13. Datový typ řetězec v jazyce JAVA (deklarace, práce s ním, metoda toString)

Nejde o primitivní datový typ!

Třída String

V Javě jsou řetězce prezentovány v objektu typu **String,** který v sobě uchovává řetězec charů.

```
String muj_retezec = "Hello Word";
String muj_retezec = new String("Hello Word");
```

Operace s řetězci

Objekty typu String lze sčítat pomocí **+**.

```
String novy = "Hello Word" + "Gutten Tag";
```

Nebo pomocí metody **concat(String s):**

```
String novy = "Hello word".concat("Guten Tag");
```

Pokud se řetězce sčítají, nemění se původní instance, ale pouze se vytváří nová instance (**imutabilita**). Což je celkem náročná operace, a pokud se provádí častěji, je lepší využít **StringBuffer**, který je pro tento případ vylepšený.

Pokud se dva řetězce porovnávají, nelze použít **==**, ale musí se použít metoda **.equals(String s)**, jelikož == by vyzkoušela, zda se jedná o tentýž objekt.

```
String Hello = "Hello Word";
String Gutten = "Gutten Tag";
boolean stejny = Hello.equals(Gutten);
```

Pokud je potřeba jenom část řetězce, je třeba použít metodu **subString(int start, int end).**

```
String ahoj = "Zdravíčko".substring(0, 4);
```

Pohyb v řetězci

Pomocí metody **charAt(int i)** lze přistupovat k jednotlivým znakům řetězce. První znak má index 0, poslední **length()– 1**.

Metody **indexOf(char)** a **lastIndexOf(char)** vrací pozici prvku v řetězci, první vrátí první výskyt znaku, druhá zase poslední.

Metoda **compareTo(String s)** porovnává řetězce **lexikograficky** (znak po znaku)

StringBuffer

Třída pro řetězce, která uchovává větší pole. Také si pamatuje operace s řetězcem a pomocí speciální metody určí, když je příliš velký nebo malý a určí novou velikost pole.