**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

по дисциплине: «Объектно-ориентированное проектирование и программирование»

на тему: **«**Делегаты и события**»**

Выполнил: студент гр. ИТИ-21

Бахонько А. М.

Принял: преподаватель

Башаримов Ю. С.

Гомель 2024

**Цель работы**: изучить работу с делегатами и лямбда-выражениями; научится создавать динамический интерфейс без использования конструктора.

**Задание:**

• Разработать графическое приложение (*WinForm*) без использования конструктора.

• Интерфейс должен генерироваться динамически в коде программы.

• События должны добавляться несколькими способами:

– лямбда-выражения;

– методы реализующие делегат.

|  |  |
| --- | --- |
| № | Условие |
| 2 | Текстовый редактор. Добавить компонент для работы с текстом. Поместить кнопки для открытия, сохранения и создания нового файла. |

**Ход работы**

На рисунке 1 представлено окно текстового редактора в приложении *WindowForm*.

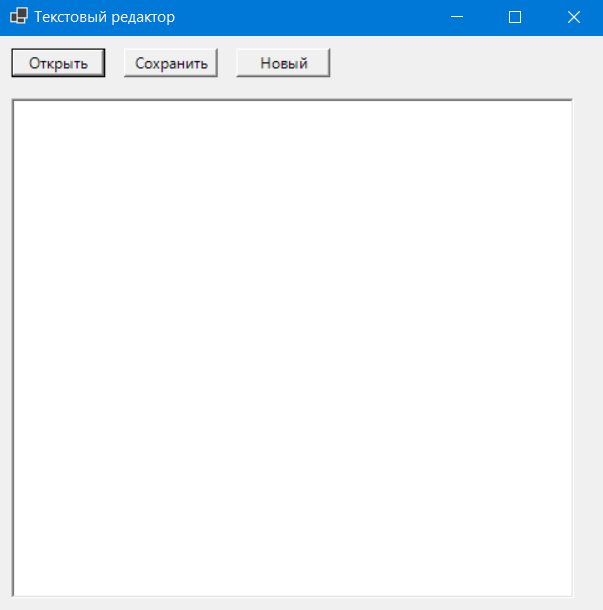


Рисунок 1 – Текстовый редактор

В текстовом редакторе есть три кнопки, что реализуют события с помощью лямбда-выражений и текстовое поле, которое содержит делегат для метода, что совершает действия при написании определённых слов. Сначала кнопка «Открыть**»**. При нажатии на эту кнопку появится окно с выбором тестового файла, содержимое которого потом будет отображено в текстовом поле редактора. Это окно представлено на рисунке 2.

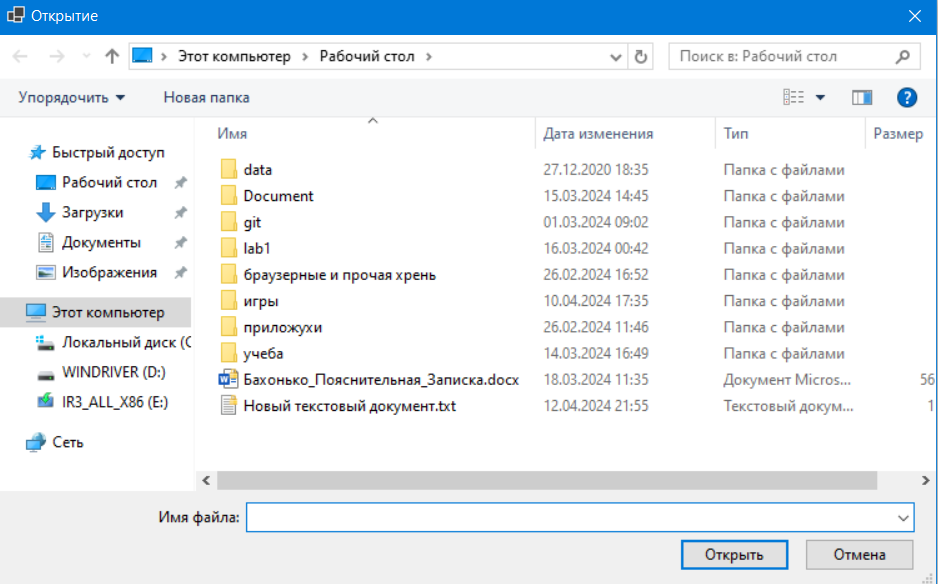


Рисунок 2 – Окно открытия файла

После выбора файла окно закрывается, и внутри редактора появляется текст. Результат представлен на рисунке 3.

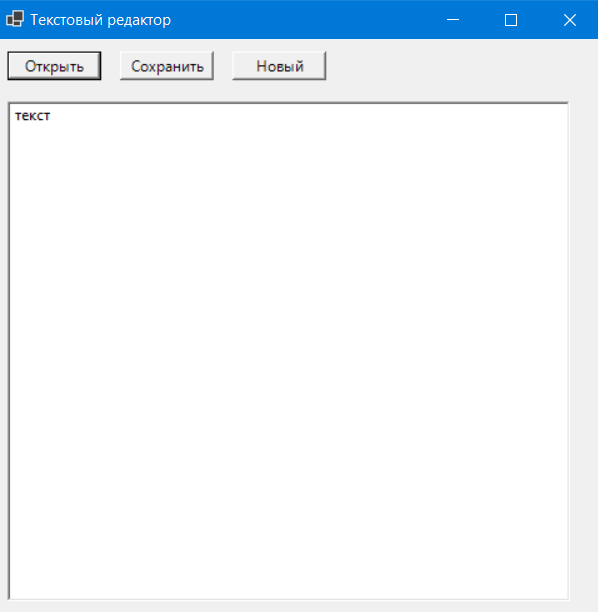


Рисунок 3 – Результат открытия файла

Далее будет кнопка «Сохранить**»**. При нажатии на эту кнопку появится окно с выбором, куда сохранить файл с содержимым текстового поля редактора. После выбора места, в данном месте появится файл с текстом. Результат представлен на рисунке 4.

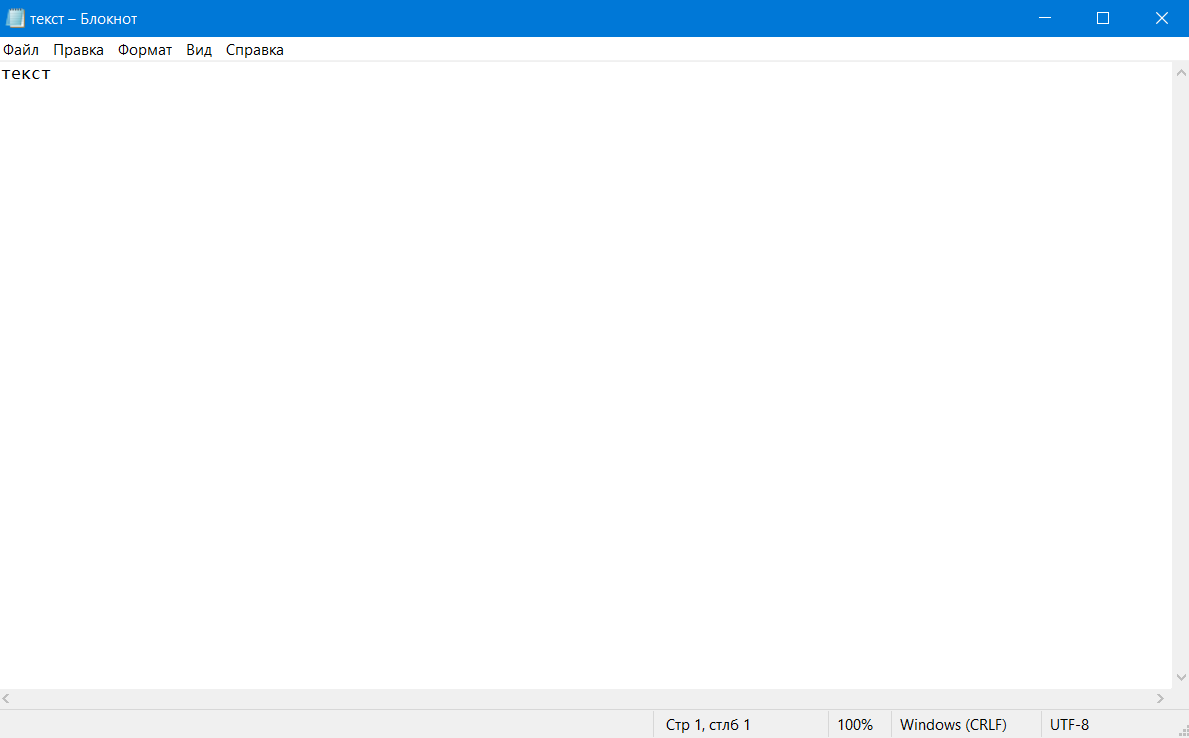


Рисунок 4 – Результат сохранения файла

Теперь кнопка «Новый**»**. При нажатии на эту кнопку текстовое поле редактора станет пустым. Результат представлен на рисунках 5.

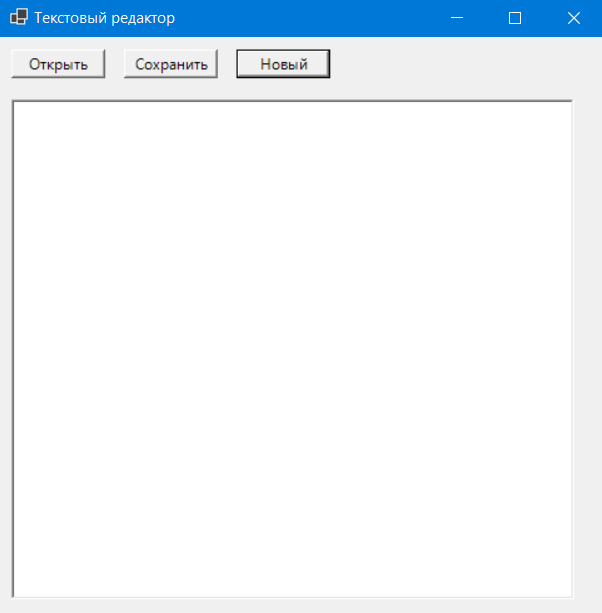


Рисунок 5 – Результат создания нового файла

Также такие же действия могут осуществляться и при написании определённых слов внутри текстового поля редактора. Например, при написании слова «сохранить**»** появится окно сохранения файла. Результат представлен на рисунках 6.

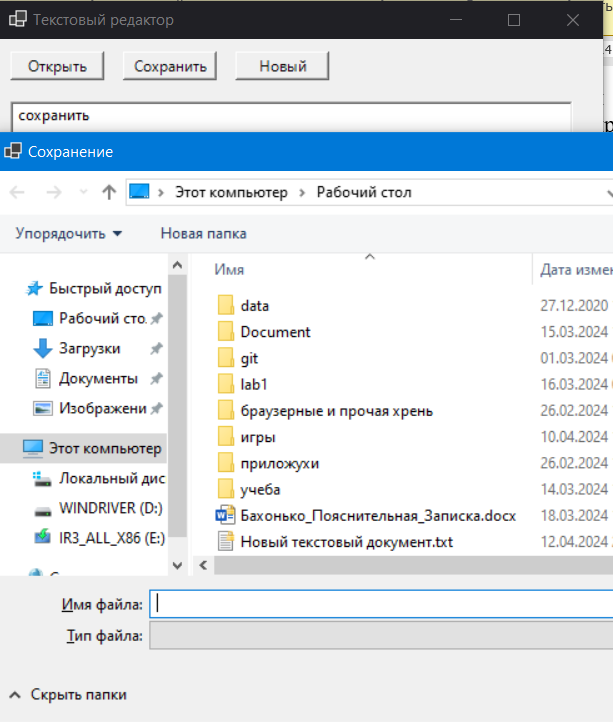


Рисунок 6 – Результат написания слова в текстовом поле

В приложении А представлен код программы.

На рисунке 7 представлен репозиторий проекта.

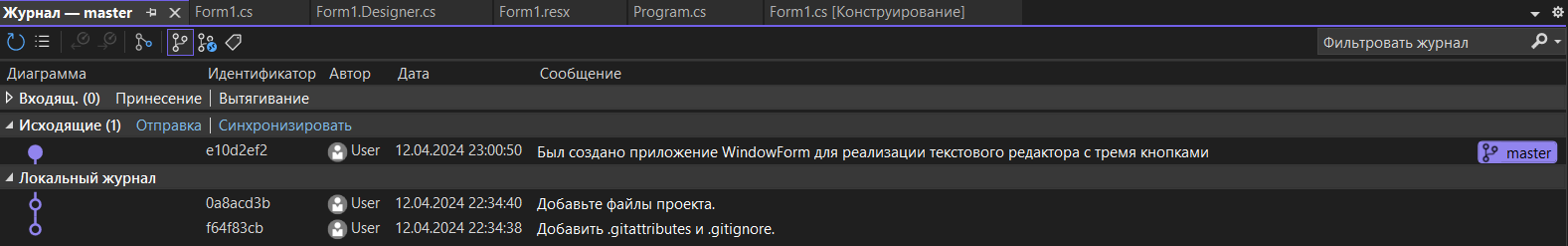


Рисунок 7 – Репозиторий проекта

**Вывод:** в результате выполнения лабораторной работы создали приложение *WindowForm* без конструктора; реализовали динамический интерфейс для генерация приложения; осуществили работу с событиями с помощью лямбд-выражений и делегатов.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Текст программы**

**Program.cs**

namespace TextEditor

{

internal static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

// Создание динамического интерфейса

Form form = new Form();

form.Text = "Текстовый редактор";

form.Size = new Size(500, 500);

form.StartPosition = FormStartPosition.CenterScreen;

// Добавление кнопок для открытия, сохранения и создания нового файла

Button openBtn = new Button();

openBtn.Text = "Открыть";

openBtn.Location = new Point(10, 10);

Button saveBtn = new Button();

saveBtn.Text = "Сохранить";

saveBtn.Location = new Point(100, 10);

Button newBtn = new Button();

newBtn.Text = "Новый";

newBtn.Location = new Point(190, 10);

// Добавление компонента для работы с текстом

TextBox textBox = new TextBox();

textBox.Multiline = true;

textBox.Size = new Size(450, 400);

textBox.Location = new Point(10, 50);

textBox.ReadOnly = false;

// Добавление событий с помощью лямбда-выражений

openBtn.Click += (sender, e) =>

{

OpenFileDialog openFileDialog = new OpenFileDialog();

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

string fileName = openFileDialog.FileName;

textBox.Text = File.ReadAllText(fileName); // код для открытия файла

}

};

saveBtn.Click += (sender, e) =>

{

SaveFileDialog saveFileDialog = new SaveFileDialog();

if (saveFileDialog.ShowDialog() == DialogResult.OK)

{

string fileName = saveFileDialog.FileName; // код для сохранения файла

File.WriteAllText(fileName, textBox.Text);

}

};

newBtn.Click += (sender, e) =>

{

textBox.Clear(); // очистка текстового поля

};

// Добавление событий с помощью методов, реализующих делегат

textBox.TextChanged += new EventHandler(TextChanged);

void TextChanged(object sender, EventArgs e)

{

if (textBox.Text.Contains("сохранить") || textBox.Text.Contains("Cохранить") || textBox.Text.Contains("save") || textBox.Text.Contains("Save"))

{

SaveFileDialog saveFileDialog = new SaveFileDialog();

if (saveFileDialog.ShowDialog() == DialogResult.OK)

{

string fileName = saveFileDialog.FileName;

File.WriteAllText(fileName, textBox.Text);

}

}

else if (textBox.Text.Contains("открыть") || textBox.Text.Contains("Открыть") || textBox.Text.Contains("open") || textBox.Text.Contains("Open"))

{

OpenFileDialog openFileDialog = new OpenFileDialog();

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

string fileName = openFileDialog.FileName;

textBox.Text = File.ReadAllText(fileName);

}

}

else if (textBox.Text.Contains("новый") || textBox.Text.Contains("Новый") || textBox.Text.Contains("new") || textBox.Text.Contains("New"))

{

textBox.Clear();

}

}

// Добавление компонентов на форму

form.Controls.Add(openBtn);

form.Controls.Add(saveBtn);

form.Controls.Add(newBtn);

form.Controls.Add(textBox);

// Отображение формы

Application.Run(form);

}

}

}