Министерство образования Республики Беларусь
Учреждение образования
“Брестский государственный технический университет”
Кафедра ИИТ

Отчёт
По лабораторной работе №8
По дисциплине СПП

Выполнил
Студент группы ПО-3
3-го курса
Кулинкович И. Т.

Проверил
Крощенко А. А.

Брест 2021

Лабораторная работа №8

ВАРИАНТ 13

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.

$$\sum_{k=0}^{n} \frac{1}{k!} = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!} \tag{1}$$

Код программы

live.ilyusha.Main

```
package live.ilyusha;

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;
    double prevElement = 1;
    int currentIteration = 0;
    Text sum = new Text("");
    TextField inputCount = 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:");
        grid.add(labelCount, 0, 1, 1, 1);
        grid.add(inputCount, 1, 1, 1, 1);
        grid.add(start, 0, 2);
        grid.add(pause, 1, 2);
        grid.add(stop, 2, 2);
    }

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

    public double calculate() {
        return prevElement * 1.0 / (currentIteration == 0 ? 1 :
currentIteration);
    }

    public long factorial(int n) {
        if(n == 0 || n == 1) {
            return 1;
        }
        return n <= 2 ? n : n * factorial(n - 1);
    }

    public void startCalculate() {
        Thread task = new Thread(() -> {
            try {
                int count = Integer.parseInt(inputCount.getText());
                start.setDisable(true);

                for (int i = currentIteration; i < count; i++) {
                    try {
                        this.currentIteration = i + 1;

                        currentValue = calculate();
                        prevElement = currentValue;

                        sum.setText(Double.toString(currentValue));
                        Thread.sleep(100);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
                start.setDisable(false);
            } catch (NumberFormatException e) {
```

```
                sum.setText("Error");
            }

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

    public void stopCalculate() {
        backgroundThread.stop();
        this.currentValue = 0.0;
        this.prevElement = 1.0;
        this.currentIteration = 0;
        this.sum.setText("");
        inputCount.setText("");
    }

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

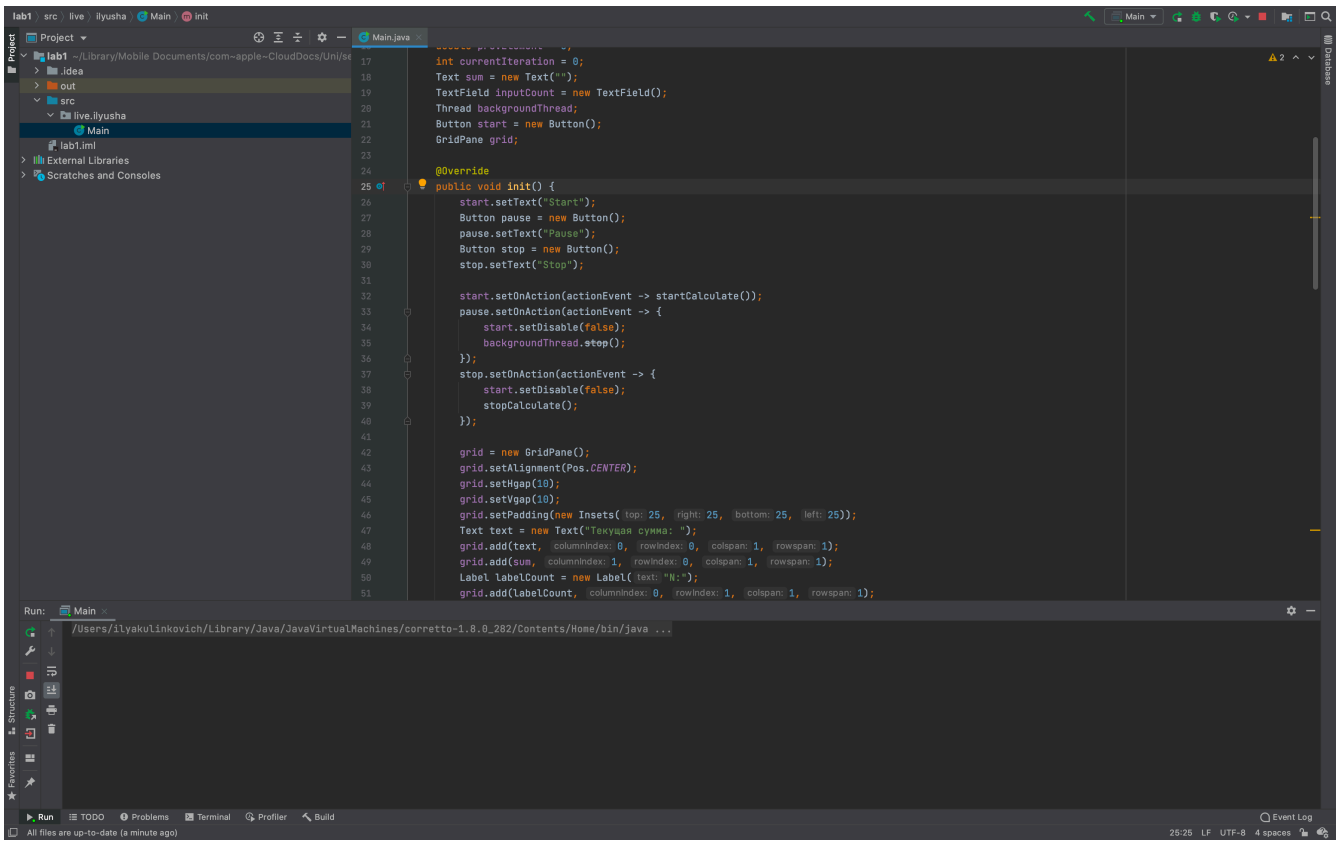
Спецификация ввода

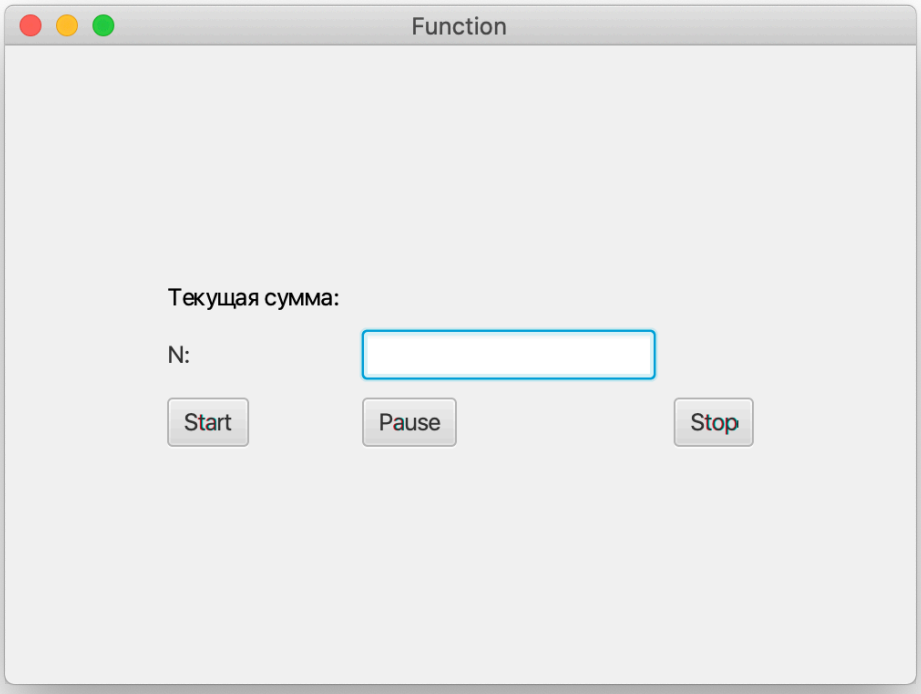
Приложение не требует ввода данных для запуска, при работе приложения можно ввести число для подстановки в формулу.

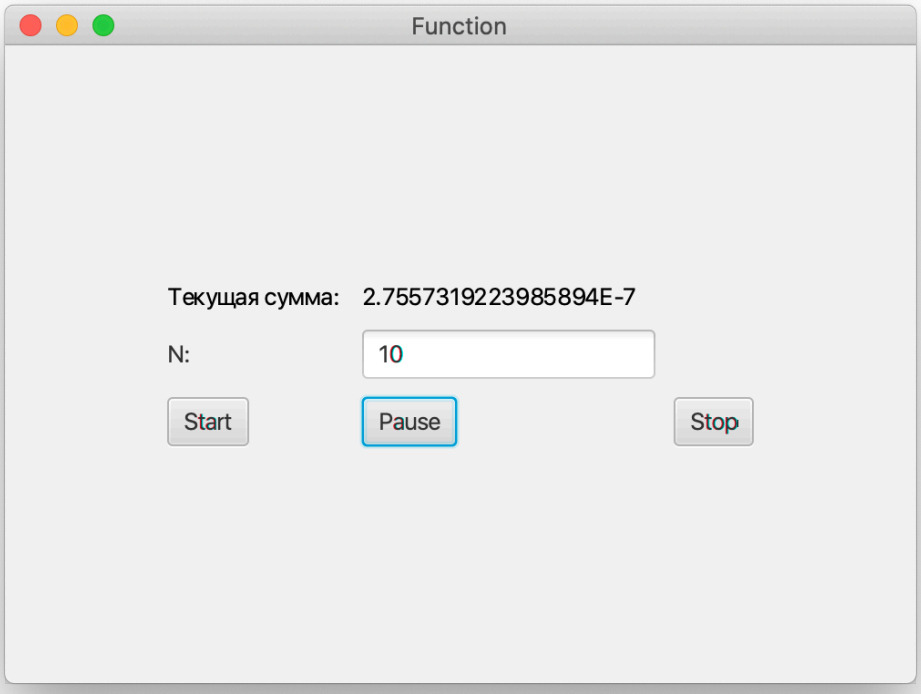
Спецификация вывода

Графическое приложение

Рисунки с результатами работы программы







Вывод

В данной лабораторной работе я приобрел навыки написания простого оконного многопоточного приложения с использованием Java API.