МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Програмна інженерія та інформаційні технології управління»

Звіт з лабораторної роботи №1

З предмету «Якість і тестування програмного забезпечення»

Варіант №19

Виконав:

Студент групи КН-36а

Рубан Ю. Д.

Перевірив:

Копп А. М.

Харків – 2018

**Тема:** Изучение метрик качества программного обеспечения.

**Цель работы:** По заданию преподавателя подготовить три программы на языках программирования Java, C#и С++.

**Задание на лабораторную работу:**

Составить алгоритм и программу, которые в одну и ту же переменную вводят вещественные значения Х>0 (количество вводимых Х заранее неизвестно) и для каждого Х находят значение произведения членов последовательности, умножая их до тех пор, пока очередное слагаемое не станет меньше 0,001: 

**Документация в нотации UML**

В нотации UML были составлены диаграммы использования, классов и деятельности. Они показаны на рисунках 1 – 4.

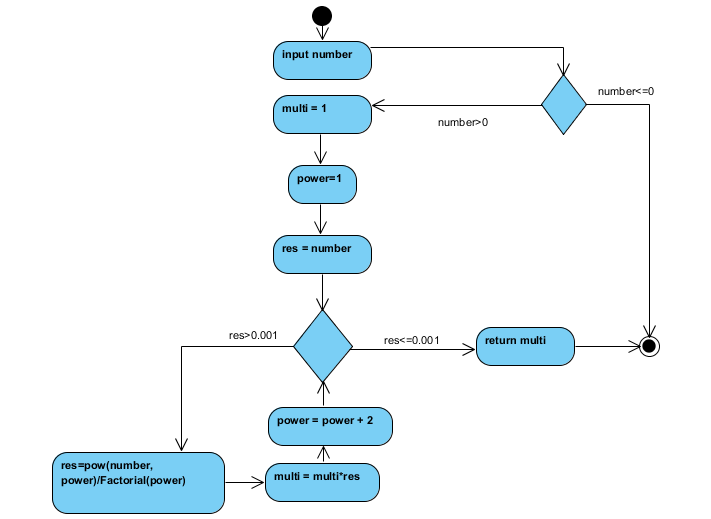


Рисунок 1 – Диаграмма деятельности №1

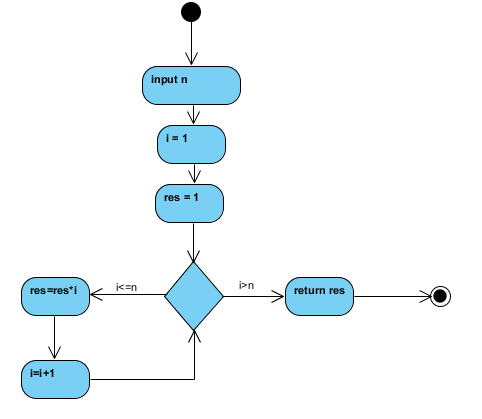


Рисунок 2 – Диаграмма деятельности №2

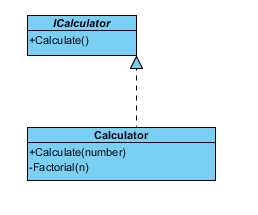


Рисунок 3 – Диаграмма классов

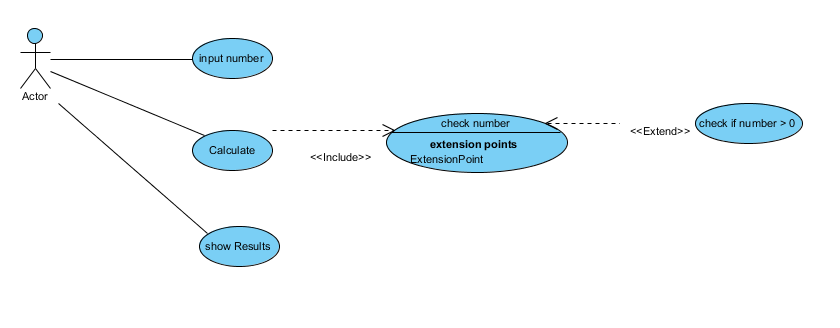


Рисунок 4 – Диаграмма использования

**Листинги разработанных программ**

Код написанный на языке C#:

Файл ICalculator.cs

namespace lab1TESTS\_v2

{

interface ICalculator

{

double Calculate(double number);

}

}

Файл Calculator.cs

using System;

namespace lab1TESTS\_v2

{

class Calculator : ICalculator

{

public double Calculate(double number)

{

if (number <= 0) throw new ArgumentException();

int power = 1;

double res = number;

double multi = 1;

while (res > 0.001)

{

res = Math.Pow(number, power) / Factorial(power);

multi \*= res;

power += 2;

}

return multi;

}

private double Factorial(int n)

{

double res = 1;

for (int i = 1; i <= n; i++)

{

res \*= i;

}

return res;

}

}

}

Файл Form1.cs

using System;

using System.Windows.Forms;

namespace lab1TESTS\_v2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

double number;

try

{

number = Convert.ToDouble(textBox1.Text);

}

catch (Exception)

{

MessageBox.Show("Не правильно введено число");

label3.Text = "";

return;

}

try

{

label3.Text = (new Calculator()).Calculate(number).ToString();

}

catch (ArgumentException)

{

MessageBox.Show("Число должно быть строго больше нуля");

label3.Text = "";

return;

}

}

}

}

Код разработанный на языке С++

Файл icalculator.h

#ifndef ICALCULATOR\_H

#define ICALCULATOR\_H

class ICalculator

{

public:

virtual double ***Calculate***(double number)=0;

};

#endif // ICALCULATOR\_H

Файл calculator.h

#ifndef CALCULATOR\_H

#define CALCULATOR\_H

#include"icalculator.h"

#include"math.h"

#include"stdexcept"

class Calculator : public ICalculator

{

private:

double **Factorial**(int n);

public:

double ***Calculate***(double number);

};

#endif // CALCULATOR\_H

Файл mainwindow.h

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QMessageBox>

#include "calculator.h"

namespace Ui {

class MainWindow;

}

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

explicit MainWindow(QWidget \*parent = 0);

~***MainWindow***();

private slots:

void **on\_pushButton\_clicked**();

private:

Ui::MainWindow \*ui;

};

#endif // MAINWINDOW\_H

Файл calculator.cpp

#include "calculator.h"

double Calculator::***Calculate***(double number)

{

if (number <= 0) throw std::invalid\_argument("");

int power = 1;

double res = number;

double multi = 1;

while (res > 0.001)

{

res = pow(number, power) / Factorial(power);

multi \*= res;

power += 2;

}

return multi;

}

double Calculator::**Factorial**(int n)

{

double res = 1;

for (int i = 1; i <= n; i++)

{

res \*= i;

}

return res;

}

Файл mainwindow.cpp

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::**MainWindow**(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**on\_pushButton\_clicked**()

{

bool check=false;

double number = ui->lineEdit->text().toDouble(&check);

if(!check)

{

QMessageBox b(this);

b.setInformativeText("Не правильно введено число");

ui->lineEdit->setText("");

b.show();

return;

}

ICalculator\* calc = new Calculator();

try

{

ui->label\_3->setText(QString::number(calc->*Calculate*(number)));

}catch(std::invalid\_argument& e)

{

QMessageBox b(this);

b.setInformativeText("Число должно быть строго больше нуля");

ui->lineEdit->setText("");

b.show();

delete calc;

return;

}

delete calc;

}

Код разработанный на языке Java

Файл ICalculator.java

package model;  
  
public interface ICalculator {  
 Double Calculate(Double number);  
}

Файл Calculator.java

package model;  
public class Calculator implements ICalculator {  
 @Override  
 public Double Calculate(Double number) {  
 if(number<=0)throw new IllegalArgumentException();  
 int power = 1;  
 double res = number;  
 double multi = 1.;  
 while (res>0.001)  
 {  
 res = Math.*pow*(number,power)/Factorial(power);  
 multi\*=res;  
 power+=2;  
 }  
 return multi;  
 }  
 private Double Factorial(Integer n)  
 {  
 Double res = 1.;  
 for(int i = 1;i<=n;i++)  
 {  
 res\*=i;  
 }  
 return res;  
 }  
}

Файл Controller.java

package sample;  
  
import javafx.fxml.FXML;  
import javafx.scene.control.\*;  
import model.Calculator;  
import model.ICalculator;  
  
public class Controller {  
 ICalculator calc;  
 @FXML  
 private TextField text;  
 @FXML  
 private Button start;  
 @FXML  
 private Label multi;  
 @FXML  
 private void initialize() {  
 }  
 @FXML  
 public void startClick()  
 {  
 Double number = new Double(0);  
 try {  
 number = Double.*parseDouble*(text.getText());  
 }catch (Exception e)  
 {  
 Alert alert = new Alert(Alert.AlertType.*ERROR*,"Не правильно введено число");  
 alert.show();  
 multi.setText("");  
 return;  
 }  
 calc=new Calculator();  
 try {  
 multi.setText(calc.Calculate(number).toString());  
 }catch (IllegalArgumentException e)  
 {  
 Alert alert = new Alert(Alert.AlertType.*ERROR*,"Число должно быть строго больше нуля");  
 alert.show();  
 multi.setText("");  
 return;  
 }  
 }  
  
}

**Таблица метрик**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | n1 | n2 | N1 | N2 | n | N |  | V | V\* | n2\* |
| C++ | 22 | 24 | 51 | 49 | 46 | 100 | 208,14659 | 552,35619 | 8 | 2 |
| C# | 21 | 19 | 46 | 38 | 40 | 84 | 172,94928 | 447,04195 | 8 | 2 |
| Java | 20 | 17 | 43 | 38 | 37 | 81 | 155,92543 | 421,96572 | 8 | 2 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | L |  | I | E | T | B | Mх | Мм |
| C++ | 0,014483 | 0,11586 | 24,5947 | 38137 | 3813 | 0,18411 | 0,33333 | 2 |
| C# | 0,017895 | 0,14316 | 21,28771 | 24980 | 2498 | 0,14901 | 0,33333 | 2 |
| Java | 0,018958 | 0,15167 | 18,87741 | 22256 | 2225 | 0,140655 | 0,33333 | 2 |

**Выводы**

В данной лабораторной работе были написаны программы в соответствии с заданием. Были составлены диаграммы в нотации UML. Были вычислены метрики качества для программ, написанных на разных языках программирования.