**Национальный исследовательский университет**

**«МИЭТ»**

Институт СПИНТех

Кафедра ИПОВС

**Контрольное домашнее задание**

по дисциплине

«Верификация и аттестация программного обеспечения»

Вариант № 31

**Выполнил:**

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.)

Группа\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Дата сдачи работы и подпись студента)

**Проверил:**

Москва – 2020

Содержание

[1. Формулировка задания 2](#_Toc59995512)

[2. Описание программы 3](#_Toc59995513)

[2.1 Постановка задачи 3](#_Toc59995514)

[2.2 Алгоритм решения задачи 4](#_Toc59995515)

[2.3 Исходный текст программы 7](#_Toc59995516)

[3. Характеристики разработанной программы на основе метрик Холстеда 17](#_Toc59995517)

[4. Оценка структурной сложности программы на основе трех критериев и метрики Маккейба 19](#_Toc59995518)

[Вывод 28](#_Toc59995519)

# 1. Формулировка задания

Задание заключается в выполнении следующих задач:

1. Разработать программу по заданной тематике в среде Visual Studio на языке C# с оконным интерфейсом.

2. Составить описание разработанной программы (при подготовке описания и формирования разделов документа руководствоваться требованиями ГОСТ РД 50-34.698-90, 19.404-79, 19.101-77, 19.502-78, 19.701-90) в следующем объеме: - постановка задачи; - алгоритм решения задачи; - исходный текст программы (комментарии в программе обязательны).

3. Определить характеристики разработанной программы на основе метрик Холстеда:

- словарь программы;

- длину и объем программы;

- уровень реализации;

4. Оценить структурную сложность программы на основе трех критериев и метрики Маккейба.

# 2. Описание программы

## 2.1 Постановка задачи

Программа разрабатывается в виде приложения Windows с оконным интерфейсом и формированием исполняемого файла с расширением .exe. Программа представляет собой решение двух задач согласно варианту 31. Выбор решаемой задачи происходит в главном окне программы. Предусмотрена проверка входных данных и при ошибке ввода формируется соответствующее сообщение. Предусмотрены кнопки возврата к главному окну программы и ее завершению.

Решаемые с помощью программы задачи согласно варианту 31:

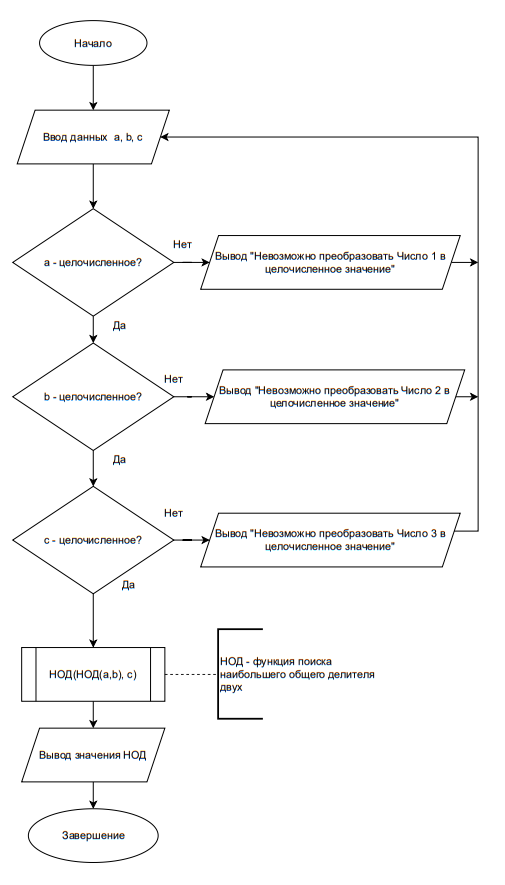
Задание 1: с клавиатуры вводится массив целых чисел до нажатия клавиши «Esc». Какое число в массиве встретится раньше: максимальное или минимальное? Если таких чисел несколько, то должны быть учтены самые первые из них. Определить размах значений элементов массива, его среднее арифметическое и среднее геометрическое значение. Результаты вывести на экран.

Задание 2: С клавиатуры по запросу программы вводятся три целых числа. Вывести на экран их наибольший общий делитель, имея в виду, что НОД(a, b, с) = НОД(НОД(a, b), с). При отсутствии решения сформировать соответствующее сообщение.

## 2.2 Алгоритм решения задачи

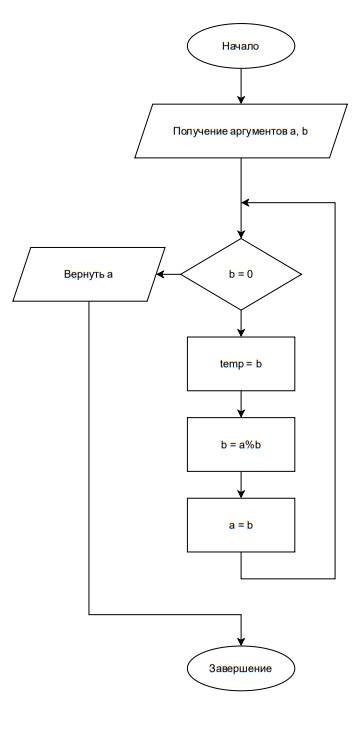
Задание 1

Начало выполнения основной задачи происходит при вызове метода SolveWithValidation(). Его блок-схема:



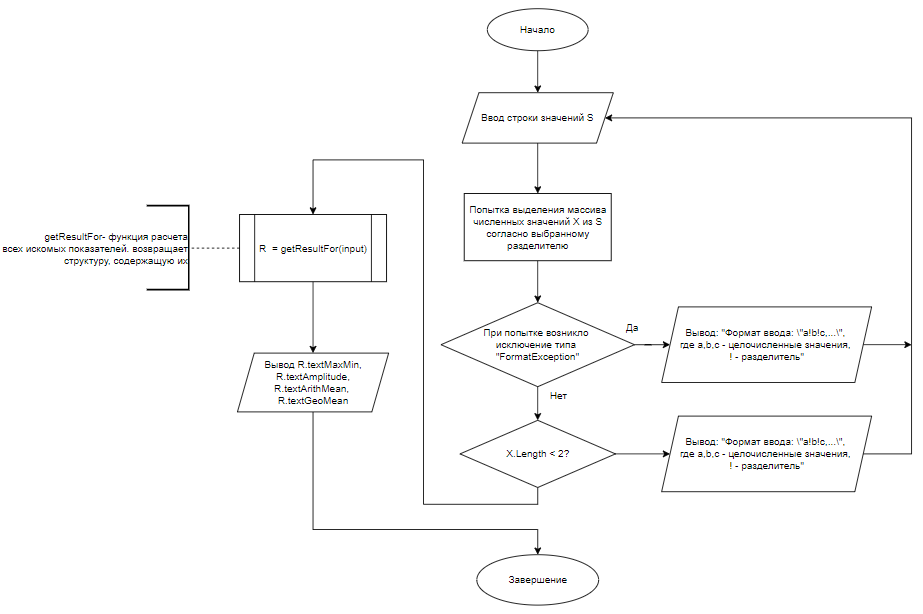
Видно, что после проверки данных вызывается метод НОД(НОД()), что равносильно последовательному двойному вызову метода НОД. Результат первого вызова метода НОД передается как аргумент для второго его вызова.

Блок-схема метод НОД(в программе в методе использована рекурсия, в блок-схеме для расчета сложности использован цикл):

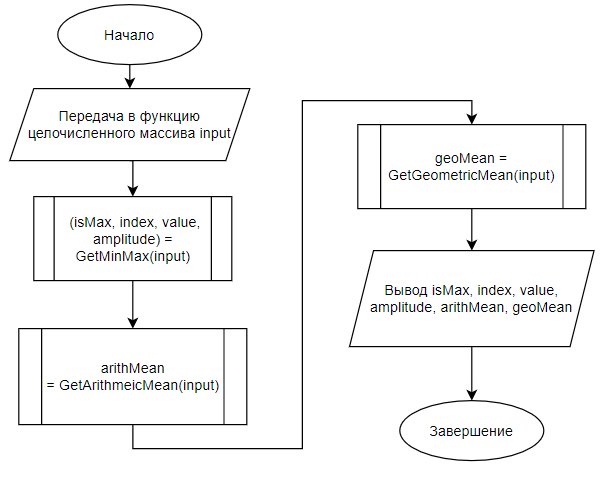


Задание 2

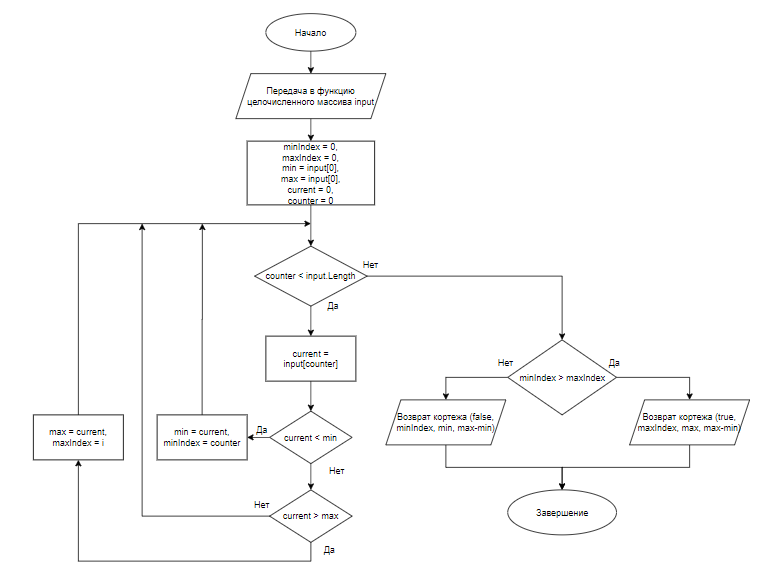
Начало выполнения основной задачи происходит при вызове метода SolveWithValidation(). Его блок-схема(также содержит схему метода валидации - GetValidatedInputNumbers):



Структура алгоритма метода getResultFor():



Блок схема метода GetMinMax():



Методы GetArithmeicMean() и GetGeometricMean() содержат последовательные вызовы встроенных методов C# без наличия ветвлений, циклов. Соответственно, их алгоритмическая сложность согласно трем критериям равна 0.

## 2.3 Исходный текст программы

Главное окно:

using System;

using System.Drawing;

using System.Windows.Forms;

namespace ПИН\_22М\_Постоев\_КДЗ\_31\_вариант

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

ConfigureUIElements();

}

private void ConfigureUIElements()

{

FormBorderStyle = FormBorderStyle.FixedDialog;

labelIntro.Text = "ПИН-22М Постоев И.Е. КДЗ Вариант 31";

labelIntro.Font = new Font(labelIntro.Font.Name, AppConstants.mainFontSize);

buttonLCD.Text = "Задание 1";

buttonLCD.Font = new Font(buttonLCD.Font.Name, AppConstants.mainFontSize);

buttonLCD.Click += new EventHandler(OnButtonLCD\_Click);

buttonMinMax.Text = "Задание 2";

buttonMinMax.Font = new Font(buttonMinMax.Font.Name, AppConstants.mainFontSize);

buttonMinMax.Click += new EventHandler(OnButtonMinMax\_Click);

buttonExit.Text = "Выход";

buttonExit.Font = new Font(buttonExit.Font.Name, AppConstants.mainFontSize);

buttonExit.Click += new EventHandler(OnButtonExit\_Click);

}

private void OnButtonLCD\_Click(Object sender, EventArgs e)

{

Hide();

var lcdForm = new LCDForm(this);

lcdForm.ShowDialog();

}

private void OnButtonMinMax\_Click(Object sender, EventArgs e)

{

Hide();

var minMaxForm = new MinMaxForm(this);

minMaxForm.ShowDialog();

}

private void OnButtonExit\_Click(Object sender, EventArgs e)

{

Close();

}

}

}

Задание 1:

using System;

using System.Drawing;

using System.Windows.Forms;

namespace ПИН\_22М\_Постоев\_КДЗ\_31\_вариант

{

/\*

\* Класс для решения Задачи 1

\*/

public partial class LCDForm : Form

{

/\*

\* Структура, содержащая входные значения

\*/

struct InputNumbers

{

public int number1;

public int number2;

public int number3;

public InputNumbers(int number1, int number2, int number3) {

this.number1 = number1;

this.number2 = number2;

this.number3 = number3;

}

}

private MainForm parent;

public LCDForm(MainForm mainForm)

{

parent = mainForm;

InitializeComponent();

ConfigureUIElements();

}

private void ConfigureUIElements()

{

SetupLabelsAppearance();

SetupTextBoxesAppearance();

SetupSolveButton();

}

/\*

\* Настраивает надписи в пользовательском интерфейсе

\*/

private void SetupLabelsAppearance()

{

introLabel.Text = "Введите 3 числа для определения их НОД";

labelNumber1.Text = "Число 1";

labelNumber2.Text = "Число 2";

labelNumber3.Text = "Число 3";

labelResult.Text = "Результат";

introLabel.Font = new Font(introLabel.Font.Name, AppConstants.mainFontSize);

labelNumber1.Font = new Font(labelNumber1.Font.Name, AppConstants.mainFontSize);

labelNumber2.Font = new Font(labelNumber2.Font.Name, AppConstants.mainFontSize);

labelNumber3.Font = new Font(labelNumber3.Font.Name, AppConstants.mainFontSize);

labelResult.Font = new Font(labelResult.Font.Name, AppConstants.mainFontSize);

}

/\*

\* Настраивает текстовые поля в пользовательском интерфейсе

\*/

private void SetupTextBoxesAppearance()

{

FormBorderStyle = FormBorderStyle.FixedDialog;

taskText.Multiline = true;

taskText.BorderStyle = BorderStyle.None;

taskText.BackColor = DefaultBackColor;

taskText.ReadOnly = true;

taskText.Font = new Font(taskText.Font.Name, AppConstants.mainFontSize);

taskText.Text = "С клавиатуры по запросу программы вводятся три целых числа." +

"Вывести на экран их наибольший общий делитель, имея в виду, что НОД(a, b, с) = НОД(НОД(a, b), с)." +

"При отсутствии решения сформировать соответствующее сообщение.";

taskText.Select(0, 0);

resultText.ReadOnly = true;

resultText.Font = new Font(resultText.Font.Name, AppConstants.mainFontSize);

textNumber1.Font = new Font(textNumber1.Font.Name, AppConstants.mainFontSize);

textNumber2.Font = new Font(textNumber2.Font.Name, AppConstants.mainFontSize);

textNumber3.Font = new Font(textNumber3.Font.Name, AppConstants.mainFontSize);

}

/\*

\* После закрытия текущего окна, отображает главное окно

\*/

protected override void OnFormClosing(FormClosingEventArgs e)

{

base.OnFormClosing(e);

parent.Show();

}

/\*

\* Настраивает кнопку "Решить"

\*/

private void SetupSolveButton()

{

buttonSolve.Text = "Решить";

buttonSolve.Font = new Font(buttonSolve.Font.Name, AppConstants.mainFontSize);

buttonSolve.Click += new EventHandler(OnButtonSolve\_Click);

}

private void OnButtonSolve\_Click(Object sender, EventArgs e)

{

SolveWithValidation();

}

/\*

\* Проводит поиск решения с учетом проверки входных значений.

\* Заполняет текстовое поле ответа

\*/

private void SolveWithValidation()

{

InputNumbers? numbers = getValidatedInputNumbers();

if (numbers.HasValue)

{

int result = getResultFor(numbers.Value);

resultText.Text = result.ToString();

}

}

/\*

\* Проверяет применимость входных значений к поиску решения.

\* Возвращает структуру из проверенных входные значений, если каждое из них соответствуют требованиям.

\* Возвращает null, если хотя бы одно не соответствует требованиям.

\* т.е. возвращаемое значение - опциональное(либо действительное значение, либо null в одном представлении)

\*/

private InputNumbers? getValidatedInputNumbers()

{

int? input1;

int? input2;

int? input3;

//Получение проверенных значений.

try

{

input1 = getValidatedInputWith(textNumber1.Text, labelNumber1.Text);

input2 = getValidatedInputWith(textNumber2.Text, labelNumber2.Text);

input3 = getValidatedInputWith(textNumber3.Text, labelNumber3.Text);

}

catch (Exception e)

{

//Обработка исключений: отображение окон соотетствующего ошибке содержания.

if (e is FormatException)

{

MessageBox.Show("Невозможно преобразовать " + e.Message + " в целочисленное значение");

}

else

{

MessageBox.Show("Ошибка входных данных");

}

return null;

}

//"Раскрытие" опциональных значений - проверка на содержание действительного значения

if (!input1.HasValue || !input2.HasValue || !input3.HasValue)

{

return null;

}

return new InputNumbers(input1.Value, input2.Value, input3.Value);

}

/\*

\* Проверка входного значения.

\* Возвращает входное значение в целочисленном формате, либо null, если оно не соответсвует требованиям.

\*/

private int? getValidatedInputWith(String input, String errorSource)

{

int num;

try

{

num = Int32.Parse(input);

}

catch

{

throw new FormatException(errorSource);

}

return num;

}

/\*

\* Обработка значений для работы с методом поиска НОД

\*/

private int getResultFor(InputNumbers numbers)

{

int a = Math.Abs(numbers.number1);

int b = Math.Abs(numbers.number2);

int c = Math.Abs(numbers.number3);

return getLCDFor(a, b, c);

}

/\*

\* Возвращает НОД трех чисел

\*/

private int getLCDFor(int a, int b, int c)

{

return Euqlid(Euqlid(a, b), c);

}

/\*

\* Возвращает НОД двух чисел

\*/

private int Euqlid(int a, int b)

{

if (b == 0)

{

return a;

}

else

{

return Euqlid(b, a % b);

}

}

}

}

Задание 2:

using System;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Windows.Forms;

namespace ПИН\_22М\_Постоев\_КДЗ\_31\_вариант

{

/\*

\* Структура, содержащая выходные данные программы

\*/

struct Result {

public string textMaxMin;

public string textAmplitude;

public string textArithMean;

public string textGeoMean;

}

/\*

\* Класс для решения Задачи 2

\*/

public partial class MinMaxForm : Form

{

private const string WHITESPACE\_NAME = "пробел";

private MainForm parent;

public MinMaxForm(MainForm mainForm)

{

parent = mainForm;

InitializeComponent();

ConfigureUIElements();

}

private void ConfigureUIElements()

{

SetupTextBoxesAppearance();

SetupLabelsAppearance();

SetupSolveButton();

SetupDividerCheckBox();

}

/\*

\* Настраивает надписи в пользовательском интерфейсе

\*/

private void SetupLabelsAppearance()

{

labelMaxMin.Text = "Первое из min/max";

labelAmplitude.Text = "Размах";

labelArithMean.Text = "Арифметическое среднее";

labelGeoMean.Text = "Геометрическое среднее";

introLabel.Text = "Введите массив чисел";

labelResult.Text = "Результат";

labelDivider.Text = "Разделитель:";

labelMaxMin.Font = new Font(labelMaxMin.Font.Name, AppConstants.mainFontSize);

labelAmplitude.Font = new Font(labelAmplitude.Font.Name, AppConstants.mainFontSize);

labelArithMean.Font = new Font(labelArithMean.Font.Name, AppConstants.mainFontSize);

labelGeoMean.Font = new Font(labelGeoMean.Font.Name, AppConstants.mainFontSize);

introLabel.Font = new Font(introLabel.Font.Name, AppConstants.mainFontSize);

labelResult.Font = new Font(labelResult.Font.Name, AppConstants.mainFontSize);

labelDivider.Font = new Font(labelDivider.Font.Name, AppConstants.mainFontSize);

}

/\*

\* Настраивает текстовые поля в пользовательском интерфейсе

\*/

private void SetupTextBoxesAppearance()

{

FormBorderStyle = FormBorderStyle.FixedDialog;

taskText.Multiline = true;

taskText.BorderStyle = BorderStyle.None;

taskText.BackColor = DefaultBackColor;

taskText.ReadOnly = true;

taskText.Text = "Условие: с клавиатуры вводится массив целых чисел до нажатия клавиши «Esc». " +

"Какое число в массиве встретится раньше: максимальное или минимальное? " +

"Если таких чисел несколько, то должны быть учтены самые первые из них. " +

"Определить размах значений элементов массива, его среднее арифметическое и среднее геометрическое значение." +

"Результаты вывести на экран.";

taskText.Select(0, 0);

textMaxMin.ReadOnly = true;

textAmplitude.ReadOnly = true;

textArithMean.ReadOnly = true;

textGeoMean.ReadOnly = true;

textInputNumbers.Font = new Font(textInputNumbers.Font.Name, AppConstants.mainFontSize);

taskText.Font = new Font(taskText.Font.Name, AppConstants.mainFontSize);

textMaxMin.Font = new Font(textMaxMin.Font.Name, AppConstants.mainFontSize);

textAmplitude.Font = new Font(textAmplitude.Font.Name, AppConstants.mainFontSize);

textArithMean.Font = new Font(textArithMean.Font.Name, AppConstants.mainFontSize);

textGeoMean.Font = new Font(textGeoMean.Font.Name, AppConstants.mainFontSize);

textInputNumbers.KeyDown += TextInputNumbers\_KeyDown;

}

/\*

\* Обработчик нажатия "Esc"

\*/

private void TextInputNumbers\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Escape)

{

SolveWithValidation();

}

}

/\*

\* После закрытия текущего окна, отображает главное окно

\*/

protected override void OnFormClosing(FormClosingEventArgs e)

{

base.OnFormClosing(e);

parent.Show();

}

/\*

\* Настраивает кнопку "Решить"

\*/

private void SetupSolveButton()

{

buttonSolve.Text = "Решить";

buttonSolve.Font = new Font(buttonSolve.Font.Name, AppConstants.mainFontSize);

buttonSolve.Click += new EventHandler(OnButtonSolve\_Click);

}

/\*

\* Настраивает элемент выбора разделителя

\*/

private void SetupDividerCheckBox() {

string[] dividers = {WHITESPACE\_NAME, ";", ",", ":"};

comboBoxDivider.Items.AddRange(dividers);

comboBoxDivider.SelectedIndex = 0;

comboBoxDivider.Font = new Font(comboBoxDivider.Font.Name, AppConstants.mainFontSize);

}

private void OnButtonSolve\_Click(Object sender, EventArgs e)

{

SolveWithValidation();

}

/\*

\* Решает задачу с обработкой входных значений

\* Заполняет текстовые поля выходных данных

\*/

private void SolveWithValidation()

{

var numbers = GetValidatedInputNumbers();

// если произошла ошибка валидации или введено малое количество чисел, к решению не приступать

if (numbers.Length >= 2)

{

var result = GetResultFor(numbers);

textMaxMin.Text = result.textMaxMin;

textAmplitude.Text = result.textAmplitude;

textArithMean.Text = result.textArithMean;

textGeoMean.Text = result.textGeoMean;

}

}

/\*

\* Возвращает массив проверенных значений

\* Если возникла ошибка проверки - отображает соответствующее окно и возвращает пустой массив

\*/

private int[] GetValidatedInputNumbers()

{

string input = textInputNumbers.Text;

string selectedState = comboBoxDivider.SelectedItem.ToString();

char divider;

if (selectedState.Equals(WHITESPACE\_NAME))

{

divider = ' ';

}

else {

divider = selectedState.ToCharArray()[0];

}

try

{

int[] numbers = input.Trim(divider).Split(divider).Select(n => Convert.ToInt32(n)).ToArray();

return numbers.Length >= 2 ? numbers : new int[0];

}

catch (FormatException e)

{

MessageBox.Show("Формат ввода: \"a!b!c,...\" где a,b,c - целочисленные значения, ! - разделитель");

return new int[0];

}

catch (Exception)

{

MessageBox.Show("Ошибка входных данных");

return new int[0];

}

}

/\*

\* Возвращает подготовленные для отображения выходные данные

\*/

private Result GetResultFor(int[] input)

{

var result = new Result();

(bool isMax, int index, int value, int amplitude) = GetMinMax(input);

double arithMean = GetArithmeicMean(input);

double geoMean = GetGeometricMean(input);

string textMaxMin = isMax ? "Первое - максимальное" : "Первое - минимальное";

textMaxMin += Environment.NewLine + "Значение: " + value +

Environment.NewLine + "Позиция: " + (index + 1);

result.textMaxMin = textMaxMin;

result.textAmplitude = amplitude.ToString();

result.textArithMean = arithMean.ToString();

result.textGeoMean = geoMean.ToString();

return result;

}

/\*

\* Определяет максимальное/минимальное значения последовательности.

\* Возвращает кортеж из значений:

\* (первый-максимум?, индекс значения, значение, разброс)

\*/

private (bool, int, int, int) GetMinMax(int[] input)

{

int minIndex = 0,

maxIndex = 0;

int min = input[0],

max = input[0];

int current = 0;

for(int i = 0; i < input.Length; i++) {

current = input[i];

if (current < min)

{

min = current;

minIndex = i;

}

else if (current > max)

{

max = current;

maxIndex = i;

}

}

if (minIndex < maxIndex)

{

return (false, minIndex, min, max - min);

}

else

{

return (true, maxIndex, max, max - min);

}

}

/\*

\* Возвращает арифметическое среднее

\*/

private double GetArithmeicMean(int[] input)

{

int sum = input.Sum();

int length = input.Length;

return sum / length;

}

/\*

\* Возвращает геометрическое среднее

\*/

private double GetGeometricMean(int[] input)

{

double multipliedValues = input.Aggregate((x, y) => x \* y);

int length = input.Length;

return Math.Pow(multipliedValues, 1.0 / length);

}

}

}

# 3. Характеристики разработанной программы на основе метрик Холстеда

Метрики рассчитаны для аналитической части программы(не UI)

Словарь программы

Словарь операторов = 15

= 34

+ 16

- 11

/ 2

\* 1

[] 7

>= 1

? 2

: 2

< 3

> 2

++ 1

% 1

== 2

=> 2

Общее количество операторов = 85

Словарь операндов = 21

numbers 13

result 17

num 3

a 7

b 7

c 4

isMax 2

index 2

value 2

amplitude 2

minIndex 3

maxIndex 3

min 6

max 6

length 4

sum 2

0 7

1 1

x 2

y 2

i 6

Общее количество операндов = 101

Общий словарь программы

*= + = 36*

Длина реализации программы

N = + = 186

Длина программы

= =58,5 + 92,2 ≈ 151

Объем программы

V = = 186 \* 5,17 ≈ 961

Потенциальный размер словаря операторов

*=19*

Потенциальный объем программы

= 92

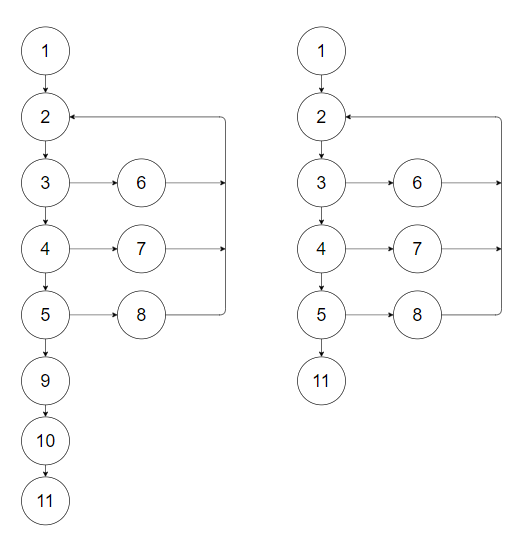
Уровень реализации программы

L = ≈ 0.1

# 4. Оценка структурной сложности программы на основе трех критериев и метрики Маккейба

Задание 1

Исходный(слева) и упрощенный(справа) управляющие графы метода SolveWithValidation



Обоснование упрощения:

= 13 – 11 + 2 = 4

= 11 – 9 + 2 = 4

Оценка алгоритмической сложности

Задание 1

Первый критерий

m1: 1-2-**3**-6-2-**3**-**4**-**5**-11 ; p1 = 4

m2: 1-2-**3**-**4**-7-2-**3**-**4**-**5**-11 ; p2 = 5

m3: 1-2-**3**-**4**-**5**-8-2-**3**-**4**-**5**-11 ; p3 = 6

S1 = p1 + p2 + p3 = 15

Второй критерий

Ациклические маршруты:

m1 = 1-2-**3**-**4**-**5**-11; p1 = 3

Циклические маршруты:

m1 = 2-**3**-6-2;p2 = 1;

m2 = 2**-3-4**-7-2**;** p3 = 2

m4 = 2**-3-4**-**5**-8-2; p4 = 3

S2 = p1 + p2 + p3 + p4 = 9

Третий критерий

m1: 1-2-**3**-6-2-**3**-**4**-**5**-11 ; p1 = 4

m2: 1-2-**3**-**4**-7-2-**3**-**4**-**5**-11 ; p2 = 5

m3: 1-2-**3**-**4**-**5**-8-2-**3**-**4**-**5**-11 ; p3 = 6

m4: 1-2-**3**-6-2-**3**-**4**-7-2-**3**-**4**-**5**-11 ; p4 = 6

m5: 1-2-**3**-6-2-**3**-**4**-**5**-8-2-**3**-**4**-**5**-11 ; p5 = 7

m6: 1-2-**3**-**4**-7-2-**3**-**4**-**5-**8-2-**3**-**4**-**5**-11 ; p6 = 8

m7: 1-2-**3**-6-2-**3**-**4**-7-2-**3**-**4**-**5-**8-2-**3**-**4**-**5**-11 ; p7 = 9

S3 = p1 + p2 + p3 + p4 + p5 + p6 + p7 = 45

Оценка сложности согласно метрике Маккейба

S = Z = 4

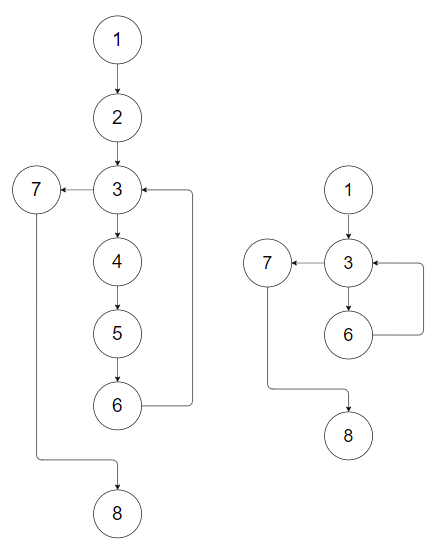
Матрица смежности:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 11 |
| 1 |  |  |  |  |  |  |  |  |  |
| 2 | 1 |  |  |  |  | 1 | 1 | 1 |  |
| 3 |  | 1 |  |  |  |  |  |  |  |
| 4 |  |  | 1 |  |  |  |  |  |  |
| 5 |  |  |  | 1 |  |  |  |  |  |
| 6 |  |  | 1 |  |  |  |  |  |  |
| 7 |  |  |  | 1 |  |  |  |  |  |
| 8 |  |  |  |  | 1 |  |  |  |  |
| 11 |  |  |  |  | 1 |  |  |  |  |

Матрица достижимости:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 11 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|  |  |  |  |  |  |  |  |  |  |

Исходный(слева) и упрощенный(справа) управляющие графы метода НОД:



Обоснование упрощения:

= 8 – 8 + 2 = 2

= 5 - 5 + 2 = 2

Оценка алгоритмической сложности

Задание 1

Первый критерий

m1: 1-**3**-6-7-8 ; p1 = 1

S1 = p1 = 1

Второй критерий

Ациклические маршруты:

m1 = 1-**3**-7-8; p1 = 1

Циклические маршруты:

m2 = **3**-6-**3**;p2 = 2;

S2 = p1 + p2 = 3

Третий критерий

m1 = 1-**3**-7-8; p1 = 1

m2 = 1-**3**-6-**3**-7-8; p1 = 2

S3 = p1 + p2 = 3

Оценка сложности согласно метрике Маккейба

S = Z = 2

Матрица смежности:

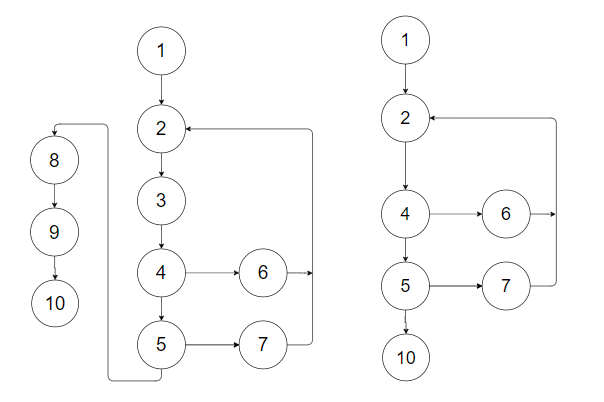
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 3 | 6 | 7 | 8 |
| 1 |  |  |  |  |  |
| 3 | 1 |  | 1 |  |  |
| 6 |  | 1 |  |  |  |
| 7 |  | 1 |  |  |  |
| 8 |  |  |  | 1 |  |

Матрица достижимости:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 3 | 6 | 7 | 8 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 1 | 1 |
| 8 | 1 | 1 | 1 | 1 | 1 |

Задание 2

Исходный(слева) и упрощенный(справа) управляющие графы метода SolveWithValidation



Обоснование упрощения:

= 11 – 10 + 2 = 3

= 8 - 7 + 2 = 3

Оценка алгоритмической сложности

Первый критерий

m1: 1-2-**4**-6-2-**4**-**5**-10; p1 = 3

m2: 1-2-**4**-**5**-7-2-**4**-**5**-10; p2 = 4

S1 = p1 + p2 =7

Второй критерий

Ациклические маршруты:

m1 = 1-2-**4**-**5**-10; p1 = 2

Циклические маршруты:

m2 = 2-**4**-6-2; p2 = 1

m3 = 2-**4**-**5**-7-2; p3 = 2

S2 = p1 + p2 + p3 = 6

Третий критерий

m1: 1-2-**4**-**5**-10; p1 = 2

m2: 1-2-**4**-6-2-**4**-**5**-10; p2 = 3

m3: 1-2-**4**-**5**-7-2-**4**-**5**-10; p2 = 4

m4: 1-2-**4-**6-2-**4**-**5**-7-2-**4**-**5**-10; p2 = 5

S3 = p1 + p2 + p3 + p4= 14

Оценка сложности согласно метрике Маккейба

S = Z = 3

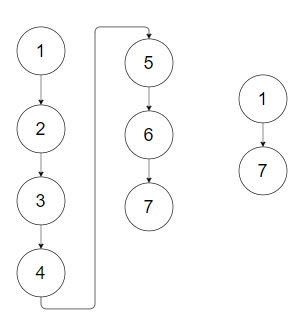
Матрица смежности:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 4 | 5 | 6 | 7 | 10 |
| 1 |  |  |  |  |  |  |  |
| 2 | 1 |  |  |  | 1 | 1 |  |
| 4 |  | 1 |  |  |  |  |  |
| 5 |  |  | 1 |  |  |  |  |
| 6 |  |  | 1 |  |  |  |  |
| 7 |  |  |  | 1 |  |  |  |
| 10 |  |  |  | 1 |  |  |  |

Матрица достижимости:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 4 | 5 | 6 | 7 | 10 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Исходный(слева) и упрощенный(справа) управляющие графы метода getResultFor



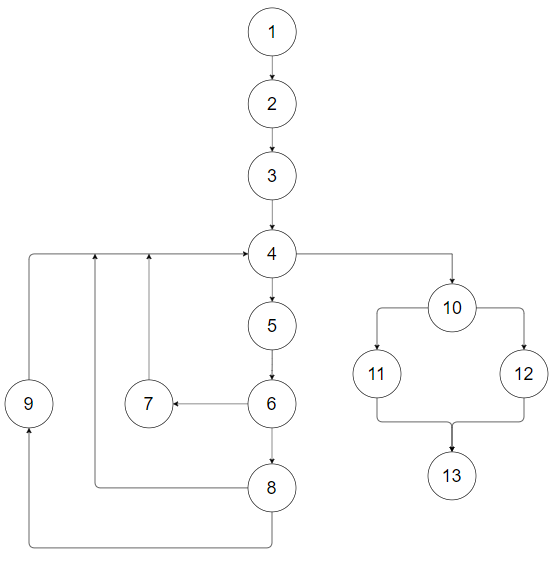
Обоснование упрощения:

= 6 – 7 + 2 = 1

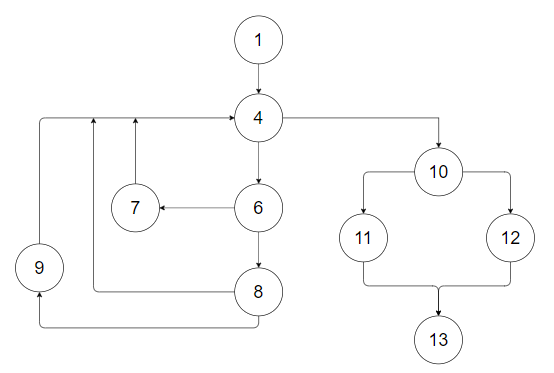
= 1 - 2+ 2 = 1

При отсутствии ветвлений, циклов, алгоритмическая сложность графа равна 0.

Исходный управляющиq граф метода GetMinMax



Упрощенный управляющиq граф метода GetMinMax



Обоснование упрощения:

= 16 – 13 + 2 = 5

= 13 - 10 + 2 = 5

Оценка алгоритмической сложности

Первый критерий

m1: 1-**4**-**10**-11-13; p1 = 2

m2: 1-**4**-**10**-12-13; p2 = 2

m3: 1-**4**-**6** -7-**4**-10-12-13; p3= 3

m4: 1-**4**-**6** -**8**-9-**4**-10-12-13; p4 = 4

S1 = p1 + p2+ p3 + p4 = 11

Второй критерий

Ациклические маршруты:

m1: 1-**4**-**10**-11-13; p1 = 2

m2: 1-**4**-**10**-12-13; p2 = 2

Циклические маршруты:

m3 = **4**-**6**-7-**4**; p3 = 3

m4 = **4**-**6**-**8**-**4**; p4 = 4

m5 = **4**-**6**-**8**-9-**4**; p5 = 4

S2 = p1 + p2 + p3 + p4 + p5 = 15

Третий критерий

m1: 1-**4**-**10**-11-13; p1 = 2

m2: 1-**4**-**10**-12-13; p2 = 2

m3 =1-**4**-**6**-7-**4-10**-12-13; p3 = 4

m4 =1-**4**-**6**-**8**-**4-10**-12-13; p4 = 5

m5 =1-**4**-**6**-**8**-9-**4-10**-12-13; p5 = 5

m6 =1-**4**-**6**-7-**4-10**-11-13; p6 = 4

m7 =1-**4**-**6**-**8**-**4-10**-11-13; p7 = 5

m8 =1-**4**-**6**-**8**-9-**4-10**-11-13; p8 = 5

m9 =1-**4**-**6**-7-**4**-**6**-**8**-**4-10**-12-13; p9 = 7

m10 =1-**4**-**6**-7-**4**-**6**-**8**-9-**4-10**-12-13; p10 = 7

m11 =1-**4**-**6**-**8**-**4**-**6**-**8**-9-**4-10**-12-13; p11 = 8

m12 =1-**4**-**6**-7-**4**-**6**-**8-4-6**-**8**-9-**4-10**-12-13; p12 = 10

m13 =1-**4**-**6**-7-**4**-**6**-**8**-**4-10**-11-13; p13 = 7

m14 =1-**4**-**6**-7-**4**-**6**-**8**-9-**4-10**-11-13; p14 = 7

m15 =1-**4**-**6**-**8**-**4**-**6**-**8**-9-**4-10**-11-13; p15 = 8

m16 =1-**4**-**6**-7-**4**-**6**-**8-4-6**-**8**-9-**4-10**-11-13; p16 = 10

S3 = 96

Оценка сложности согласно метрике Маккейба

S = Z = 5

Матрица смежности:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 4 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 4 | 1 |  |  | 1 | 1 | 1 |  |  |  |  |
| 6 |  | 1 |  |  |  |  |  |  |  |  |
| 7 |  |  | 1 |  |  |  |  |  |  |  |
| 8 |  |  | 1 |  |  |  |  |  |  |  |
| 9 |  |  |  |  | 1 |  |  |  |  |  |
| 10 |  | 1 |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  | 1 |  |  |  |
| 12 |  |  |  |  |  |  | 1 |  |  |  |
| 13 |  |  |  |  |  |  |  | 1 | 1 |  |

Матрица достижимости:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 4 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 12 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 13 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

# Вывод

В результате проведенной работы была создана программа, соответствующая требованиям варианта 31.

Были сформированы блок-схемы основных алгоритмов программы, не считая инициализации пользовательского интерфейса.

На основе алгоритмов созданы управляющие графы программ, проведено их упрощение с проверкой цикломатических чисел графов.

Был проведен анализ согласно метрике Холстеда, в ходе которого было определено, что длина реализации программы на 19% больше, чем длина программы, рассчитанная исходя из размеров словарей. Это говорит о возможном наличии несовершенств кода. Уровень реализации программы мал – составил всего 0.1, что говорит о малой степени компактности программы.

Были проведены оценки сложности алгоритмов согласно трем критериям. Наибольшая сложность оказалась у метода GetMinMax: S1 = 11, S2 = 15, S3 = 96. Оценка сложности согласно критерию Маккейба(Z = 5) показала, что этот метод имеет относительно малую сложность.