Podstawy i języki programowania

Laboratorium 6 - klasa BigDecimal i String oraz tablice

mgr inż. Krzysztof Szwarc

krzysztof@szwarc.net.pl

Sosnowiec, 16 listopada 2018

Klasa BigDecimal

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

Zadania

1. Napisz konwerter złotówek na euro z wykorzystaniem klasy BigDecimal. Znajdź w dokumentacji metodę wykorzystywaną do wykonania operacji mnożenia.

Klasa String

Tworzenie obiektu klasy String

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

```
String zmienna;
```

Przykład inicjalizacji:

```
String zmienna = "206 team";
```

W Javie obiekty klasy String są niemodyfikowalne.

Konkatenacja

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 zmienna = "Pycha kroczy";
zmienna += " przed upadkiem";
System.out.print(zmienna); // "Pycha
   kroczy przed upadkiem"
```

Co naprawdę się wydarzyło?

```
String zmienna = "Pycha kroczy";
zmienna = new
    StringBuilder(zmienna).append(" przed
    upadkiem").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 zmienna = new StringBuilder
("Pycha kroczy");
zmienna.append(" przed upadkiem");
System.out.print(zmienna);
```

Zadania

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 StringBuildera.
4. Napisz program pobierający od użytkownika linijkę tekstu i zastępujący co drugi znak znakiem nowej linii (’\n’). Wypisz otrzymaną wartość. Wykorzystaj StringBuildera.

Tablice

Opis

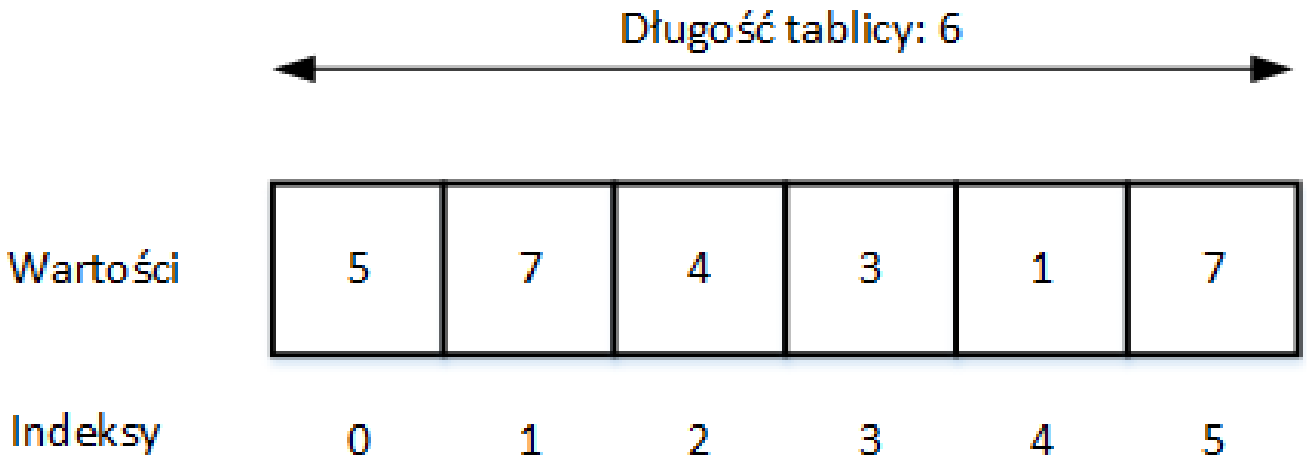
Tablice są strukturami pozwalającymi na przechowywanie określonej liczby zmiennych danego typu w uporządkowanej formie. W zależności od wymiaru tablicy dzielimy je na jednowymiarowe i wielowymiarowe. Tablice posiadają pole length zwracające liczbę przechowywanych elementów. Przykład deklaracji tablicy jednowymiarowej:

```
typ[] nazwaTablicy;
```

Sposób inicjalizacji tablicy jednowymiarowej:

```
typ[] nazwaTablicy = new
   typ[liczbaElementow];
typ[] nazwaTablicy = {elementJeden,
   elementDwa};
```

Reprezentacja tablicy



Przykłady

```
int[] tablica = new int[2];
System.out.println(tablica.length); //2
tablica[0] = 1;
System.out.println(tablica[0]); //1
System.out.println(tablica[1]);
//0 (domyslna wartosc int)
System.out.println(tablica[2]);
//java.lang.ArrayIndexOutOfBoundsException
```

Przykłady cz. 2

```
int[] tablica = {2};
System.out.println(tablica.length); //1
System.out.println(tablica[0]); //2
System.out.println(tablica[1]);
//java.lang.ArrayIndexOutOfBoundsException
```

Przykład wypisania wszystkich elementów tablicy

```
int[] tab = {2, 5, 7};
for (int i=0; i<tab.length; i++)
    System.out.println(tab[i]);
//2,5,7
```

Tablice wielowymiarowe

Opis

Tablice wielowymiarowe obsługujemy analogicznie do tablic jednowymiarowych, a liczbę wymiarów określa liczba nawiasów kwadratowych. Przykład deklaracji tablicy dwuwymiarowej.

```
typ[][] nazwaTablicy;
```

Sposób inicjalizacji tablicy dwuwymiarowej:

```
typ[][] nazwaTablicy = new
   typ[liczbaWierszy][liczbaKolumn];
typ[][] nazwaTablicy =
   {{el1,el2},{el3,el4}};
```

Przykłady

```
int[][] tab = {{1,2},{2,3,4}};
System.out.println(tab[1][0]); // 2
```

Przykład wypisania wszystkich elementów tablicy

```
int[][] tab = {{1,2},{2,3}};
for (int i=0; i<tab.length; i++)
    for (int j=0; j<tab[i].length; j++)
        System.out.println(tab[i][j]);
//1,2,2,3
```

Tablice poszarpane

Opis

Tablice poszarpane są tablicami wielowymiarowymi mającymi różną liczbę wierszy w zależności od indeksu kolumny.
Przykładowy sposób inicjalizacji:

```
int [][] tablica = new int[2][];
for(int i=0; i<tablica.length; i++)
    tablica[i] = new int[i];
```

Przykład

```
int [][] tablica = new int[2][];
for(int i=0; i<tablica.length; i++)
    tablica[i] = new int[i];
System.out.println(tablica[0][0]); //
   ArrayIndexOutOfBoundsException
System.out.println(tablica[1][0]); // 0
```

Zadania

1. Napisz program tworzący dwudziestoelementową tablicę wypełnioną losowymi wartościami z zakresu 0-1 (typu zmiennoprzecinkowego). Aplikacja powinna wypisać średnią arytmetyczną, najmniejszą oraz największą z liczb znajdujących się w tablicy.
2. Napisz program konwertujący liczbę w systemie dziesiętnym na liczbę w systemie o podstawie 2, 4 oraz 8. Niech użytkownik ma możliwość wyboru żądanego systemu.
3. Napisz program pobierający od użytkownika linijkę tekstu i zamieniającą w niej wszystkie znaki zgodnie z szyfrem Cezara (przesunięcie 13).

Zadania

4. Napisz program tworzący tablicę jednowymiarową o rozmiarze 20 wypełnioną wylosowanymi liczbami całkowitymi z przedziału [0;10]. Niech aplikacja posortuje wartości i wypisze je na ekran w porządku leksykograficznym.

Przykład użycia metody split klasy String

```
String zdanie = "Ala_ma_kota";
String[] tablica = zdanie.split("_");
System.out.print(tablica[0]); // Ala
System.out.print(tablica[1]); // ma
System.out.print(tablica[2]); // kota
System.out.print(tablica.length); // 3
```

Kopiowanie tablic

```
System.arraycopy(tablicaDoPrzekopiowania,
    pierwszyIndeksDoPrzekopiowania,
    tablicaDocelowa,
    pierwszyIndeksDoWklejeniaZawartosci,
    liczbaElementowDoPrzekopiowania);
```

Przykład

```
String[] literki = {"f","g","h","i"};
String[] wynikowa = new String[2];
System.arraycopy(literki, 1, wynikowa,
   0, 2);
for (int i=0; i<wynikowa.length; i++)
   System.out.print(wynikowa[i]); // gh
```

Zadania

1. Napisz program kopiujący zawartość jednej tablicy do drugiej, bez użycia metody arraycopy oraz z jej wykorzystaniem.
2. Napisz program, który pobiera od użytkownika pięć liczb całkowitych, oddzielonych spacją (pobierz jedną linijkę tekstu) i korzystając z metody split oraz valueOf przypisz je do tablicy typu int o rozmiarze 5.

Dziękuję za uwagę