

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
КАФЕДРА ИИТ

Лабораторная работа №2
по дисциплине «Современные платформы программирования»

Выполнила:
Андросюк М.М.
Группа: ПО-5
Проверил:
Крощенко А.А.

Брест 2022

Цель работы: приобрести навыки написания простого оконного многопоточного приложения 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.

Текст программы

```
package com.example.demo;

import
javafx.application.Platform;
import javafx.fxml.FXML;
import javafx.scene.control.Label;
import
javafx.scene.control.TextField;

public class HelloController {
    @FXML
    private Label
resultText; @FXML
    private Label
messageText; @FXML
    private TextField valueText;

    private Thread
thread; private
Double result;
    final Object lock = new Object();
    boolean paused;
    int value;

    @FXML
    protected void
onStartButtonClick() { if
(thread != null) {
        thread.interrupt();
    }

    try {
        value = Integer.parseInt(valueText.getText());

        if (value < 0) {
            throw new IllegalArgumentException();
        }
    }
    catch (IllegalArgumentException e) {
```

```

        messageText.setText("Invalid input");

        return;
    }

    result = 0.0;
    paused = false;
    messageText.setText("Started");

    thread = new Thread(() ->
    { double previousTerm =
      1.0;

      for (int i = 0; i <= value;
        i++) { synchronized (lock)
        {
          try {
            Thread.sleep(1000);

            if (paused) {
              lock.wait();
            }

            previousTerm = i == 0 ? 1.0 : previousTerm /
              i; result += previousTerm;

            Platform.runLater(() -> resultText.setText(result.toString()));
          } catch (InterruptedException e) {
            break;
          }
        }
      }
    });

    thread.start();
}

@FXML
protected void onPauseButtonClick() {
    if (thread != null && thread.isAlive())
    { paused = true;
      messageText.setText("Paused");
    }
}

@FXML
protected void onResumeButtonClick() {
    if (paused && thread != null && thread.isAlive()) {
        synchronized (lock) {
            paused = false;
            lock.notify();
            messageText.setText("Resumed");
        }
    }
}

```

```

    }

@FXML
protected void
    onInterruptButtonClick() { if
        (thread != null) {
            thread.interrupt();
            messageText.setText("Stopped");
        }
    }
}

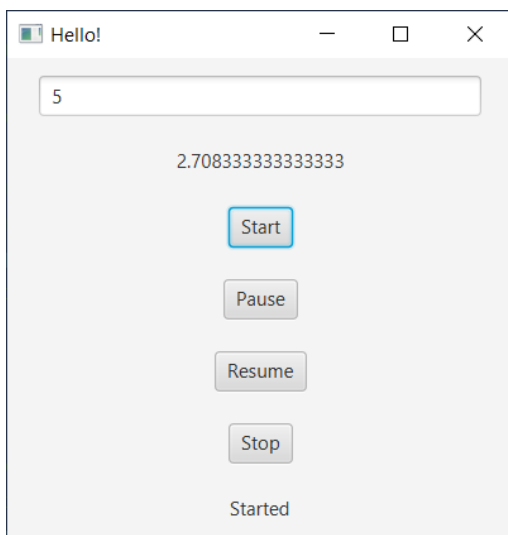
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.VBox?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.TextField?>
<VBox alignment="CENTER" spacing="20.0" xmlns:fx="http://javafx.com/fxml"
    fx:controller="com.example.demo.HelloController">
    <padding>
        <Insets bottom="20.0" left="20.0" right="20.0" top="20.0"/>
    </padding>

    <TextField fx:id="valueText" />
    <Label fx:id="resultText"/>
    <Button text="Start" onAction="#onStartButtonClick"/>
    <Button text="Pause" onAction="#onPauseButtonClick"/>
    <Button text="Resume" onAction="#onResumeButtonClick"/>
    <Button text="Stop" onAction="#onInterruptButtonClick"/>
    <Label fx:id="messageText"/>
</VBox>

```



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