

Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №8

По дисциплине «Современные платформы программирования»

Выполнила:

Студентка 3 курса

Группы ПО-3

Пивчик В.Г.

Проверил:

Крощенко А.А.

Брест 2020 г.

Вариант 9

Цель работы:

приобрести навыки написания простого оконного многопоточного приложения с использованием Java API.

Ход работы

Задание 1:

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.

$$\frac{\pi}{2} - \sum_{k=0}^n (-1)^k \frac{x^{2k+1}}{2k+1} = \frac{\pi}{2} - \left(x - \frac{x^3}{3} + \frac{x^5}{5} - \dots + (-1)^n \frac{x^{2n+1}}{2n+1} \right)$$

Текст программы:

```
package com.company;
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;
    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.stop();
        });
        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 double calculate(int xValue) {
        int intermediateValue = 2 * currentIteration + 1;
        return Math.pow(-1, currentIteration) * Math.pow(xValue,
intermediateValue) /
            intermediateValue + currentValue;
    }
    public void startCalculate() {
        Thread task = new Thread(() -> {
            try {
                int count = Integer.parseInt(inputCount.getText());
                int xValue = Integer.parseInt(inputValue.getText());
                start.setDisable(true);
                for (int i = currentIteration; i < count; i++) {
                    try {
                        currentValue = calculate(xValue);

                        sum.setText(Double.toString(Math.PI/2 -
currentValue));

                        currentIteration = i + 1;
                        Thread.sleep(100);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
            }
            start.setDisable(false);
        });
    }

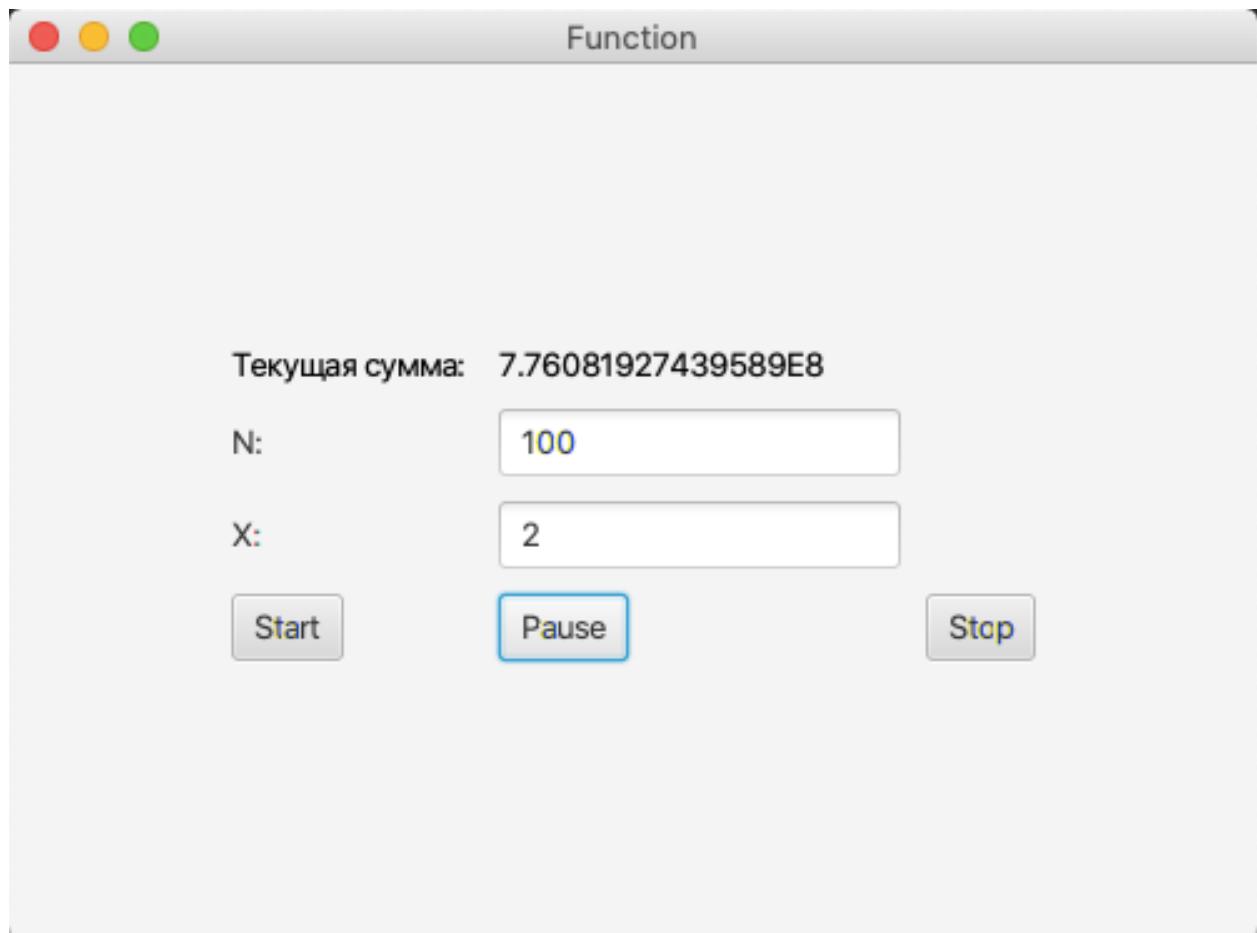
```

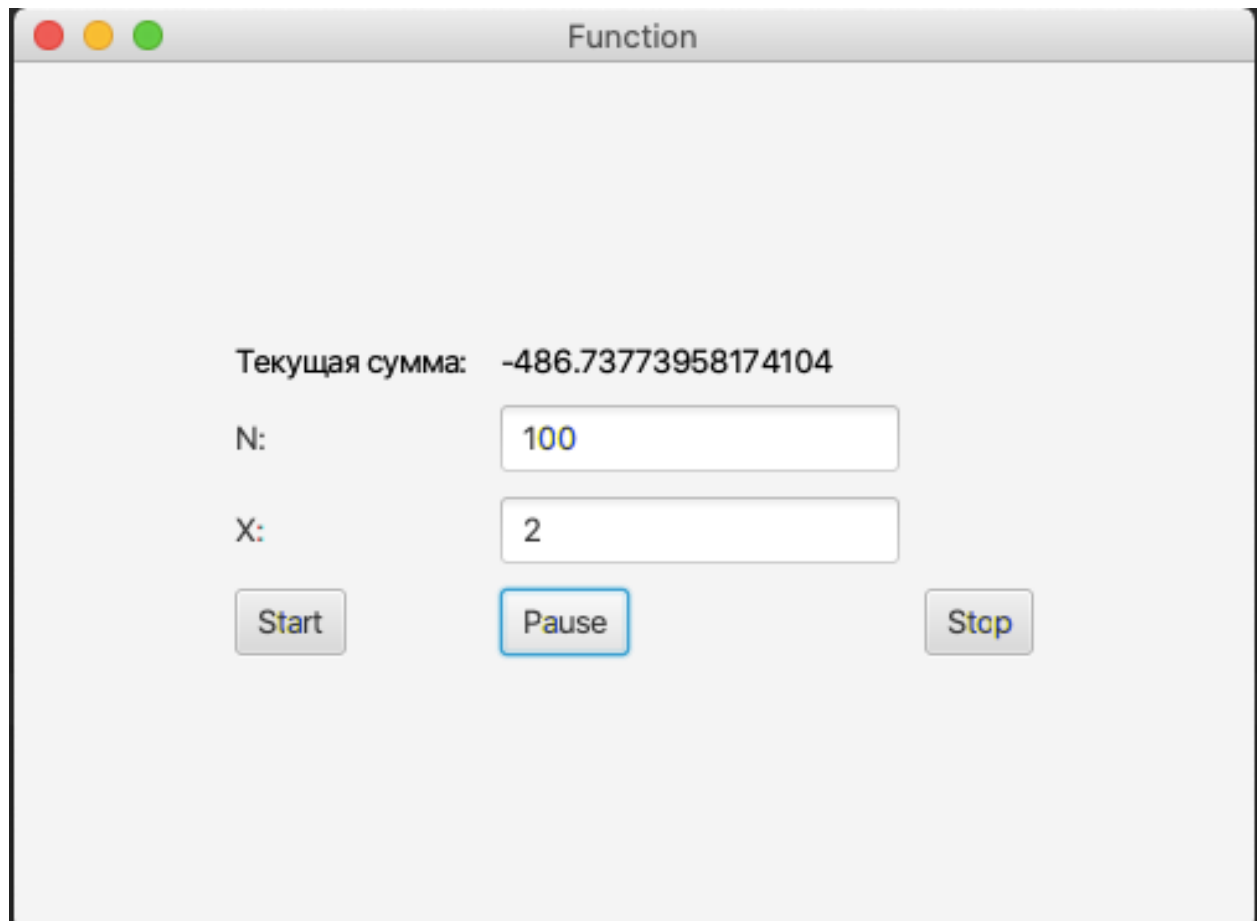
```

        } catch (NumberFormatException e) {
            sum.setText("Error in input!!!");
        });
    backgroundThread = new Thread(task);
    backgroundThread.setDaemon(true);
    backgroundThread.start();
}
public void stopCalculate() {
    backgroundThread.stop();
    this.currentValue = 0;
    this.sum.setText("");
    this.currentIteration = 0;
    inputCount.setText("");
    inputValue.setText("");
}
public static void main(String[] args) {
    launch(args);
}
};

```

Рисунок с результатом работы программы:





Выводы:

Приобрела практические навыки в области написания многопоточного Java-приложения.