

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра интеллектуальных информационных технологий

Тема: «Многопоточные оконные приложения»

Отчёт лабораторной работы №2
по дисциплине «Современные платформы программирования»
за II семестр

Выполнил:
студент 3-его курса
VI-го семестр
факультета ЭИС
группы ПО-4(1)
зачётная книжка №190333
Галанин П. И.
«__» _____ 2022 г.

Проверил:
ассистент
кафедры ИИТ
Монтик Н. С.
«__» _____ 2022 г.

Отчёт лабораторной работы №2

Тема: «Многопоточные оконные приложения»

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

Что нужно сделать:

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.

Вариант 5:
$$\sum_{k=1}^n \frac{1}{(2k-1)(2k+1)} = \frac{1}{1*3} + \frac{1}{3*5} + \dots + \frac{1}{(2n-1)(2n+1)}$$

Разработка дизайна:

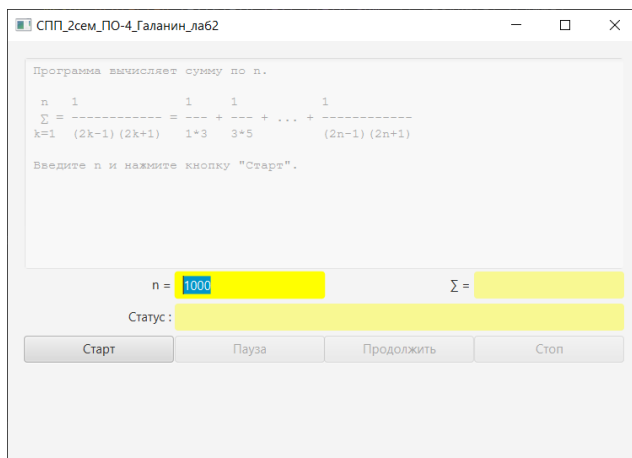


Рисунок 1 – Ввод

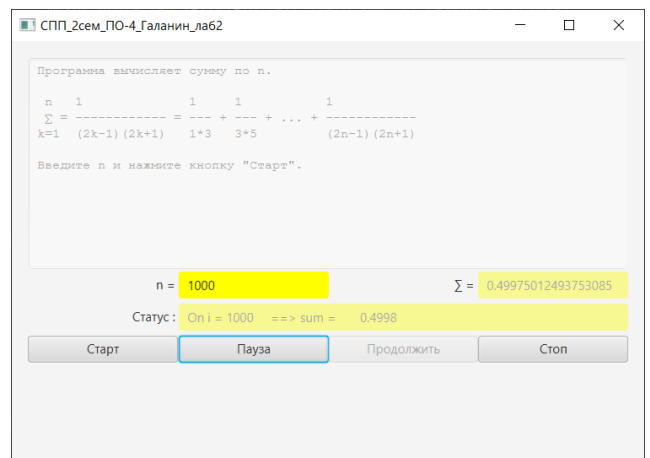


Рисунок 2 – Вывод

Исходный код:

- **Main.java** - Основная функция программы. Вызывает главное окно.
- **MainApplication.java** - Вызов главного окна. Подключаем дизайн. Задаем заголовок окну.
- **MainController.java** - Контролер главного окна. Управление кнопок и текстовых полей. Создания потока с задачей суммы.
- **CalcTask.java** - Задача расчёта суммы.
- **main-view.fxml** - Дизайн главного окна

Листинг: src/main/java/com/example/.../Main.java

```
1 package com.example.spp_2sem_po4_galanin_lab2;
2
3 public class Main {
4     public static void main(String[] args) {
5         MainApplication.main(args);
6     }
7 }
```

Листинг: src/main/java/com/example/.../MainApplication.java

```
1 package com.example.spp_2sem_po4_galanin_lab2;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.scene.Scene;
6 import javafx.stage.Stage;
7
8 import java.io.IOException;
9
10 public class MainApplication extends Application {
11     @Override
12     public void start(Stage stage) throws IOException {
13         FXMLLoader fxmlLoader = new
14         FXMLLoader(MainApplication.class.getResource("main-view.fxml"));
15         Scene scene = new Scene(fxmlLoader.load());
16         stage.setTitle("СПИ_2сем_Ю-4_Галанин_лаб2");
17         stage.setScene(scene);
18         stage.show();
19     }
20
21     public static void main(String[] args) {
22         launch();
23     }
24 }
```

Листинг: src/main/java/com/example/.../MainController.java

```
1 package com.example.spp_2sem_po4_galanin_lab2;
2
3 import javafx.fxml.FXML;
4 import javafx.scene.control.Button;
5 import javafx.scene.control.TextField;
6
7 public class MainController {
8     @FXML protected TextField text_field_n; // До какой итерации считать сумму
9     @FXML protected TextField text_field_sum; // Результат (сумма)
10    @FXML protected TextField text_field_status; // Статус задачи
11    @FXML protected Button button_start;
12    @FXML protected Button button_pause;
13    @FXML protected Button button_resume;
14    @FXML protected Button button_stop;
15    protected Thread thread; // Поток
16    protected CalcTask task; // Задача
17
18    // Функция, которая начинает вычислять сумму
19    @FXML protected void onStartButtonClick() {
20        button_start.setDisable(true);
21        button_pause.setDisable(false);
22        button_resume.setDisable(true);
23        button_stop.setDisable(false);
24
25        // Получаем число итераций суммы из текстового поля
```

```

26         int n = 0;
27         try {
28             n = Integer.parseInt(text_field_n.getText());
29         }
30         catch (Exception e) {
31             System.out.println(e);
32         }
33
34         // Создаем задачу
35         task = new CalcTask(n);
36
37         // Соединяем текстовое поле с задачей
38         // В текстовое поле будут выводиться сообщения
39         text_field_status.textProperty().bind(task.messageProperty());
40
41         // Функция, при старте задачи
42         task.setOnRunning((succeesesEvent) -> {
43             button_start.setDisable(true);
44             text_field_sum.setText("");
45         });
46
47         // Функция, при завершении задачи
48         task.setOnSucceeded((succeesesEvent) -> {
49             text_field_sum.setText(task.getValue().toString());
50             button_start.setDisable(false);
51         });
52
53         // Создаём поток из задачи
54         thread = new Thread(task);
55
56         // Стартуем поток
57         thread.start();
58     }
59
60     // Функция, которая ставит вычисление на паузу
61     @FXML protected void onButtonPauseClick() {
62         button_start.setDisable(true);
63         button_pause.setDisable(true);
64         button_resume.setDisable(false);
65         button_stop.setDisable(false);
66         task.isPause = true;
67     }
68
69     // Функция, которая продолжает вычисление после паузы
70     @FXML protected void onButtonResumeClick() {
71         button_start.setDisable(true);
72         button_pause.setDisable(false);
73         button_resume.setDisable(true);
74         button_stop.setDisable(false);
75         task.isPause = false;
76     }
77
78     // Функция, которая завершает поток вычисления суммы
79     @FXML protected void onButtonStopClick() {
80         button_start.setDisable(false);
81         button_pause.setDisable(true);
82         button_resume.setDisable(true);
83         button_stop.setDisable(true);
84         thread.stop();
85     }
86 }

```

Листинг: src/main/java/com/example/.../CalcTask.java

```
1 package com.example.spp_2sem_po4_galanin_lab2;
2
3 import javafx.concurrent.Task;
4
5 public class CalcTask extends Task<Double> {
6     protected int n; // Сумма от 1 до n
7     public boolean isPause = false; // Вычисление на паузе или нет?
8
9     // Конструктор, который указывает число n
10    public CalcTask(int n) {
11        this.n = n;
12    }
13
14    // Функция, которая считает один элемент a_i
15    protected double calculate_a(int i) {
16        return (double) 1 / ((2 * i - 1) * (2 * i + 1));
17    }
18
19    // Функция, которая считает сумму
20    protected double calculate_sum(int n) {
21        double sum = 0;
22        int i = 1;
23        while (true) {
24            if (isPause == true) { // Задача на паузе?
25                continue;
26            }
27
28            sum += calculate_a(i);
29
30            // Выводим результат в сообщение таски.
31            // Сообщение таски будет выводить в TextEdit как статус бары
32            String str = String.format("On i = %-8d ==> sum = %12.4f", i, sum);
33            updateMessage(str);
34
35            // Спим 10 миллисекунд, чтобы поток не выполнялся очень быстро
36            sleep(10);
37
38            if (i < n == false) { // Дошли до последнего элемента суммы?
39                break;
40            }
41
42            i += 1;
43        }
44        return sum;
45    }
46
47    // Функция, которая спит время time
48    protected void sleep(int time) {
49        try {
50            Thread.sleep(time);
51        }
52        catch (Exception e) {
53            System.out.println(e);
54        }
55    }
56
57    // Функция, которая возвращает результат суммы после выполнения таски
58    @Override
59    protected Double call() throws Exception {
60        double result = calculate_sum(n);
61        return result;
62    }
```

Листинг: src/main/resources/com/example/.../main-view.fxml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.geometry.Insets?>
4  <?import javafx.scene.control.Button?>
5  <?import javafx.scene.control.Label?>
6  <?import javafx.scene.control.TextArea?>
7  <?import javafx.scene.control.TextField?>
8  <?import javafx.scene.layout.ColumnConstraints?>
9  <?import javafx.scene.layout.GridPane?>
10 <?import javafx.scene.layout.RowConstraints?>
11 <?import javafx.scene.layout.VBox?>
12 <?import javafx.scene.text.Font?>
13
14
15 <VBox maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
    prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/18"
    xmlns:fx="http://javafx.com/fxml/1"
    fx:controller="com.example.spp_2sem_po4_galanin_lab2.MainController">
16 <children>
17     <GridPane>
18         <columnConstraints>
19             <ColumnConstraints halignment="RIGHT" hgrow="SOMETIMES" maxWidth="200.0" minWidth="10.0"
20                 prefWidth="100.0" />
21             <ColumnConstraints hgrow="SOMETIMES" maxWidth="200.0" minWidth="10.0" prefWidth="100.0"
22                 />
23             <ColumnConstraints halignment="RIGHT" hgrow="SOMETIMES" maxWidth="200.0"
24                 minWidth="10.0" prefWidth="100.0" />
25             <ColumnConstraints hgrow="SOMETIMES" maxWidth="200.0" minWidth="10.0"
26                 prefWidth="100.0" />
27         </columnConstraints>
28         <rowConstraints>
29             <RowConstraints minHeight="10.0" prefHeight="200.0" vgrow="SOMETIMES" />
30             <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
31             <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
32             <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
33         </rowConstraints>
34         <children>
35             <TextArea disable="true" prefHeight="200.0" prefWidth="200.0" text="Программа вычисляет
36                 сумму по n.&#10;&#10; n      1      1      1      1&#10; ∑ = ----- = --- +
37                 --- + ... + -----&#10;k=1  (2k-1)(2k+1)  1*3    3*5
38                 (2n-1)(2n+1)&#10;&#10;Введите n и нажмите кнопку &quot;Старт&quot;." GridPane.columnSpan="4">
39                 <font>
40                     <Font name="Courier New" size="12.0" />
41                 </font>
42             </TextArea>
43             <Label text="n = " GridPane.rowIndex="1" />
44             <TextField fx:id="text_field_n" style="-fx-background-color: yellow;" text="1000"
45                 GridPane.columnIndex="1" GridPane.rowIndex="1" />
46             <Label text="∑ = " GridPane.columnIndex="2" GridPane.rowIndex="1" />
47             <TextField fx:id="text_field_sum" disable="true" style="-fx-background-color: yellow;"
48                 GridPane.columnIndex="3" GridPane.rowIndex="1" />
49             <Label text="Старт : " GridPane.rowIndex="2" />
50             <TextField fx:id="text_field_status" disable="true" prefWidth="414.0"
51                 style="-fx-background-color: yellow;" GridPane.columnIndex="1" GridPane.columnSpan="3"
52                 GridPane.rowIndex="2" />
53             <Button fx:id="button_start" mnemonicParsing="false" onAction="#onButtonStartClick"
54                 prefWidth="200.0" text="Старт" GridPane.rowIndex="3" />
55             <Button fx:id="button_pause" disable="true" mnemonicParsing="false"
56                 onAction="#onButtonPauseClick" prefWidth="200.0" text="Пауза" GridPane.columnIndex="1"
57                 GridPane.rowIndex="3" />

```

```

44         <Button fx:id="button_resume" disable="true" mnemonicParsing="false"
onAction="#onButtonResumeClick" prefWidth="200.0" text="Продолжить" GridPane.columnIndex="2"
GridPane.rowIndex="3" />
45         <Button fx:id="button_stop" disable="true" mnemonicParsing="false"
onAction="#onButtonStopClick" prefWidth="200.0" text="Стоп" GridPane.columnIndex="3"
GridPane.rowIndex="3" />
46     </children>
47 </GridPane>
48 </children>
49 <padding>
50     <Insets bottom="16.0" left="16.0" right="16.0" top="16.0" />
51 </padding>
52 </VBox>

```

Вывод: Реализовали оконное приложение с многопоточностью: основной поток - GUI; другой поток - задача, которая вычисляет сумму ряда. Реализовали вывод промежуточных результатов в TextEdit (как статус бар). Реализовали кнопки старта потока, приостановки и продолжения задачи, завершения потока.

Список использованных источников:

1. Как создать исполняемый jar файл в IntelliJ IDEA - YouTube [Электронный ресурс] - Режим доступа: https://www.youtube.com/watch?v=tA8rEz_xFrQ. Дата доступа: 01.05.2022.
2. Setup IntelliJ IDEA (2021) for JavaFX & SceneBuilder and Create Your First JavaFX Application - YouTube [Электронный ресурс] - Режим доступа: <https://www.youtube.com/watch?v=ZfaPMLdgJxQ>. Дата доступа: 01.05.2022.
3. JavaFX - Opening an FXML file in New Window - YouTube [Электронный ресурс] - Режим доступа: https://www.youtube.com/watch?v=ZzwvQ6pa_tk. Дата доступа: 01.05.2022.
4. How To Fix JavaFX runtime components are missing and are required to run this application - YouTube [Электронный ресурс] - Режим доступа: https://www.youtube.com/watch?v=sdkW_cUH3hw. Дата доступа: 01.05.2022.
5. Export JavaFX 11, 15 or 17 projects into an executable jar file with IntelliJ [2022] - YouTube [Электронный ресурс] - Режим доступа: <https://www.youtube.com/watch?v=F8ahBtXkQzU>. Дата доступа: 01.05.2022.