МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,

СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА» (СПбГУТ)

Санкт-Петербургский колледж телекоммуникаций им. Кренкеля

##### Лабораторно-практическая работа № 14

по профессиональному модулю

**ПМ.01 «ПРИКЛАДНОЕ ПРОГРАММИРОВАНИЕ»**

Студент гр. 502\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Савва В.И.

(подпись)

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Параничев

(оценка и подпись)

Санкт-Петербург

2022 год

РАЗРАБОТКА КЛАССА ДЛЯ ВЫВОДА ПОЛЬЗОВАТЕЛЬСКОЙ ОТЛАДОЧНОЙ ИНФОРМАЦИИ В ТЕКСТОВЫЙ ФАЙЛ

Код:

Form1.cs

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;

using System.IO;

using WinFormsApp\_LogFiles;

namespace lab14

{

public partial class form\_Main : Form

{

LogData \_logData = null; // ссылка на экземляр класса данных лог‐файла

bool \_log\_click = false; // флаг записи в лог по событию "Click"

bool \_log\_mouseclick = false; // флаг записи в лог по событию "MouseClick"

bool \_enabled\_log = false; // флаг разрешения записи в лог

public form\_Main()

{

InitializeComponent();

{ // инициализируем экземпляр класса данных лог‐файла

\_logData = new LogData();

InitializeComponent();

openFileDialog.InitialDirectory =

Directory.GetCurrentDirectory();// начальный каталог для открытия файлов

}

//‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

// ‐‐‐ END OF USER DEFINED METHODS

}

private void button\_MouseClick(object sender, MouseEventArgs e)

{

Control c = (Control)sender;

string str\_add = "Событие MouseClick экземпляра компонента" +

c.Name + "координаты (" +

e.X + "," + e.Y + ")";

\_logData.AddToLog(str\_add + "\n");

toolStripStatusLabel.Text = str\_add;

richTextBox\_ShowLog.AppendText(str\_add);//добавляем в список

}

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

{

Control c = (Control)sender;

string str\_add = "Событие Click экземпляра компонента" + c.Name

+ "координаты не определены";

\_logData.AddToLog(str\_add + "\n");

toolStripStatusLabel.Text = str\_add;

}

private void button\_SwitchLog\_Click(object sender, EventArgs e)

{

\_enabled\_log = !\_enabled\_log; // инвертируем флаг разрешения лога

if (\_enabled\_log) // если флаг теперь разрешен

{

button\_SwitchLog.Text = "Выключить лог";

}

else

{

button\_SwitchLog.Text = "Включить лог";

}

}

private void checkBox\_Click\_CheckedChanged(object sender, EventArgs e)

{

\_log\_click = checkBox\_Click.Checked && \_enabled\_log; // получаем флаг о разрешении клика

foreach (Control c in this.Controls) // для всех контролов

{

//if c.

if (\_log\_click) // добавляем событие Click, если разрешено

{

c.Click += new EventHandler(button\_Click);

}

else // иначе удаляем событие Click

{

c.Click -= new EventHandler(button\_Click);

}

}

}

private void comboBox\_MouseClick\_CheckedChanged(object sender, EventArgs e)

{

\_log\_mouseclick = comboBox\_MouseClick.Checked && !\_enabled\_log;

// получаем флаг о разрешении клика

foreach (Control c in this.Controls) // для всех контролов

{

//if c.

if (\_log\_mouseclick) // добавляем событие Click, если разрешено

{

c.MouseClick += new MouseEventHandler(button\_MouseClick);

}

else // иначе удаляем событие Click

{

c.MouseClick += new MouseEventHandler(button\_MouseClick);

}

}

}

private void button\_OpenLog\_Click(object sender, EventArgs e)

{

if (openFileDialog.ShowDialog() == DialogResult.OK) // если выбран файл загрузки

{

richTextBox\_ShowLog.Clear();//очищаем список

string str\_file = openFileDialog.FileName; // имя выбранного пользователем файла

string[] str\_dat = LogData.ShowLogData(str\_file); //загружаем данные файла

richTextBox\_ShowLog.Lines = str\_dat;

}

}

private void button\_ClearList\_Click(object sender, EventArgs e)

{

richTextBox\_ShowLog.Clear();

}

private void timer\_Tick(object sender, EventArgs e)

{

\_logData.SaveLog(); // записываем новые строки в лог‐файл

}

}

}

LogData.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

namespace WinFormsApp\_LogFiles

{

public class LogData // класс для работы с отладочными данными

{

private StringBuilder \_sb; // "построитель" строки

private string \_log\_file; // путь к лог файлу

public LogData(string log\_file = null) // общедоступный конструктор

{

\_sb = new StringBuilder(); // инициализируем объект "построителя"строки

if (log\_file == null) // если задан пустой лог, формируем его имя

{

DateTime current\_time = DateTime.Now; // текущее время

string current\_dir = Directory.GetCurrentDirectory(); // текущий каталог

\_log\_file = current\_dir + current\_time.Year.ToString() + "\_" +

current\_time.Month.ToString() +

current\_time.Day.ToString() +

current\_time.Hour.ToString() +

current\_time.Minute.ToString() + ".log"; // имя лог‐файла

}

else // иначе запоминаем текущее имя

{

\_log\_file = log\_file;

}

}

public void AddToLog(string str\_add) // добавление строки в лог‐файл

{

\_sb.Append(str\_add);

}

public void SaveLog() // сохранение лога в файл

{ // добавляем данные в файл

File.AppendAllText(\_log\_file, \_sb.ToString(),

Encoding.Default);

\_sb.Clear(); // очищаем уже добавленные строки лог‐файла

}

public static string[] ShowLogData(string log\_file)

{ // загрузка одного из лог файлов во внешние переменные

string[] str\_ret = File.ReadAllLines(log\_file,

Encoding.Default);

return str\_ret; // возвращаем все считанные строки

}

}

}

Результат:



