МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут **ІКНІ**

Кафедра **ПЗ**

ЗВІТ

До лабораторних робіт № 1-4

На тему: «Середовище програмування»

З дисципліни «Об’єктно-орієнтоване програмування»

Лектор: доцент каф. ПЗ

Коротєєва Т.О.

Виконала: ст.гр. ПЗ-23

Кохман О.В.

Прийняла: доцент каф. ПЗ

Коротєєва Т.О.

«_____»_____2022р.

∑ ____ _____.

Львів - 2022

Лабораторна робота №1

Тема: Ознайомлення із середовищем розробки Visual Studio 2022. Створення проекту та налаштування його властивостей.

Мета: Засвоїти принцип візуального програмування шляхом створення та налаштування проекту.

Індивідуальне завдання

1. Ознайомитись із середовищем Visual Studio 2022.
2. Створити новий проект. Зберегти його двома способами – через комбінації швидких клавіш та через меню.
3. Проглянути у вікні інспектора об’єктів властивості форми. Змінити назву форми та її розміри.
4. Запустити на виконання застосування.
5. Відкрити опції проекту, змінити налаштування на закладках General, Configuration Manager, NuGet Packages Manager. Запустити на виконання застосування.

Теоретичні відомості

Для того, щоб створити проект у Visual Studio 2022 обираємо Create a new project -> CLR Empty Project(.NET Framework) . Якщо такого пункту немає, то потрібно перейти до встановки інших інструментів і опцій і обрати Desktop Development with C++ -> C++/CLI support for v143 build tools(Latest). Створюємо новий файл: Source files -> Add -> New Item -> UI -> Windows Form.

Створяться 2 файли: MyForm.h і MyForm.cpp. В другому вводимо наступний код:

```
using namespace System;

using namespace System::Windows::Forms;

[STAThread]

void main(array<String^>^ arg) {

    Application::EnableVisualStyles();

    Application::SetCompatibleTextRenderingDefault(false);

    Project2::MyForm form;

    Application::Run(% form);
```

}

Далі перезапускаємо проект та заходимо в Properties -> Linker -> System -> SubSystem і обираємо свою платформу та Linker -> Advanced -> Empty Point пишемо main. В іншому випадку програма не запуститься.

Усі властивості форми і додаткових інструментів та дії над ними будуть у Properties і Events. Додактові інструменти , такі як button, textbox, checkbox тощо , можна знайти , натиснувши в лівому верхньому куті на ToolBox.

Закладка Properties, натискаючи на проект, містить вкладку General, де можна змінити імя, також директорію , тип конфігурації, обрати версію C/C++. Зверху можна обрати відповідну конфігурацію (Debug або Release), а в C/C++ -> Optimization змінити налаштування конфігурації.

Закладка Tools -> Options містить NuGet Package Manager, де є можливість змінити налаштування завантаження бібліотек.

Код програми

Назва файлу: MyForm.cpp

```
#include "MyForm.h"
using namespace System;
using namespace System::Windows::Forms;
[STAThread]
void main(array<String^>^ arg) {
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);
    Lab01::MyForm form;
    Application::Run(% form);
}
```

Назва файлу: MyForm.h

```
#pragma once
namespace Lab01 {
      using namespace System;
      using namespace System::ComponentModel;
      using namespace System::Collections;
      using namespace System::Windows::Forms;
      using namespace System::Data;
      using namespace System::Drawing;
      public ref class MyForm : public System::Windows::Forms::Form
      {
      public:
            MyForm(void)
            {
                  InitializeComponent();
            }
      protected:
            ~MyForm()
```

```
            {
                  if (components)
                  {
                        delete components;
                  }
            }
      private: System::Windows::Forms::Button^ button1;
      private: System::Windows::Forms::Button^ button2;
      private: System::Windows::Forms::TextBox^ textBox1;
      private: System::Windows::Forms::TextBox^ textBox2;
      private: System::Windows::Forms::TextBox^ textBox3;
      private: System::Windows::Forms::Button^ button3;
            System::ComponentModel::Container ^components;
#pragma region Windows Form Designer generated code
            void InitializeComponent(void)
            {
                  //creating components
            }

#pragma endregion
      private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e) {
            this->Text = "Changed name";
            this->Size = System::Drawing::Size(550, 550);
            this->BackColor = System::Drawing::Color::Violet;
      }
      private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e) {
            int a, b, sum;
            a = System::Convert::ToInt16(textBox1->Text);
            b = System::Convert::ToInt16(textBox2->Text);
            sum = a + b;
            textBox3->Text = System::Convert::ToString(sum);
      }
private: System::Void button3_Click(System::Object^ sender,
System::EventArgs^ e) {
      button3->Width = 100;
      button3->Height = 100;
      button3->Location = System::Drawing::Point(200, 200);
}
private: System::Void button1_MouseClick(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) {
}
};
}
```

Протокол роботи

Рис. 1.1 Форма при запуску програми

Рис. 1.2 Результат натискання кнопки з текстом “to change size and name of form”

Рис. 1.3 Результат натискання кнопки з текстом “to sum”

Рис. 1.4 Результат натискання кнопки з текстом “to move”

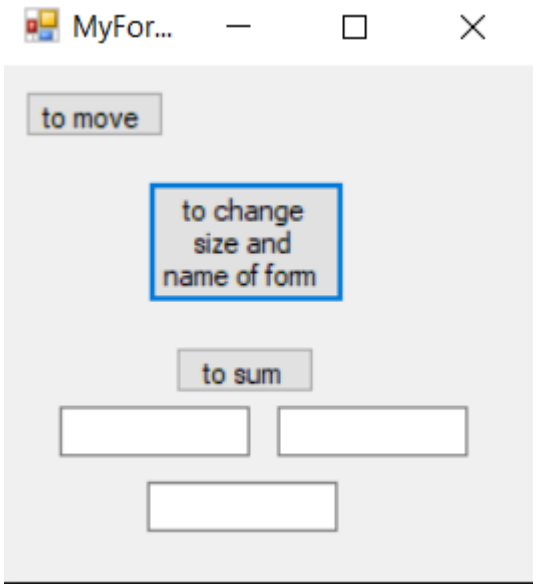

Рис. 1.1 Форма при запуску програми з доданими кнопками та текстбоксами

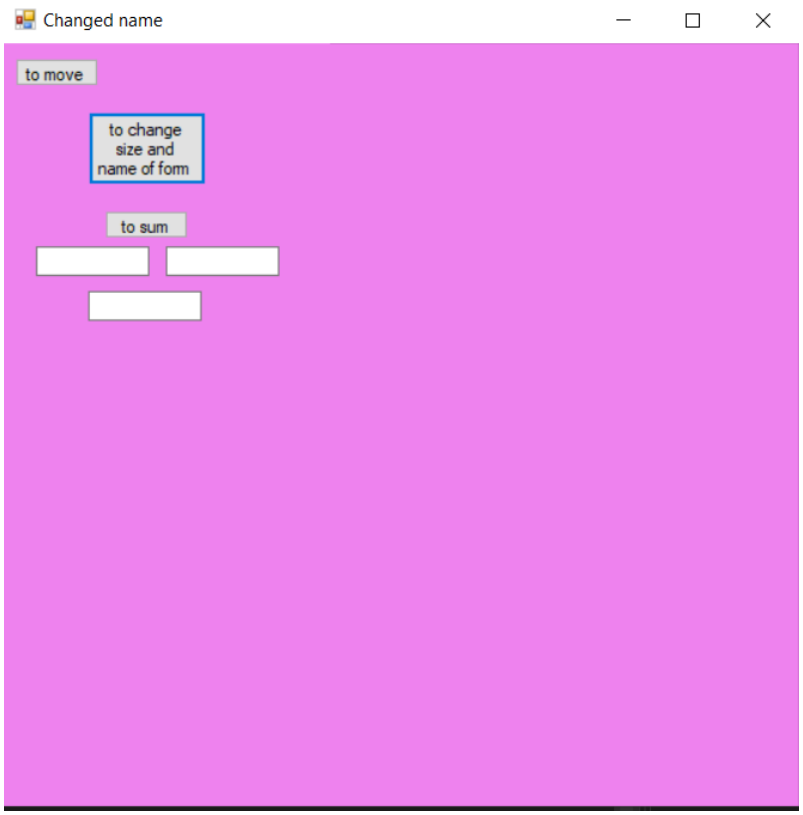

Рис. 1.2 Результат натискання кнопки з текстом “to change size and name of form” . Змінилась назва форми на “Changed name”, а також колір та розмір форми.

Рис. 1.3 Результат натискання кнопки з текстом “to sum”. Введено числа 2 і 3 в текстбоксах 1 і 2, а в текстбоксі 3 виводиться результат, що дорівнює 5, при натисканні на кнопку

Рис. 1.4 Результат натискання кнопки з текстом “to move”

Висновок

Завдяки цій лабораторній роботі я навчилась працювати з середовищем розробки та з його можливостями та функціями.

Лабораторна робота №2

Тема. Базові візуальні компоненти Visual Studio 2022 . Створення проекту із використанням візуальних компонент.

Мета. Створити віконний проект та продемонструвати використання візуальних компонент Visual Studio 2022.

Індивідуальне завдання

1. Ознайомитись із палітрою компонент Visual Studio 2022.
2. Створити віконний проект, додати розглянуті візуальні компоненти.
3. Реалізувати калькулятор.

Теоретичні відомості

Елементи керування Label використовуються для відображення тексту або зображень, які користувач не може редагувати. Вони використовуються для ідентифікації об’єктів у формі - для надання опису того, що певний елемент керування зробить, якщо клацнути, наприклад, або для відображення інформації у відповідь на подію або процес виконання у вашій програмі.
Елемент керування button дозволяє користувачеві натиснути її, щоб виконати дію. Кнопка управління може відображати як текст, так і зображення. При натисканні кнопки виглядає так, ніби її натискають і відпускають.
TextBox використовуються для отримання вводу від користувача або для відображення тексту. Елемент керування TextBox зазвичай використовується для редагованого тексту, хоча його також можна зробити лише для читання. Текстові поля можуть відображати кілька рядків, обтікати текст за розміром елемента керування та додавати базове форматування. Елемент керування TextBox забезпечує єдиний формат тексту, що відображається або вводиться в елемент керування.
Подія - це дія, на яку ви можете відповісти або "обробити" у коді. Події можуть генеруватися дією користувача, наприклад, клацанням миші або натисканням клавіші, програмним кодом або системою.

Код програми

Назва файлу: MyForm.cpp

```
#include "MyForm.h"
using namespace System;
```

```
using namespace System::Windows::Forms;

[STAThread]
void main(array<String^>^ arg) {
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);
    Project2::MyForm form;
    Application::Run(% form);
}
```

Назва файлу: MyForm.h

```
#pragma once
#include <string>
#include <cmath>
namespace Project2 {
      using namespace System;
      using namespace System::ComponentModel;
      using namespace System::Collections;
      using namespace System::Windows::Forms;
      using namespace System::Data;
      using namespace System::Drawing;
      public ref class MyForm : public System::Windows::Forms::Form
      {
      public:
            MyForm(void)
            {
                  InitializeComponent();
            }

      protected:
            ~MyForm()
            {
                  if (components)
                  {
                        delete components;
                  }
            }

      protected:
      private: System::Windows::Forms::TextBox^ textBox1;
      private: System::Windows::Forms::Button^ button1;
      private: System::Windows::Forms::Button^ button2;
      private: System::Windows::Forms::Button^ button3;
      private: System::Windows::Forms::Button^ button4;
      private: System::Windows::Forms::Button^ button5;
      private: System::Windows::Forms::Button^ button6;
      private: System::Windows::Forms::Button^ button7;
      private: System::Windows::Forms::Button^ button8;
      private: System::Windows::Forms::Button^ button9;
      private: System::Windows::Forms::Button^ button10;
      private: System::Windows::Forms::Button^ button11;
      private: System::Windows::Forms::Button^ button12;
      private: System::Windows::Forms::Button^ button13;
      private: System::Windows::Forms::Button^ button14;
```

```
      private: System::Windows::Forms::Button^ button15;
      private: System::Windows::Forms::Button^ button16;
      private: System::Windows::Forms::Button^ button17;
      private: System::Windows::Forms::Button^ button18;
      private: System::Windows::Forms::Button^ button19;
      private: System::Windows::Forms::Button^ button20;
      private: System::Windows::Forms::Button^ button21;
      private: System::Windows::Forms::Button^ button22;
      private: System::Windows::Forms::Button^ button23;
      private: System::Windows::Forms::Button^ button24;
      private: System::Windows::Forms::MenuStrip^ menuStrip1;
      private: System::Windows::Forms::ToolStripMenuItem^
fileToolStripMenuItem;
      private: System::Windows::Forms::ToolStripMenuItem^
standardToolStripMenuItem;
      private: System::Windows::Forms::ToolStripMenuItem^
extendedToolStripMenuItem;
      private: System::Windows::Forms::Label^ label1;
      private:
            System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
            void InitializeComponent(void) {
  //creating components
      }

#pragma endregion
            double first_num;
            double second_num;
            char operat;
            double sum = 1;
            double num;
private: System::Void button3_Click(System::Object^ sender,
System::EventArgs^ e) {
      num = System::Convert::ToDouble(textBox1->Text);
      num = -1 * num;
      textBox1->Text = System::Convert::ToString(num);
      }
private: System::Void standardToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
      this->Width = 267;
      textBox1->Width = 239;
}
private: System::Void extendedToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
      this->Width = 330;
      textBox1->Width = 300;
}
private: System::Void button18_Click(System::Object^ sender,
System::EventArgs^ e) {
      if (!textBox1->Text->Contains(".")) {
            textBox1->Text = textBox1->Text + ".";
      }
}
```

```
private: System::Void numbers_Click(System::Object^ sender,
System::EventArgs^ e) {
	System::Windows::Forms::Button^ button = (Button^)sender;
	if (textBox1->Text == "0") {
		textBox1->Text = button->Text;
	}
	else {
		textBox1->Text = textBox1->Text + button->Text;
	}
}
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e) {
	textBox1->Text = "0";
	label1->Text = "";
}
private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e) {
	if (textBox1->Text->Length > 1) {
		textBox1->Text = textBox1->Text->Substring(0, (textBox1-
>Text->Length - 1));
	}
	else {
		textBox1->Text = "0";
	}
}
private: System::Void math_operations(System::Object^ sender,
System::EventArgs^ e) {
	System::Windows::Forms::Button^ button = (Button^)sender;
	operat = System::Convert::ToChar(button->Text);
	first_num = System::Convert::ToDouble(textBox1->Text);
	textBox1->Text = "";
}
private: System::Void other_operations(System::Object^ sender,
System::EventArgs^ e) {
	operat = '~';
	textBox1->Text = "";
}
private: System::Void factorial(System::Object^ sender,
System::EventArgs^ e) {
	first_num = System::Convert::ToDouble(textBox1->Text);
	System::Windows::Forms::Button^ button = (Button^)sender;
	operat = System::Convert::ToChar(button->Text);
	textBox1->Text = "";
}
private: System::Void button24_Click(System::Object^ sender,
System::EventArgs^ e) {
	operat = '|';
	textBox1->Text = "";
}
	private: System::Void button19_Click(System::Object^ sender,
System::EventArgs^ e) {
		if (textBox1->Text != "") {
			second_num = System::Convert::ToDouble(textBox1-
>Text);
```

```
            }
            else {
                  second_num = 1;
            }

                  switch (operat) {
                  case '+':
                        sum = first_num + second_num;
                        label1->Text =
System::Convert::ToString(first_num) + " + " +
System::Convert::ToString(second_num) + " = ";
                        textBox1->Text =
System::Convert::ToString(sum);
                        break;
                  case '-':
                        sum = first_num - second_num;
                        label1->Text =
System::Convert::ToString(first_num) + " - " +
System::Convert::ToString(second_num) + " = ";
                        textBox1->Text =
System::Convert::ToString(sum);
                        break;
                  case '*':
                        sum = first_num * second_num;
                        label1->Text =
System::Convert::ToString(first_num) + " * " +
System::Convert::ToString(second_num) + " = ";
                        textBox1->Text =
System::Convert::ToString(sum);
                        break;
                  case '/':
                        if (second_num != 0) {
                              sum = first_num / second_num;
                              label1->Text =
System::Convert::ToString(first_num) + " / " +
System::Convert::ToString(second_num) + " = ";
                              textBox1->Text =
System::Convert::ToString(sum);
                              break;
                        }
                        else {
                              textBox1->Text = "division by 0";
                              label1->Text = "";
                              break;
                        }
                  case '%':
                        sum = first_num * second_num / 100;
                        label1->Text =
System::Convert::ToString(first_num) + " % " +
System::Convert::ToString(second_num) + " = ";
                        textBox1->Text =
System::Convert::ToString(sum);
                        break;
                  case '^':
```

```
                              sum = pow(first_num, second_num);
                              label1->Text =
System::Convert::ToString(first_num) + " ^ " +
System::Convert::ToString(second_num) + " = ";
                              textBox1->Text =
System::Convert::ToString(sum);
                              break;
                        case '~': //sqrt
                              if (second_num >= 0) {
                                    sum = sqrt(second_num);
                                    label1->Text = " square root of " +
System::Convert::ToString(second_num) + " = ";
                                    textBox1->Text =
System::Convert::ToString(sum);
                                    break;
                              }
                              else {
                                    textBox1->Text = "invalid value";
                                    label1->Text = "";
                                    break;
                              }
                        case '!':
                              if (first_num > 0) {
                                    sum = 1;
                                    for (int i = 1; i <= first_num; i++) {
                                          sum *= i;
                                    }
                                    textBox1->Text =
System::Convert::ToString(sum);
                                    label1->Text =
System::Convert::ToString(first_num) + " ! " + " = ";
                                    break;
                              }
                              else {
                                    textBox1->Text = "invalid value";
                                    label1->Text = "";
                                    break;
                              }
                        case '|':
                              if (second_num >= 0) {
                                    sum = second_num;
                              }
                              else {
                                    sum = -1 * second_num;
                              }
                              label1->Text = " | " +
System::Convert::ToString(second_num) + " | " + " = ";
                              textBox1->Text =
System::Convert::ToString(sum);
                              break;
                        }
            }
};
}
```

Протокол роботи

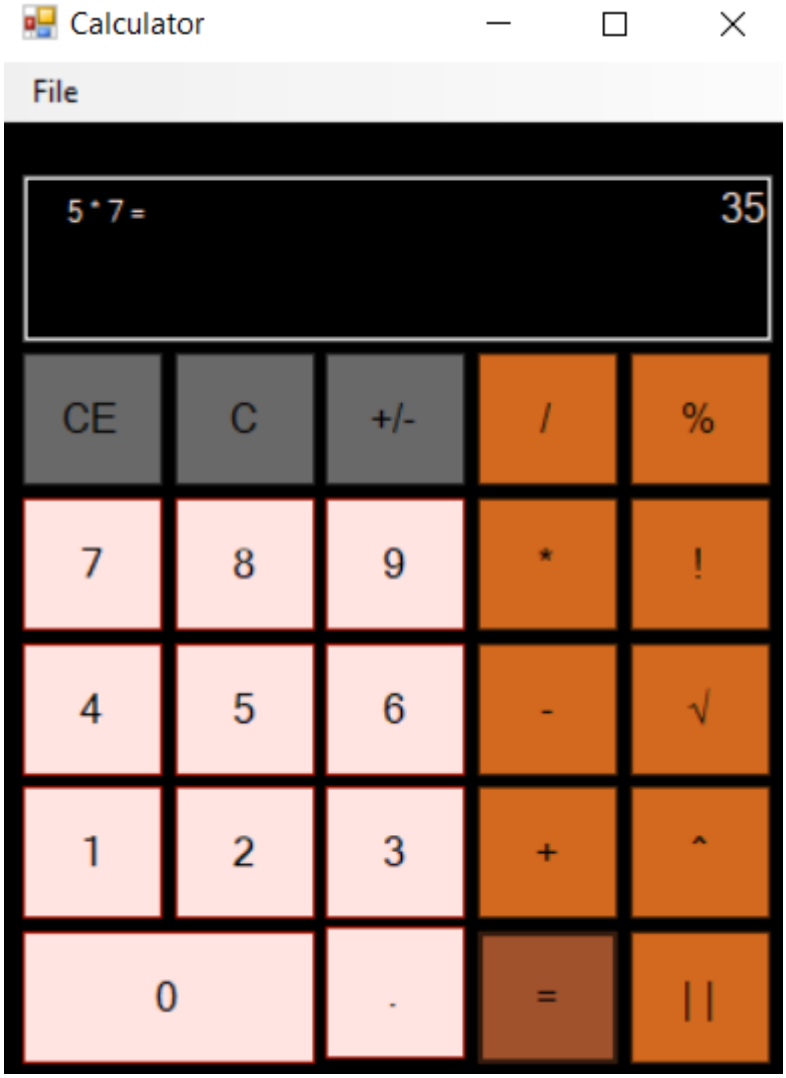

Рис. 1.1 Результат виконання програми

Висновок

На цій лабораторній роботі я навчилась реалізовувати компоненти Visual Studio 2022 , створивши калькулятор.

Лабораторна робота №3

Тема. Огляд невізуальних компонент Visual Studio 2022. Створення проекту із невізуальними компонентами та їх використання.

Мета. Створити віконний проект та продемонструвати використання невізуальних компонент Visual Studio 2022.

Індивідуальне завдання

1. Створити віконний проект. Додати головне та контекстне меню, необхідні системні діалоги.
2. Реалізувати текстовий редактор і переглядач графічних файлів.

Теоретичні відомості

У Visual Studio доступна велика кількість різних системних діалогів: OpenFileDialog(діалог для відкривання файлів), SaveFileDilog(діалог для збереження файлів), ColorDialog(діалог для вибору кольору), FontDialog(діалог для зміни шрифту його розмірі та типу написання). Також існують MessegeBox(маленькі випливаючі діалоги, в основному використовуються для підтвердження дії).

Код програми

Назва файлу: MyForm.cpp

```
#pragma once
#include <string>
namespace NotePad {
	using namespace System;
	using namespace System::ComponentModel;
	using namespace System::Collections;
	using namespace System::Windows::Forms;
	using namespace System::Data;
	using namespace System::Drawing;
	using namespace System::IO;
	using namespace std;
	public ref class MyForm : public System::Windows::Forms::Form
	{
	public:
		MyForm(void)
		{
			InitializeComponent();
		}
	protected:
```

```
		~MyForm()
		{
			if (components)
			{
				delete components;
			}
		}
	private: System::Windows::Forms::ToolStripPanel^
BottomToolStripPanel;
	protected:
	private: System::Windows::Forms::ToolStripPanel^
TopToolStripPanel;
	private: System::Windows::Forms::ToolStripPanel^
RightToolStripPanel;
	private: System::Windows::Forms::ToolStripPanel^
LeftToolStripPanel;
	private: System::Windows::Forms::MenuStrip^ menuStrip1;
	private: System::Windows::Forms::ToolStripMenuItem^
fileToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
newFileToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
newWindowToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
openFileToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
openPhotoToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
saveToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
saveAsToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
printToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
exitToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
editToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
undoToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
redoToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
cutToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
copyToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
pasteToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
findToolStripMenuItem;
	private: System::Windows::Forms::ToolStripSeparator^
toolStripMenuItem1;
	private: System::Windows::Forms::ToolStripMenuItem^
selectAllToolStripMenuItem;
```

```
	private: System::Windows::Forms::ToolStripMenuItem^
dateTimeToolStripMenuItem;
	private: System::Windows::Forms::ToolStripSeparator^
toolStripMenuItem2;
	private: System::Windows::Forms::ToolStripMenuItem^
toolsToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
fontToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
boldToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
italicToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
underlinedToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
largerToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
smallerToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
colourToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
themeToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
autoToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
darkToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
lightToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
adaptiveToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^
aboutToolStripMenuItem;
	private: System::Windows::Forms::RichTextBox^ richTextBox1;
	private: System::Windows::Forms::ToolStripMenuItem^
inverseToolStripMenuItem;
		System::ComponentModel::Container^ components;
#pragma region Windows Form Designer generated code
		void InitializeComponent(void)
		{

//creating components
		}

#pragma endregion
	private: System::Void
exitToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^
e) {
		Close();
	}
	private: System::Void
newFileToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
		String^ message = "Do you want to save this file?";
		String^ caption = "";
		MessageBoxButtons buttons = MessageBoxButtons::YesNo;
```

```
            System::Windows::Forms::DialogResult result =
MessageBox::Show(this, message, caption, buttons);
            if (result == System::Windows::Forms::DialogResult::Yes) {
                  SaveFileDialog^ save_file_dialog = gcnew
SaveFileDialog();
                  save_file_dialog->Filter = "Text File|*.txt|Any
File|*.*";
                  if (save_file_dialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK) {
                        StreamWriter^ stream_writer = gcnew
StreamWriter(save_file_dialog->FileName);
                        stream_writer->Write(richTextBox1->Text);
                        stream_writer->Close();
                  }
                  richTextBox1->Clear();
            }
            else {
                  richTextBox1->Clear();
            }
      }
    private: System::Void
openFileToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
            OpenFileDialog^ open_file_dialog = gcnew OpenFileDialog();
            open_file_dialog->Filter = "Text File|*.txt|Any File|*.*";
            if (open_file_dialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK) {
                  StreamReader^ stream_reader = gcnew
StreamReader(open_file_dialog->FileName);
                  richTextBox1->Text = stream_reader->ReadToEnd();
                  stream_reader->Close();
            }
      }
      private: System::Void
saveToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^
e) {
            StreamWriter^ stream_writer = gcnew StreamWriter(this-
>Text);
            stream_writer->Write(richTextBox1->Text);
            stream_writer->Close();
      }
      private: System::Void
saveAsToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
            SaveFileDialog^ save_file_dialog = gcnew SaveFileDialog();
            save_file_dialog->Filter = "Text File|*.txt|Any File|*.*";
            if (save_file_dialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK) {
                  StreamWriter^ stream_writer = gcnew
StreamWriter(save_file_dialog->FileName);
                  stream_writer->Write(richTextBox1->Text);
                  stream_writer->Close();
            }
      }
```

```
      private: System::Void
printToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
            PrintDialog^ print_dialog = gcnew PrintDialog();
            if (print_dialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK) {

            }
      }
      private: System::Void
undoToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^
e) {
            richTextBox1->Undo();
      }
      private: System::Void
pasteToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
            richTextBox1->Paste();
      }
      private: System::Void
copyToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^
e) {
            richTextBox1->Copy();
      }
      private: System::Void
redoToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^
e) {
            richTextBox1->Redo();
      }
      private: System::Void cutToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
            richTextBox1->Cut();
      }
      private: System::Void
findToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^
e) {
            richTextBox1->Clear();
      }
      private: System::Void
colourToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
            ColorDialog^ color_dialog = gcnew ColorDialog();
            if (color_dialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK) {
                  richTextBox1->ForeColor = color_dialog->Color;
            }
      }
      private: System::Void
boldToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^
e) {
            System::Drawing::Font^ current_font = richTextBox1-
>SelectionFont;
            System::Drawing::FontStyle bold_style_font;
            bold_style_font = FontStyle::Bold;
```

```
            richTextBox1->SelectionFont = gcnew
System::Drawing::Font(current_font->FontFamily, current_font->Size,
bold_style_font);
      }
      private: System::Void
italicToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
            System::Drawing::Font^ current_font = richTextBox1-
>SelectionFont;
            System::Drawing::FontStyle italic_style_font;
            italic_style_font = FontStyle::Italic;
            richTextBox1->SelectionFont = gcnew
System::Drawing::Font(current_font->FontFamily, current_font->Size,
italic_style_font);
    }
private: System::Void autoToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
      richTextBox1->BackColor = System::Drawing::Color::DarkSlateBlue;
      richTextBox1->ForeColor = System::Drawing::Color::White;
}
private: System::Void darkToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
      richTextBox1->BackColor = System::Drawing::Color::Black;
      richTextBox1->ForeColor = System::Drawing::Color::White;
}
private: System::Void lightToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
      richTextBox1->BackColor = System::Drawing::Color::White;
      richTextBox1->ForeColor = System::Drawing::Color::Black;
}
private: System::Void adaptiveToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
      ColorDialog^ color_dialog = gcnew ColorDialog();
      if (color_dialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK) {
            richTextBox1->BackColor = color_dialog->Color;
      }
      richTextBox1->ForeColor = System::Drawing::Color::White;
}
private: System::Void selectAllToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
      richTextBox1->SelectAll();
}
private: System::Void dateTimeToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
      DateTime date_time = DateTime::Now;
      richTextBox1->Text = System::Convert::ToString(date_time);
}
private: System::Void
underlinedToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
      System::Drawing::Font^ current_font = richTextBox1-
>SelectionFont;
      System::Drawing::FontStyle underlined_style_font;
```

```
        underlined_style_font = FontStyle::Underline;
        richTextBox1->SelectionFont = gcnew
System::Drawing::Font(current_font->FontFamily, current_font->Size,
underlined_style_font);
}
private: System::Void largerToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
        System::Drawing::Font^ current_font = richTextBox1-
>SelectionFont;
        FontStyle current_font_style = richTextBox1->Font->Style;
        float current_size = richTextBox1->SelectionFont->Size;
        current_size += 1.0F;
        richTextBox1->SelectionFont = gcnew
System::Drawing::Font(current_font->FontFamily, current_size,
current_font_style);
}
private: System::Void smallerToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
        System::Drawing::Font^ current_font = richTextBox1-
>SelectionFont;
        FontStyle current_font_style = richTextBox1->Font->Style;
        float current_size = richTextBox1->SelectionFont->Size;
        current_size -= 1.0F;
        richTextBox1->SelectionFont = gcnew
System::Drawing::Font(current_font->FontFamily, current_size,
current_font_style);
}
private: System::Void newWindowToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
        MyForm^ new_form = gcnew MyForm();
        new_form->Show();
}
private: System::Void openPhotoToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
        OpenFileDialog^ open_file_dialog = gcnew OpenFileDialog();
        open_file_dialog->Filter = "All supported graphics | *.jpg;
*.jpeg; *.png | " +
                "JPEG (*.jpg;*.jpeg)|*.jpg;*.jpeg|" +
                "Portable Network Graphic (*.png)|*.png";
        if (open_file_dialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK) {
                Clipboard::SetImage(Image::FromFile(open_file_dialog-
>FileName));
                richTextBox1->Paste();
        }
}
private: System::Void aboutToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
        String^ message = "\nCreated by Olesia Kokhman for educational
purposes\n";
        String^ caption = "NotePad";
        MessageBoxButtons buttons = MessageBoxButtons::OK;
        System::Windows::Forms::DialogResult result;
        result = MessageBox::Show(this, message, caption, buttons);
```

```
}
private: System::Void inverseToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
      System::Drawing::Color temp;
      temp = richTextBox1->BackColor;
      richTextBox1->BackColor = richTextBox1->ForeColor;
      richTextBox1->ForeColor = temp;
}
};
}
```

Протокол роботи

Рис. 3.1 Реалізація графічного редактору

Рис. 3.2 Вигляд форми після зміни теми та шрифту

Рис. 3.3-3.5 Функції текстового редактору

Рис. 3.6 Функція ‘about’

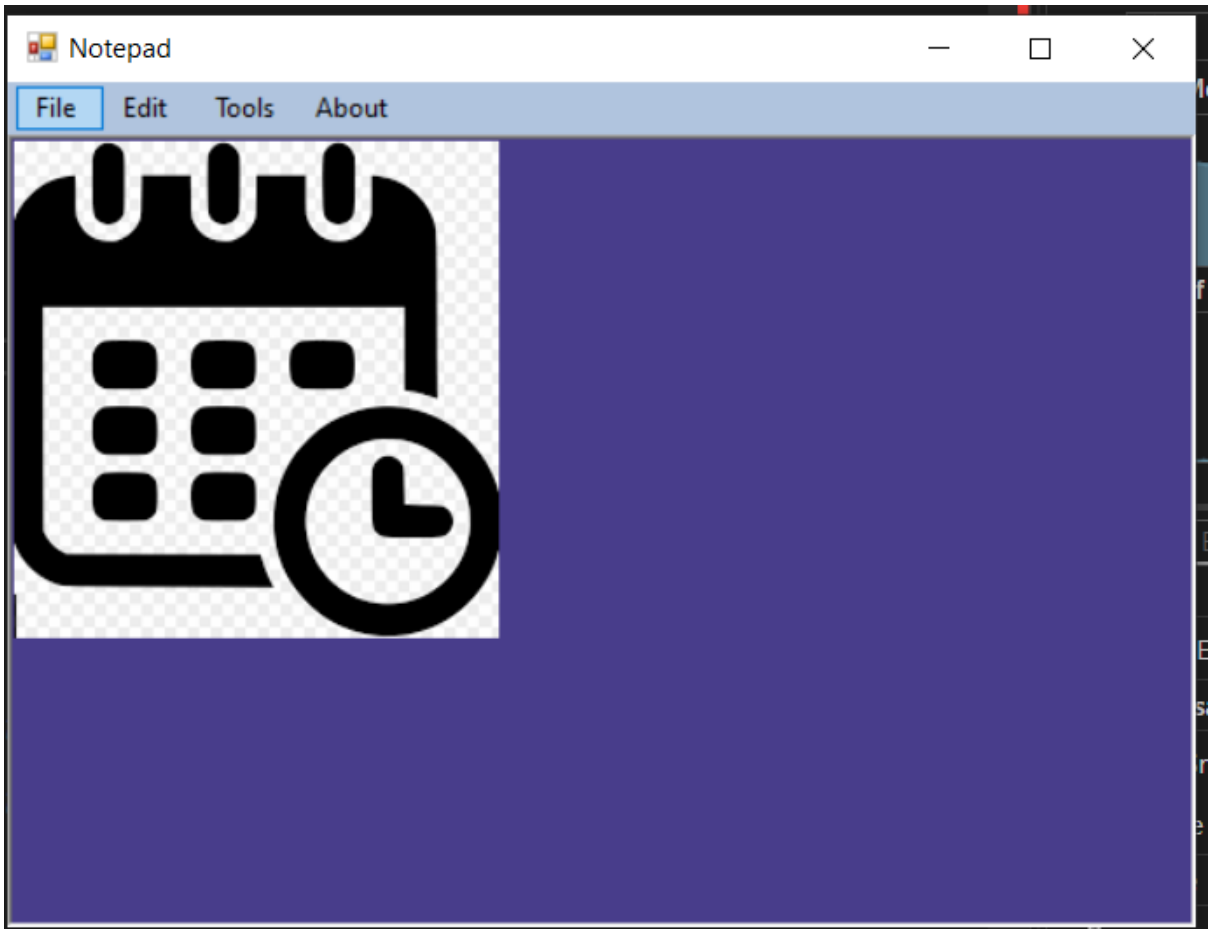

Рис. 3.1 Реалізація графічного редактору

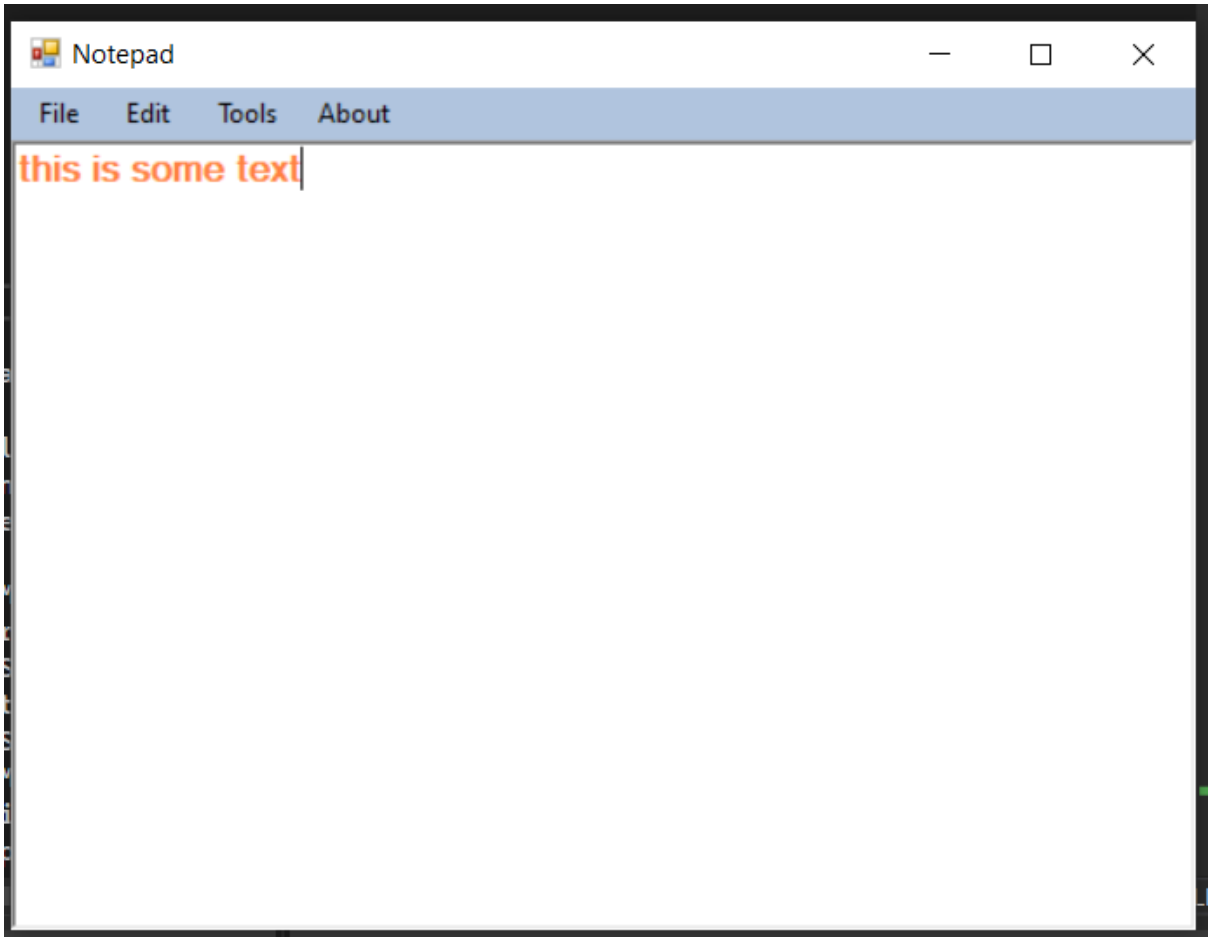

Рис. 3.2 Вигляд форми після зміни теми та шрифту

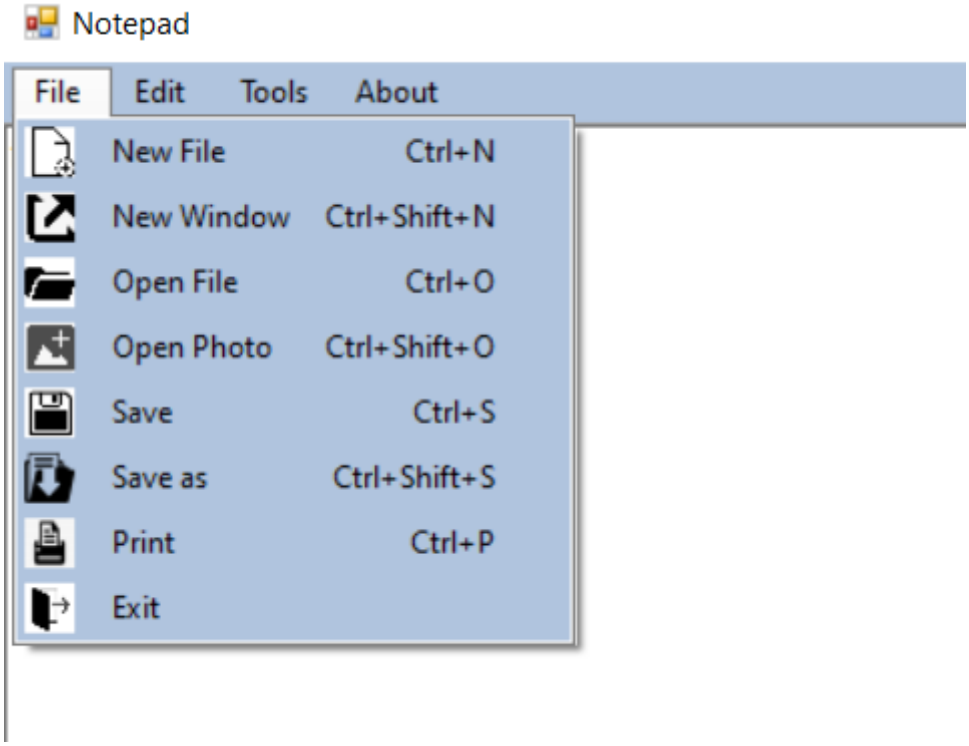

Рис. 3.3 Функції текстового редактору

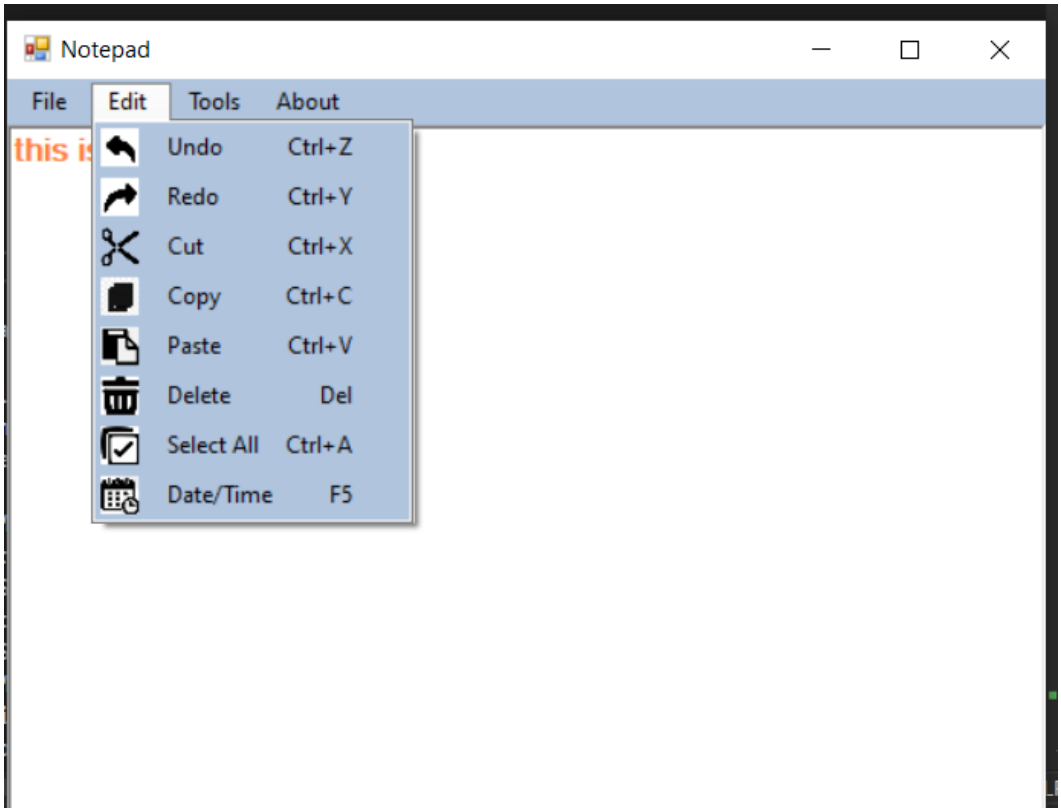

Рис. 3.4 Функції текстового редактору

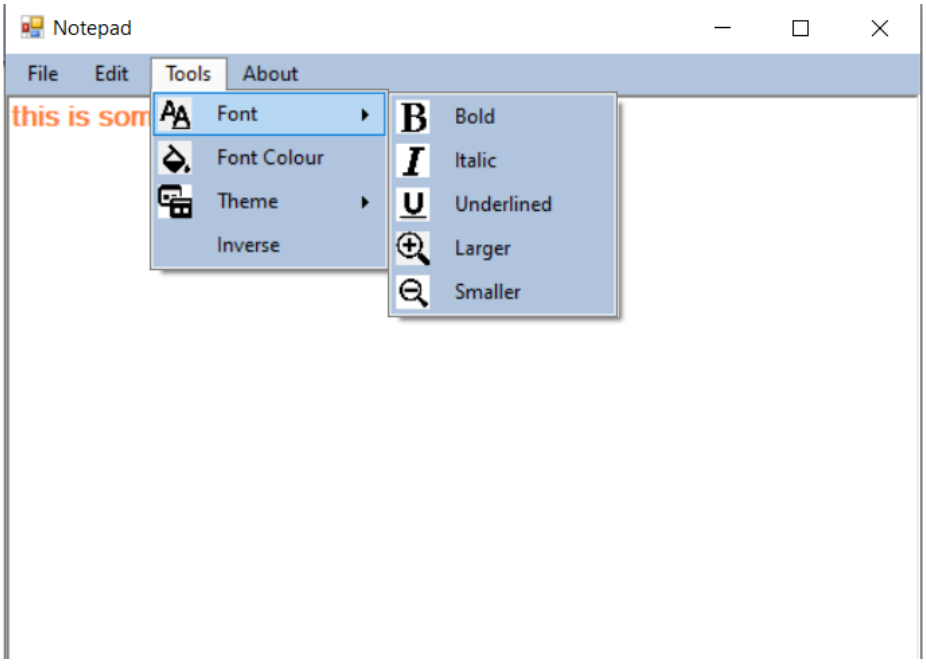

Рис. 3.5 Функції текстового редактору

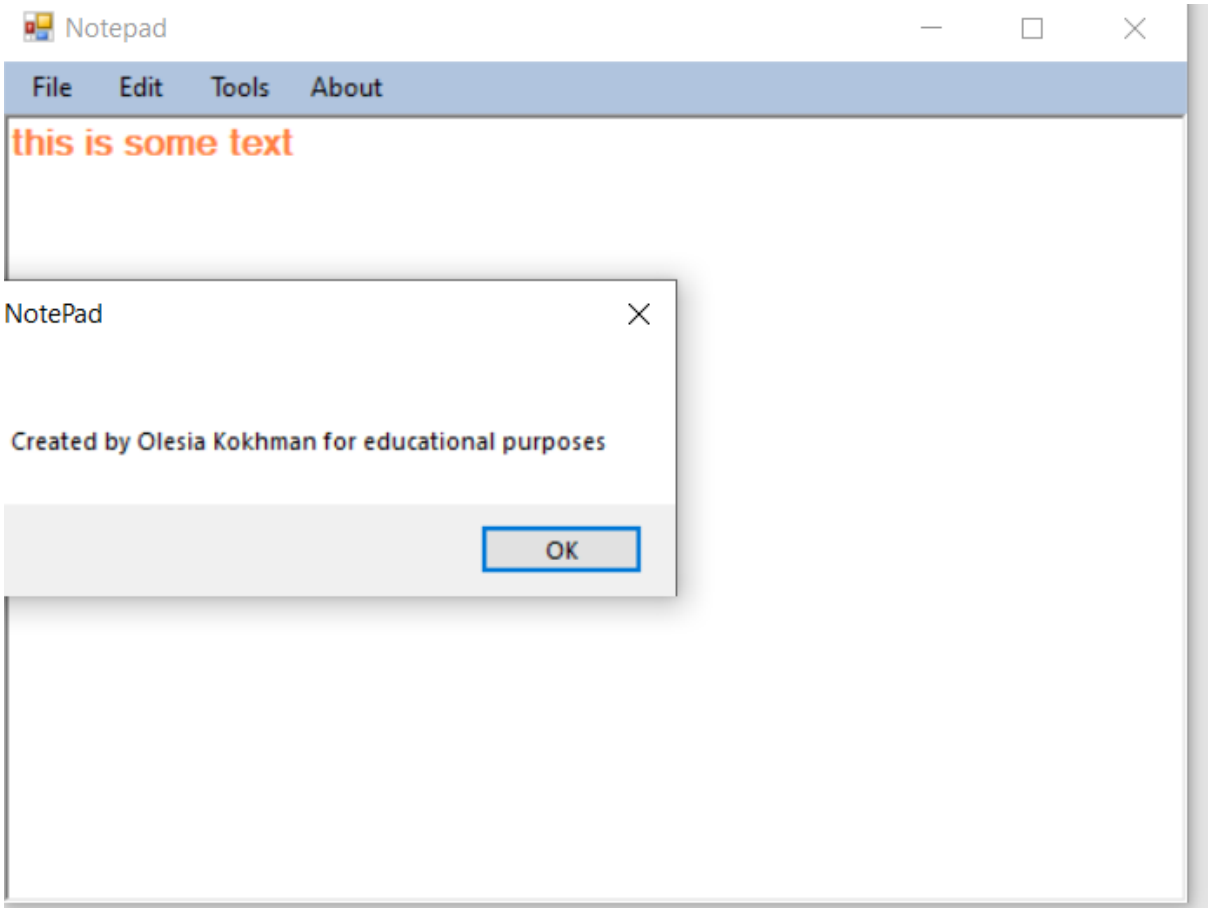

Рис. 3.6 Функція ‘about’

Висновок

На цій лабораторній роботі я ознайомилась ще із великою кількістю компонент та можливостей Visual Studio 2022 та реалізувала текстовий та графічний редактор.

Лабораторна робота №4

Тема: Компоненти для представлення даних.

Мета: Створити віконний проект та продемонструвати використання компонент призначених для відображення та опрацювання даних.

Індивідуальне завдання

1. Ознайомитись із компонентою **DataGridView.**
2. Реалізувати гру.

Код програми

Назва файлу: GameForm.h

```
#pragma once
#include "Game.h"
namespace lab04 {
	using namespace System;
	using namespace System::ComponentModel;
	using namespace System::Collections;
	using namespace System::Windows::Forms;
	using namespace System::Data;
	using namespace System::Drawing;
	public ref class GameForm : public System::Windows::Forms::Form
	{
	public:
		GameForm(void)
		{
			InitializeComponent();
		}
	protected:
		~GameForm()
		{
			if (components)
			{
				delete components;
			}
		}
	private: System::Windows::Forms::DataGridView^ dataGridView1;
	protected:
	private: System::Windows::Forms::DataGridViewImageColumn^ Column1;
	private: System::Windows::Forms::DataGridViewImageColumn^ Column2;
	private: System::Windows::Forms::DataGridViewImageColumn^ Column3;
	private: System::Windows::Forms::ImageList^ imageList1;
	private: System::ComponentModel::IContainer^ components;
	private: System::Windows::Forms::Button^ restartButton;
	protected:
	private: Game* game;
	private:
#pragma region Windows Form Designer generated code

		void InitializeComponent(void)
		{
			this->components = (gcnew
System::ComponentModel::Container());
			System::Windows::Forms::DataGridViewCellStyle^
dataGridViewCellStyle1 = (gcnew System::Windows::Forms::DataGridViewCellStyle());
```

```
                    System::ComponentModel::ComponentResourceManager^ resources =
(gcnew System::ComponentModel::ComponentResourceManager(GameForm::typeid));
                    this->dataGridView1 = (gcnew
System::Windows::Forms::DataGridView());
                    this->Column1 = (gcnew
System::Windows::Forms::DataGridViewImageColumn());
                    this->Column2 = (gcnew
System::Windows::Forms::DataGridViewImageColumn());
                    this->Column3 = (gcnew
System::Windows::Forms::DataGridViewImageColumn());
                    this->imageList1 = (gcnew
System::Windows::Forms::ImageList(this->components));
                    this->restartButton = (gcnew
System::Windows::Forms::Button());

      (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>dataGridView1))->BeginInit();
                    this->SuspendLayout();
                    //
                    // dataGridView1
                    //
                    this->dataGridView1->AllowUserToAddRows = false;
                    this->dataGridView1->AllowUserToDeleteRows = false;
                    this->dataGridView1->AllowUserToResizeColumns = false;
                    this->dataGridView1->AllowUserToResizeRows = false;
                    this->dataGridView1->ColumnHeadersHeight = 21;
                    this->dataGridView1->ColumnHeadersVisible = false;
                    this->dataGridView1->Columns->AddRange(gcnew cli::array<
System::Windows::Forms::DataGridViewColumn^  >(3) {
                          this->Column1,
                                this->Column2, this->Column3
                    });
                    this->dataGridView1->Cursor =
System::Windows::Forms::Cursors::Hand;
                    this->dataGridView1->Location = System::Drawing::Point(62,
72);
                    this->dataGridView1->Margin =
System::Windows::Forms::Padding(0);
                    this->dataGridView1->MultiSelect = false;
                    this->dataGridView1->Name = L"dataGridView1";
                    this->dataGridView1->ReadOnly = true;
                    this->dataGridView1->RowHeadersVisible = false;
                    this->dataGridView1->RowHeadersWidth = 21;
                    dataGridViewCellStyle1->SelectionBackColor =
System::Drawing::Color::White;
                    this->dataGridView1->RowsDefaultCellStyle =
dataGridViewCellStyle1;
                    this->dataGridView1->RowTemplate->Height = 69;
                    this->dataGridView1->RowTemplate->Resizable =
System::Windows::Forms::DataGridViewTriState::False;
                    this->dataGridView1->ShowEditingIcon = false;
                    this->dataGridView1->Size = System::Drawing::Size(303, 211);
                    this->dataGridView1->TabIndex = 0;
                    this->dataGridView1->CellClick += gcnew
System::Windows::Forms::DataGridViewCellEventHandler(this,
&GameForm::dataGridView1_CellClick);
                    this->dataGridView1->ColumnAdded += gcnew
System::Windows::Forms::DataGridViewColumnEventHandler(this,
&GameForm::dataGridView1_ColumnAdded);
                    //
                    // Column1
                    //
                    this->Column1->HeaderText = L"Column1";
```

```
                    this->Column1->Name = L"Column1";
                    this->Column1->ReadOnly = true;
                    //
                    // Column2
                    //
                    this->Column2->HeaderText = L"Column2";
                    this->Column2->Name = L"Column2";
                    this->Column2->ReadOnly = true;
                    //
                    // Column3
                    //
                    this->Column3->HeaderText = L"Column3";
                    this->Column3->Name = L"Column3";
                    this->Column3->ReadOnly = true;
                    //
                    // imageList1
                    //
                    this->imageList1->ImageStream =
(cli::safe_cast<System::Windows::Forms::ImageListStreamer^>(resources-
>GetObject(L"imageList1.ImageStream")));
                    this->imageList1->TransparentColor =
System::Drawing::Color::Transparent;
                    this->imageList1->Images->SetKeyName(0, L"cross.png");
                    this->imageList1->Images->SetKeyName(1, L"circle.png");
                    this->imageList1->Images->SetKeyName(2, L"white.png");
                    //
                    // restartButton
                    //
                    this->restartButton->Location = System::Drawing::Point(146,
310);
                    this->restartButton->Name = L"restartButton";
                    this->restartButton->Size = System::Drawing::Size(114, 42);
                    this->restartButton->TabIndex = 1;
                    this->restartButton->Text = L"Restart";
                    this->restartButton->UseVisualStyleBackColor = true;
                    this->restartButton->Click += gcnew
System::EventHandler(this, &GameForm::restartButton_Click);
                    //
                    // GameForm
                    //
                    this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
                    this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
                    this->ClientSize = System::Drawing::Size(445, 364);
                    this->Controls->Add(this->restartButton);
                    this->Controls->Add(this->dataGridView1);
                    this->Name = L"GameForm";
                    this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
                    this->Text = L"Tic-Tac-Toe";
                    this->Load += gcnew System::EventHandler(this,
&GameForm::GameForm_Load);

       (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>dataGridView1))->EndInit();
                    this->ResumeLayout(false);

             }
#pragma endregion
private: System::Void dataGridView1_CellContentClick(System::Object^ sender,
System::Windows::Forms::DataGridViewCellEventArgs^ e) {
}
```

```
private: System::Void dataGridView1_ColumnAdded(System::Object^ sender,
System::Windows::Forms::DataGridViewColumnEventArgs^ e) {
	this->dataGridView1->Rows->Add();
}
private: System::Void GameForm_Load(System::Object^ sender, System::EventArgs^ e)
{
	this->dataGridView1->Rows->Add();
	this->dataGridView1->Rows->Add();
	this->dataGridView1->Rows->Add();
	dataGridView1->ClearSelection();
	dataGridView1->CurrentCell = nullptr;
	SetupNewGame();
}
private: void SetupNewGame() {
	game = new Game();
	BuildEmptyCells();
}
private: void BuildEmptyCells() {
	for (int i = 0; i < dataGridView1->Rows->Count; i++) {
		for (int j = 0; j < dataGridView1->Rows[i]->Cells->Count; j++) {
			dataGridView1->Rows[i]->Cells[j]->Value = imageList1-
>Images[2];
		}
	}
}
private: System::Void dataGridView1_CellClick(System::Object^ sender,
System::Windows::Forms::DataGridViewCellEventArgs^ e) {
	this->dataGridView1->ClearSelection();
	dataGridView1->CurrentCell = nullptr;
	if (game->Items[e->RowIndex][e->ColumnIndex] != Empty) {
		return;
	}
	game->MakeMove(e->RowIndex, e->ColumnIndex);
	int imageIndex;
	if (game->Items[e->RowIndex][e->ColumnIndex] == Cross) {
		imageIndex = 0;
	}
	else if (game->Items[e->RowIndex][e->ColumnIndex] == Nought) {
		imageIndex = 1;
	}
	else {
		imageIndex = 2;
	}
	dataGridView1->Rows[e->RowIndex]->Cells[e->ColumnIndex]->Value =
imageList1->Images[imageIndex];
	MessageBoxButtons buttons = MessageBoxButtons::OK;
	System::Windows::Forms::DialogResult result;
	if (game->CheckTie()) {
		result = MessageBox::Show(this, "Tie", "Game Result", buttons);
	}
	if (game->Winned) {
		result = MessageBox::Show(this, (game->CurrentPlayer == 0 ? "Cross"
: "Nought") + " is winner!", "Game Result", buttons);
	}
}
private: System::Void restartButton_Click(System::Object^ sender,
System::EventArgs^ e) {
	SetupNewGame();
}
};
}
```

Назва файлу: GameForm.cpp

```
#include "GameForm.h"
using namespace lab04;
int main() {
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);
    Application::Run(gcnew GameForm());
    return 0;
}
```

Назва файлу: Game.h

```
#pragma once
#ifndef GAME_H
#define GAME_H
enum FieldState {
    Empty,
    Cross,
    Nought
};
class Game {
public:
    FieldState Items[3][3];
    int CurrentPlayer = 0;
    bool Winned = false;
    Game();
    void MakeMove(int row, int col);
    bool CheckTie();
private:
    bool CheckWinner(FieldState state);
    bool AllCellsAreFull();
};
#endif
```

Назва файлу: Game.cpp

```
#include "Game.h"
Game::Game() {
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            Items[i][j] = Empty;
        }
    }
}
void Game::MakeMove(int row, int col) {
    if (Items[row][col] != Empty) {
        return;
    }
    if (Winned) {
        return;
    }
    Items[row][col] = CurrentPlayer == 0 ? Cross : Nought;
    if (CheckWinner(Cross) || CheckWinner(Nought)) {
        Winned = true;
        return;
    }
    CurrentPlayer ^= 1;
}
bool Game::CheckWinner(FieldState state){
    for (int i = 0; i < 3; i++) {
        if (Items[i][0] == state && Items[i][1] == state && Items[i][2] == state)
{
            return true;
        }
```

```
        if (Items[0][i] == state && Items[1][i] == state && Items[2][i] == state)
{
            return true;
        }
    }
    if (Items[0][0] == state && Items[1][1] == state && Items[2][2] == state) {
        return true;
    }
    if (Items[0][2] == state && Items[1][1] == state && Items[2][0] == state) {
        return true;
    }
    return false;
}
bool Game::AllCellsAreFull() {
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            if (Items[i][j] == Empty) {
                return false;
            }
        }
    }
    return true;
}
bool Game::CheckTie() {
    return !Winned && AllCellsAreFull();
}
```

Протокол роботи

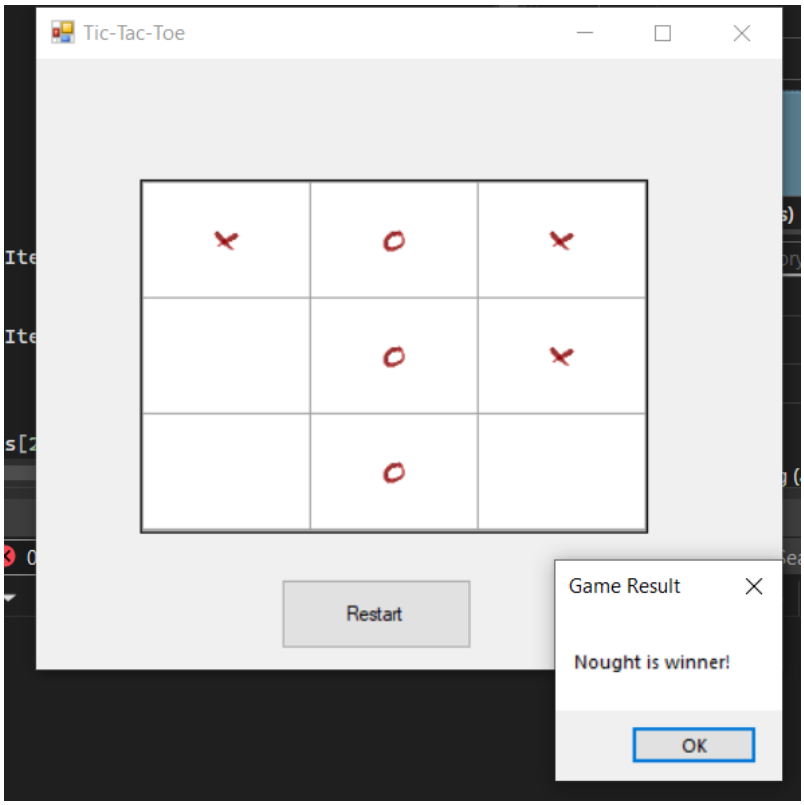

Рис.1 Результат роботи програми, коли нолики перемагають.

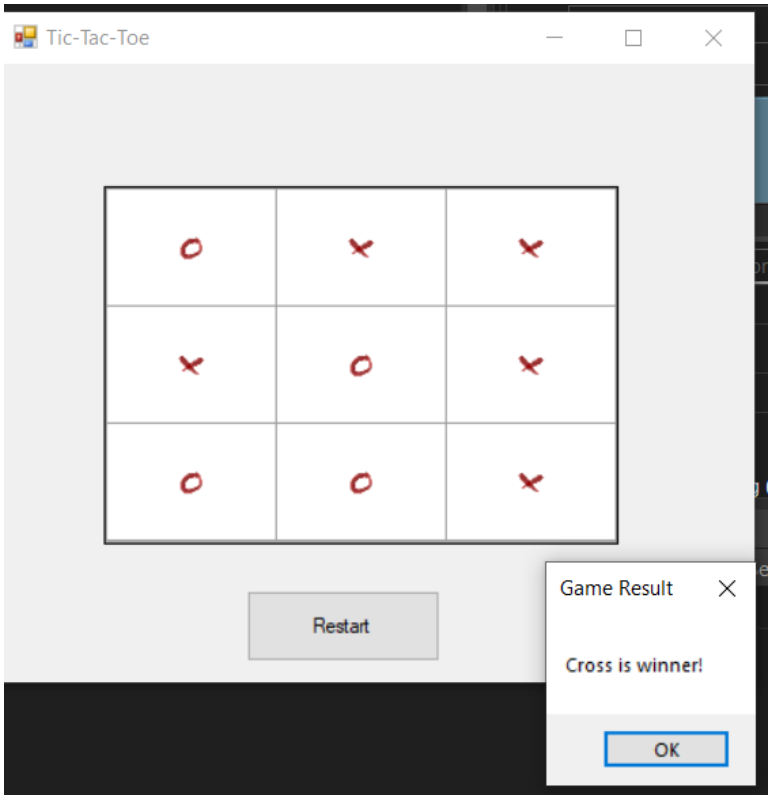

Рис.2 Результат роботи програми, коли хрестики перемагають.

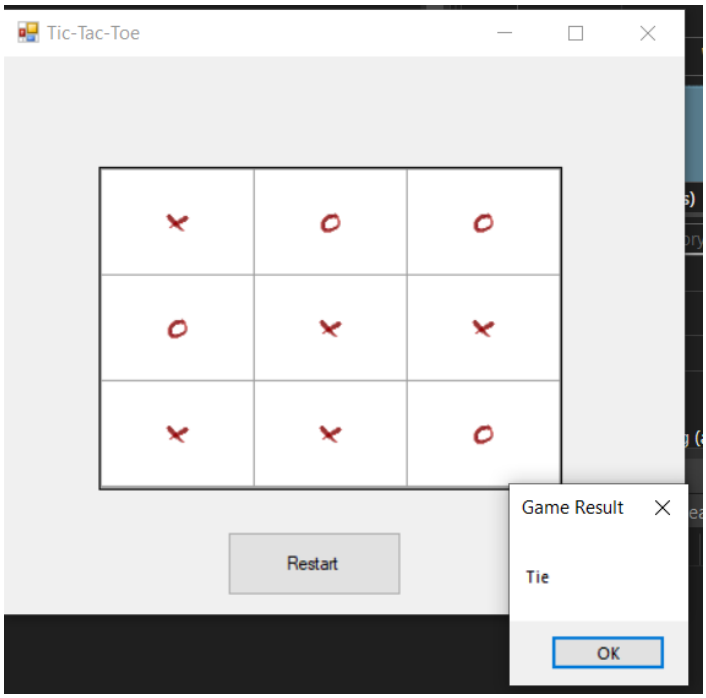

Рис. 3 Результат роботи програми, коли нічия.

Висновок

На цій лабораторній роботі я ознайомилась із компонентою DataGridView та реалізувала гру «хрестики-нолики».