Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет» Кафедра ИИТ

Лабораторная работа №8

По дисциплине «СПП» за 6 семестр

Выполнил:

Студент группы ПО-3

Кабачук Д. С.

Проверил:

Крощенко А. А.

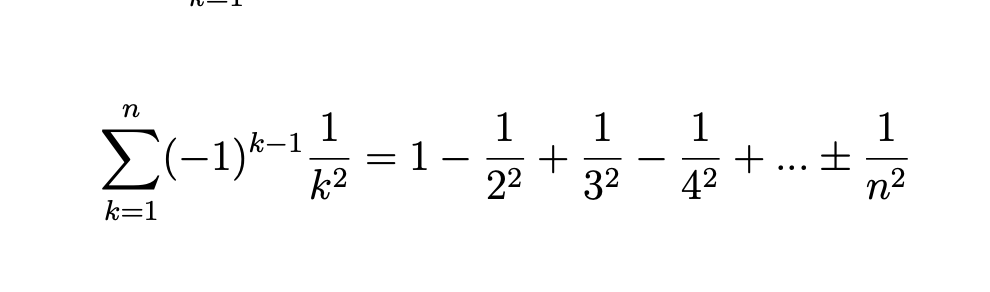
Брест 2021

**Вариант 1**

**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:**

**Main**

**package** sample;  
  
**import** javafx.application.Application;  
**import** javafx.fxml.FXMLLoader;  
**import** javafx.scene.Parent;  
**import** javafx.scene.Scene;  
**import** javafx.stage.Stage;  
  
**public class** Main **extends** Application {  
  
 @Override  
 **public void** start(Stage primaryStage) **throws** Exception {  
 Parent root = FXMLLoader.*load*(getClass().getResource(**"sample.fxml"**));  
 primaryStage.setTitle(**"Number series"**);  
 primaryStage.setScene(**new** Scene(root, 520, 380));  
 primaryStage.show();  
 }  
  
  
 **public static void** main(String[] args) {  
 *launch*(args);  
 }  
}

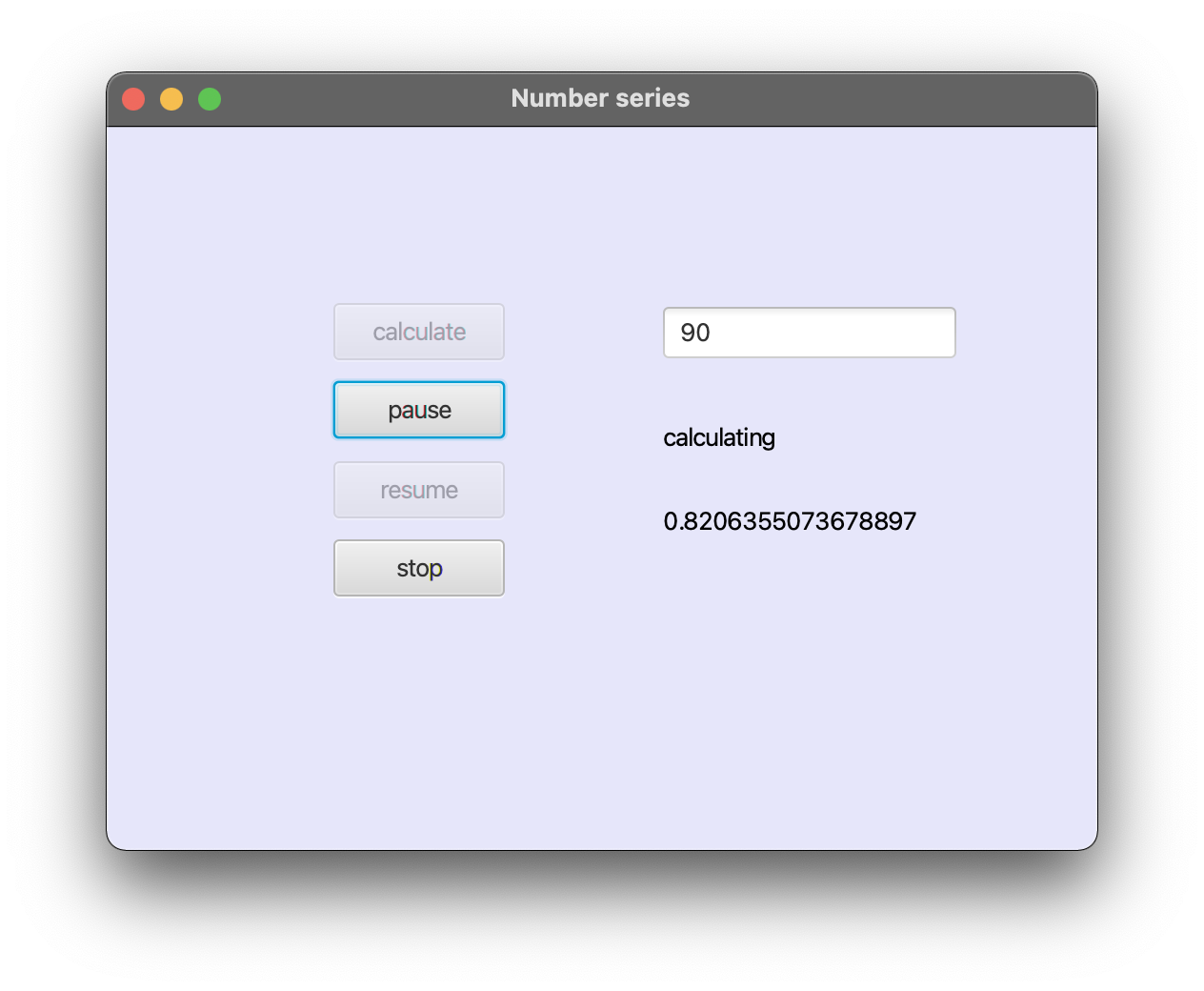
**Calculator**

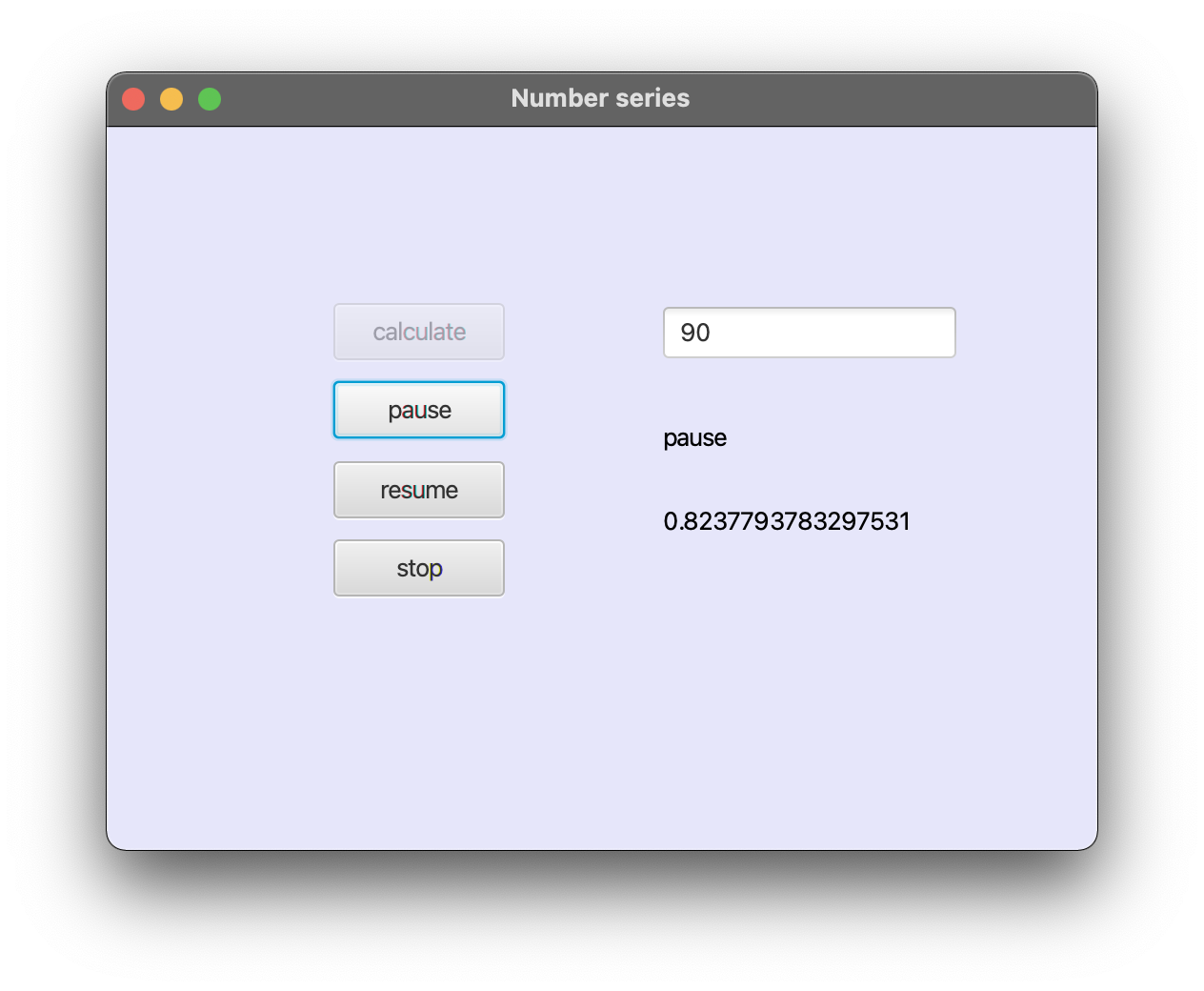
**package** sample;  
  
**import static** java.lang.Math.*pow*;  
  
**public class** Calculator **extends** Thread {  
  
 **private int n**;  
 **private final** Controller **controller**;  
  
 **public** Calculator(**int** n, Controller controller) {  
 **this**.**n** = n;  
 **this**.**controller** = controller;  
 }  
  
 @Override  
 **public void** run() {  
 **int** k = 1;  
 **double** sum = 0;  
 **try** {  
 **while** (k < **n** && !isInterrupted()) {  
 **controller**.updateResult(sum);  
 **double** s = *pow*(-1, k - 1) \* 1 / *pow*(k, 2);  
 sum += s;  
 k++;  
 **controller**.updateResult(sum);  
  
 *sleep*(300);  
 }  
 } **catch** (InterruptedException exception) {  
 System.***out***.println(**"Interrupted"**);  
 }  
 }  
}

**Controller**

**package** sample;  
  
**import** javafx.event.ActionEvent;  
**import** javafx.fxml.FXML;  
**import** javafx.scene.control.Button;  
**import** javafx.scene.control.TextField;  
**import** javafx.scene.text.Text;  
  
**import** java.util.ResourceBundle;  
  
**public class** Controller {  
  
 **private** Thread **calculator**;  
 **boolean isRunning**;  
 @FXML  
 **private** ResourceBundle **resources**;  
  
 @FXML  
 **private** TextField **textFildN**;  
  
 @FXML  
 **private** Button **calc**;  
  
 @FXML  
 **private** Button **resume**;  
  
 @FXML  
 **private** Button **pause**;  
  
 @FXML  
 **private** Button **stop**;  
  
 @FXML  
 **private** Text **result**;  
  
 @FXML  
 **private** 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**);  
 **isRunning** = **false**;  
 }  
 }  
  
 **public void** resumeClick(ActionEvent actionEvent) {  
  
 **if** (!**isRunning**) {  
 **message**.setText(**"calculating"**);  
 **calculator**.resume();  
 **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.