

# Programowanie w JAVA

## Lab. 3 – Algorytmy

Każde zadanie zaimplementuj w osobnej paczce. Zadbaj o przejrzystość kodu (w metodzie main powinno być tylko uruchomienie zadania, zadanie powinno być zaimplementowane w dedykowanej klasie, testy powinny być w osobnej klasie).\

1. Zaimplementuj z wykorzystaniem wzorca projektowego **strategia** pięć algorytmów dla sortowania tablic. Wynikiem ma być (dla każdego sortowania) zmierzony czas dla wersji: pesymistycznej, optymistycznej oraz oczekiwanej.

<https://www.geeksforgeeks.org/sorting-algorithms/>

2. Ciąg „PAYPALISHIRING” jest zapisywany zygzakiem w określonej liczbie wierszy w następujący sposób: (dla lepszej czytelności możesz wyświetlić ten wzór stałą czcionką)

```
P A H N
A P L S I I G
Y I R
```

A następnie przeczytaj wiersz po wierszu: „PAHNAPLSIIGYIR”

Napisz kod, który pobierze ciąg znaków i dokonaj konwersji, biorąc pod uwagę liczbę wierszy:

string convert(string s, int numRows);

Przykład 1:

Input: s = "PAYPALISHIRING", numRows = 3

Output: "PAHNAPLSIIGYIR"

Przykład 2:

Input: s = "PAYPALISHIRING", numRows = 4

Output: "PINALSIGYAHRPI"

3. Biorąc pod uwagę dwie posortowane tablice num1 i num2 o rozmiarach odpowiednio m i n, zwróć medianę dwóch posortowanych tablic.

Ogólna złożoność czasu wykonania powinna wynosić  $O(\log(m+n))$ .

Przykład 1:

Wejście: num1 = [1,3], num2 = [2]

Wyjście: 2,00000

Objaśnienie: połączona tablica = [1,2,3], a mediana wynosi 2.

Przykład 2:

Wejście: num1 = [1,2], num2 = [3,4]

Wyjście: 2,50000

Objaśnienie: połączona tablica = [1,2,3,4], a mediana wynosi  $(2 + 3) / 2 = 2,5$ .

4. Biorąc pod uwagę tablicę liczb całkowitych nums, zwróć wszystkie trójki [nums[i], nums[j], nums[k]] takie, że  $i \neq j$ ,  $i \neq k$  oraz  $j \neq k$ , poza tym  $nums[i] + nums[j] + nums[k] == 0$ .

Należy zauważyć, że zbiór rozwiązań nie może zawierać zduplikowanych trójek.

Przykład 1:

Wejście: nums = [-1,0,1,2,-1,-4]

Przykład 2:

Wejście: liczby = [0,1,1]

<p>Wyjście: <code>[[-1,-1,2],[-1,0,1]]</code></p> <p>Wyjaśnienie:</p> <p><math>liczby[0] + liczby[1] + liczby[2] = (-1) + 0 + 1 = 0.</math></p> <p><math>nums[1] + nums[2] + nums[4] = 0 + 1 + (-1) = 0.</math></p> <p><math>nums[0] + nums[3] + nums[4] = (-1) + 2 + (-1) = 0.</math></p> <p>Odrębne trojaczki to <code>[-1,0,1]</code> i <code>[-1, -1,2]</code>.</p> <p>Zauważ, że kolejność wyników i kolejność trójek nie ma znaczenia.</p>	<p>Wyjście: <code>[]</code></p> <p>Objaśnienie: Jedyna możliwa trójka nie sumuje się do 0.</p> <p>Przykład 3:</p> <p>Wejście: <code>liczby = [0,0,0]</code></p> <p>Wyjście: <code>[[0,0,0]]</code></p> <p>Wyjaśnienie: Jedyna możliwa trójka sumuje się do 0.</p>
--	---

5. Dana jest macierz o wymiarach  $n \times m$ . Wypisz elementy macierzy jako spiralą, przeciwnie do kierunku ruchu wskazówek zegara.
6. **Zaimplementuj własne klasy wyjątków na wypadek nie spełnienia któregoś z wymogów algorytmu.**

Przykład do zad. 1:

```
long tStart = System.currentTimeMillis();  
// uruchom sortowanie.  
long tEnd = System.currentTimeMillis();  
long tDelta = tEnd - tStart;  
double elapsedSeconds = tDelta / 1000.0;
```

Aby wyniki były porównywalne wielkość tablicy powinna być relatywnie duża. Tablicę w przypadku realistycznym należy wypełnić losowymi liczbami (zastosuj klasę *Random*)

Przykład do zad. 5

```
int[][] tab = {  
    { 1, 2, 3, 4, 5},  
    { 6, 7, 8, 9, 10},  
    {11, 12, 13, 14, 15},  
    {16, 17, 18, 19, 20},  
    {21, 22, 23, 24, 25}  
};  
printClockwise(tab);  
// 1 ,6, 11, 16, 21, 22, 23, 24, 25, 20, 15, 10, 5, 4, 3, 2, 7, 12, 17, 18, 19, 14, 9, 8, 13
```

**Teoria:**

1. Własne typy wyjątków, hierarchia dziedziczenia wyjątków
2. Do czego służy klasa Object i jakie ma zastosowanie ?
3. Jak budować własne kolekcje i struktury danych w Java?
4. Generyczność i hierarchia dziedziczenia kolekcji

<https://www.javatpoint.com/collections-in-java>

Po uzyskaniu zaliczenia na zajęciach, prześlij źródła w archiwum **zgodnie z konwencją nazewnictw** (patrz prezentacja 0) do chmury na adres:

<https://cloud.kisim.eu.org/s/f3GKkgXRX89QsmT>

najpóźniej do następnych zajęć