

Міністерство освіти і науки України  
Рівненський державний гуманітарний університет  
Кафедра інформаційно-комунікаційних технологій  
та методики викладання інформатики

*Степанія Михайлівна Бабич*  
*Ігор Станіславович Войтович*  
*Юлія Валеріївна Батишкіна*

**Лабораторний практикум з інформатики**

**ЕЛЕМЕНТИ ВІЗУАЛЬНОГО ПРОГРАМУВАННЯ  
В СЕРЕДОВИЩІ VISUAL STUDIO**

**Visual C++**

Рівне – 2020

*Розглянуто та схвалено на засіданні кафедри інформаційно-комунікаційних технологій та методики викладання інформатики (протокол № 1 від 28 січня 2020 р.)*

**Укладачі:**

- *Бабич С. М.*, кандидат технічних наук, доцент кафедри інформаційно-комунікаційних технологій та методики викладання інформатики РДГУ
- *Войтович І. С.*, доктор педагогічних наук, професор, завідувач кафедри інформаційно-комунікаційних технологій та методики викладання інформатики РДГУ
- *Батишкіна Ю. В.*, кандидат технічних наук, доцент кафедри інформаційно-комунікаційних технологій та методики викладання інформатики РДГУ

**Рецензенти:**

- *Сяський В.А.*, кандидат технічних наук, доцент кафедри інформатики та прикладної математики Рівненського державного гуманітарного університету;
- *Бомба А.Я.*, доктор технічних наук, професор кафедри прикладної математики Національного університету водного господарства та природокористування МОН України.

*Бабич С. М., Войтович І. С., Батишкіна Ю. В.* Лабораторний практикум з інформатики «Елементи візуального програмування в середовищі Visual Studio. Visual C++». – Рівне: РДГУ, 2020, – 32 с.

Призначається для студентів спеціальності *Середня освіта (Математика)*.

# ЗМІСТ

ПЕРЕДМОВА .....	4
Лабораторна робота ПРОГРАМУВАННЯ КНОПОК. ОБ'ЄКТИ: ФОРМА, ТЕКСТОВЕ ПОЛЕ, ЗОБРАЖЕННЯ, КНОПКА .....	5
Лабораторна робота ПРОГРАМУВАННЯ РОЗГАЛУЖЕНЬ. ОБ'ЄКТИ: ПОЛЯ РЕДАГУВАННЯ, ПЕРЕМИКАЧІ ТА ЇХНІ ВЛАСТИВОСТІ.....	14
Лабораторна робота ПРОГРАМУВАННЯ ЦИКЛІВ. ОБ'ЄКТИ: ТЕХТВОХ, MENUSTRIP, CONTEXTMENUSTRIP, CHECKBOX, GROUPBOX.....	21
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	31

## ПЕРЕДМОВА

Лабораторний практикум призначений для проведення лабораторних занять з розділу «Елементи візуального програмування» навчальної дисципліни *Інформатика*, яка викладається для студентів 2-го курсу спеціальності *Середня освіта (Математика)*.

Тематика лабораторних робіт цілком відповідає робочій програмі базового курсу. У якості середовища розробки програм пропонується Visual Studio C++. Всі лабораторні роботи розроблені у вигляді інструкцій з коментарями і містять вправи для самостійного виконання.

Посібник побудовано на матеріалах лекційних і лабораторних занять, які проводяться на кафедрі інформаційно-комунікаційних технологій та методики викладання інформатики Рівненського державного гуманітарного університету.

# Лабораторна робота


## ПРОГРАМУВАННЯ КНОПОК. ОБ'ЄКТИ: ФОРМА, ТЕКСТОВЕ ПОЛЕ, ЗОБРАЖЕННЯ, КНОПКА

**Мета роботи.** Створити форму "Анкета студента" з даними про себе й двома фотографіями (портретною та художньою), які перекривають одна одну і мають з'являтися в результаті натискання на кнопки (рис. 2).

Ознайомитися з такими об'єктами: форма (Form), текстове поле (Label), зображення (PictureBox), кнопка (Button) та їхніми основними властивостями: підпис (Text), колір (Color), шрифт (Font), видимість (Visible), доступність (Enabled), розмір (Size) та ін.

### ХІД РОБОТИ

**1) Завантажте середовище візуального програмування Visual Studio.**

Запуск системи візуального програмування Visual Studio виконують клацанням на піктограмі  або за допомогою каскадного меню **Пуск** ⇒ **Visual Studio 2012**.

**2) Створіть новий проект**

*Создать проект → Пустой проект CLR.*

Назвіть проект, наприклад, **Project1**. Тоді виконайте наступні дії:

*Обозреватель решений → Добавить → Создать элемент.*

*Visual C++ → UI → Форма Windows Forms.*

Вкажіть ім'я форми, наприклад **Form1**.

Відкрийте файл вихідного коду **Form1.cpp** і вставте код:

```
#include "Form1.h" //Цей рядок має бути вже наявний
using namespace Project1;
[STAThreadAttribute]
int main(array<System::String ^> ^args)
{
```

```

Application::EnableVisualStyles();
Application::SetCompatibleTextRenderingDefault(false);
Application::Run(gcnew Form1());
return 0;
}

```

Далі виконайте такі дії:

- *Проект* → *Свойства: Project1...* → *Свойства конфигурации* → *Компоновщик* → *Система* → *Подсистема* → *Windows (/SUBSYSTEM:WINDOWS)*;
- *Компоновщик* → *Дополнительно* → *Точка входа*. Напишіть **main**;
- *Застосувати* → *ОК*.

### **Зауваження 1.**

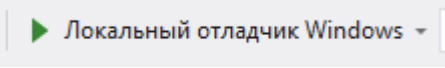
Відкрити *Панель елементов* у разі її відсутності можна через меню *Вид* (або комбінацією клавіш **Ctrl+Alt+X**); вікно *Свойства* – меню *Вид* → *Другие окна* → *Окно свойств (Alt+ENTER)*.

Перейти до редактора коду із вікна форми можна за допомогою комбінації клавіш **<Ctrl>+<Alt>+<0>**, або виконавши команду *Код* головного меню *Вид*.

Із режиму написання коду в режим дизайнера форми можна переключитися комбінацією клавіш **<Shift>+<F7>**.

### **3) Запустіть програму Project1 на виконання і розгляньте вікно порожньої поки-що форми. Поекспериментуйте з вікном форми.**

Запустити програму можна декількома способами:


- виконати команду *Отладка* ⇔ *Начать отладку* головного меню;
- клацнути на кнопці  панелі інструментів;
- натиснути функціональну клавішу.

Виконайте такі дії: максимізуйте вікно, відновіть його попередній розмір, мінімізуйте та знову розгорніть вікно, пересуньте на робочому столі та змініть його розміри, викличте системне меню (натиснути піктограму або **Alt + Пробіл**). Виконайте ті самі дії за допомогою

команд *Перемістити*, *Розмір* та інших і клавіатури.

*Висновок*: вікно форми володіє усіма властивостями стандартного вікна операційної системи Windows.

#### 4) **Збережіть створену програму.**

Для цього виберіть команду головного меню *File* → *Сохранить все* (**Ctrl+Shift+S**) або натисніть на кнопку  панелі інструментів.

#### 5) **Візуально ознайомтеся з властивістю форми Size (Width, Height).**

Змініть розміри форми. Переконайтесь, що тепер змінюються властивості *Width* (ширина) та *Height* (висота) форми у вікні властивостей об'єкта.

Спробуйте безпосередньо ввести відповідне значення ширини і висоти форми у пікселях і натиснути на клавішу **Enter**.

#### 6) **Змініть колір фону форми.**

Для цього у вікні властивостей форми *Свойства* у рядку *Appearance* (*BackColor*) виберіть значення кольору фону одним із двох способів:

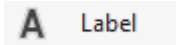
- подвійним клацанням мишею на поточному значенні властивості *BackColor*.
- за допомогою спадаючого меню.

#### 7) **Дослідіть властивість форми *FormBorderStyle*. Встановіть її значення, яке бажаєте.**

#### 8) **Змініть властивість *Text*, наприклад, на «Анкета» чи «Форма А».**

#### 9) **Виконайте програму ще раз (див. п. 3).**

#### 10) **Вставте у форму текстове поле (об'єкт типу *Label*) з текстом "Анкета студента".**

Двічі клацніть мишею на піктограмі  на закладці *Стандартные элементы управления* панелі елементів або перетягніть його у форму. Розташуйте вставлений об'єкт, наприклад, так, як

показано на рис. 3, переміщуючи його мишею. Активізуйте елемент Label1. Змініть значення властивості Text з Label1 на текст "АНКЕТА СТУДЕНТА" без лапок. Змініть значення властивості Font (шрифт) цього текстового поля на такі:

Font: Candara;

Bold: True;

Size: 14;

ForeColor: Maroon (вкладка Web).

**Зауваження 2.** У вікні *Свойства* відображається список властивостей лише активного у певний момент об'єкта.

**11) Аналогічно вставте у форму ще декілька текстових полів з вашими біографічними даними (група, факультет, університет, рік народження, адреса, ...).**


Один із варіантів розміщення текстових полів показаний на рис. 1-2.

**12) Вставте у форму об'єкт типу PictureBox (зображення).**

Для цього двічі клацніть лівою клавшею миші на піктограмі PictureBox закладки *Стандартные элементы управления* панелі елементів або перетягніть у форму. На формі з'явиться **квадратна** область для майбутнього зображення (фотографії). Розташуйте її, наприклад, у нижньому правому куті форми. Якщо потрібно, змініть розмір форми чи вставленого об'єкта та добийтеся якнайкращого розташування на ній створених раніше об'єктів. Змінювати розміри об'єкта можна методом їх "розтягування" за маркери (білі габаритні квадратики). Запам'ятайте назву, яку Visual Studio присвоїть цьому об'єкту (значення властивості (Name)) або замініть її на свій розсуд. За замовчуванням цей об'єкт матиме стандартну назву pictureBox1.

**13) Вставте свою портретну фотографію за допомогою властивості Picture (ілюстрація) об'єкта pictureBox1.**



Для цього виокремите об'єкт pictureBox1 та активізуйте рядок Image у вікні *Свойства*. Клацнувши на кнопці ..., викличте діалогове вікно вибору малюнка. Або натисніть на кнопку меню  області зображення. З'явиться меню PictureBox Tasks, натисніть Choose Image. Завантажте попередньо підготовлену фотографію чи картинку. Задайте властивість SizeMode.

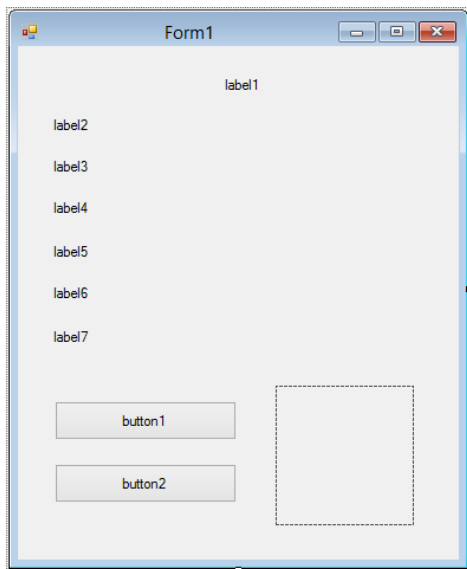


Рис. 1 Макет форми «Анкета студента»



Рис. 2 Готова форма

**14) Вставте свою художню фотографію у форму поверх наявної, скориставшись ще одним об'єктом типу PictureBox.**

Один із варіантів розташування фотографії показаний на рис. 1. Вважатимемо, що цей об'єкт має назву pictureBox2.

**Зауваження 3.** Під час накладання об'єктів може виникнути потреба використати команди *На передній план* чи *На задній план*, які є в їх контекстних меню.

**15) Поекспериментуйте з властивістю Visible (видимість) обох зображень, кожного разу виконуючи програму (див. пункт 3). Після цього встановіть значення властивості Visible у False для обох зображень.**

**16) Вставте у форму кнопки для засвічування фотографій — два об'єкти типу Button з назвами button1 і button2.**

Піктограма об'єкта типу Button (кнопка) знаходиться у вкладці *Стандартные элементы управления* панелі елементів. Поміняйте підписи на кнопках (змінить властивість Text) на "Портретна фотографія" та "Художня фотографія" відповідно. Виберіть найкращий, на ваш розсуд, кирилізований шрифт для підписів. Якщо використано інші картинки, виберіть для кнопок цікаві підписи. Один із варіантів розміщення кнопок показаний на рис. 1-2.

**17) Запрограмуйте кнопку "Портретна фотографія" так, щоб після її натискання у формі з'являлась портретна фотографія.**

Для програмування кнопки button1 необхідно двічі клацнути на ній лівою клавішею миші. У результаті активізується вікно тексту програми із заготовкою функції button1\_Click, яка опрацьовуватиме подію клацання на кнопці button1:

```
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e) {
}
```

У заготовку необхідно вставити текст програми реакції на цю подію. Процедура матиме такий вигляд:

```
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e) {
    pictureBox1->Visible=true;    // Портретна
    фотографія стає видимою
    pictureBox2->Visible=false;  // Художня
    фотографія стає невидимою
}
```

За допомогою функції властивість видимості для об'єкта pictureBox1 задаємо, і цю ж властивість для об'єкта pictureBox2 забираємо. Для кнопки "Художня фотографія" дії будуть протилежні.

**18) Запрограмуйте кнопку "Художня фотографія" відповідно до її призначення (див. п. 17).**

Текст функції для цієї кнопки матиме такий вигляд:

```
private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e) {
    pictureBox1->Visible=false;    // Портретна
    фотографія стає невидимою
    pictureBox2->Visible=true;    // Художня
    фотографія стає видимою
}
```

Щоб створити таку функцію швидко, можна скопіювати дві команди присвоєння з попередньої функції у нову і поміняти вирази праворуч.

**19) Виконайте програму і переконайтесь, що кнопки виконують свої функції. Закрийте вікно програми "Анкета студента".**

**20) Збережіть створену програму.**

**21) Закрийте Visual Studio C++, виконайте створену програму і поекспериментуйте з побудованими кнопками.**

Запустіть exe-файл з іменем проекту (Project1/Debug/Project1.exe).

**22) Продемонструйте створену форму викладачеві. Закінчіть роботу.**

## ВПРАВИ

1. Вставте у форму третю фотографію (фото вашого будинку, машини, чи університету) і ще одну кнопку з відповідним підписом, яка її висвітлюватиме.
2. Додайте у форму ще одну кнопку "Забрати фотографію". Запрограмуйте її відповідно до нового призначення. Виконайте програму і переконайтесь у правильності її роботи.

*Підказка.* У тексті функцій, що описують роботу кнопок, можна скористатися такими командами:

```
if (pictureBox1->Visible==true) ... // Якщо видимість = true
```

3. Поміняйте сценарій роботи програми з п. 2 на такий:

- відразу після запуску програми фотографій не видно, тільки є дві кнопки "Портретна фотографія" і "Забрати фотографію". Доступною є лише перша кнопка;
- після клацання на кнопці "Портретна фотографія" у формі з'являється портретне фото, підпис на першій кнопці змінюється на "Художня фотографія", стає доступною кнопка "Забрати фотографію";
- після клацання на кнопці "Художня фотографія" фотографія у формі змінюється на художню, а підпис на цій кнопці змінюється на "Третя фотографія";
- після клацання на кнопці "Третя фотографія" фотографія у формі змінюється на третю, а підпис на цій кнопці змінюється на "Портретна фотографія";
- після клацання на кнопці "Забрати фотографію" фотографія зникає і ця кнопка стає невидимою (або недоступною);
- виконайте програму і переконайтесь у правильності її роботи.

*Підказка.* У тексті функцій, які описують роботу кнопок, можна скористатися командами, що змінюють властивості кнопок Text, Visible, Enabled.

4. Доповніть програмний код розв'язування задачі з п. 3 так, щоб після вимкнення фотографій напис на першій кнопці завжди відповідав фотографії, яка повинна з'явитися після її натискання.
5. Виходячи з умови задачі з п. 4, добийтеся того, щоб послідовність перемикання фотографій не порушувалася внаслідок їх

вимкнення, а також додайте текстовий підпис з назвою фотографії, видимою у поточний момент.

6. Скопіюйте програмний код у звіт.

**Зауваження 4.** При бажанні форму можна доповнити додатковими текстовими полями та зображеннями (рис. 3).



The image shows a web browser window titled "Анкета" (Survey). The main content is a form titled "АНКЕТА СТУДЕНТА" (STUDENT SURVEY). The form contains the following text:

**Комбель Стефанія Михайлівна**  
Група: **МЕФІ-21**  
Факультет: **математики та інформатики**  
Рівненський державний гуманітарний університет  
Рік народження: **2000**  
Адреса: **м. Рівне, вул. Платова, 29/604А**

Below the text, there is a date "2017 р." and a photograph of a woman with long brown hair holding yellow tulips. To the left of the photo are two buttons: "Художня фотографія" (Artistic photograph) and "Забрати фотографію" (Remove photograph). At the bottom of the form, there is a decorative red flourish.

Рис. 3 Доповнена форма (тут використано 14 текстових полів, 4 кнопки та 4 зображення)

## Лабораторна робота ПРОГРАМУВАННЯ РОЗГАЛУЖЕНЬ. ОБ'ЄКТИ: ПОЛЯ РЕДАГУВАННЯ, ПЕРЕМИКАЧІ ТА ЇХНІ ВЛАСТИВОСТІ

**Мета роботи.** Створити форму з назвою "Обмін валюти", на якій можна змоделювати операції обміну валюти в обмінному пункті. Застосувати поля редагування (TextBox) та перемикачі (RadioButton, дослівно радіокнопка), а також кнопки для виконання обчислень і закінчення роботи програми (див. зразок форми на рис. 5).


### ХІД РОБОТИ

1) Завантажте середовище візуального програмування Visual Studio C++ та створіть новий проект.

2) Відмовтесь від можливості змінювати розміри вікна програми, надавши властивості форми `FormBorderStyle` значення `FixedToolWindow`.

Задавши це значення, виконайте програму і переконайтеся, що не можна змінити розмір форми. Зверніть увагу на відсутність у вікні кнопок мінімізації і максимізації, а також системного меню. Завершіть роботу програми.

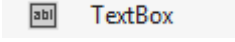
3) Вставте у форму два об'єкти типу `RadioButton` (перемикач), як це показано на рис. 1.

Для цього клацніть на піктограмі  `RadioButton` (перемикач), яка розміщена у вкладці *Стандартные элементы управления* панелі елементів, а після цього клацніть на потрібному місці на формі. Бажаного результату також можна досягнути, перетягнувши перемикач чи двічі клацнути на ньому. Повторіть ці дії, щоб вставити другу радіокнопку.

4) Задайте початкове значення другого перемикача як активне.

Для цього клацніть на правому перемикачі і значення його властивості Checked (контроль вибору) задайте як True.

**5) Вставте у форму два поля редагування — об'єкти textBox1 та textBox2.**

Для цього клацніть на піктограмі  об'єкта типу TextBox (редагування), яка знаходиться у вкладці *Стандартные элементы управления* панелі елементів, а потім клацніть у потрібному місці на формі. Вставте другий об'єкт (рис. 1).

Запустіть програму і поекспериментуйте зі вставленими об'єктами: клацніть у полі редагування, введіть деяке число, вилучіть його. Закрийте вікно програми.

**6) Розташуйте у формі два текстові поля — об'єкти Label1 та Label2 (рис. 4).**

**7) Вставте у форму два поля редагування — об'єкти textBox3 та textBox4 (рис. 4).**

**8) Вставте у форму дві кнопки - об'єкти типу Button (рис. 4).**

**9) Вставте у форму ще два текстові поля — об'єкти Label3 і Label4 (рис. 4).**

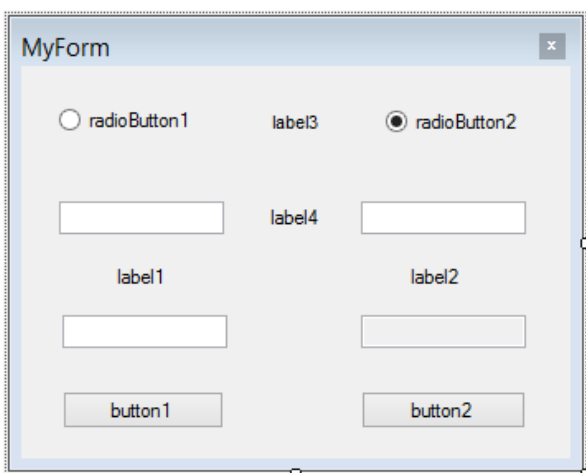


Рис. 4 Макет форми «Обмін валют»

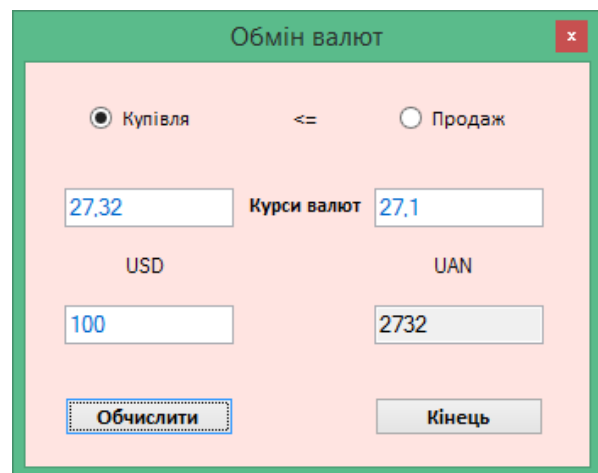


Рис. 5 Готова форма «Обмін валют»

**10) Збережіть створену у цей момент форму у своїй папці.**

Імена файлу тексту програми та проекту занотуйте у звіт. У подальшому періодично, зокрема перед черговими запусками проекту на виконання, зберігайте файли програми (Save All).

**11) Змініть назву форми на "Обмін валюти".**

Для цього змініть значення властивості Text форми. Клацніть на формі і на рядку Text у вікні *Свойства*. Введіть назву форми без лапок. Зверніть увагу на те, що для об'єктів багатьох типів (зокрема, Label, Button, Form, RadioButton та інших) значення властивостей Text та (Name) збігаються.

**12) Змініть підписи Text на об'єктах типу RadioButton, Label та Button так, як показано на рис. 5.**

Для цього по черзі вибирайте об'єкти (клацайте на них) і змінюйте значення властивості Text.

**13) Вирівняйте текстові поля, поля редагування та кнопки на формі. Збережіть роботу.**

Деякі властивості, н-д, Font, можна змінювати одночасно для декількох об'єктів. Для цього потрібно виділити об'єкт на формі, затиснути клавішу Shift і виділити інші. Після активізації властивості її значення – це значення властивості першого виокремленого об'єкта створеної групи. Щоб зняти виокремлення, потрібно клацнути на вільному місці форми або натиснути клавішу Esc.

**14) Задайте значення курсів купівлі та продажу валюти, а також кількість валюти, яку потрібно купити чи продати.**

Для цього введіть потрібне число, наприклад 5, як значення властивості Text об'єкта TextBox1. Повторіть це для об'єкта TextBox2 (значення 5.2) та TextBox3 (значення 20). Для набору символу десяткової крапки використайте символ, передбачений операційною системою комп'ютера (див. Start (Пуск) ⇨ Settings (Налаштування) ⇨ Control Panel (Панель керування) ⇨ Region (Регіональні настройки) ⇨ Формати ⇨ Додаткові налаштування ⇨ Закладка «Числа», Рядок



«Десятковий роздільник».

**15) Заблокуйте можливість введення даних для поля TextBox4, задавши його властивість Readonly як True, оскільки це поле міститиме результат.**

Змінювати значення певної властивості можна подвійним клацанням на ній лівою клавішею миші. Збережіть роботу (Save All). Виконайте програму і переконайтеся, що не можна ввести чи редагувати дані у полі textBox4.

**16) Запрограмуйте радіокнопки так, щоб напрямок стрілки показував на вид операції: купівля чи продаж. Зробіть активним поле textBox3.**

Клацніть *двічі* на правому перемикачі radioButton2 (Продаж). Отримаєте заготовку функції radioButton2\_CheckedChanged. У тілі цієї функції опишіть дії, які мають відбутися у результаті клацання на правому перемикачі radioButton2:

```
private: System::Void  
radioButton2_CheckedChanged(System::Object^ sender,  
System::EventArgs^ e) {  
    label13->Text="=>"; // Змінюємо напрямок стрілки  
    textBox3->Focus(); // Активізуємо поле TextBox3  
}
```

Аналогічно запрограмуйте подію Click клацанням на лівому перемикачі RadioButton1, врахувавши, що стрілка має показувати на ліве поле ('<=').

**17) Запустіть програму і переконайтесь, що перемикач виконує свої функції згідно з п. 17.**

**18) Запрограмуйте кнопку "Кінець".**

Скористайтесь функцією закриття потоку Close():

```
private: System::Void button2_Click(System::Object^ sender,
```

```

System::EventArgs^ e) {
    Close(); // Закінчуємо роботу програми
}

```

### 19) Запрограмуйте кнопку "Обчислити".

Інформація у полі редагування – це значення властивості Text об'єкта типу рядок. Для перетворення цього даного у числовий дійсний тип (double) скористайтесь класом Convert (див. нижче). Опишіть основні і додаткові змінні.

```

private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e) {
    double f1=Convert::ToDouble(textBox1->Text); //Одержуємо
значення курсу купівлі
    double f2=Convert::ToDouble(textBox2->Text); //Одержуємо
значення курсу продажу
    double f3=Convert::ToDouble(textBox3->Text); //Одержуємо
значення суми в USD
    double f4; //Змінна для суми в гривнях
    if (radioButton1->Checked) f4=f1*f3; else f4=f2*f3;
    textBox4->Text=Convert::ToString(f4); //Одержане число суми
в гривнях перетворюємо у текстовий формат і результат присвоюємо
властивості Text поля textBox4
}

```

### 20) Збережіть роботу (Save All).

### 21) Виконайте програму і проекспериментуйте з різними грошовими сумами й операціями купівлі чи продажу. Закрийте вікно програми "Обмін валюти".

Перевірте роботу програми для випадку, коли вхідні дані не введені. Повинно відбутися переривання програми.

Щоб уникнути такої помилки, потрібно попередньо здійснювати перевірку, чи поле редагування є непорожнім (наприклад, об'єкт textBox1):

```

if (textBox1->Text!="") ... // У цьому випадку можна
проводити обчислення

```

**22) Змініть розміри та кольори символів (зокрема об'єктів Label3 і TextBox3), розташування об'єктів, фон форми (властивість BackColor) так, щоб форма виглядала якнайкраще.**

**23) Забезпечте появу підказки "Введіть суму в доларах" після переміщення вказівника миші до поля TextBox3.**

У панелі елементів два рази клацніть на об'єкті ToolTip. Створиться об'єкт toolTip1. Виділіть поле редагування TextBox3 і введіть у поле властивості ToolTip on toolTip1 текст без лапок «Введіть суму в доларах».

Збережіть роботу, запустіть програму і переконайтеся, що підказка з'являється.

**24) Закрийте Visual Studio C++, запустіть створену програму і виконайте обчислення для різних початкових даних.**

Запустіть exe-файл з іменем проекту з папки проекту.

**25) Продемонструйте створену форму викладачеві. Закінчіть роботу.**

## **ВПРАВИ**

1. Передбачте у створеній програмі ще одну кнопку для очищення полів грошових сум. Виконайте програму і переконайтесь у правильності її роботи.

*Підказка.* Для об'єктів TextBox3, TextBox4 у функції опрацювання події натискання на цю кнопку використайте команду присвоєння їхнім властивостям Text порожнього рядка ("").

2. Забезпечте появу підказки "Введіть курс купівлі" та "Введіть курс продажу" після переміщення вказівника миші до полів TextBox1 та TextBox2 відповідно (див. п. 23).

*Підказка.* Для створення підказок до полів TextBox1 та TextBox2 можна використати раніше створений об'єкт toolTip1.

3. Модифікуйте програму, передбачивши додаткову можливість

зміни типу операцій (купівля, продаж) унаслідок клацання мишею на стрілці. Запишіть у звіт фрагмент програмного коду, який реалізує цю можливість. Виконайте програму.

**Підказка.** Для цього двічі клацніть на текстовому полі стрілки. Відкриється вікно програмного коду із заготовкою функції Label3Click (опис дій у випадку клацання на об'єкті Label3). У тілі цієї функції можна скористатися командами вигляду:

```
if (radioButton1->Checked==true)    /* Встановлюємо перемикач у
праве положення, змінюючи значення властивості Checked об'єкта
RadioButton2 */
else    /* Встановлюємо перемикач у ліве положення, змінюючи
значення властивості Checked об'єкта RadioButton1 */
```

4. Спростіть форму (вилучіть зайві об'єкти) та змініть код кнопки "Обчислити" так, щоб її можна було використати для переведення миль у кілометри чи навпаки залежно від положення перемикача (1 миля = 1,609344 км). Або створіть нову форму. Назвіть її «Конвертер» (рис. 6).
5. У створену для задачі форму (п. 3) вставте групу перемикачів (об'єкт GroupBox, яка містить групу об'єктів RadioGroup) для вибору типу милі з двох можливих значень: морської чи звичайної (1 морська миля = 1,852 км) (рис. 6).

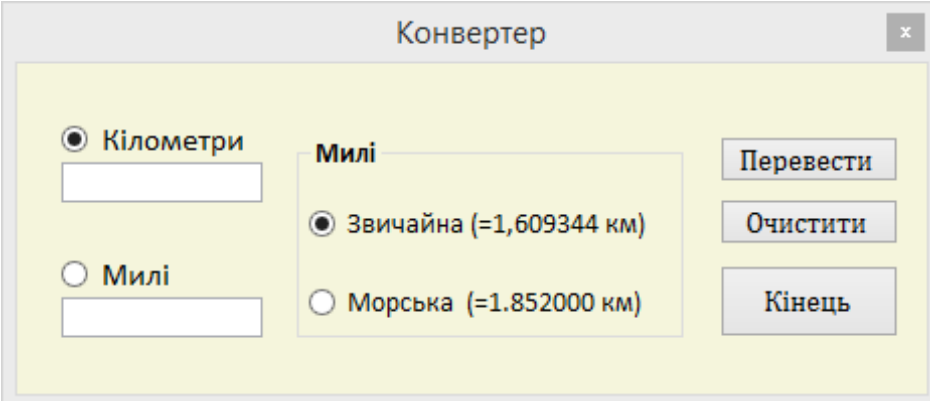


Рис. 6 Форма «Конвертер»

## Лабораторна робота ПРОГРАМУВАННЯ ЦИКЛІВ. ОБ'ЄКТИ: TEXTBOX, MENUSTRIP, CONTEXTMENUSTRIP, CHECKBOX, GROUPBOX

**Мета роботи.** Створити форму для розв'язування задачі табулювання функції. Створити у ній головне меню (об'єкт типу MenuStrip) з командами: закінчити роботу програми, табулювати функцію, очистити поля виведення результатів, а також контекстне меню (ContextMenuStrip), що міститиме команду для очистки поля виведення. Результати табулювання вивести у багаторядкове поле редагування (об'єкт типу TextBox). Передбачити можливість виведення результатів на екран, у файл, у масив. Напрямок виведення задати за допомогою трьох прапорців (об'єктів типу CheckBox), розташованих на панелі групи об'єктів (типу GroupBox) (див. рис. 7).

### ХІД РОБОТИ

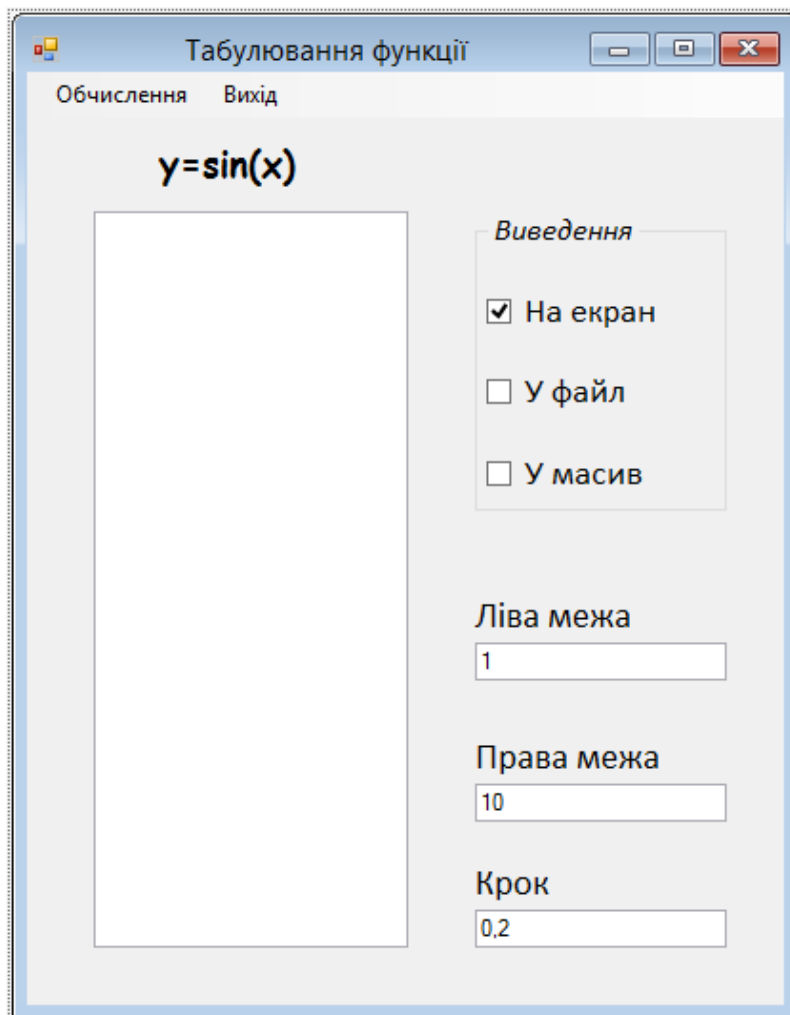
1. Завантажте середовище візуального програмування Visual Studio C++.
2. Створіть форму для розв'язування задачі табулювання функції.
3. Змініть заголовок Text форми на "Табулювання функції" (без лапок) і збільшіть розміри форми у вертикальному напрямку.
4. Розташуйте у формі поля редагування TextBox1, TextBox2, TextBox3 і відповідні їм текстові поля "Ліва межа", "Права межа", "Крок", а також текстове поле для вигляду заданої (згідно варіанту) функції, наприклад,  $y = \sin x$  (див. рис. 7, читай зауваження).

*Зауваження.* Для того, щоб швидко вставити у форму декілька однотипних об'єктів, клацніть на піктограмі цього об'єкта, утримуючи натиснутою клавішу Ctrl. Тепер вставляння у форму всіх об'єктів цього типу відбуватиметься без повторного вибору піктограми. Якщо

випадково вставите зайвий об'єкт, то вилучіть його за допомогою клавіші Delete. Щоб відмовитися від такого режиму, клацніть на зображенні стрілки на палітрі компонентів. Розмір, стиль і колір шрифтів виберіть на власний розсуд так, щоб форма виглядала якнайкраще. Збережіть форму.

**5. Вставте у форму панель групи об'єктів (об'єкт типу GroupBox).**

Для цього використайте компоненту GroupBox із закладки Standard. Змініть значення властивості Text цього об'єкта на слово "Виведення" (без лапок). Розмір, стиль і колір шрифту виберіть на власний розсуд. Збільшіть панель групи.



*Рис. 7 Готова форма*

**6. Вставте у панель три прапорці (об'єкти типу CheckBox).**

Для цього використайте компоненту CheckBox із закладки *Стандартные элементы управления* панелі елементів. Вирівняйте прапорці. Змініть значення властивості Text цих об'єктів на такі, як показано на рис. 7. Стель і колір шрифтів виберіть на власний розсуд.

**7. Встановіть прапорець "На екран" у положення "увімкнено".**

Для цього виокремте цей об'єкт та змініть значення його властивості Checked (контроль вибору) на True.

**8. Вставте у форму поле редагування (об'єкт TextBox) та зробіть його багаторядковим.**

Для цього вставте компоненту TextBox (у цьому прикладі об'єкт матиме назву textBox7). Збільшіть розміри поля. Встановіть властивість Multiline у значення True або ж встановіть прапорець Multiline на області об'єкта. Властивість ScrollBars (наявність смуг прокручування) цього об'єкта задайте як Both. Збережіть роботу.

**9. Задайте початкові значення для полів редагування лівої і правої меж аргумента функції та для кроку зміни цього аргументу, наприклад, такі, як на рис. 7.**

Для цього змініть властивість Text цих об'єктів. Для набору символу десяткової крапки використайте символ, передбачений операційною системою комп'ютера.

**10. Вставте у форму головне і контекстне меню (об'єкти типу MenuStrip і ContextMenuStrip).**

Для цього використайте компоненти MenuStrip і ContextMenuStrip із закладки *Меню и панели инструментов*. Вони розташуються під формою і на етапі виконання програми будуть невидимими.

**11. Введіть назви команд головного меню форми (див. рис. 8 і рис. 9).**

Для цього виберіть об'єкт menuStrip1 і задайте значення його

властивості Text. Інший шлях – виділіть сам об'єкт. У вікні, яке відкриється, вибирайте мишею рамку команди і записуйте її назву, наприклад, "Обчислення", як значення властивості Text у вікні *Свойства*. Створіть команди даного меню. Закрийте вікно створення команд головного меню MenuStrip1, клацнувши лівою кнопкою миші на вільному місці форми.

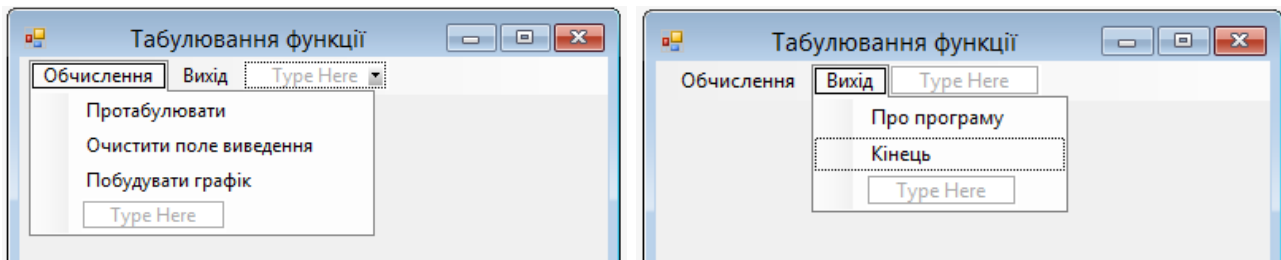


Рис. 8, 9 Створення головного меню

## 12. Введіть назви команд контекстного меню форми.

Для цього аналогічно змініть значення властивості Text об'єкта ContextMenuStrip. У вікні *Свойства* введіть текст "Очистити поле виведення" без лапок як значення властивості Text (рис. 10). За бажанням можете придумати ще якусь команду. Закрийте вікно створення меню. Збережіть форму.

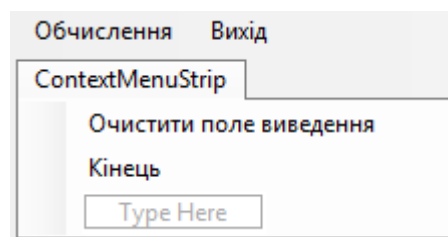


Рис. 10 Створення контекстного меню

## 13. Запрограмуйте команду "Очистити поле виведення" головного меню, скориставшись методом Clear об'єкта textBox7.

Методи об'єкта — це набір функцій, які, аналогічно до властивостей, застосовують до об'єкта.

Виконайте команду меню "Очистити поле виведення", не запускаючи програму на виконання (двічі клацніть на об'єкт). З'явиться



заготовка функції реакції на подію виклику цієї команди. У ній запишіть команду виклику методу Clear для очищення поля виведення об'єкта textBox7:

```
private: System::Void toolStripTextBox1_Click(System::Object^
sender, System::EventArgs^ e) {
    textBox7->Clear(); //Викликати метод об'єкта
textBox7, який очищає //багаторядкове поле редагування
}
```

**14. Запрограмуйте команду "Кінець" головного меню, скориставшись стандартною функцією Close().**

```
private: System::Void toolStripTextBox5_Click(System::Object^
sender, System::EventArgs^ e) {
    Close();//Закриваємо вікно програми
}
```

Збережіть виконану на даний момент форму у своїй папці (Файл => Сохранить все).

**15. Запрограмуйте команду "Очистити поле виведення" контекстного меню.**

Двічі клацніть на команді контекстного меню "Очистити поле виведення" у вікні ContextMenuStrip1. Текст функції очищення такий самий, як і у команди головного меню, а саме: textBox7->Clear(); (див. п. 11).

**16. "Прив'яжіть" контекстне меню contextMenuStrip1 до форми.**

Клацніть на формі та задайте властивість форми ContextMenuStrip як contextMenuStrip1.

*Зауваження.* Значення властивості форми MainMenuStrip автоматично встановлюється як menuStrip1 у момент створення головного меню (п. 13).

## 17. Запустіть створену програму та дослідіть її роботу.

Поекспериментуйте з багаторядковим полем редагування TextBox, вводячи та коректуючи у ньому будь-який текст: виокремте фрагмент тексту, скопіюйте, перемістіть чи вилучте цей фрагмент. Витріть текст за допомогою команди головного меню "Очистити поле виведення". Ще раз введіть текст і витріть його за допомогою контекстного меню (для виклику контекстного меню форми потрібно клацнути правою клавішею миші на вільному місці форми). Закінчіть роботу програми, клацнувши на команді *Кінець* головного меню.

## 18. Запрограмуйте команду "Протабулювати".

Властивість **Lines** об'єкта типу TextBox є комплексною, тобто також є об'єктом зі своїми властивостями та методами. Результат роботи цієї програми – це таблиця, що складається з декількох рядків. Щоб додати у поле TextBox новий рядок до таблиці, треба змінити значення комплексної властивості Lines (рядки) за допомогою її методу AppendText.

Виконайте команду "Протабулювати" з головного меню, клацнувши на ній один раз. З'явиться заготовка функції, яку заповніть так:

```
private: System::Void toolStripComboBox1_Click(System::Object^
sender, System::EventArgs^ e) {
    double a=Convert::ToDouble(textBox1->Text);
    double b=Convert::ToDouble(textBox2->Text);
    double h=Convert::ToDouble(textBox3->Text);
    if (checkBox1->Checked)
    {
        textBox7->AppendText("      X | Y \n");
        for (double x=a; x<b+h/2; x+=h)
            textBox7->AppendText(x.ToString("0.00")+
"+sin(x).ToString("0.000")+"\n");
            textBox7->AppendText(" \n");
    }
}
```

19. Виконайте програму і поекспериментуйте з різними значеннями лівої, правої межі та кроку аргументу. Закрийте вікно програми "Таблювання функції".
20. Збережіть створену програму у своїй папці.
21. Закрийте Visual Studio, запустіть створену програму і виконайте обчислення для різних початкових даних.
22. Продемонструйте створену форму викладачеві. Закінчіть роботу.

### ВПРАВИ

1. Додайте до контекстового меню команду "Кінець" і запрограмуйте її.
2. Створіть *ще одне* контекстне меню з командами "Зняти всі прапорці", "Встановити всі прапорці", "Встановити інверсивно" і запрограмуйте його. Таке меню має з'являтися після клацання правою клавішею миші на довільному місці панелі з прапорцями. Не забудьте "прив'язати" це меню (contextMenuStrip2) до панелі з прапорцями (див. п. 16).
3. Визначте кількість елементів масиву більших, ніж 0,5, і менших, ніж 1.
4. Передбачте у створеній програмі додаткову можливість для визначення максимального та мінімального значень функції.

*Підказка.* У тілі функції, що описує програмний код кнопки "Протабулювати", скористайтеся командами:

```
double max=sin(a); // Початкове максимальне значення
функції
...
    if (max<sin(x)) max=sin(x); // У циклі табулювання
...

```

```
textBox4->Text=Convert::ToString(max); // Виведення  
максимального значення у відповідне поле форми
```

Аналогічно для мінімального значення.

6. Змініть функції команди "Протабулювати" так, щоб для увімкненого прапорця "У масив" виведення результатів виконувалось в одновимірний масив.

7. Змініть функції команди "Протабулювати" так, щоб для увімкненого прапорця "У файл" виведення результатів виконувалось в текстовий файл.

*Підказка.* Скористайтеся такими командами:

```
FILE *ff;  
ff = fopen ("sin.txt","wt"); //Відкриваємо файл sin.txt для  
запису  
if (ff==NULL) textBox7->Text="Не вдалося відкрити файл!";  
else  
{  
    ● ● ●  
    // У циклі табулювання:  
    fprintf (ff, "%.2f   %.2f\n",x,sin(x));  
    fclose(ff); //Закриваємо файл  
    ● ● ●  
}
```

8. Вставте у форму об'єкт Chart із вкладки Данные для побудови графіка функції (рис. 11).

У розділі Chart, у полі Series⇒ChartType виберіть піктограму потрібного графіка. Набору даних, що відповідатиме цьому типу графіка, присвоється стандартне ім'я chart1. Створіть додатковий пункт меню або вставте кнопку "Намалювати графік" для отримання графіка і запрограмуйте так:

```
● ● ●  
double a=Convert::ToDouble(textBox1->Text);
```

```
//Одержуємо числові значення лівої
double b=Convert::ToDouble(textBox2->Text); //та
правої меж
double h=Convert::ToDouble(textBox3->Text);
//Одержуємо числове значення кроку
chart1->Series["Series1"]->Points->Clear();
for (double x=a; x<b; x+=h)
    chart1->Series["Series1"]->Points-
>AddXY(x,sin(x));
•••
```

Поекспериментуйте з різними вкладками з розділу Chart, щоразу запускаючи програму на виконання. Заберіть легенду (Legends). Придумайте та вставте заголовок (Titles) та налаштуйте осі (ChartAreas⇒Axes).

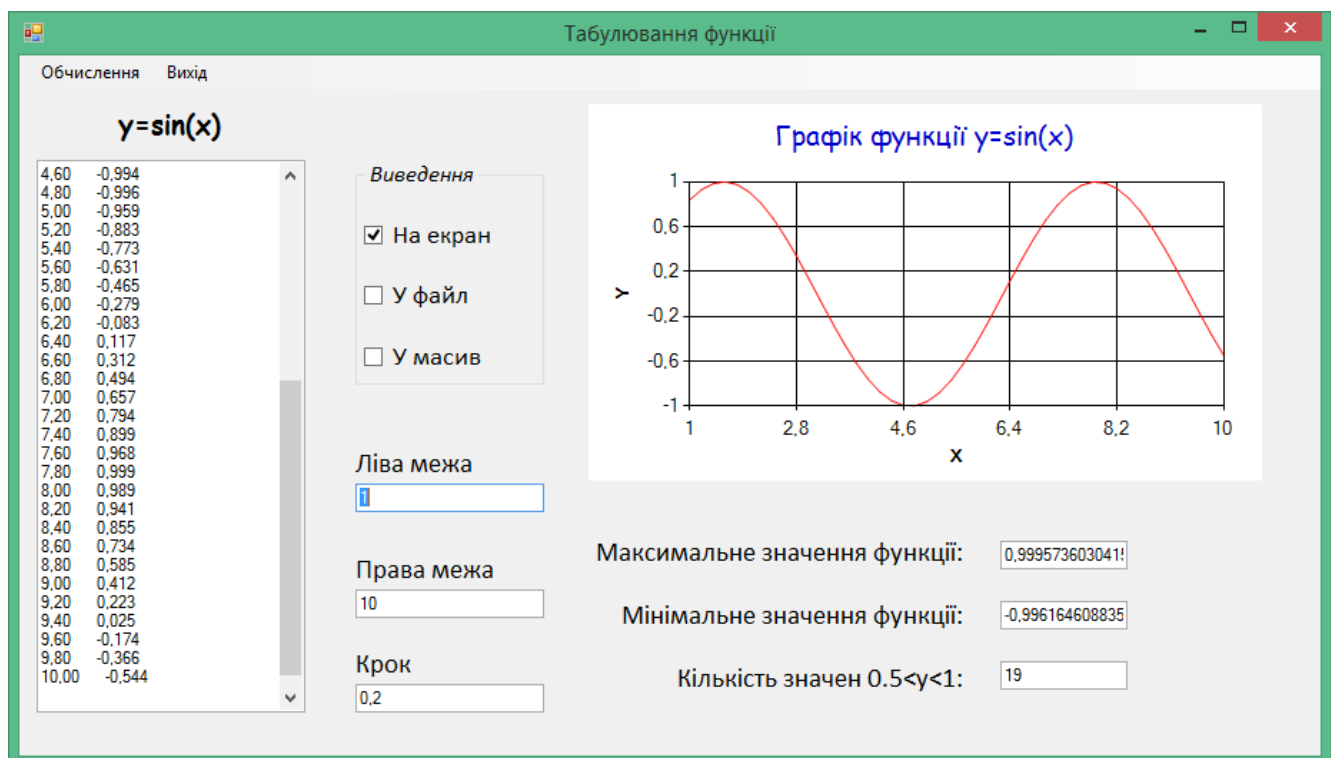


Рис. 11 Доповнена форма

9. Сумістіть побудову графіка з табулюванням функції.

### Варіанти:

1.  $Y = \sin x - 0.5;$

2.  $Y = \cos x - 0.2;$

3.  $Y = \sin 2x;$

4.  $Y = \cos 3x;$

5.  $Y = \sin^2 x;$

6.  $Y = \cos^2 x;$

7.  $Y = \sin x + \cos x;$

8.  $Y = \cos x - \sin x;$

9.  $Y = \sin (x+3);$

10.  $Y = \cos (x-5);$

11.  $Y = \sin (\cos (x));$

12.  $Y = \cos (\sin (x));$

13.  $Y = \sin x^2;$

14.  $Y = \cos x^2;$

15.  $Y = \sin x * \cos x.$

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Глинський Я.М., Анохін В.Є., Рязська В.А. С++ і С++ Builder. Львів : Деол, СПД Глинський, 2006. 192 с.
2. Кривцова О.П. Інформатика. Основи програмування у середовищі Microsoft Visual С++Express : навч. посіб. Полтава : ПНПУ імені В.Г. Короленка, 2018. 161 с.
3. Ткачук В.М. Основи візуального програмування : лабораторний практикум. URL: <http://docplayer.net/43751340-Osnovi-vizualnogo-programuvannya-laboratorniy-praktikum-ukladach-kandidat-f-m-nauk-docent-tkachuk-v-m-ivano-frankivsk-2015-r-s.html> (дата звернення: 02.12.2019).

Редакційно-видавничий відділ  
Рівненського державного гуманітарного університету