Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра програмування та захисту інформації

**Звіт**

з виконаної лабораторної роботи № 2

дисципліни “ Системне програмування”

на тему

«**Проектування інтерфейсних додатків в середовищі RAD.**»

Виконав :

студент академічної групи КІ-15

Аннаєв А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перевірив :

Викладач

Константинова Л.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий- 2017

**Лабораторна робота №2**

ТЕМА: Проектування інтерфейсних додатків в середовищі RAD.

МЕТА: Отримати практичні навики в проектуванні користувальницького інтерфейсу в середовищі RAD.

ЗНАТИ: Мову програмування C або Pascal.

ВМІТИ: Інсталювати RAD (наприклад Builder або Delphi). Знати основи розробки програм під

ОС Windows.

ЗАВДАННЯ

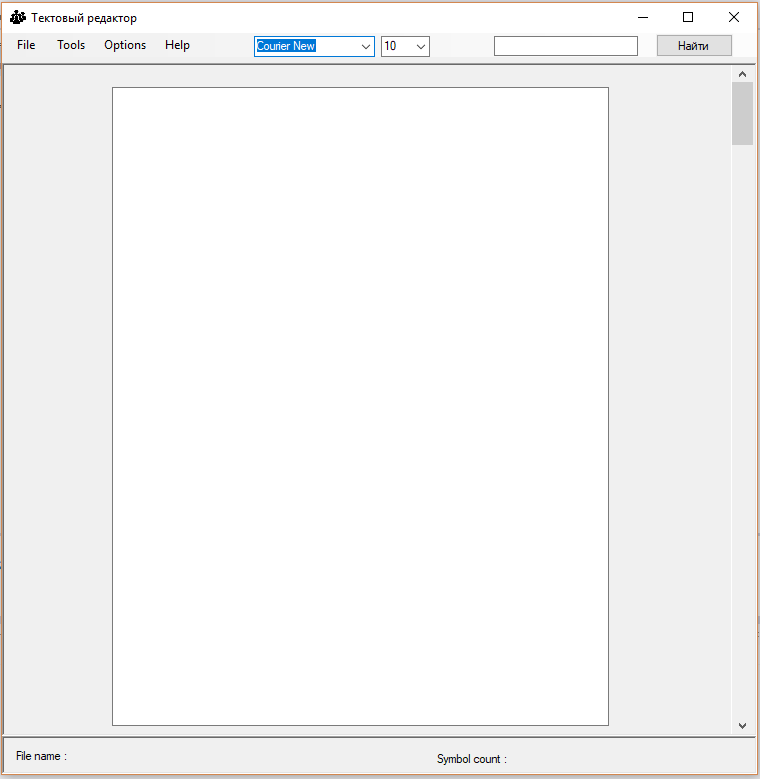
Спроектувати найпростіший текстовий редактор з наступними виконавчими функціями: відкрити, зберегти, виконувати пошук і заміну тексту, міняти шрифт, розмір і колір тексту. Використовувати стандартні компоненти TLabel, TButton, TMemo, TOpenDialog, TSaveDialog, TFontDialog, TFindDialog чи інші аналоги.

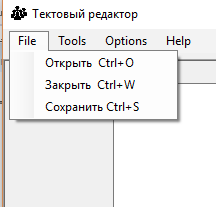
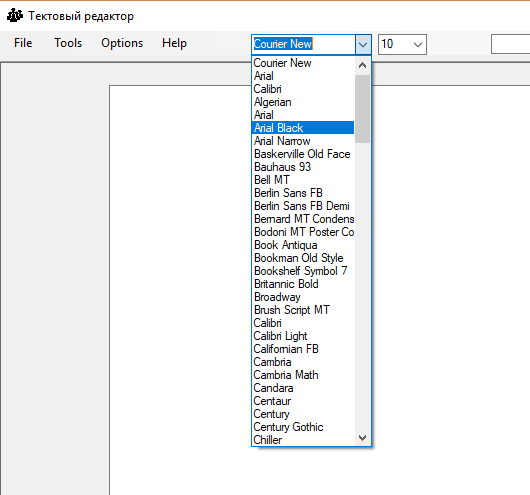
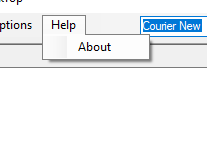
Для здачі ЛР необхідно:

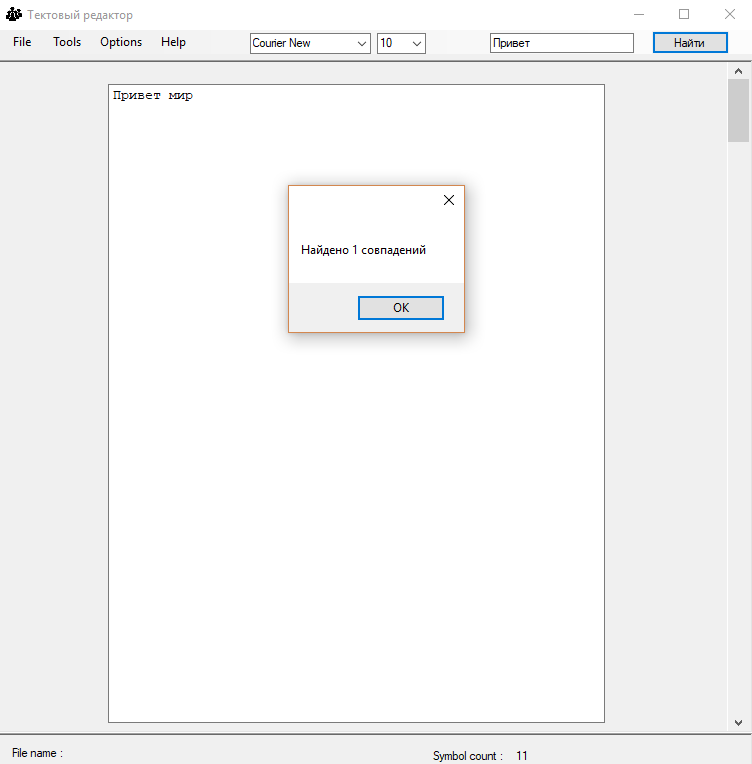
1. Реалізувати програму;

2. Відповісти на контрольні запитання.

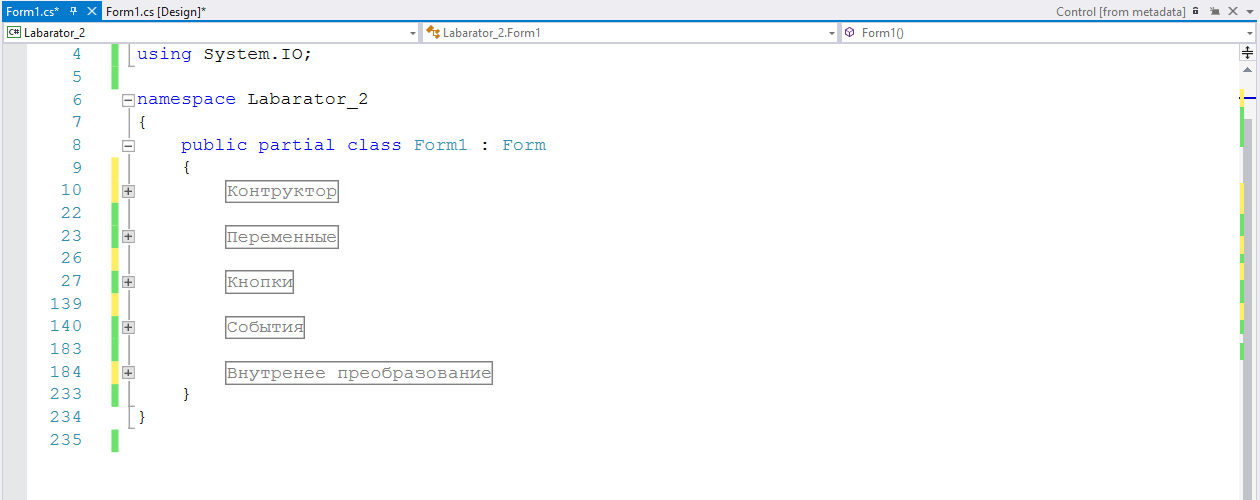
У звіт ЛР помістити вихідний код програми, графічний матеріал, відповіді на контрольні запитання. ЛР оформляти по загальноприйнятій формі аналогічно до інших предметів. Для підвищення балів ЛР, необхідно на колоквіумі відповісти на додаткові запитання, що стосуються ЛР.



Функционал редактора



Поиск слова



Структура проекта

#region Контруктор

/// <summary>

/// Конструктор по умолчанию

/// </summary>

public Form1()

{

InitializeComponent( );

InitFont( );

}

#endregion

#region Переменные

string FileName = string.Empty;

#endregion

#region Кнопки

/// <summary>

/// Открытие файла

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

OpenFileDialog openFileDialog = new OpenFileDialog( );

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

{

FileName = openFileDialog.FileName;

StreamReader str = new StreamReader(FileName);

txtEditor.Text = str.ReadToEnd( );

str.Close( );

lblFileName.Text = FileName;

lblFileName.Visible = true;

lblSymCount.Text = Convert.ToString(txtEditor.Text.Length + 1);

lblSymCount.Visible = true;

}

}

/// <summary>

/// Закрытие файла

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

// перед закрытиев приложения справшиваем пользователя сохранить ли документ

var result = MessageBox.Show("Сохранить документ ?", "Сохранено", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

// если результат положительный

if (result != DialogResult.OK)

SaveFile( );

Close( );

}

/// <summary>

/// Сохранение файла

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

SaveFile( );

}

/// <summary>

/// Поиск слова в тексте

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void btnFind\_Click\_1( object sender, EventArgs e )

{

// текст полученые с редактора

string text = txtEditor.Text;

// слово которое необходимо найти

string find\_word = txtWord.Text;

// количество найденных слов

int iterator = 0;

// если текстовый редактор не пуст

if (!string.IsNullOrEmpty(text))

{

// и пользователь ввел слово

if (!string.IsNullOrEmpty(find\_word))

{

// получаем массив все слов в редакторе

string[] str = text.Split(new Char[] { ' ' });

//проход по массиву

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

// если нашли совпадение

if (String.Compare(str[i], find\_word) == 0)

// увеличиваем итератор показателя слов

iterator++;

// после прохода проверяем нашли ли мы совпадения

if (iterator > 0)

{

string message = "Найдено " + Convert.ToString(iterator) + " совпадений";

MessageBox.Show(message);

}

else

MessageBox.Show("Совпадений не найдено");

}

else

MessageBox.Show("Введите слово для поиска");

}

else

MessageBox.Show("Страница пустая");

}

/// <summary>

/// О разработчике

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

Form2 form = new Form2( );

form.Show( );

}

#endregion

#region События

/// <summary>

/// Подсчет введенных символов

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void txtEditor\_KeyPress( object sender, KeyPressEventArgs e )

{

lblSymCount.Text = Convert.ToString(txtEditor.Text.Length + 1);

lblSymCount.Visible = true;

}

/// <summary>

/// Выбор шрифта

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void cmbFont\_SelectedValueChanged( object sender, EventArgs e )

{

string str = cmbKegle.Text;

if (str != "")

{

int size = int.Parse(str);

txtEditor.Font = new Font(cmbFont.Text, (float)size);

}

else

txtEditor.Font = new Font(cmbFont.Text, txtEditor.Font.Size);

}

/// <summary>

/// Выбор размера шрифта

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void cmbKegle\_SelectedValueChanged( object sender, EventArgs e )

{

txtEditor.Font = new Font(cmbFont.Text, (float)Convert.ToInt32(cmbKegle.Text));

}

#endregion

#region Внутренее преобразование

/// <summary>

/// Инициализация шрифтов

/// </summary>

private void InitFont()

{

// отображаем все установленные на машине шрифты

foreach (FontFamily oneFontFamily in FontFamily.Families)

cmbFont.Items.Add(oneFontFamily.Name);

// по умолчанию устанавливаем вторые элементы списка

cmbFont.SelectedIndex = 1;

cmbKegle.SelectedIndex = 1;

}

/// <summary>

/// Сохранение документа данная сигнатура используется в нескольких местаъ

/// для оптимизации и читабельности кода данная инструкция выведена в отдельную функцию

/// </summary>

private void SaveFile()

{

// диалог сохранения

SaveFileDialog saveFileDialog = new SaveFileDialog( );

// указываем расширения для сохраняемых файлов

saveFileDialog.Filter = "txt files (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";

// если все успешно

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

{

// получаем имя которое выбрал пользователь

string name = saveFileDialog.FileName;

try

{

// открываем поток чтения

StreamWriter streamWriter = new StreamWriter(name);

// формальное представление текта

string allData = txtEditor.Text;

// записываем в файл

streamWriter.WriteLine(allData);

// закрываем поток

streamWriter.Close( );

}

catch { }

// запрещаем отображение параметров файла который уже сохранен

lblSymCount.Visible = false;

lblFileName.Visible = false;

}

}

#endregion

**Контрольні запитання**

1. Що таке компонент RAD?

Компонент – один із елементів в палітрі компонентів, який використовується для створення інтерфейсу програми(Button, CheckBox, ListBox тощо.)

2. Які можливості надають компоненти RAD?

Вони допомагають створити і налагодити інтерфейс програми.

3. Які можливості мають стандартні компоненти RAD?

Вони дозволяють створити досить функціональний та інтуїтивно зрозумілий для користувача інтерфейс програми, дозволяючи додавати у вікно такі елементи, як кнопки, списки, перемикачі тощо.

4. Які можливості мають компоненти сторонніх розробників RAD?

Вони дозволяють впроваджувати в програму різноманітні додатки, які вдосконалюють програму(різноманітні плеєри, карти і т.д.)

5. Які основні сполучення клавіш використовуються для роботи в обраній RAD?

Ctrl C/Ctrl V – Копіювати/вставити

Ctrl F – Пошук в документі

Ctrl P – Друк коду

Ctrl F5 – Запуск без від лагодження програми

Ctrl F4 – Закриття поточного вікна

6. Що таке технологія програмування і життєвий цикл програми?

Життєвий цикл програми – сукупність окремих етапів робіт, що проводяться у заданому порядку протягом періоду часу, який починається з вирішення питання про розроблення програмного забезпечення і закінчується припиненням його використання.

Технологія програмування – комплекс програмних та організаторських заходів, що проводяться з метою створення програмного забезпечення та забезпечення його працездатності.