

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №2**

**з дисципліни «Проектний практикум»**

**на тему:**

**«Реалізація виділених можливостей. Розвиток проекту»**

Виконав:

студент гр. ПЗ2011

Кулик С. В.

Прийняв:

Іванов О.П.

Дніпро, 2023

**Тема.** Реалізація виділених можливостей. Розвиток проекту.

1. **Завдання**

Реалізація текстового редактора з можливістю рефакторингу вихідних текстів за завданням ЛР1. Моделювання зауважень замовника щодо внесення доопрацювань для наступної версії продукту.

1. **Код програми**
   * + 1. Form1

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 ConsoleApp1

{

public partial class Form1 : Form

{

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

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

{

string new\_name = Convert.ToString(textBoxConst.Text);

string name = Convert.ToString(textBoxNum.Text);

string text = Convert.ToString(richTextBox1.Text);

Refactor\_Test r = new Refactor\_Test();

bool inComent = checkBoxSearchInComment.Checked;

bool inString = checkBoxSearchInString.Checked;

string rez = r.changetoNum(text, name, new\_name, inString, inComent);

richTextBox1.Text = Convert.ToString(rez);

}

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

{

string name = Convert.ToString(richTextBox1.Text);

string new\_name = Convert.ToString(textBoxPar.Text);

Refactor\_Test r = new Refactor\_Test();

bool inComent = checkBoxSearchInComment.Checked;

bool inString = checkBoxSearchInString.Checked;

string text = Convert.ToString(richTextBox1.Text) + Convert.ToString(textBoxPar.Text);

string rez = r.addParametr(text, name, new\_name, inString, inComent);

richTextBox1.Text = Convert.ToString(rez);

}

}

}

* + - 1. Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

namespace ConsoleApp1

{

public class Refactor\_Test

{

public string changetoNum(string text, string value, string Newvalue, bool inString = false, bool inComment = false)

{

string[] line = text.Split(new char[] { '\"' });

string result = "";

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

{

if (i % 2 == 1)

{

if (inString)

{

result += "\"";

result += findInString(line[i], value, Newvalue);

result += "\"";

}

else

{

result += "\"";

result += line[i];

result += "\"";

}

}

else { result += findVarInLine(line[i], value, Newvalue, inComment); }

}

return result;

}

public string addParametr(string text, string name, string newName, bool inString = false, bool inComment = false)

{

string[] line = text.Split(new char[] { '\"' });

string result = "";

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

{

if (i % 2 == 1)

{

if (inString)

{

result += "\"";

result += findInString(line[i], name, newName);

result += "\"";

}

else

{

result += "\"";

result += line[i];

result += "\"";

}

}

else{result += findVarInLine(line[i], name, newName, inComment);}

}

return result;

}

private string SearchConst(string lines, string name, string newName)

{

string res = "";

bool isEnd = false;

while (!isEnd)

{

int index = lines.IndexOf(name);

if (index >= 0)

{

if (lines[index + name.Length] == ' ' || lines[index + name.Length] == ';' || lines[index + name.Length] == ')')

{

res += lines.Substring(0, index);

res += newName;

lines = lines.Substring(index + name.Length);

}

else

{

res += lines.Substring(0, index + 1);

lines = lines.Substring(index + 1);

}

}

else

{

res += lines;

isEnd = true;

}

}

return res;

}

private string findVarInLine(string text, string name, string newName, bool inComent)

{

string result = "";

bool isEnd = false;

while (!isEnd)

{

int index = text.IndexOf("//");

if (index >= 0)

{

result += SearchConst(text.Substring(0, index), name, newName);

int indexEndComent = text.Substring(index).IndexOf("\n");

if (indexEndComent >= 0)

{

if (inComent)

{

result += findInString(text.Substring(index, indexEndComent), name, newName);

}

else

{

result += text.Substring(index, indexEndComent);

}

text = text.Substring(index + indexEndComent);

}

else

{

if (inComent)

{

result += findInString(text.Substring(index), name, newName);

}

else

{

result += text.Substring(index);

}

isEnd = true;

}

}

else

{

result += SearchConst(text, name, newName);

isEnd = true;

}

}

return result;

}

private string findInString(string text, string name, string newName)

{

string result = "";

bool isEnd = false;

while (!isEnd)

{

int index = text.IndexOf(name);

if (index >= 0 && text[index + name.Length] == ' ')

{

result += text.Substring(0, index);

result += newName;

text = text.Substring(index + name.Length);

}

else

{

result += text;

isEnd = true;

}

}

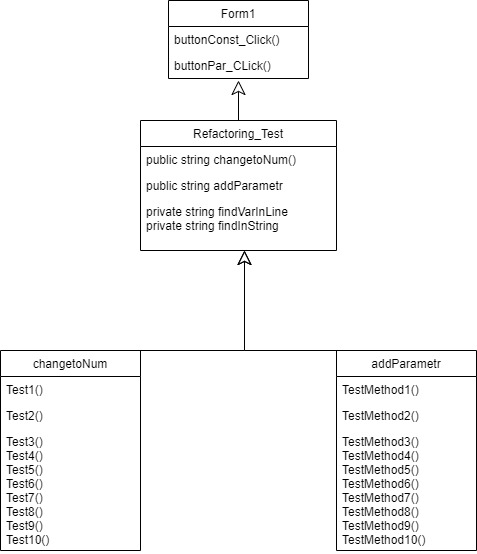
return result;

}

}

}

1. **Діаграма класів**

****

1. Опис побажань покращення функціональних можливостей системи.

* Покращення інтерфейсу додатку
* Додавання більш розгорнутих коментарів до коду
* Додавання функції очищення введених даних в текстбокси

**Висновок**

За тестуванням програми було збудовано список покращень які будуть виправлені та реалізовані покращення для більш зручної роботи програми. **Модульне тестування (Unit testing)** – тестування кожної атомарної функціональності додатку окремо, в штучно створеному середовищі. Саме потреба у створенні штучної робочого середовища для певного модуля, вимагає від тестувальника знань в автоматизації тестування програмного забезпечення, деяких навичок програмування. Дане середовище для деякого юніта створюється за допомогою драйверів і заглушок.