

# Podstawy programowania

## Laboratorium 6 - klasa BigDecimal i String

mgr inż. Krzysztof Szwarz

krzysztof@szwarz.net.pl

Sosnowiec, 19 kwietnia 2021

## Tworzenie obiektu klasy BigDecimal

Klasa **BigDecimal** przeznaczona jest do przechowywania liczb zmiennoprzecinkowych o ściśle określonej precyzji.

Wykorzystuje się ją do wykonywania operacji finansowych. Aby korzystać z klasy należy ją zaimportować następującą instrukcją:

```
import java.math.BigDecimal;
```

# Przykład

```
BigDecimal numb = new BigDecimal("0.1");  
BigDecimal frac = new BigDecimal("0.1");  
numb = numb.add(frac);  
numb = numb.add(frac);  
System.out.print(numb); // 0.3
```

- 1 Napisz konwerter złotych na euro z wykorzystaniem klasy `BigDecimal`. Znajdź w **dokumentacji** metodę wykorzystywaną do wykonania operacji mnożenia.

## Tworzenie obiektu klasy String

Obiekty klasy **String** przeznaczone są do przechowywania ciągów znaków. Przykład deklaracji:

```
String variable
```

Przykład inicjalizacji:

```
String variable = "Ala ma kota";
```

W Javie obiekty klasy String są niemodyfikowalne.

## Łączenie wyrażeń

Konkatenacja polega na łączeniu ze sobą wyrażeń. Jej operatorem w Javie jest `+`. W momencie połączenia ciągu znaków przechowywanego przez obiekt klasy `String` następuje utworzenie nowego obiektu `StringBuilder`, wywołanie jego metody `append` i konwersja na obiekt klasy `String`. Przykład:

```
String var = "Ala ma";  
var += " kota";  
System.out.print(var); // "Ala ma kota"
```

# Co naprawdę się wydarzyło?

```
String var = "Ala ma";  
var = new StringBuilder(var).append(" kota").toString();
```

Metoda	Opis
length()	Zwraca liczbę znaków.
contains(ciąg)	Zwraca wartość logiczną czy ciąg występuje.
charAt(indeks)	Zwraca znak znajdujący się pod indeksem.
indexOf(ciąg)	Zwraca pierwszy indeks, w którym występuje ciąg (lub -1).
substring (indeksOd, indeksDo)	Zwraca podciąg znajdujący się pomiędzy wskazanymi indeksami.
toLowerCase(), toUpperCase()	Konwersja znaków na małe/duże.
equals(), equalsIgnoreCase()	Sprawdzenie czy ciągi zawierają ten sam tekst.
replace(aktualny, nowy)	Zamienia ciąg aktualny na nowy.
split(separator)	Dzieli ciąg wejściowy wg separatora.



# Użycie StringBuildera

```
StringBuilder var = new StringBuilder("Ala");  
var.append(" ma kota");  
System.out.print(var);
```

- 1 Napisz program pobierający od użytkownika linijkę tekstu (metoda `nextLine` klasy `Scanner`) i przypisz ją do zmiennej typu `String`.
- 2 Zamień wszystkie znaki na małe w zmiennej utworzonej w pierwszym punkcie i wypisz jej nową wartość.
- 3 Napisz program pobierający od użytkownika kolejne linijki tekstu i tworzący z nich jeden ciąg do momentu wystąpienia w zdaniu słowa "stop" (usuń z otrzymanej wartości wszystkie znaki od słowa "stop" włącznie). Wykorzystaj `StringBuilder`.
- 4 Napisz program pobierający od użytkownika linijkę tekstu i zastępujący co drugi znak znakiem nowej linii (`'\n'`). Wypisz otrzymaną wartość. Wykorzystaj `StringBuilder`.

Dziękuję za uwagę