

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №8
По дисциплине «СПП»

Выполнил
студент 3 курса
группы ПО-3:
Григорьева В.А.
Проверил:
Крощенко А.А.

Брест, 2021

Вариант 8

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

Ход работы:

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. При вычислении суммы рекомендуется находить следующее слагаемое применением рекуррентной формулы.

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

Код программы:

Main.java

```
package sample;

import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.scene.text.Text;
import javafx.stage.Stage;

public class Main extends Application {
    double currentValue = 0;
    private double prevElement=0;
    private double currentSum=1;
    private int N = 3;

    int currentIteration = 0;
    Text sum = new Text("");
    TextField inputCount = new TextField();
    TextField inputValue = new TextField();
    Thread backgroundThread;
    Button start = new Button();
    GridPane grid;

    @Override
    public void init() {
        start.setText("Start");
        Button pause = new Button();
        pause.setText("Pause");
        Button stop = new Button();
        stop.setText("Stop");
    }
}
```

```

start.setOnAction(actionEvent -> startCalculate());
pause.setOnAction(actionEvent -> {
    start.setDisable(false);
    backgroundThread.suspend();
});
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("N:");
Label labelValue = new Label("X:");
grid.add(labelCount, 0, 1, 1, 1);
grid.add(inputCount, 1, 1, 1, 1);
grid.add(labelValue, 0, 2, 1, 1);
grid.add(inputValue, 1, 2, 1, 1);
grid.add(start, 0, 3);
grid.add(pause, 1, 3);
grid.add(stop, 2, 3);
}

@Override
public void start(Stage primaryStage) {
    primaryStage.setTitle("Function");
    Scene scene = new Scene(grid, 500, 350);
    primaryStage.setScene(scene);
    primaryStage.show();
}

public void startCalculate() {
    if (backgroundThread != null) {
        backgroundThread.resume();
    } else {
        Thread task = new Thread(() -> {
            try {
                int i = 1;
                int count = Integer.parseInt(inputCount.getText());
                int xValue = Integer.parseInt(inputValue.getText());
                start.setDisable(true);
                while (i <= count) {
                    try {
                        prevElement = Math.pow(xValue * Math.log10(N), i) /
factorial(i);

                        currentSum += prevElement;
                        sum.setText(String.valueOf(currentSum));
                        i++;
                        Thread.sleep(500);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
                start.setDisable(false);
            } catch (NumberFormatException e) {
                sum.setText("Error in input!!!");
            }
        });
    }

    backgroundThread = new Thread(task);
    backgroundThread.setDaemon(true);
    backgroundThread.start();
}

```

```

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

public void stopCalculate() {
    backgroundThread.stop();
    backgroundThread = null;
    this.currentValue = 0;

    this.sum.setText("");
    this.currentSum = 1;
    this.prevElement = 0;
    this.currentIteration = 0;
    inputCount.setText("");
    inputValue.setText("");
}

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

```

sample.fxml

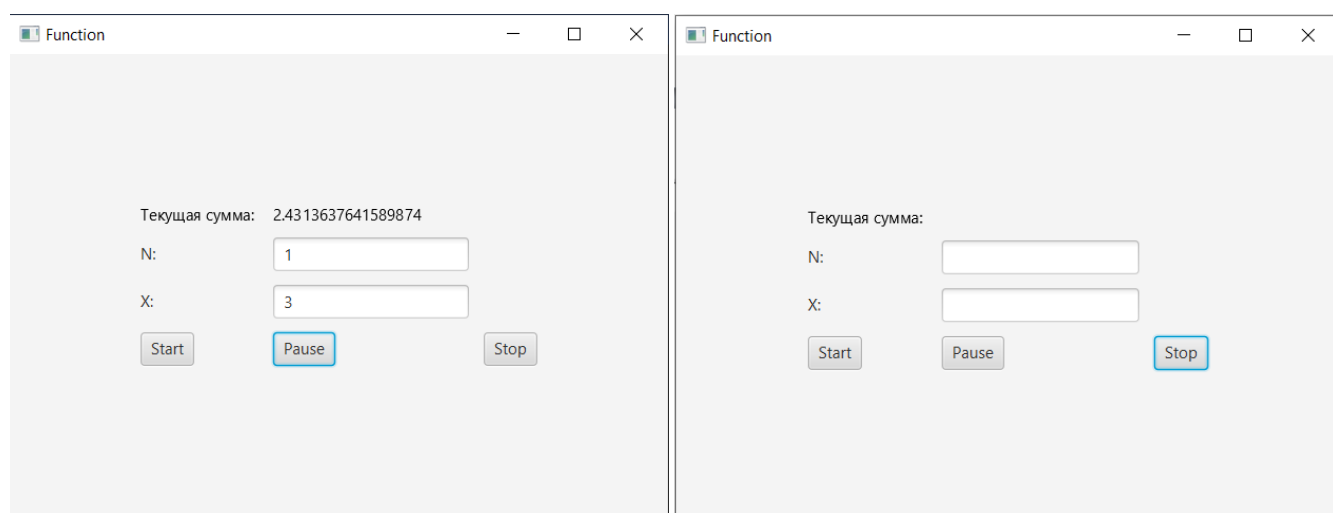
```

<?import javafx.scene.layout.GridPane?>

<GridPane fx:controller="sample.Controller"
    xmlns:fx="http://javafx.com/fxml" alignment="center" hgap="10" vgap="10">
</GridPane>

```

Результат выполнения:



Function

Текущая сумма: 546.3708957615962

N: 10

X: 20

Start

Pause

Stop

Function

Текущая сумма: 8921.933038742054

N: 10

X: 20

Start

Pause

Stop