Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет» Кафедра ИИТ

Лабораторная работа №8

Выполнил:

Студент группы ПО-5

Пищик А. В.

Проверил:

Крощенко А. А.

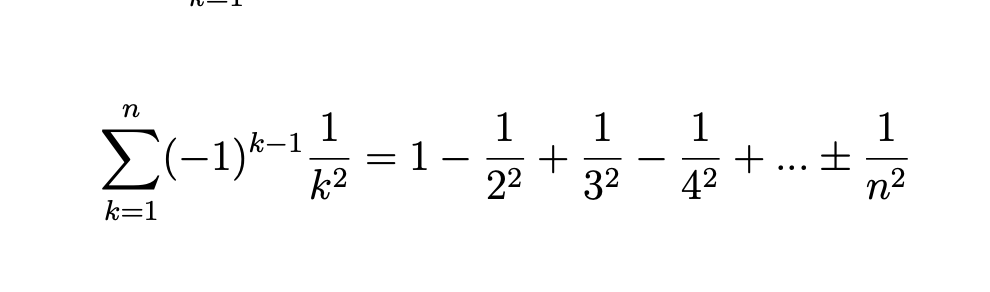
Брест 2022

**Вариант 11**

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

**Задание**

Разработать оконное приложение с использованием Java API, использующее один вспомогатель- ный поток, вычисляющий заданную сумму и выполняющий вывод результата вычисления (как ко- нечный, 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.



**Текст программы:**

**HelloApplication**

package com.example.lab1;  
  
import javafx.application.Application;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Scene;  
import javafx.stage.Stage;  
import java.io.IOException;  
  
public class HelloApplication extends Application {  
 @Override  
 public void start(Stage stage) throws IOException {  
 FXMLLoader fxmlLoader = new FXMLLoader(HelloApplication.class.getResource("lab1.fxml"));  
 Scene scene = new Scene(fxmlLoader.load(), 520, 380);  
 stage.setTitle("Lab1-2!");  
 stage.setScene(scene);  
 stage.show();  
 }  
  
 public static void main(String[] args) {  
 *launch*();  
 }  
}

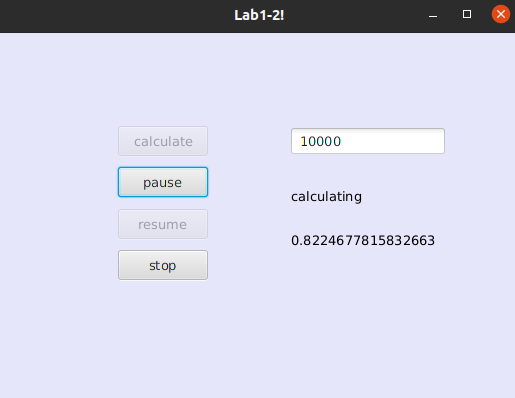
**Calculator**

package com.example.lab1;  
  
import javafx.application.Platform;  
  
public class Calculator extends Thread {  
 private int n;  
 private Double sum = 0.0;  
 private final HelloController controller;  
  
 public Calculator(int n, HelloController controller) {  
 this.n = n;  
 this.controller = controller;  
 }  
  
 @Override  
 public void run() {  
 int k = 1;  
 try {  
 while (k < n && !isInterrupted()) {  
 Platform.*runLater*(() -> controller.updateResult(sum));  
 double s = Math.*pow*(-1, k - 1) \* 1 / Math.*pow*(k, 2);  
 sum += s;  
 k++;  
 Platform.*runLater*(() -> controller.updateResult(sum));  
  
 *sleep*(10);  
  
 if(k == n) {  
 controller.pause.setDisable(true);  
 controller.resume.setDisable(true);  
 controller.stop.setDisable(true);  
 controller.isRunning = false;  
 controller.textFildN.setText("Enter n");  
 controller.calc.setDisable(false);  
 }  
 }  
 } catch (InterruptedException exception) {  
 System.*out*.println("Interrupted");  
 }  
 }  
}

**HelloController**

package com.example.lab1;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.TextField;  
import javafx.scene.text.Text;  
  
  
public class HelloController {  
 private Thread calculator;  
 boolean isRunning;  
  
 @FXML  
 public TextField textFildN;  
  
 @FXML  
 public Button calc;  
  
 @FXML  
 public Button resume;  
  
 @FXML  
 public Button pause;  
  
 @FXML  
 public Button stop;  
  
 @FXML  
 private Text result;  
  
 @FXML  
 public Text message;  
  
 @FXML  
 void initialize() {  
  
 }  
  
 public void calcClick(ActionEvent actionEvent) {  
  
 try {  
 int n = Integer.*parseInt*(textFildN.getText());  
  
 calculator = new Calculator(n, this);  
  
 calculator.start();  
 isRunning = true;  
  
 message.setText("calculating");  
 pause.setDisable(false);  
 calc.setDisable(true);  
 resume.setDisable(true);  
  
 } catch (NumberFormatException e) {  
 message.setText("enter the number");  
 }  
 }  
  
 public void pauseClick(ActionEvent actionEvent) throws InterruptedException {  
  
 if (isRunning) {  
 calculator.suspend();  
 message.setText("pause");  
 calc.setDisable(true);  
 resume.setDisable(false);  
 pause.setDisable(true);  
 isRunning = false;  
 }  
 }  
  
 public void resumeClick(ActionEvent actionEvent) {  
  
 if (!isRunning) {  
 message.setText("calculating");  
 calculator.resume();  
 resume.setDisable(true);  
 pause.setDisable(false);  
 isRunning = true;  
 }  
 }  
  
 public void stopClick(ActionEvent actionEvent) {  
  
 calculator.interrupt();  
 message.setText("stopped");  
 textFildN.setText("Enter n");  
 calc.setDisable(false);  
 pause.setDisable(true);  
 resume.setDisable(true);  
 }  
  
 public void updateResult(double sum) {  
 result.setText(String.*valueOf*(sum));  
 }  
}

**Результат выполнения:**



****

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