Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра інформатики

Звіт з лабораторної роботи №1

з предмету: «Теорія програмування»

Виконав:

Студент групи ІТІНФ-21-2

Марченко Богдан Олександрович

Lab 1. Refactoring, clean code (20 points)

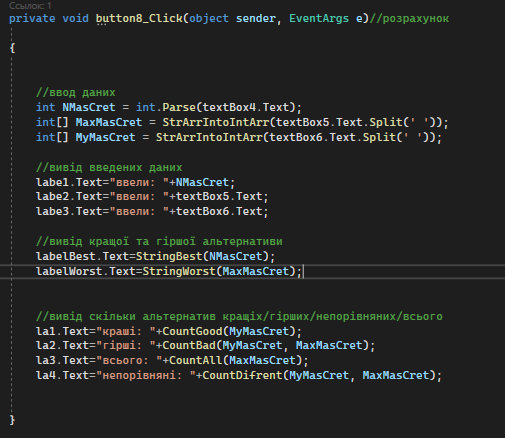
Task: Take any dirty code, cover it with tests and refactor.

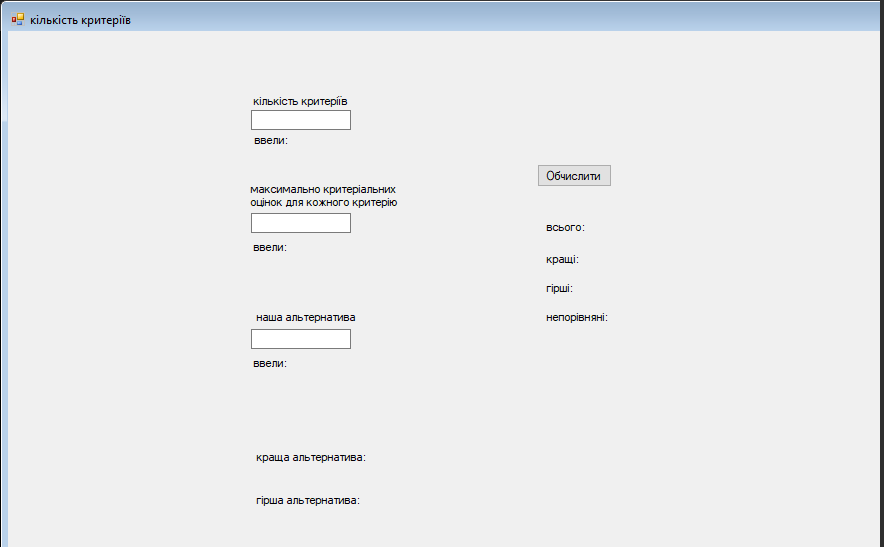
Завдання: Візьміть будь-який брудний код, покрийте його тестами та виконайте рефакторинг.

Візьмемо код програми для лаб роб 1 з курсу Теорія прийняття рішень:

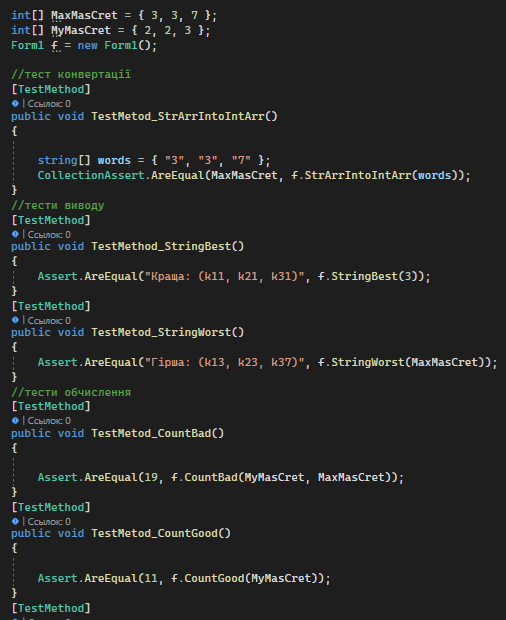


Код має невпорядкований вигляд, все написано майже без використання функцій, розіб’ємо код на окремі функції, маємо(також був прибраний зайвий функціонал):



Інтерфейс програми: 

Додамо по одному unit тесту для кожного методу:



Висновок: на лабораторній роботі ми взяли брудний код, покрили його тестами та виконали рефакторинг.

Код програми:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Краші\_гірші\_непорівняня\_альтернатив\_V2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

//МЕТОДИ:

public int[] StrArrIntoIntArr(string[] words)

{

int[] result = new int[words.Length];

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

{

result[i]=int.Parse(words[i]);

}

return result;

}

//вивід

public string StringBest(int NmasCret)

{

string str="";

for (int i = 0; i<NmasCret; i++)

{

str+="k"+(i+1)+1+", ";

}

str=str.Remove(str.Length - 2);

str="Краща: ("+str+")";

return str;

}

public string StringWorst(int[] MaxMasCret)

{

string str = "";

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

{

str+="k"+(i+1)+MaxMasCret[i]+", ";

}

str=str.Remove(str.Length - 2);

str="Гірша: ("+str+")";

return str;

}

//обрахунок

public int CountGood(int[] MyMasCret)

{

int bad = 1;

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

{

bad\*= MyMasCret[i];

}

bad -= 1;

return bad;

}

public int CountBad(int[] MyMasCret, int[] MaxMasCret)

{

int good = 1;

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

{

good\*=MaxMasCret[i]-MyMasCret[i]+1;

}

good-=1;

return good;

}

public int CountAll(int[] MaxMasCret)

{

int all = 1;

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

{

all \*=MaxMasCret[i];

}

return all;

}

public int CountDifrent(int[] MyMasCret, int[] MaxMasCret)

{

return CountAll(MaxMasCret)-CountGood(MyMasCret)-CountBad(MyMasCret, MaxMasCret)-1;

}

private void button8\_Click(object sender, EventArgs e)//розрахунок

{

//ввод даних

int NMasCret = int.Parse(textBox4.Text);

int[] MaxMasCret = StrArrIntoIntArr(textBox5.Text.Split(' '));

int[] MyMasCret = StrArrIntoIntArr(textBox6.Text.Split(' '));

//вивід введених даних

labe1.Text="ввели: "+NMasCret;

labe2.Text="ввели: "+textBox5.Text;

labe3.Text="ввели: "+textBox6.Text;

//вивід кращої та гіршої альтернативи

labelBest.Text=StringBest(NMasCret);

labelWorst.Text=StringWorst(MaxMasCret);

//вивід скільки альтернатив кращіх/гірших/непорівняних/всього

la1.Text="краші: "+CountGood(MyMasCret);

la2.Text="гірші: "+CountBad(MyMasCret, MaxMasCret);

la3.Text="всього: "+CountAll(MaxMasCret);

la4.Text="непорівняні: "+CountDifrent(MyMasCret, MaxMasCret);

}

}

}

Код Unit tests:

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using Краші\_гірші\_непорівняня\_альтернатив\_V2;

namespace UnitTestProject1

{

[TestClass]

public class UnitTest1

{

int[] MaxMasCret = { 3, 3, 7 };

int[] MyMasCret = { 2, 2, 3 };

Form1 f = new Form1();

//тест конвертації

[TestMethod]

public void TestMetod\_StrArrIntoIntArr()

{

string[] words = { "3", "3", "7" };

CollectionAssert.AreEqual(MaxMasCret, f.StrArrIntoIntArr(words));

}

//тести виводу

[TestMethod]

public void TestMethod\_StringBest()

{

Assert.AreEqual("Краща: (k11, k21, k31)", f.StringBest(3));

}

[TestMethod]

public void TestMetod\_StringWorst()

{

Assert.AreEqual("Гірша: (k13, k23, k37)", f.StringWorst(MaxMasCret));

}

//тести обчислення

[TestMethod]

public void TestMetod\_CountBad()

{

Assert.AreEqual(19, f.CountBad(MyMasCret, MaxMasCret));

}

[TestMethod]

public void TestMetod\_CountGood()

{

Assert.AreEqual(11, f.CountGood(MyMasCret));

}

[TestMethod]

public void TestMetod\_CountAll()

{

Assert.AreEqual(63, f.CountAll(MaxMasCret));

}

[TestMethod]

public void TestMetod\_CountDifrent()

{

Assert.AreEqual(32, f.CountDifrent(MyMasCret,MaxMasCret));

}

}

}