



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційні систем та технологій

**Лабораторна робота №3**  
із дисципліни *«Основи програмування»*  
**Тема:** «Управління потоком виконання»

Виконали:  
Студенти групи ІА-24  
Красношапка Р.О.  
Бакалець А.І.  
Орловська А.В.

Перевірив:  
Колеснік В.М.

## Хід роботи:

1. Виконати три завдання з таблиці 2 відповідно до свого варіанту у таблиці 1.

- Кожне завдання має бути реалізовано як окремий клас.
- Кожен клас має складатись щонайменше з двох методів.

## Завдання 3

3	Обчислити суму ряду: $\sum_{i=1}^k \left( \frac{1}{\sqrt{z \cdot i}} + \operatorname{tg} \left( \frac{k}{i} \right) \right), \quad k < 15$
---	--

## Код програми:

```
public class lab3 {
    1 usage
    static double function(double z, int k) {
        if (k < 1 || z < 0) {
            throw new IllegalArgumentException("param a = " + k);
        } else {
            double res = 0;
            for (int i = 1; i <= k; i++) {
                res += (1 / Math.sqrt(z * i)) + (Math.tan(k / i));
            }
            return res;
        }
    }

    static int getFactorial(int n) {
        int factorial = 1;
        for (int i = 1; i <= n; i++) {
            factorial *= i;
        }
        return factorial;
    }

    public static void main(String[] args) {
        printResults(z: 3.26, k: 1);
        printResults(z: 55.1, k: -1);
        printResults(z: 15.6, k: -20);
        printResults(z: -1, k: 20);
    }

    public static void main(String[] args) {
        printResults(z: 3.26, k: 1);
        printResults(z: 55.1, k: -1);
        printResults(z: 15.6, k: -20);
        printResults(z: -1, k: 20);
    }

    static String @kryglenis(double res) {
        String result = String.format("Сума = " + "%.3f", res);
        return result;
    }

    4 usages
    private static double printResults(double z, int k) {
        System.out.print("k : " + k);
        try {
            System.out.println(function(z, k));
        } catch (IllegalArgumentException e) {
            System.out.println("EXCEPTION! " + e.getMessage());
        }
        return 0;
    }
}
```

## Результат:

```
2022.2.2\bin" -Dfile.encoding=UTF-8 -c
k : 12.1112565002766135
k : -1EXCEPTION! param a = -1
k : -20EXCEPTION! param a = -20
k : 20EXCEPTION! param a = 20
```

## Завдання 8

8	$x(t, n, l) = \sum_{i=1}^n \begin{cases} \frac{1}{t} \cdot l, & l = 1, 3, 5, \dots \\ \frac{1}{\sqrt{t}} \cdot l, & l = 2, 4, 6, \dots \end{cases}$ <p>Обчислити значення функції:</p>
---	--

## Код програми:

```
public class lab3 {
    public static void main(String[] args) {
        double res = function(10, 0.25);
        System.out.println(res);
    }

    static double function(int n, double t) {
        double firstBrack = 0;
        double secondBrack = 0;
        double res = 0;
        int[] l1 = {1, 3, 5};
        int[] l2 = {2, 4, 6};
        for (int j = 0; j < l1.length; j++) {
            for (int i = 1; i <= n; i++) {
                firstBrack = 1 / t * l1[j];
                secondBrack = (1 / Math.sqrt(t)) * l2[j];
                if (firstBrack == secondBrack) {
                    res += firstBrack;
                }
            }
        }
    }
}
```

## Результат:

```
C:\Program Files\Java\jdk-17.0.4\bin\jav
2022.2.2\bin" -Dfile.encoding=UTF-8 -class
40.0
```

## Завдання 13

13	$\sum_{i=1}^{\infty} \frac{(-1)^i}{i!}$ <p>Обчислити нескінченну суму <math>\sum_{i=1}^{\infty} \frac{(-1)^i}{i!}</math> із заданою точністю <math>\epsilon</math> (<math>\epsilon &gt; 0</math>). Вважати, що необхідна точність досягнута якщо черговий доданок виявився по модулю меншим, ніж <math>\epsilon</math>. Цей і усі наступні доданки можна не враховувати.</p>
----	--

## Код програми:

```

public class Lab3 {

    public static void main(String[] args) {
        printResults(e: 0);
        printResults(e: -1);
        printResults(e: 9);
        printResults(e: 0.019);
    }

    2 usages
    static double dodanok (int i) {
        double funk = 0;
        for (int k = 1; k <= i; k++) {
            funk += k;
        }

        return (Math.pow(-1, i) / funk);
    }

    1 usage
    public static double fid (double e) {
        double res = 0;
        if (e <= 0) {
            throw new IllegalArgumentException("param e = " + e);
        }
        int i = 1;
        while (Math.abs(dodanok(i)) > e) {
            res += dodanok(i);
            i++;
        }
        return res;
    }

    4 usages
    static void printResults(double e) {
        System.out.print("e: " + e + " result ");
        try {
            System.out.println(fid(e));
        } catch (IllegalArgumentException er) {
            System.out.println("EXCEPTION! " + er.getMessage());
        }
    }
}

```

## Результат:

```

2022.2.2\bin" -Dfile.encoding=UTF-8 -classpath
e: 0.0 result EXCEPTION! param e = 0.0
e: -1.0 result EXCEPTION! param e = -1.0
e: 9.0 result 0.0
e: 0.019 result -0.7825396825396825

```

**Висновок:** під час виконання лабораторної роботи, ознайомилися з управлінням потоком даних .