

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Кафедра ИИТ

ЛАБОРАТОРНАЯ РАБОТА №8

По дисциплине: «Современные платформы программирования»

Выполнил:

Студент ФЭИС

3-го курса, группы ПО-5

Прокопчик Е.А.

Проверил:

Крощенко А.А.

Брест 2022

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

Общее 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.

Вариант 12

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

Код программы:

HelloController.java

```
package com.example.spplabl;

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

public class HelloController {
    @FXML
    private Label welcomeText;
    @FXML
    private TextField xText;
    @FXML
    private TextField cycleText;
    @FXML
    private Button Pause;
    @FXML
    private Button Resume;
    @FXML
    private Button Interrupt;

    private Thread thread;
    private double n;
```

```

private double nprev;
private Double result;
private int x;
private int cycle;
final Object lock = new Object();
boolean paused;
boolean thisprocessing;

@FXML
protected void initialize() {
    Pause.setDisable(true);
    Resume.setDisable(true);
    Interrupt.setDisable(true);
}

@FXML
protected void onStartButtonClick() {

    try {
        x = Integer.parseInt(xText.getText());

        if (x < 0) {
            throw new IllegalArgumentException();
        }

        cycle = Integer.parseInt(cycleText.getText());

        if (cycle < 0) {
            throw new IllegalArgumentException();
        }
    }
    catch (IllegalArgumentException e) {
        Platform.runLater(() -> welcomeText.setText("Invalid input"));

        return;
    }

    thread = new Thread(() -> {

        Pause.setDisable(false);
        Interrupt.setDisable(false);

        thisprocessing = true;
        n = 1.0;
        nprev = 1.0;
        result = 1.0;
        while (n < cycle + 1 && thisprocessing == true) {

            try {
                Thread.sleep(200);
            } catch (InterruptedException ignored) {
            }

            synchronized (lock) {
                if (paused) {
                    try {
                        lock.wait();
                    } catch (InterruptedException e) {
                        break;
                    }
                } else {
                    result += (Math.pow(x, n) / (n * nprev));
                    nprev *= n;
                    n++;
                }
            }
        }
    });
}

```

```

        Platform.runLater(() ->
welcomeText.setText(result.toString()));
        }
    }

    Platform.runLater(() -> welcomeText.setText("finish = " +
result.toString()));
    });

    thread.start();
}

@FXML
protected void onPauseButtonClick() {
    Resume.setDisable(false);
    welcomeText.setText(result.toString());
    paused = true;
}

@FXML
protected void onResumeButtonClick() {
    Resume.setDisable(true);
    synchronized (lock) {
        paused = false;
        lock.notify();
    }
}

@FXML
protected void onInterruptButtonClick() {
    thisprocessing= false;
}
}

```

hello-view.fxml

```

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

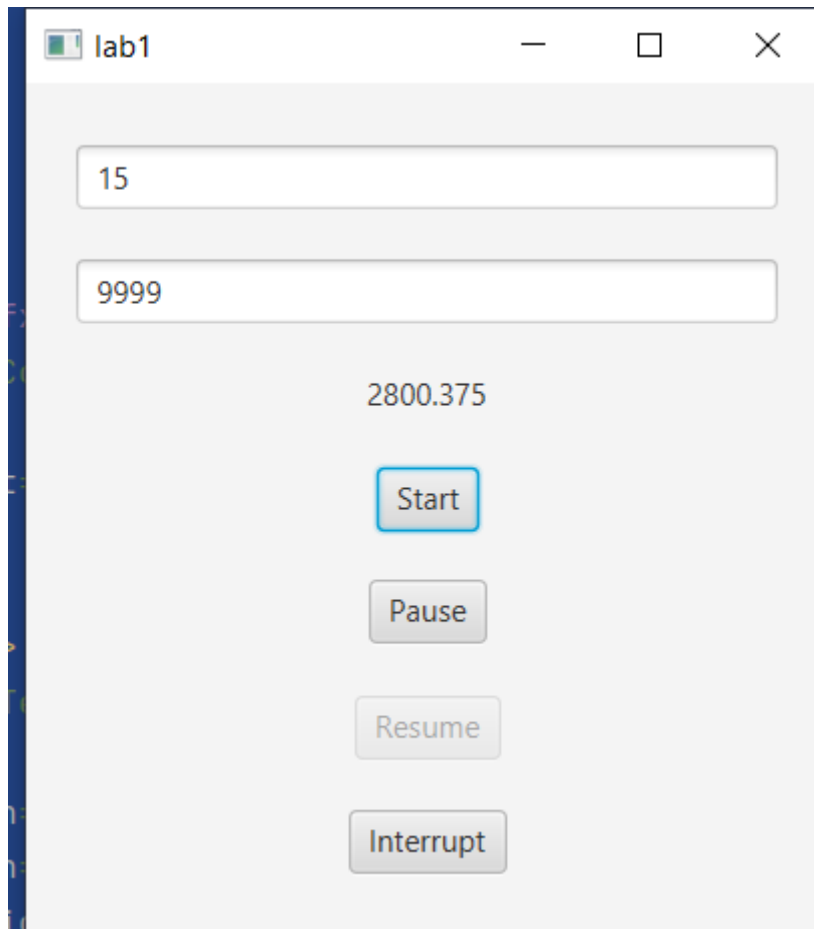
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.TextField?>

<VBox alignment="CENTER" spacing="20.0" xmlns:fx="http://javafx.com/fxml"
    fx:controller="com.example.spplabl.HelloController">
    <padding>
        <Insets bottom="20.0" left="20.0" right="20.0" top="20.0"/>
    </padding>

    <TextField promptText="x" fx:id="xText" />
    <TextField promptText="cycle" fx:id="cycleText" />
    <Label fx:id="welcomeText"/>
    <Button text="Start" fx:id="Start" onAction="#onStartButtonClick"/>
    <Button text="Pause" fx:id="Pause" onAction="#onPauseButtonClick"/>
    <Button text="Resume" fx:id="Resume" onAction="#onResumeButtonClick"/>
    <Button text="Interrupt" fx:id="Interrupt"
onAction="#onInterruptButtonClick"/>
</VBox>

```

Результат:



Вывод:

приобрел навыки написания простого оконного многопоточного приложения с использованием Java API