrých důvodů — tytéž jednou definované řetězce lze používat souběžně z více míst programu, nehrozí modifikace.
- Nicméně i negativní stránky — někdy větší režie spojená s nemodifikovatelností:

Kolik se vytvoří objektů?

```
String s = "Hello " + "World";
```

- Kód vytvoří 3 objekty: `"Hello "`, `"World"` a `"Hello World"`.
- Cože? To je tak neefektivní?
- Pokud by vadilo, lze místo `String` použít `StringBuilder`, který je modifikovatelný (mutable), viz dále

String pod lupou

- Podívejme se na rozdíl `"Hello"` a `new String("Hello")`.

`new String("Hello")`
: vytvoří pokaždé nový objekt

`"Hello"`
: funguje na principu: *jestli takový objekt zatím neexistuje, tak ho vytvořím, jinak vrátím odkaz na již existující objekt* (uložen v paměti *String constant pool*)

Rovnost řetězců

Identické

```
String s1 = "Hello";
String s2 = "Hello";
boolean bTrue = (s1 == s2); // true, identical objects
```

Neidentické, ačkoli stejné znaky

```
s1 = new String("Hello");
s2 = new String("Hello");
boolean bFalse = (s1 == s2); // false, different objects though with same value
```

Metody třídy `String`

- `char charAt(int index)` — vráti prvek na daném indexu
- `static String format(String format, Object⋯ args)` — stejné jako `printf` v C
- `boolean isEmpty()` — vráti `true` jestli je prázdný
- `int length()` — velikost řetězce
- `matches`, `split`, `indexOf`, `startsWith` …

Více metod najdete v dokumentaci třídy String.

- Doporučujeme javadoc prostudovat, používáním existujících metod si ušetříte spoustu práce!

Víceřádkové řetězcové literály = textové bloky

- V nových verzích Javy (15+) je možnost použít speciální formy řetězcových literálů:

```
String s = """
    toto je první slovo řetězce, druhé,
    druhý řádek,
    třetí""";
```

- Překladač vytvoří řetězec včetně oddělovačů řádku (dříve bylo možné pomocí znaku `\n`).
- Vynechá počáteční mezery (leading spaces), takže skutečně slovo "toto" bude prvními znaky řetězce `s`.

Třída `StringBuilder`

```
StringBuilder builder = new StringBuilder("Hello ");
builder.append("cruel ").append("world"); // method chain
builder.append("!");
String result = builder.toString();
```

- `StringBuilder` se průběžně modifikuje, přidáváme do něj další znaky
- Na závěr vytvoříme výsledný řetězec
- `StringBuilder` není *thread safe*, proto existuje její varianta `StringBuffer`.

Návrhový vzor *Builder*

Třída `StringBuilder` odpovídá návrhovému vzoru *Builder*

- Builder obecně umožňuje konstruovat složitý objekt po částech a až poté získat celý výsledek
- Metody `append` slouží pro přidávání kousků textu, tj. odpovídají metodě `buildPart()` ze vzoru.
- Metoda `toString` slouží k získání výsledného textu, tj. odpovídá metodě `getResult()` ze vzoru.
- `Director` je náš kód používající `StringBuilder`