МИНЕСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ИИТ

**ЛАБОРАТОРНАЯ РАБОТА №8**

«Многопоточные приложения с использованием

Java API»

Выполнил

Студент ФЭИС

3- го курса, группы ПО-5

Брич М.Н.

Проверил

Крощенко А. А.

2021

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

Вариант №3

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

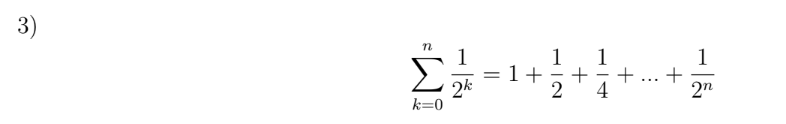
в соответствующие визуальные компоненты. В программе должны быть предусмотрены функции

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.



Код программы:

package com.example.lab8;

import javafx.application.Application;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.GridPane;

import javafx.scene.text.Text;

import javafx.stage.Stage;

public class HelloApplication extends Application {

double currentValue = 0;

int currentIteration = 0;

Text sum = new Text("");

TextField inputCount = new TextField();

Thread backgroundThread;

Button start = new Button();

Button pause = new Button();

Button stop = new Button();

GridPane grid;

@Override

public void init() {

start.setText("Start");

pause.setText("Pause");

stop.setText("Stop");

pause.setDisable(true);

stop.setDisable(true);

start.setOnAction(actionEvent -> startCalculate());

pause.setOnAction(actionEvent -> {

start.setDisable(false);

backgroundThread.suspend();

});

stop.setOnAction(actionEvent -> {

pause.setDisable(true);

stop.setDisable(true);

start.setDisable(false);

stopCalculate();

});

grid = new GridPane();

grid.setAlignment(Pos.CENTER);

grid.setHgap(10);

grid.setVgap(10);

grid.setPadding(new Insets(25, 25, 25, 25));

Text text = new Text("Текущая сумма: ");

grid.add(text, 0, 0, 1, 1);

grid.add(sum, 1, 0, 1, 1);

Label labelCount = new Label("N:");

grid.add(labelCount, 0, 1, 1, 1);

grid.add(inputCount, 1, 1, 1, 1);

grid.add(start, 0, 3);

grid.add(pause, 1, 3);

grid.add(stop, 2, 3);

}

@Override

public void start(Stage primaryStage) {

primaryStage.setTitle("Function");

Scene scene = new Scene(grid, 500, 350);

primaryStage.setScene(scene);

primaryStage.show();

}

public void startCalculate() {

pause.setDisable(false);

stop.setDisable(false);

if (backgroundThread != null) {

backgroundThread.resume();

} else {

Thread task = new Thread(() -> {

try {

int count = Integer.parseInt(inputCount.getText());

start.setDisable(true);

if (count == 0) {

sum.setText(Double.toString(1.0));

} else {

sum.setText(Double.toString(this.currentValue));

for (int i = 0; i <= count; i++) {

try {

this.currentValue += 1 / Math.pow(2.0, i);

Thread.sleep(500);

sum.setText(Double.toString(this.currentValue));

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

pause.setDisable(true);

inputCount.setText("");

} catch (NumberFormatException e) {

sum.setText("Error in input!!!");

pause.setDisable(true);

start.setDisable(true);

}

});

backgroundThread = new Thread(task);

backgroundThread.setDaemon(true);

backgroundThread.start();

}

}

public void stopCalculate() {

backgroundThread.stop();

this.currentValue = 0;

this.sum.setText("");

this.currentIteration = 0;

inputCount.setText("");

backgroundThread = null;

}

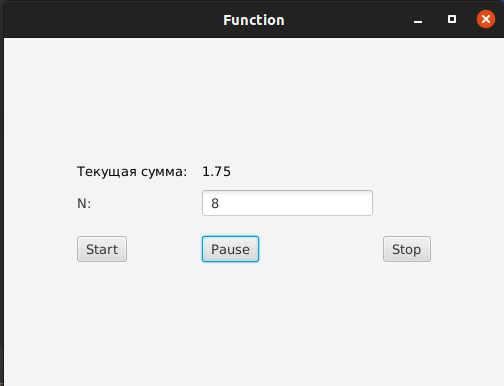
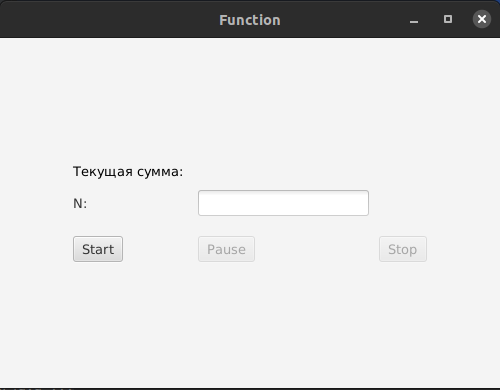
public static void main(String[] args) {

launch(args);

}

}

Тестирование:



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