

**Мета роботи:** Опанувати методику створення адаптивного розширювального віконного інтерфейсу в середовищі Visual C++ 2019 для багаторазового розв’язання набору однотипних задач, із використанням декількох чисельних методів. У проєкті є можливість додавання інших нелінійних рівнянь і нових методів їхнього розв’язання. Здійснюється контроль за правильністю вхідних даних і адаптивне налаштування інтерфейсу на вхідні дані та вихідні результати кожного методу. Проєкт є навчальним.

Особливістю проєкту є те, що він **контролює і опрацьовує** різні нестандартні ситуації, такі, як заборона запуску програми на виконання, якщо вхідні дані введено некоректно, і **ввічливо повідомляє** про те користувача.

### Частина 1. Створення віконного проєкту для двох методів розв’язання нелінійних рівнянь: методу ділення напіл (МДН) і методу Ньютона (МН)

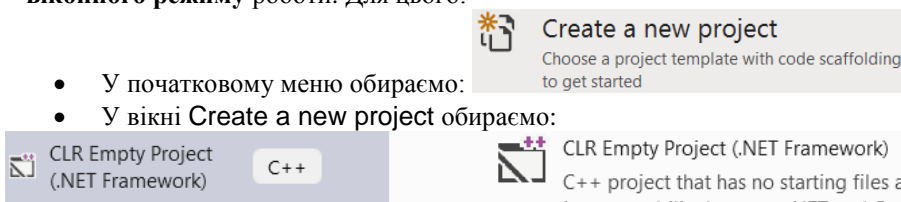
Перед початком створення цього проєкту необхідно повторити теоретичний матеріал до МДН і МН, який подано в методичних вказівках до лабораторних робіт №3 і №4. Слід також опрацювати матеріал Лекції 9 стосовно функцій C++, а також Лаб. роб 8 про консольний режим роботи у середовищі Visual C++ 2019.

У проєкті потрібно створити окрему функцію, яка реалізовує кожен чисельний метод розв’язання рівнянь, запрограмувати один із можливих механізмів вибору необхідного методу, а також інший механізм вибору нелінійного рівняння із декількох альтернатив.

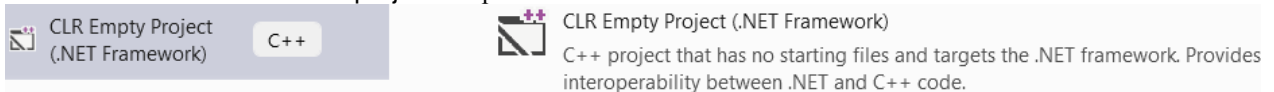
Послідовність створення віконного інтерфейсу у середовищі Visual C++ 2019, для одного із варіантів даного проєкту є такою:

1. Створюємо папку, наприклад, з іменем **XIO\_MDN\_MN\_VC++2019\_Form**.
2. Запускаємо середовище VC++ 2019 і виконуємо наступну послідовність дій, які необхідні для **створення віконного режиму** роботи. Для цього:

- У початковому меню обираємо:



- У вікні **Create a new project** обираємо:



і натискаємо **Next**. (Якщо у Вашій версії VC++ 2019 цей режим відсутній, його необхідно інстальовати!)

3. У наступному вікні вказуємо ім’я проєкту, налаштовуємось на папку, яку було для нього створено, і натискаємо **Create**.

## Configure your new project

CLR Empty Project (.NET Framework) C++ Windows Console

Project name

MyProject

Location

C:\XIO\_MDN\_MN\_VC++2019\_Form

Solution name ⓘ

MyProject

☒ Place solution and project in the same directory

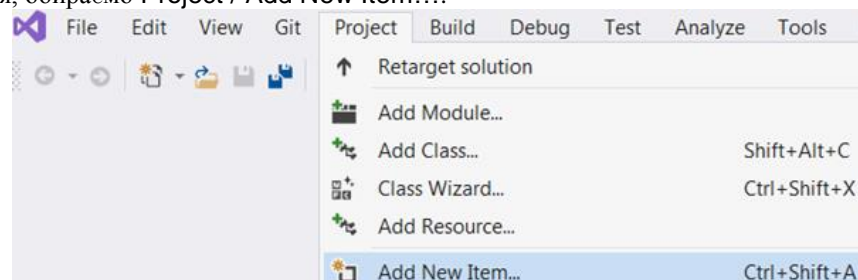
Framework

.NET Framework 4.8

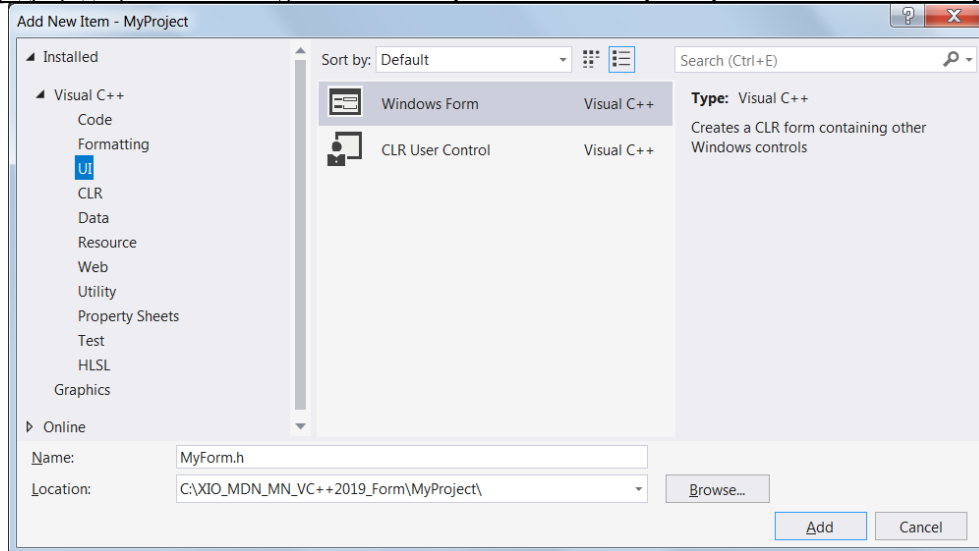
Back

Create

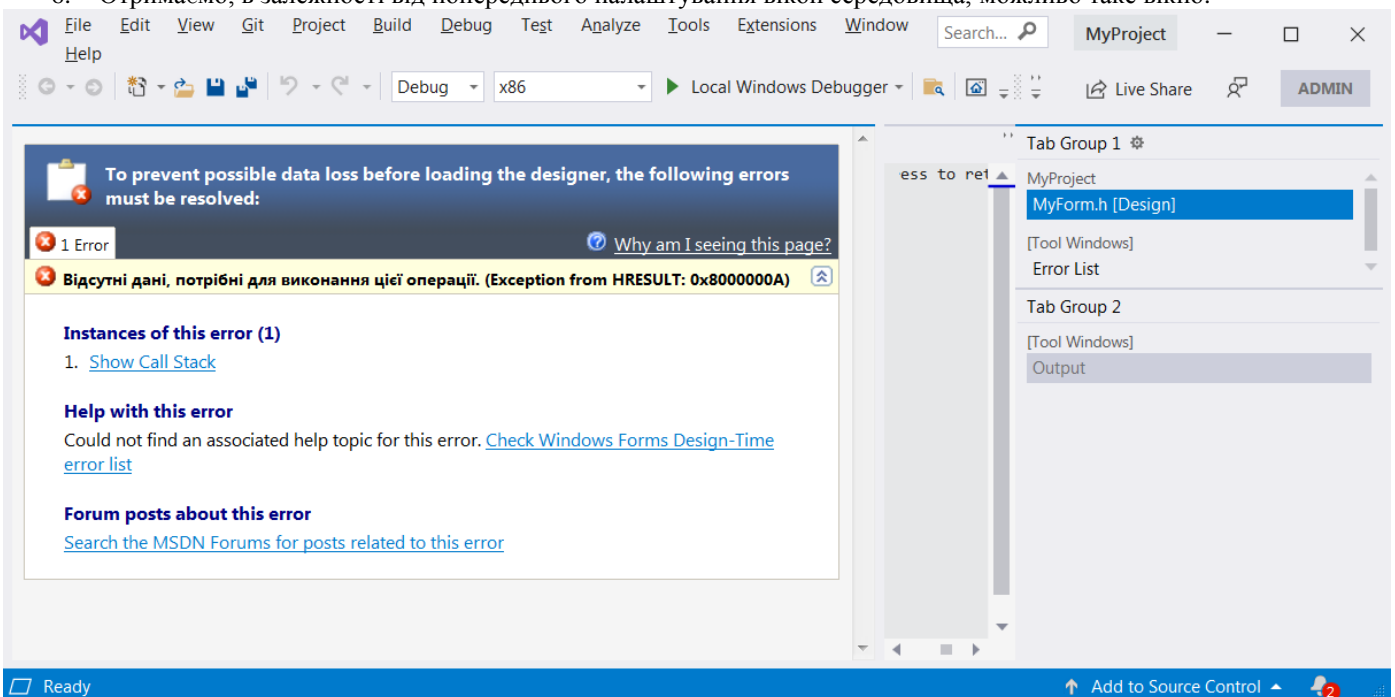
4. У вікні, що відкрилося, обираємо **Project / Add New Item...**:



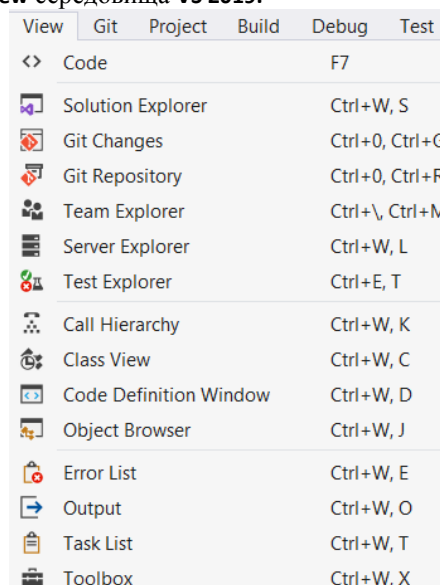
5. У наступному вікні обираємо: **UI / Windows Form** і натискаємо **Add**



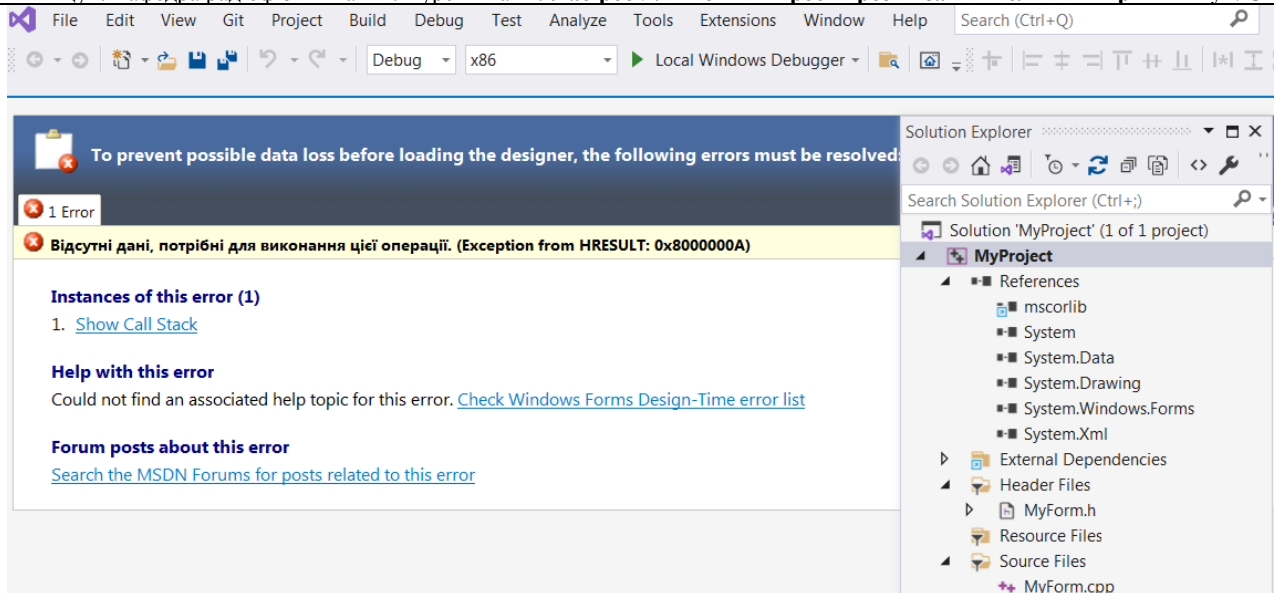
6. Отримаємо, в залежності від попереднього налаштування вікон середовища, можливо таке вікно:



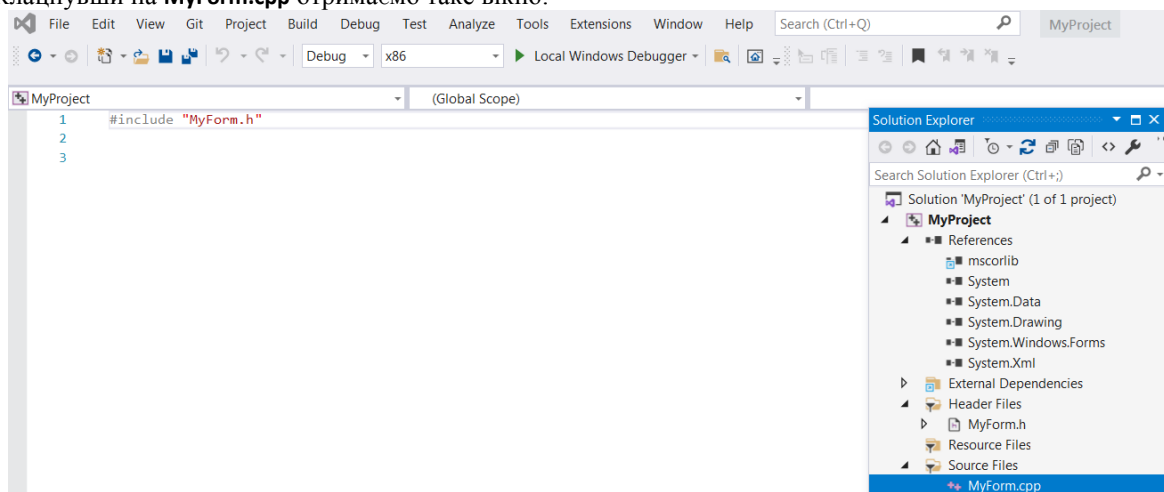
7. Нам потрібно, щоб на екрані було видно вікно **Solution Explorer**. Якщо воно не видиме, то виводимо його на екран за допомогою спадного меню **View** середовища **VS 2019**.



8. Відкривши вікно **Solution Explorer**, побачимо наступне:



9. Клацнувши на **MyForm.cpp** отримаємо таке вікно:



10. Заповнюємо його таким вмістом і закриваємо проект!:

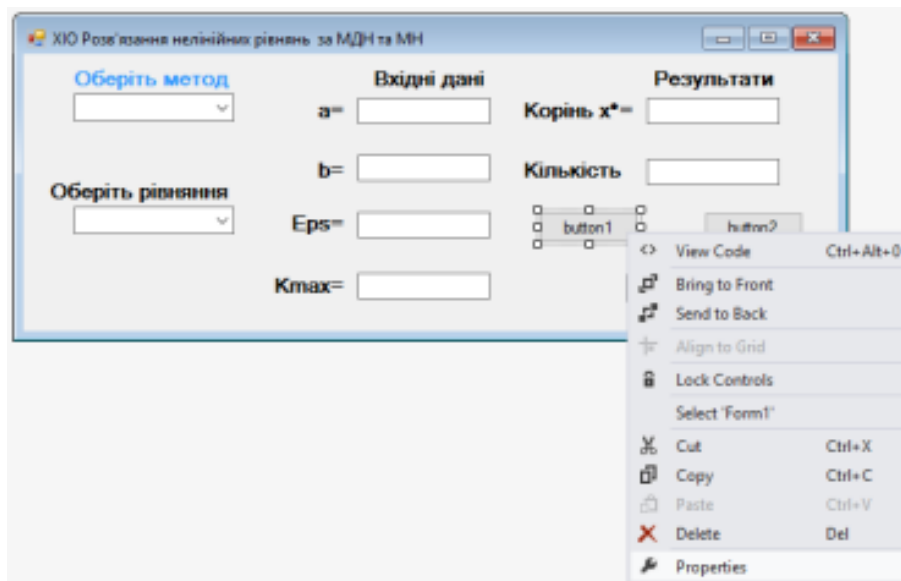
```
#include "MyForm.h"
#include <Windows.h>
#include <tchar.h>
using namespace MyProject;
// Головна функція WinMain (Точка входу)
[STAThread]
int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    Application::EnableVisualStyles(); // включаємо візуальні стилі
    // Візуалізація тексту
    Application::SetCompatibleTextRenderingDefault(false);
    Application::Run(gcnew MyForm); // запуск форми
    return 0;
}
```

Вміст папки з проектом буде таким:

Имя	Тип	Размер	Дата
..	<Папка>		31.10.2021 21:27
Debug	<Папка>		31.10.2021 21:10
MyForm	cpp	406	31.10.2021 21:27
MyProject	vcxproj	6 016	31.10.2021 21:10
MyProject.vcxproj	filters	1 101	31.10.2021 21:10
MyForm	h	1 356	31.10.2021 21:10
MyProject	sln	1 438	31.10.2021 21:08
MyProject.vcxproj	user	168	31.10.2021 21:08

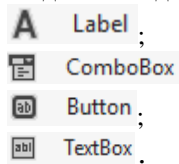
- Формуємо графічний інтерфейс проекту, використовуючи **Палітру** компонент (**Toolbox**) середовища **VC++ 2019**:
- Підписуємо форму проекту. Для цього її активізуємо, поставивши курсор миші на поле форми і один раз клацнувши лівою кнопкою. У вікні **Properties** обираємо курсором рядок **Text** і у віконці, що є праворуч, набираємо текст: **<Своє прізвище> Розв'язання нелінійних рівнянь за МДП та МН**.

3. Якщо вікно **Properties** є невидимим, то його можна вивести на екран активізувавши курсором мишки відповідну візуальну компоненту і клацнувши на ній правою кнопкою миші. Одержимо спливаюче меню, у нижній частині якого обираємо курсором рядок **Properties** :

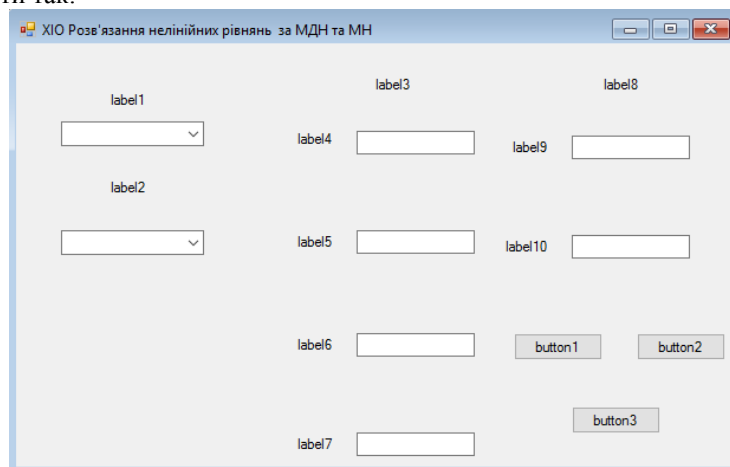


4. Розширивши за допомогою миші форму до потрібного розміру, наносимо на поле форми такі візуальні компоненти, обираючи їх на відповідних вкладках **Палітри**:

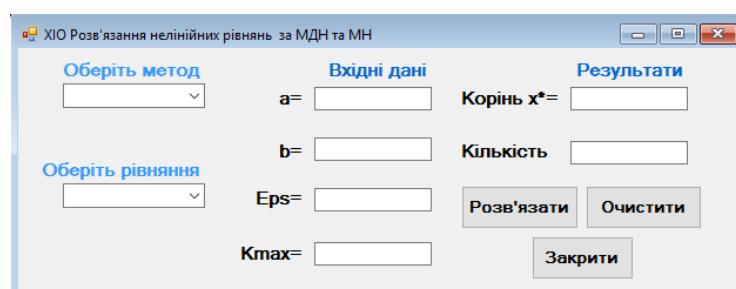
- 10 компонент **Label**
- 2 компоненти **ComboBox**
- 3 компоненти **Button**
- 6 компонент **Textbox**



Розташуємо ці компоненти так:



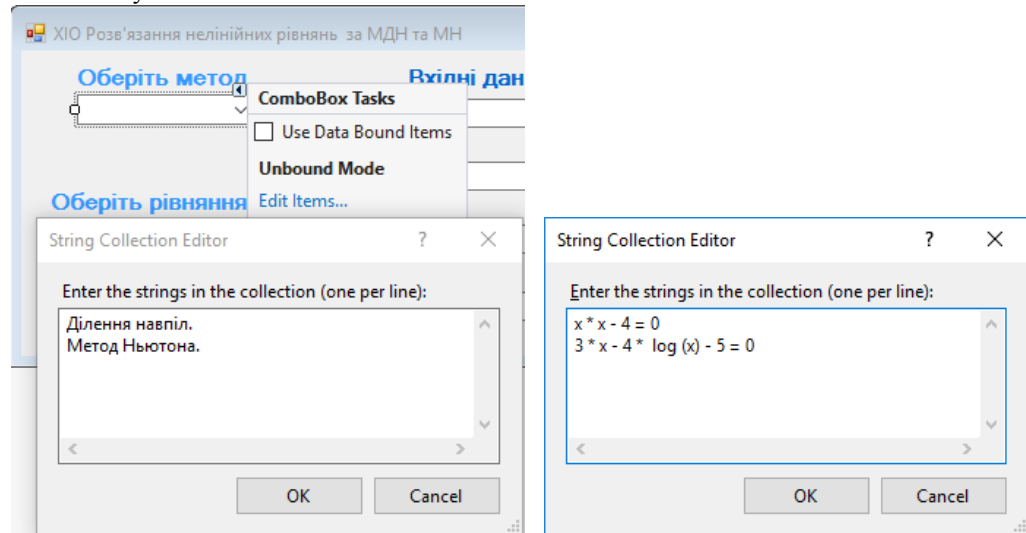
5. Активізувавши по черзі кожен із компонент **Label**, змінимо у вікні **Properties** їхні властивості **Text** так, як це подано нижче. Про цьому, використавши властивість **Font**, обираємо шрифт і колір текстів. Аналогічним чином змінюємо підписи компонентів **Button**.
6. Задаємо в обох компонентах **Combobox**, необхідний текст. У результаті, після оптимізації розміру форми, візуальний інтерфейс нашого проекту набуде такого вигляду:



7. Оскільки вхідний параметр **Kmax** для МДН не є потрібним, то зробимо спочатку його компоненти невидимими для введення. Для цього, активізувавши кожен із компонент (тобто **Label7** та **Edit4**), у вікні **Properties** значення їхніх властивостей **Visible** встановимо у **false**.

## Частина 2. Програмування візуальних елементів інтерфейсу

1. Програмуємо дві візуальні компоненти **ComboBox**. Для цього активізуємо по черзі відповідну компоненту і зробимо наступне:
  - обираємо курсором миші у правому верхньому куті активізованої компоненти маленький трикутник;
  - клацнувши лівою кнопкою миші, одержимо спадне меню, у якому клацнемо на рядок [Edit items...](#)
  - одержимо вікно, у якому задаємо, для першої компоненти, назви наших методів розв'язання нелінійних рівнянь, а для другої - наші нелінійні рівняння. Очевидно, що кількість методів і кількість рівнянь може бути більшою за 2.



2. Програмуємо обробку ситуації: зміна вмісту компонентів **ComboBox**

```
private: System::Void comboBox1_SelectedValueChanged(System::Object^ sender, System::EventArgs^ e)
{
    switch (comboBox1->SelectedIndex)
    {
        case 0:
        {
            label7 -> Visible = false;           // робимо невидимим вікно для введення Kmax
            textBox4 -> Visible = false;
            textBox1 -> Clear();
            textBox2 -> Clear();
        } break;
        case 1:
        {
            label7 -> Visible = true;           // робимо видимим вікно для введення Kmax
            textBox4 -> Visible = true;
            textBox1 -> Clear();
            textBox2 -> Clear();
        } break;
    }
}
```

3. Програмуємо функцію **f**, яка містить ліві частини обох заданих нами рівнянь. Щоб була можливість обирати потрібне нам рівняння, доцільно у заголовок функції **f** додати другий цілочисельний параметр, значення якого гратиме роль селектора в операторі **switch** для вибору необхідно рівняння.

```
double f(double x, char k1)           // ліві частини наших нелінійних рівнянь
{
    switch (k1)
    {
        case 0: return x * x - 4; break;
        case 1: return 3 * x - 4 * Math::Log(x) - 5; break;
    }
}
```

4. Описуємо функції, які обчислюють першу **f<sub>p</sub>** і другу **f<sub>2p</sub>** похідні функції **f**:

```
double fp (double x, double d, char k1) // перша похідна функції f
{
    return (f(x + d, k1) - f(x, k1)) / d;
}
double f2p (double x, double d, char k1) // друга похідна функції f
{
    return (f(x + d, k1) + f(x - d, k1) - 2 * f(x, k1)) / (d * d);
}
```

5. Описуємо функцію **MDP**, у якій реалізовано метод ділення навпіл (МДП)

```
double MDP (double a, double b, double Eps, char k1, int &L)
{
    double c, Fc;
```

```

while ( b - a > Eps )
{
    c = a + 0.5 * (b - a);
    L = L + 1; // лічильник кількості поділів інтервалу [a, b]
    Fc = f(c, k1);
    if (Math::Abs(Fc) < Eps) // перевірка, чи точка c не є поблизу кореня x*
        return c; // завершення роботи функції MDP
    if (f(a, k1) * Fc > 0)
        a = c;
    else b = c;
}
return c; // завершення роботи функції MDP
}

6. Описуємо функцію MN, у якій реалізовано метод Ньютона (МН)
double MN (double a, double b, double Eps, char k1, int Kmax, int &L)
{
    double x, Dx, D; int i;
    Dx = 0.0; D = Eps / 100.0;
    x = b;
    if (f(x, k1) * f2p(x, D, k1) < 0)
        x = a;
    if (f(x, k1) * f2p(x, D, k1) < 0)
        MessageBox::Show ("Для цього рівняння збіжність ітерацій не гарантована");
    for( i = 1; i <= Kmax; i++)
    {
        Dx = f(x, k1) / fp(x, D, k1);
        x = x - Dx;
        if (Math::Abs (Dx) < Eps)
        {
            L = i; // занесення у лічильник L результатуючої кількості ітерацій
            return x; // завершення роботи функції MN
        }
    }
    MessageBox::Show ("За задану кількість ітерацій кореня не знайдено");
    return -1000.0;
}

7. Програмуємо обробник події клацання на кнопці Button1: (Розв'язати)
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
{
    int L, Kmax, k = -1, m = -1;
    double D, Eps, a, b;
    L = 0;
    switch (comboBox1 -> SelectedIndex) // вибір методу :
    {
        case 0: m = 0; break; // метод ділення навпіл
        case 1: // метод Ньютона
        {
            m = 1;
            D = Eps / 100.0;
            label7 -> Visible = true;
            textBox4 -> Visible = true; // робимо видимим вікно для введення Kmax
        } break;
    }
    if (m == -1)
    {
        MessageBox::Show ("Оберіть метод!");
        comboBox1 -> Focus();
        return;
    }
    textBox1 -> Enabled = true;
    switch ( comboBox2 -> SelectedIndex) // вибір нелінійного рівняння
    {
        case 0: k = 0; break;
        case 1: k = 1; break;
    }
    if (k == -1)
    {
        MessageBox::Show ("Оберіть рівняння!");
        comboBox2 -> Focus();
        return;
    }
    // Перевіряємо правильність введення вхідних даних
    if (textBox1 -> Text == "")
    {
        MessageBox::Show ("Введіть число в textBox1");
        textBox1 -> Focus();
        return;
    }
    a = Convert::ToDouble (textBox1->Text);
    textBox2 -> Enabled = true;
    if (textBox2 -> Text == "")
    {

```

```

    MessageBox::Show ("Введіть число в textBox2");
    textBox2 -> Focus();
return;
}
b = Convert::ToDouble (textBox2->Text);
if (a > b)
{
    D = a;
    a = b;
    b = D;
    // міняємо місцями a і b
    textBox1 -> Text = Convert::ToString (a);
    textBox2 -> Text = Convert::ToString (b);
}
if (textBox3 -> Text == "" )
{
    MessageBox::Show ("Введіть число в textBox3");
    textBox3 -> Focus();
    return;
}
Eps = Convert::ToDouble (textBox3->Text);
if ((Eps > 1e-1) || (Eps <= 0)) // похибка Eps не повинна бути більшою за 0.1 або меншою за 0
{
    Eps = 1e-4; // задаємо примусово Eps = 0.0001
    textBox3->Text = Convert::ToString (Eps);
}
if (m == 0)
if (f(a, k) * f(b, k) > 0) // перевіряємо, чи є корінь на інтервалі [a, b]
{
    MessageBox::Show ("Введіть правильний інтервал [a, b]!");
    textBox1 -> Text = "";
    textBox2 -> Text = "";
    textBox1 -> Focus();
    return;
}
// Перевіряємо, чи межі інтервалу [a, b] не є поблизу кореня
if (Math::Abs (f(a, k)) < Eps)
{
    textBox5 -> Text = Convert::ToString (a);
    textBox6 -> Text = Convert::ToString (L);
    return;
}
if (Math::Abs (f(b, k)) < Eps)
{
    textBox5 -> Text = Convert::ToString (b);
    textBox6 -> Text = Convert::ToString (L);
    return;
}
switch (m)
{
    case 0: // обрали метод MDP - ділення навпіл
    {
        textBox5 -> Text = Convert::ToString (MDP(a, b, Eps, k, L)); // виконали метод ділення навпіл MDP
        textBox6 -> Text = Convert::ToString (L);
        label10 -> Text = "К-ть поділів=";
    } break;
    case 1: // обрали метод Ньютона
    {
        if (textBox4 -> Text == "" )
        {
            MessageBox::Show ("Введіть число в textBox4");
            textBox4 -> Focus();
            return;
        }
        Kmax = Convert::ToInt32 (textBox4 -> Text);
        textBox5 -> Text = Convert::ToString (MN (a, b, Eps, k, Kmax, L)); // виконали метод Ньютона
        textBox6 -> Text = Convert::ToString (L);
        label10 -> Text = "К-ть ітерац.=";
    } break;
}
}

```

#### 8. Програмуємо обробник події клацання на кнопці button2: (Очистити)

```

private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e)
{
    textBox1->Clear();
    textBox2->Clear();
    textBox3->Clear();
    textBox4->Clear();
    textBox5->Clear();
    textBox6->Clear();
    switch (comboBox1->SelectedIndex)
    {
        case 0:

```

```

{
    label7->Visible = false;           // робимо невидимим вікно для введення Kmax
    textBox4->Visible = false;
} break;
case 1:
{
    label7->Visible = true;           // робимо видимим вікно для введення Kmax
    textBox4->Visible = true;
} break;
}
}

```

9. Програмуємо завершення програми- обробник події клацання на кнопці **button3: (Завершити)**

```

private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e)
{
    Close();
}

```

#### Завдання:

1. Створити віконний проект, використовуючи методику, яку наведено вище.
2. Зрозуміти і вивчити кожен пункт цієї методики.
3. Провести тестування працездатності проекту для різних вхідних даних – коректних і не коректних.

#### Додаткові завдання:

4. Створити подібний проект, у якому метод ділення навпіл і метод Ньютона оформлено, як функції типу **void**.
5. Додати до проекту ще одне нелінійне рівняння продемонструвати його розв'язання методами проекту.
6. Написати електронний звіт про виконане Вами тестування проекту, у якому навести копії вікон із різними результатами.
7. Захистити звіт у викладача.