

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

Лабораторная работа №8  
По дисциплине СПП за 6 семестр  
Вариант 11

**Выполнил:**

Студент группы ПО-6(1)  
3-го курса

Мартынович Даниил

**Проверил:**

Монтик Н. С.

Брест 2023

### Цель работы:

приобрести навыки написания простого оконного многопоточного приложения с использованием Java API

### Общее 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.

$$\sum_{k=1}^n (-1)^{k-1} \frac{1}{k^2} = 1 - \frac{1}{2^2} + \frac{1}{3^2} - \frac{1}{4^2} + \dots \pm \frac{1}{n^2}$$

### Ход работы

#### 1) Текст программы

```
{
    double currentValue = 0;

    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.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:");
grid.add(labelCount, 0, 1, 1, 1);
grid.add(inputCount, 1, 1, 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 count = Integer.parseInt(inputCount.getText());
                start.setDisable(true);

                if (count == 0) {
                    sum.setText(Double.toString(1.0));
                } else {
                    sum.setText(Double.toString(this.currentValue));
                    for (int i = 0; i <= count; i++) {
                        try {
                            this.currentValue += Math.pow(-1,i)*1/Math.pow(2.0,i);
                            Thread.sleep(500);
                            sum.setText(Double.toString(this.currentValue));
                        } 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();
    }
}

```

```

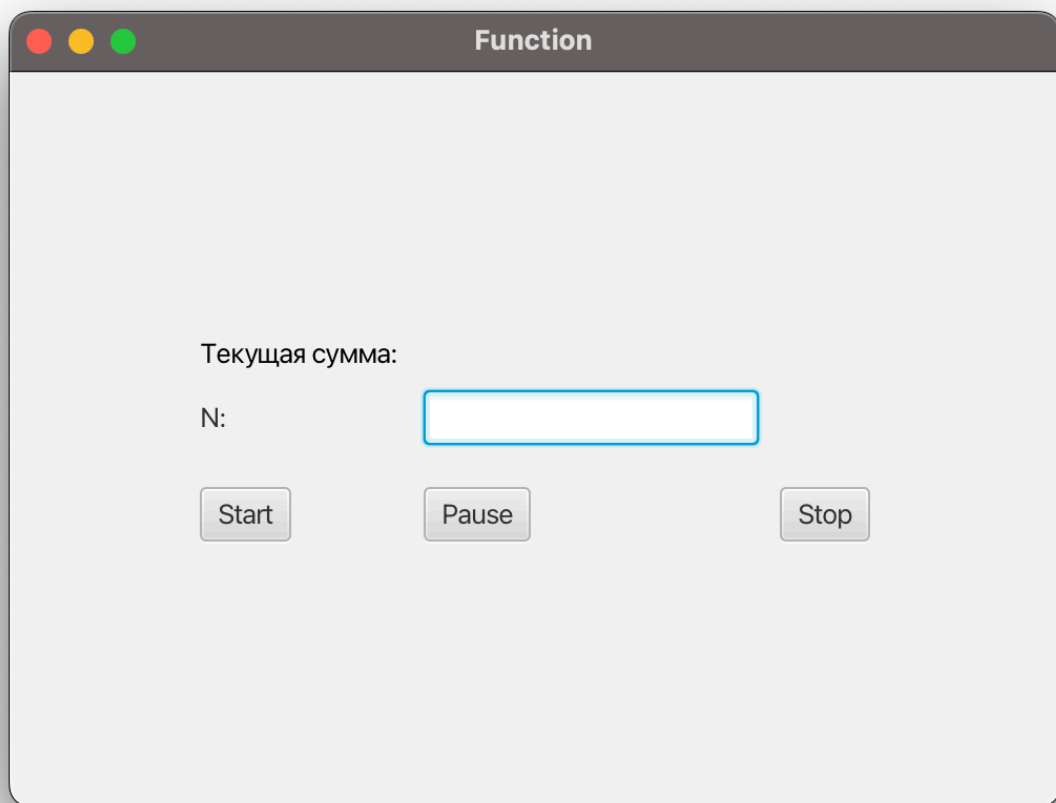
public void stopCalculate() {
    backgroundThread.stop();
}

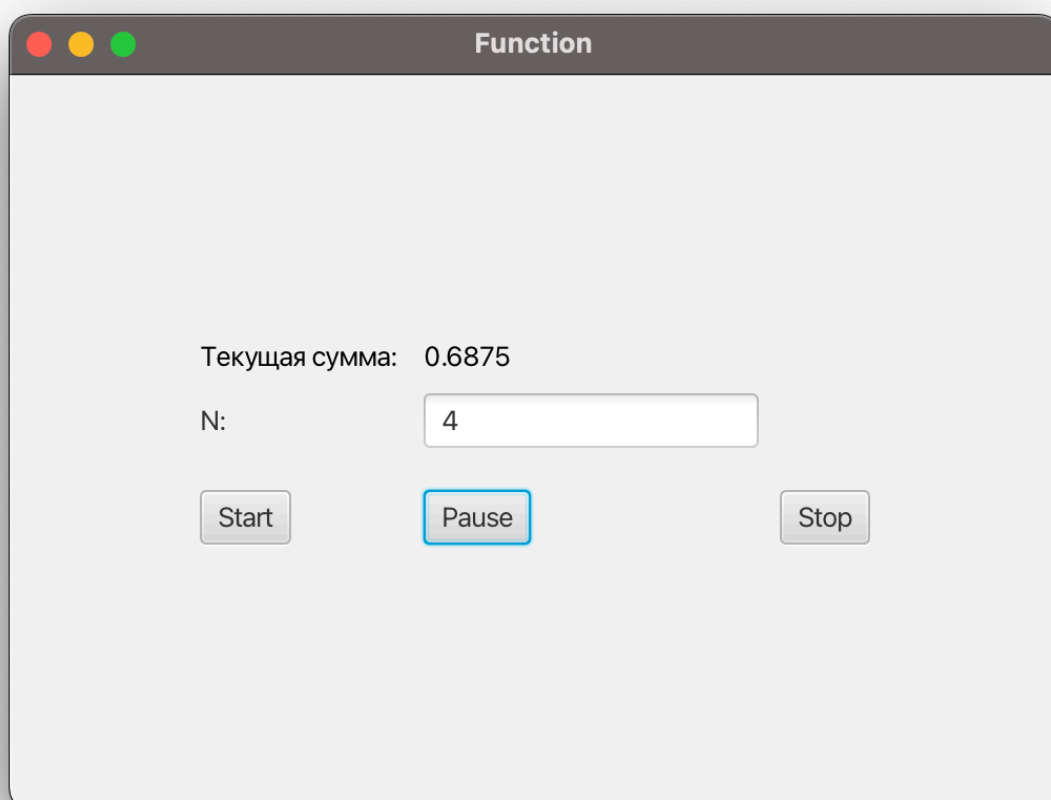
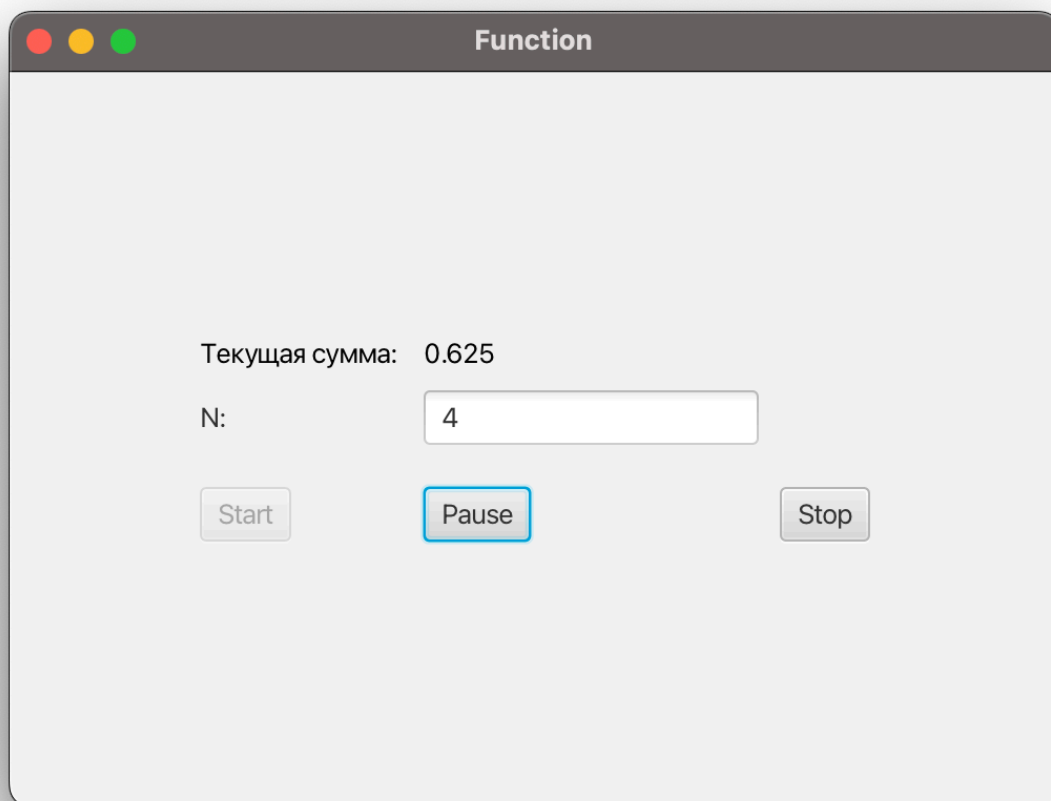
```

```
this.currentValue = 0;

this.sum.setText("");
this.currentIteration = 0;
inputCount.setText("");
backgroundThread = null;
}

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





**Вывод:** приобрел навыки написания простого оконного многопоточного приложения с использованием Java API

