Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №2

По дисциплине: «СПП»

Тема: “Многопоточные приложения Java”

Выполнил:

Студент 3 курса

Группы ПО-7

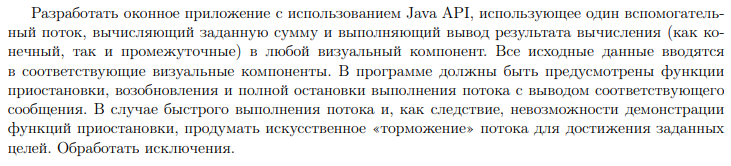
Комиссаров А.Е.

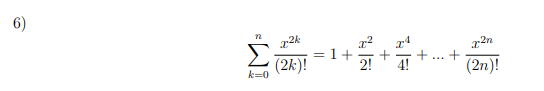
Проверил:

Монтик Н.С.

2023

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





**Ход работы.**

**Файл Main.java:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import static java.lang.Math.pow;

public class Main {

public static int factorial(int n)

{

if (n == 0) return 1;

return n \* factorial(n-1);

}

static double calculate(int x, int k){

return pow(x, 2 \* k) / factorial(2\*k);

}

static void message\_to\_result(String msg){

new Thread(() -> {

String prev\_result\_text = result\_text.getText();

result\_text.setText(msg);

button\_start.setEnabled(false);

try {

Thread.sleep(1000);

} catch (InterruptedException ex) {

throw new RuntimeException(ex);

}

button\_start.setEnabled(true);

result\_text.setText(prev\_result\_text);

}).start();

}

static JLabel title\_text = new JLabel("Введите значения переменных.");

static JLabel enter\_x\_text = new JLabel("Введите значение Х");

static JLabel enter\_n\_text = new JLabel("Введите значение n");

static JTextField input\_x = new JTextField();

static JTextField input\_n = new JTextField();

static JButton button\_start = new JButton("Начать вычисления");

static JButton button\_reset = new JButton("Сбросить значения");

static JButton button\_stop = new JButton("Пауза");

static JLabel result\_text = new JLabel("Результат: ");

static CalcThread calcThread;

static ActionListener listenerStart = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if(calcThread != null && calcThread.isAlive()){

calcThread.pause = false;

button\_start.setText("Продолжить");

button\_start.setEnabled(false);

button\_stop.setEnabled(true);

} else {

if (!input\_x.getText().isEmpty() && !input\_n.getText().isEmpty()) {

int x\_val = Integer.parseInt(input\_x.getText());

int n\_val = Integer.parseInt(input\_n.getText());

if (x\_val != 0 && n\_val != 0) {

calcThread = new CalcThread();

calcThread.SetupVars(x\_val, n\_val);

calcThread.start();

result\_text.setText("Результат: ");

} else {

if (x\_val == 0) {

message\_to\_result("X не может быть равен нулю");

return;

}

message\_to\_result("N не может быть равен нулю");

return;

}

} else {

message\_to\_result("Одно или несколько полей пусты");

}

button\_start.setEnabled(false);

button\_stop.setEnabled(true);

button\_reset.setText("Прекратить вычисления");

}

}

};

static ActionListener listenerReset = e -> {

if(button\_reset.getText().equals("Сбросить значения")){

input\_x.setText("");

input\_n.setText("");

button\_reset.setEnabled(false);

} else {

calcThread.stop = true;

calcThread = null;

Main.button\_start.setText("Начать вычисления");

Main.button\_reset.setText("Сбросить значения");

Main.button\_start.setEnabled(true);

Main.button\_reset.setEnabled(true);

Main.button\_stop.setEnabled(false);

}

};

static ActionListener listenerStop = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if(calcThread != null){

calcThread.pause = true;

button\_start.setText("Продолжить");

button\_start.setEnabled(true);

button\_stop.setEnabled(false);

}

}

};

static KeyListener numbers\_filter = new KeyAdapter() {

public void keyTyped(KeyEvent e) {

char c = e.getKeyChar();

if (!((c >= '0') && (c <= '9') ||

(c == KeyEvent.VK\_BACK\_SPACE) ||

(c == KeyEvent.VK\_DELETE))) {

e.consume();

}

button\_reset.setEnabled(true);

}

};

public static void main(String[] args) {

input\_x.addKeyListener(numbers\_filter);

input\_n.addKeyListener(numbers\_filter);

button\_start.addActionListener(listenerStart);

button\_reset.addActionListener(listenerReset);

button\_stop.addActionListener(listenerStop);

JFrame frame = new JFrame();

frame.setSize(new Dimension(550,260)); //setting window size

frame.setResizable(false);

JPanel panel = new JPanel();

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

frame.setTitle("Лабораторная 8 КОМИССАРОВ");

panel.setLayout(new BoxLayout(panel, BoxLayout.Y\_AXIS));

panel.add(Box.createRigidArea(new Dimension(0, 5))); // empty space 5px high

panel.add(title\_text);

panel.add(enter\_x\_text);

panel.add(input\_x);

panel.add(enter\_n\_text);

panel.add(input\_n);

panel.add(Box.createRigidArea(new Dimension(0, 5)));

panel.add(button\_start);

panel.add(Box.createRigidArea(new Dimension(0, 5)));

panel.add(button\_reset);

panel.add(Box.createRigidArea(new Dimension(0, 5)));

panel.add(button\_stop);

panel.add(Box.createRigidArea(new Dimension(0, 5)));

panel.add(result\_text);

panel.add(Box.createRigidArea(new Dimension(0, 10)));

frame.add(panel);

frame.setLocationRelativeTo(null); // align window to screen center

button\_start.setEnabled(true);

button\_reset.setEnabled(false);

button\_stop.setEnabled(false);

}

}

**Файл CalcThread.java:**

public class CalcThread extends Thread {

int x\_v;

int n\_v;

double prev\_v = 0.0;

boolean ready = false;

boolean pause = false;

boolean stop = false;

CalcThread() {

System.out.println("=============\nthread created");

}

public void SetupVars(int x, int n){

this.x\_v = x;

this.n\_v = n;

this.ready = true;

System.out.println("thread ready");

}

public void run() {

if(!ready){ return; }

for (int k = 0; k < n\_v;){

if(stop){

return;

}

System.out.println("thread busy");

if(!pause){

this.prev\_v += Main.calculate(x\_v, k);

k++;

if(this.prev\_v == Double.POSITIVE\_INFINITY){

Main.result\_text.setText("Результат: k=" + (k-1) + ", результат = " + Main.calculate(x\_v, k - 2) + ". (уходит в +\\-бесконечность)");

k = n\_v;

} else {

Main.result\_text.setText("Результат: k=" + k + ", результат = " + this.prev\_v + ".");

} } try {

Thread.sleep(100);

} catch (InterruptedException e) {

throw new RuntimeException(e);

}

}

Main.button\_start.setText("Начать вычисления");

Main.button\_reset.setText("Сбросить значения");

Main.button\_start.setEnabled(true);

Main.button\_reset.setEnabled(true);

Main.button\_stop.setEnabled(false);

System.out.println("thread stopped");

}

}

**Результат работы программы:**

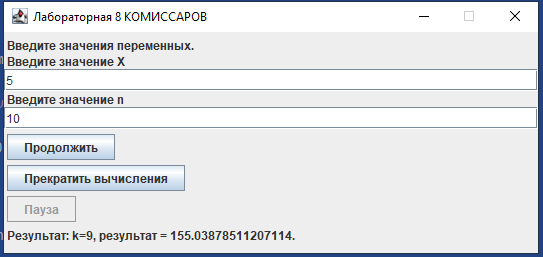
****

Рис. 1 – работа программы (нажата кнопка “Пауза”)

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