

**Министерство образования Республики Беларусь
Учреждение Образования
«Брестский Государственный Технический Университет»
Кафедра ИИТ**

**Лабораторная работа №8
По дисциплине СПП за 6 семестр
Тема: «Java»**

Выполнил:
Студент 3-го курса
Группы ПО-5
Крощук В.В.
Проверил:
Крощенко А.А.

Брест 2022

Лабораторная работа №8

Цель работы: Приобрести навыки написания простого оконного многопоточного приложения с использованием Java API.

Вариант 8.

Задание.

Разработать оконное приложение с использованием Java API, использующее один вспомогательный поток, вычисляющий заданную сумму и выполняющий вывод результата вычисления (как конечный, так и промежуточные) в любой визуальный компонент. Все исходные данные вводятся в соответствующие визуальные компоненты.

В программе должны быть предусмотрены функции приостановки, возобновления и полной остановки выполнения потока с выводом соответствующего сообщения. В случае быстрого выполнения потока и, как следствие, невозможности демонстрации функций приостановки, продумать искусственное торможение потока для достижения заданных целей. Обработать исключения.

8)

$$\sum_{k=0}^n \frac{(x \ln a)^k}{k!} = 1 + \frac{x \ln a}{1!} + \frac{(x \ln a)^2}{2!} + \dots + \frac{(x \ln a)^n}{n!}$$

Рисунок 1 - Исходная формула.

Код консольной программы:

Main.java

```
package com.company;
import java.util.Scanner;
class Tred extends Thread{
    Scanner in = new Scanner(System.in);
    Tred(String name){
        super(name);
    }

    @Override
    public void run(){
        System.out.println("Поток " + Thread.currentThread().getName() + "
стартовал..." + '\n');
        System.out.print("Введите степень k: ");
        int k = in.nextInt() + 1;
        System.out.print("Введите число x: ");
        int x = in.nextInt();
        System.out.print("Введите число a: ");
        int a = in.nextInt();
        int i = 0;
        double sum = 0;
        while(i != k){
            sum += Math.pow(x * Math.log(a), i) / factorial(i);
            System.out.println("sum = " + sum);
            try{
                System.out.println("stop on 1s");
                Thread.sleep(1);
            }
            catch (InterruptedException e){
                System.out.println("Поток " + Thread.currentThread().getName() + "
был прерван");
            }
            i++;
        }
        System.out.println("Итоговая сумма равна = " + sum + '\n');
```

```

        System.out.println("Поток " + Thread.currentThread().getName() + " завершил
работу...");
    }
    public double factorial(int num){
        if(num == 0 || num == 1)
            return 1;
        int sum = 1;
        for(int i = 2; i <= num; i++)
            return sum *= i;
        return sum;
    }
}
public class Main {

    public static void main(String[] args) {
        System.out.println("Поток Main стартовал...");
        Tred t = new Tred("Treddy");
        t.start(); // Treddy
        try{
            t.join();
        }
        catch(InterruptedException e){
            System.out.println("Поток " + t.getName() + " был прерван");
        }
        System.out.println("Поток Main завершил работу...");
        System.out.println();
    }
}

```

Результат тестирования:

```

"C:\Program Files\Eclipse Foundation\jdk-
Поток Main стартовал...
Поток Treddy стартовал...

Введите степень k: 46
Введите число x: 50
Введите число a: 100
sum = 1.0
stop on 1s
sum = 231.25850929940458
stop on 1s
sum = 26740.749061691396
stop on 1s
sum = 6130776.525942121
stop on 1s
sum = 1.4116369552206628E9
stop on 1s
sum = 3.250413944725696E11
stop on 1s
sum = 7.48435469253451E13
stop on 1s
sum = 1.7233363545683492E16
stop on 1s
sum = 3.9681286002437555E18
stop on 1s
sum = 9.136953762004599E20
stop on 1s
sum = 2.103861352776766E23
stop on 1s
sum = 4.844319788630069E25

```

...

```

sum = 5.503323240072184E84
stop on 1s
sum = 1.2671870054517905E87
stop on 1s
sum = 2.9178059087890574E89
stop on 1s
sum = 6.718496389827628E91
stop on 1s
sum = 1.5469909634551411E94
stop on 1s
sum = 3.5620783314483045E96
stop on 1s
sum = 8.201988466069969E98
stop on 1s
sum = 1.888577637488181E101
stop on 1s
sum = 4.348610715042199E103
stop on 1s
sum = 1.0013046207690344E106
stop on 1s
sum = 2.3055890933288352E108
stop on 1s
Итоговая сумма равна = 2.3055890933288352E108

Поток Treddy завершил работу...
Поток Main завершил работу...

Process finished with exit code 0

```

Код программы с графическим интерфейсом:

Culculate.java

```
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.geometry.*;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.GridPane;
import javafx.scene.text.Text;

public class Calculate extends Application {

    double currentValue = 0;
    int currentIteration = 0;
    Text sum = new Text("");
    Thread backgroundThread;
    TextField inputCount = new TextField();
    TextField inp = new TextField();
    TextField in = new TextField();
    Button start = new Button();
    GridPane grid;

    @Override
    public void init() {
        start.setText("Начать");
        Button pause = new Button();
        pause.setText("Пауза");
        Button stop = new Button();
        stop.setText("ОЧИСТИТЬ");
        start.setOnAction(actionEvent -> startCalculate());
        pause.setOnAction(actionEvent -> {
            start.setDisable(false);
            backgroundThread.suspend(); // приостанавливает
        });
        stop.setOnAction(actionEvent -> {
            start.setDisable(false);
            stopCalculate();
        });
        grid = new GridPane();
        grid.setAlignment(Pos.CENTER);
        grid.setHgap(10);
        grid.setVgap(10);
        grid.setPadding(new Insets(25, 25, 25, 25));
        Text text = new Text("Текущая сумма: ");
        grid.add(text, 0, 0, 1, 1);
        grid.add(sum, 1, 0, 1, 1);
        Label labelCount = new Label("K:");
        grid.add(labelCount, 0, 1, 1, 1);
        grid.add(inputCount, 1, 1, 1, 1);
        Label labelinp = new Label("X:");
        grid.add(labelinp, 0, 2, 1, 1);
        grid.add(inp, 1, 2, 1, 1);
        Label labelin = new Label("A: ");
        grid.add(labelin, 0, 3, 1, 1);
        grid.add(in, 1, 3, 1, 1);
        grid.add(start, 0, 5);
        grid.add(pause, 1, 5);
        grid.add(stop, 2, 5);
    }

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Function");
        Scene scene = new Scene(grid, 500, 500);
        primaryStage.setScene(scene);
    }
}
```

```

        primaryStage.show();
    }

    public void startCalculate() {
        if (backgroundThread != null) {
            backgroundThread.resume(); // продолжает
        } else {
            Thread task = new Thread(() -> {
                try {
                    int k = Integer.parseInt(inputCount.getText()) + 1;
                    int x = Integer.parseInt(inp.getText());
                    int a = Integer.parseInt(in.getText());
                    start.setDisable(true);
                    if (a == 0)
                        sum.setText("-infinity");
                    if (k == 0) {
                        sum.setText(Double.toString(1.0));
                    } else {
                        sum.setText(Double.toString(this.currentValue));
                        for (int i = 0; i < k; i++) {
                            try {
                                this.currentValue += Math.pow(x * Math.log(a), i) /
factorial(i);

                                Thread.sleep(500);
                                sum.setText(Double.toString(this.currentValue));
                            } catch (InterruptedException e) {
                                e.printStackTrace();
                            }
                        }
                    }
                    start.setDisable(false);
                } catch (NumberFormatException e) {
                    sum.setText("Ошибка ввода");
                }
            });
            backgroundThread = new Thread(task);
            backgroundThread.setDaemon(true);
            backgroundThread.start();
        }
    }

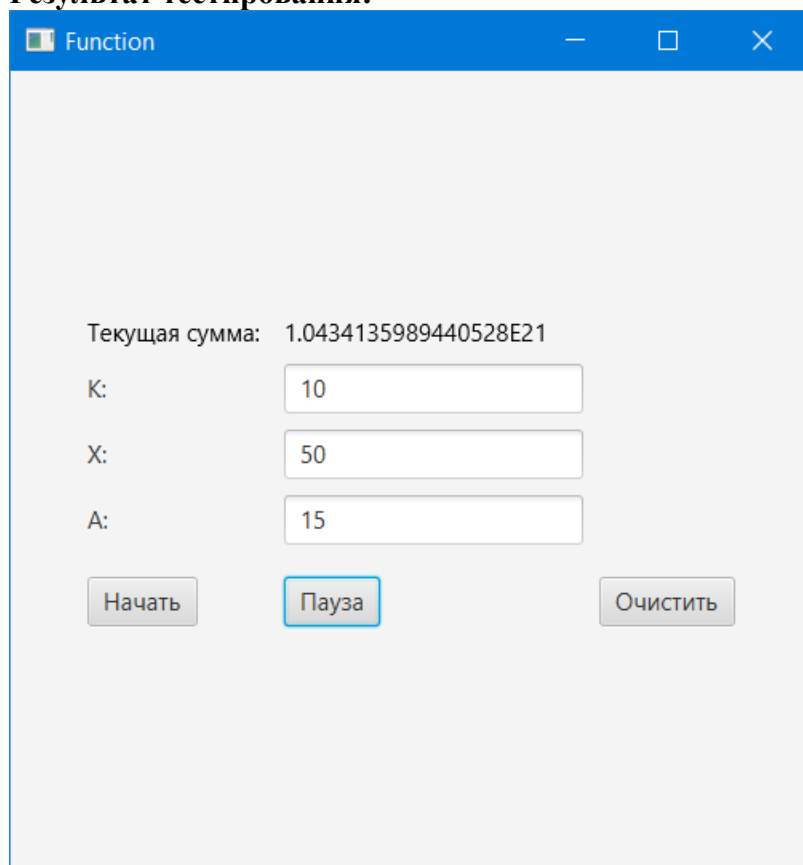
    public double factorial(int num) {
        if (num == 0 || num == 1)
            return 1;
        int sum = 1;
        for (int i = 2; i <= num; i++)
            return sum *= i;
        return sum;
    }

    public void stopCalculate() {
        backgroundThread.stop();
        this.currentValue = 0;
        this.sum.setText("");
        this.currentIteration = 0;
        inputCount.setText("");
        inp.setText("");
        in.setText("");
        backgroundThread = null;
    }

    public static void main(String[] args) {
        launch(args);
    }
}

```

Результат тестирования:



The screenshot shows a Java application window titled "Function". Inside the window, the text "Текущая сумма: 1.0434135989440528E21" is displayed. Below this, there are three input fields labeled "K:", "X:", and "A:". The "K:" field contains the value "10", the "X:" field contains "50", and the "A:" field contains "15". At the bottom of the window, there are three buttons: "Начать", "Пауза", and "Очистить". The "Пауза" button is highlighted with a blue border.

Label	Value
Текущая сумма:	1.0434135989440528E21
K:	10
X:	50
A:	15

Buttons: Начать, Пауза, Очистить

Вывод: Приобрел навыки написания простого оконного многопоточного приложения с использованием Java API.