

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЯ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №8

Специальность ПОЗ

Выполнила
Гаврилюк Р. И., студентка
группы ПОЗ

Проверил
Крощенко А. А., ст. преп.
Кафедры ИИТ

Брест 2021

Вариант 6

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

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.

6)

$$\sum_{k=0}^n \frac{x^{2k}}{(2k)!} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{x^{2n}}{(2n)!}$$

Текст программы:

Main.java

```
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("Lab 8 SPP Hauryliuk");
        primaryStage.setScene(new Scene(root, 300, 275));
        primaryStage.show();
    }

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

Calculator.java

```
package sample;

public class Calculator extends Thread{
```

```

private final Controller controller;
private double X;
private int N;

public Calculator(int X, int N, Controller controller) {
    this.X = X;
    this.N = N;
    this.controller = controller;
}

@Override
public void run() {
    double a = 1.0D;
    double sum = a;

    try {
        for (int i = 1, k = 1; i < N && !isInterrupted(); ++i, ++k) {
            controller.updateResult(sum);
            a = (a * X * X) / (double) (k * (k + 1));
            sum += a;

            Thread.sleep(500);
        }
        controller.updateResult(sum);
    } catch (InterruptedException e) {
        Thread.currentThread().interrupt();
        e.printStackTrace();
    }

    controller.reinitialize();
}
}

```

Controller.java

```

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;

public class Controller {
    @FXML
    public TextField textFieldX;
    @FXML
    public TextField textFieldN;
    @FXML
    public Button buttonStart;
    @FXML
    public Button buttonStop;
    @FXML
    public Button buttonPause;
    @FXML
    public Button buttonResume;
    @FXML
    public Text textFieldResult;
    @FXML
    public Text textFieldError;
}

```

```
private Calculator calculator;
```

```
@FXML
```

```
void initialize() {  
    buttonStart.setVisible(true);  
    buttonStop.setVisible(false);  
    buttonPause.setVisible(false);  
    buttonResume.setVisible(false);  
  
    textFieldError.setText("");  
    textFieldResult.setText("");  
    textFieldX.setText("");  
    textFieldN.setText("");  
}
```

```
public void reinitialize() {  
    buttonStart.setVisible(true);  
    buttonStop.setVisible(false);  
    buttonPause.setVisible(false);  
    buttonResume.setVisible(false);  
  
    textFieldError.setText("");  
}
```

```
@FXML
```

```
public void startClick(ActionEvent actionEvent) {  
    try {  
        buttonStop.setVisible(true);  
        buttonPause.setVisible(true);  
        textFieldError.setText("");  
        textFieldResult.setText("");  
  
        int N = Integer.parseInt(textFieldN.getText());  
        int X = Integer.parseInt(textFieldX.getText());  
        calculator = new Calculator(X, N, this);  
        calculator.start();  
    } catch (NumberFormatException e) {  
        initialize();  
        textFieldError.setText("Format error! Please, enter an integer number.");  
    }  
}
```

```
@FXML
```

```
public void pauseClick(ActionEvent actionEvent) {  
    buttonPause.setVisible(false);  
    buttonResume.setVisible(true);  
    calculator.suspend();  
}
```

```
@FXML
```

```
public void stopClick(ActionEvent actionEvent) {  
    initialize();  
    calculator.interrupt();  
}
```

```
@FXML
```

```
public void resumeClick(ActionEvent actionEvent) {
```

```

        buttonPause.setVisible(true);
        buttonResume.setVisible(false);
        calculator.resume();
    }

    public void updateResult(double sum) {
        textFieldResult.setText(Double.toString(sum));
    }
}

```

sample.fxml

```

<?import javafx.scene.layout.GridPane?>

<?import javafx.scene.layout.AnchorPane?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.text.Font?>
<?import javafx.scene.text.Text?>

<GridPane fx:controller="sample.Controller"
    xmlns:fx="http://javafx.com/fxml" alignment="center" hgap="10" vgap="10"
    prefHeight="300.0" prefWidth="500.0">
    <AnchorPane>
        <Text layoutX="10.0" layoutY="25.0" strokeType="OUTSIDE" strokeWidth="0.0"
            text="Enter X"/>
        <TextField fx:id="textFieldX" layoutX="70.0" layoutY="5.0"/>

        <Text layoutX="10.0" layoutY="55.0" strokeType="OUTSIDE" strokeWidth="0.0"
            text="Enter N"/>
        <TextField fx:id="textFieldN" layoutX="70.0" layoutY="35.0"/>

        <Button fx:id="buttonStart" layoutX="10.0" layoutY="85.0" mnemonicParsing="false"
            onAction="#startClick"
            prefHeight="30.0" prefWidth="230.0" text="Start"/>
        <Button fx:id="buttonStop" layoutX="10.0" layoutY="85.0" mnemonicParsing="false"
            onAction="#stopClick"
            prefHeight="30.0" prefWidth="230.0" text="Stop"/>
        <Button fx:id="buttonPause" layoutX="10.0" layoutY="125.0" mnemonicParsing="false"
            onAction="#pauseClick"
            prefHeight="30.0" prefWidth="230.0" text="Pause"/>
        <Button fx:id="buttonResume" layoutX="10.0" layoutY="125.0"
            mnemonicParsing="false" onAction="#resumeClick"
            prefHeight="30.0" prefWidth="230.0" text="Resume"/>

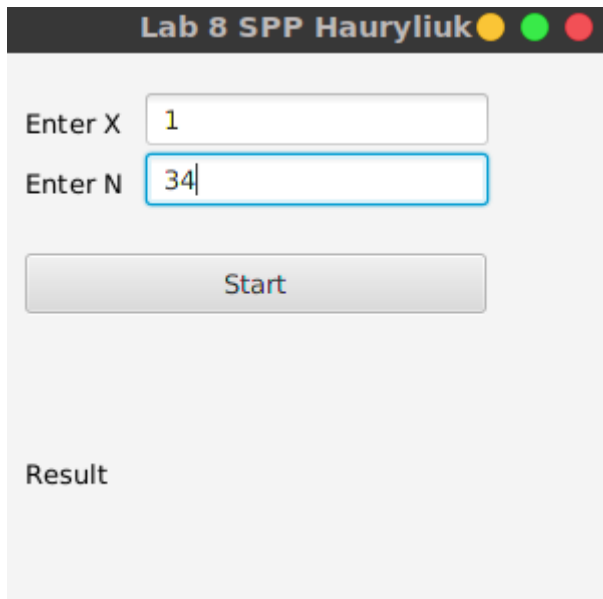
        <Text layoutX="10.0" layoutY="200.0" strokeType="OUTSIDE" strokeWidth="0.0"
            text="Result"/>
        <Text fx:id="textFieldResult" layoutX="70.0" layoutY="200.0" strokeType="OUTSIDE"
            strokeWidth="0.0"
            wrappingWidth="230"/>

        <Text fx:id="textFieldError" fill="#e80808" layoutX="10.0" layoutY="240.0"
            strokeType="OUTSIDE"
            strokeWidth="0.0" wrappingWidth="230">
            <font>
                <Font size="15.0"/>
            </font>
        </Text>
    </AnchorPane>
</GridPane>

```

```
</AnchorPane>  
</GridPane>
```

Тестирование:

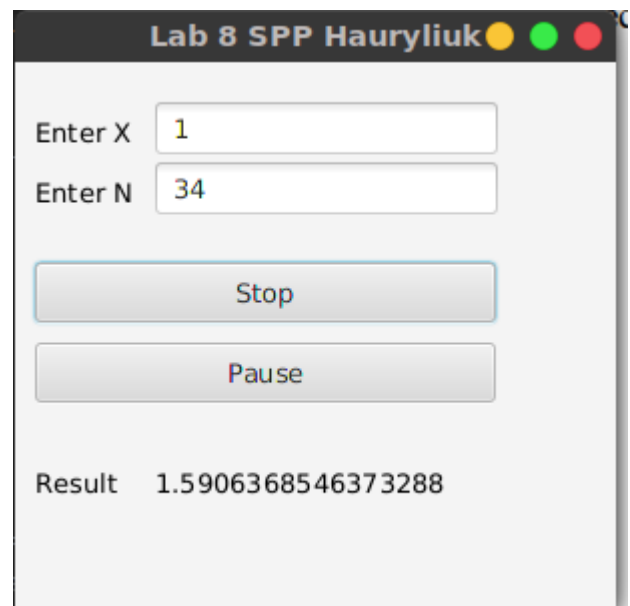


Lab 8 SPP Hauryliuk

Enter X

Enter N

Result

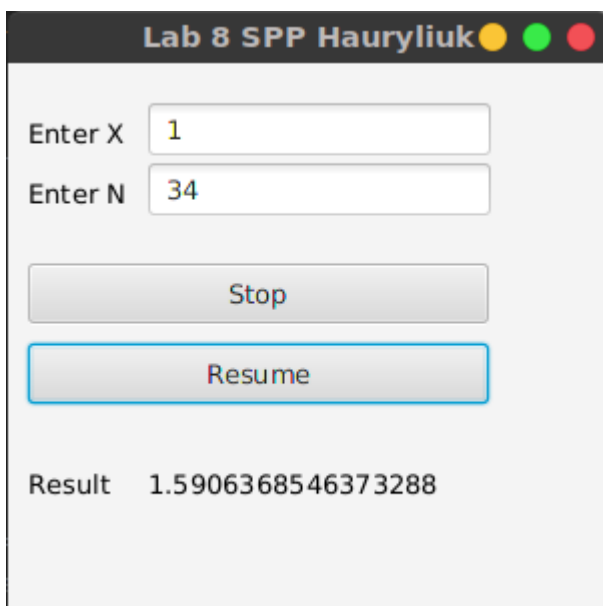


Lab 8 SPP Hauryliuk

Enter X

Enter N

Result 1.5906368546373288



Lab 8 SPP Hauryliuk

Enter X

Enter N

Result 1.5906368546373288

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