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

Ковалёва А. И.

Проверил:

Крощенко А. А.

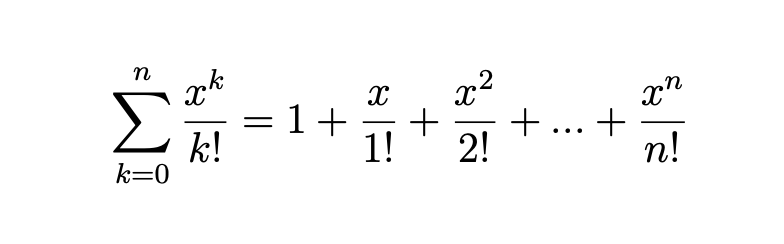
Брест 2021

**Вариант 12**

**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(**"Hello World"**);  
 primaryStage.setScene(**new** Scene(root, 550, 350)); primaryStage.show();  
 }  
  
 **public static void** main(String[] args) {  
 *launch*(args);  
 }  
}

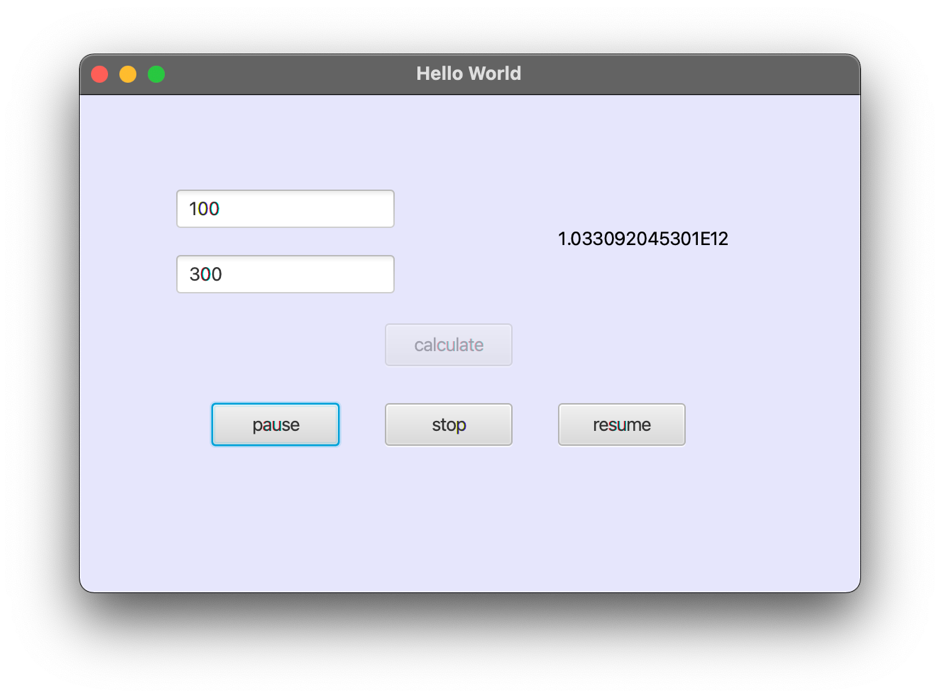
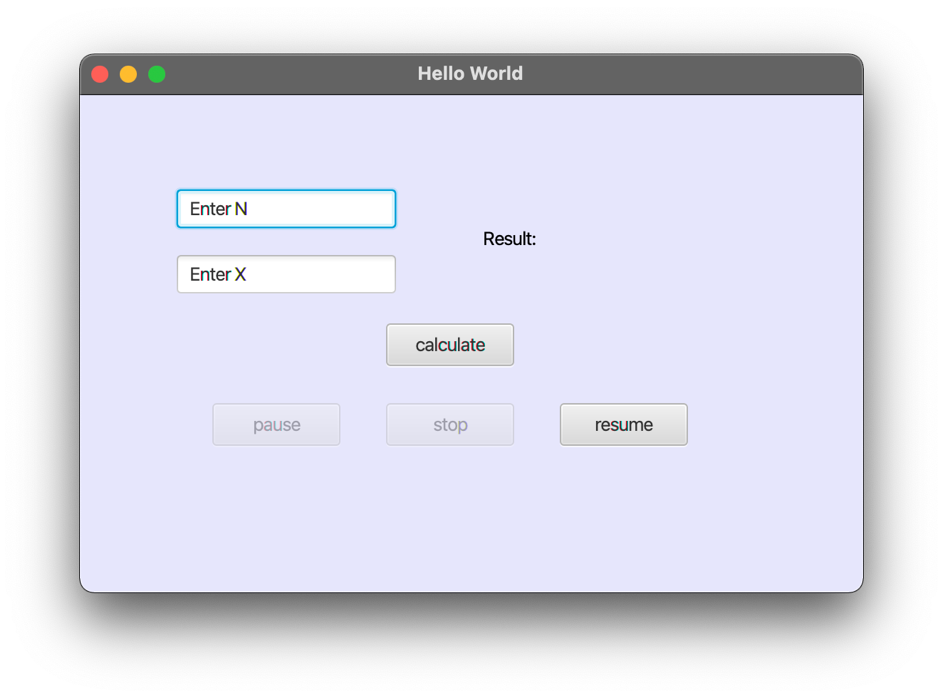
**Calculator**

**package** sample;  
  
**import static** java.lang.Math.*pow*;  
  
**public class** Calculator **extends** Thread {  
  
 **private int N**;  
 **private int X**;  
 **private final** Controller **controller**;  
  
 **public** Calculator( **int** N, **int** X, Controller controller) { **this**.**N** = N;  
 **this**.**X** = X;  
 **this**.**controller** = controller; }  
 @Override  
 **public void** run() {  
 **double** a = 1.;  
 **double** sum = a;  
 **int** i = 1;  
 **try** {  
 **while** (i < **N** && !isInterrupted()) {  
 **controller**.updateResult(sum);  
 a = a \* ((**double**) **X** / i);  
 sum += a;  
 i++;  
 **synchronized** (**controller**) {  
 **controller**.updateResult(sum);  
 }  
 Thread.*sleep*(500);  
 }  
 } **catch** (InterruptedException e) {  
 Thread.*currentThread*().interrupt();  
 e.printStackTrace();  
 }  
 }  
}

**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** Calculator **calculator**;  
 **private boolean isRunning** = **false**;  
  
 @FXML  
 **private** TextField **textFieldX**, **textFieldN**;  
  
 @FXML  
 **private** Button **startBut**, **pauseBut**, **stopBut**, **resumeBut**;  
  
 @FXML  
 **private** Text **resultText**, **errorText**;  
  
 @FXML  
 **void** initialize() {  
 **pauseBut**.setDisable(**true**);  
 **stopBut**.setDisable(**true**);  
 }  
  
 @FXML  
 **void** startClick() {  
 **try** {  
 **int** N = Integer.*parseInt*(**textFieldN**.getText());  
 **int** X = Integer.*parseInt*(**textFieldX**.getText());  
 **calculator** = **new** Calculator(N, X, **this**);  
 **startBut**.setDisable(**true**);  
 **pauseBut**.setDisable(**false**);  
 **stopBut**.setDisable(**false**);  
 **resumeBut**.setDisable(**true**);  
 **errorText**.setText(**""**);  
 **isRunning** = **true**;  
 **calculator**.start();  
 } **catch** (NumberFormatException e) {  
 **errorText**.setText(**"Format error! Please, enter an integer number."**);  
 }  
 }  
  
 @FXML  
 **synchronized void** pauseClick() **throws** InterruptedException {  
 **if** (**isRunning**) {  
 **resumeBut**.setDisable(**false**);  
 **calculator**.suspend();  
 **isRunning** = **false**;  
 }  
 }  
  
 @FXML  
 **synchronized void** resumeClick() {  
 **if** (!**isRunning**) {  
 **isRunning** = **true**;  
 **calculator**.resume();  
 **resumeBut**.setDisable(**true**);  
 }  
 }  
  
 @FXML  
 **void** stopClick() {  
 **startBut**.setDisable(**false**);  
 **pauseBut**.setDisable(**true**);  
 **stopBut**.setDisable(**true**);  
 **resumeBut**.setDisable(**true**);  
 **calculator**.interrupt();  
 }  
  
 **void** updateResult(**double** sum) {  
 **resultText**.setText(Double.*toString*(sum));  
 }}

**Результат выполнения:**

****

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