

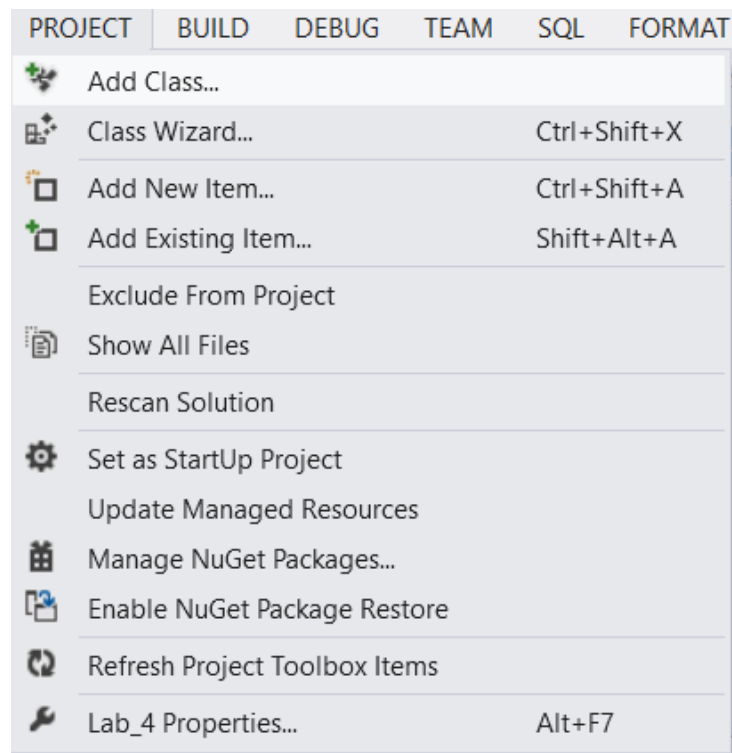
Лаб. 3. Створення власного класу. Метод дихотомії

Мета: Засвоїти структуру опису класу у мові C++, навчитись реалізувати функціонал та використовувати класи у основній програмі.

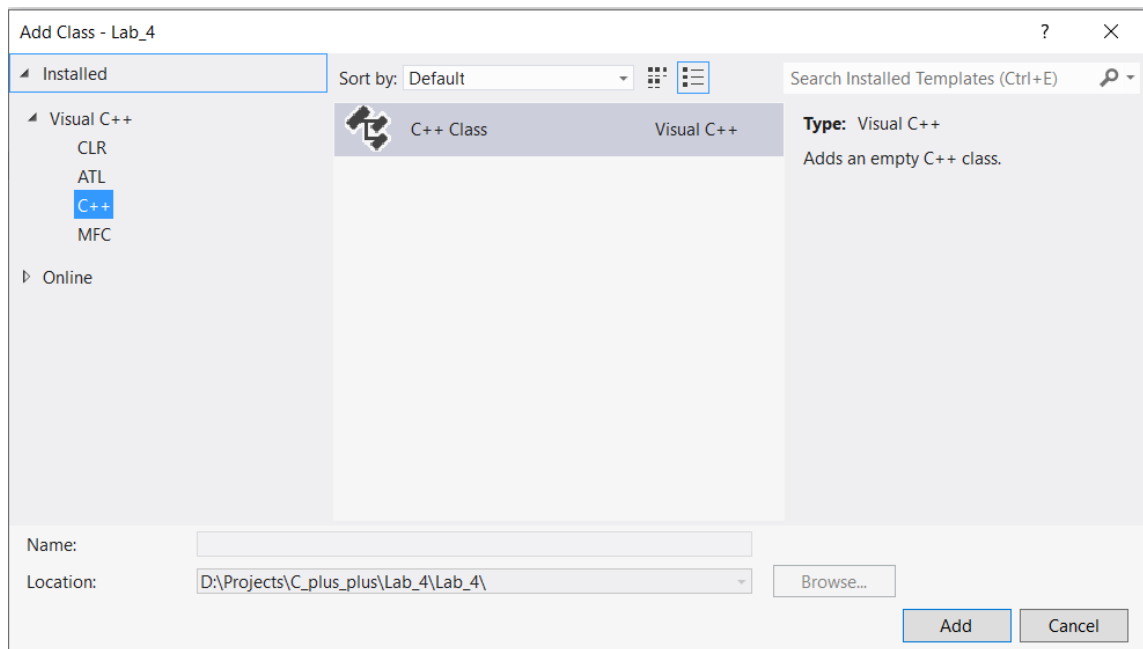
Завдання: Створити власний клас з реалізацією розв'язання нелінійного рівняння методом дихотомії (ділення відрізка навпіл) та табуляції ключової функції.

1. ClassWizard у Visual Studio

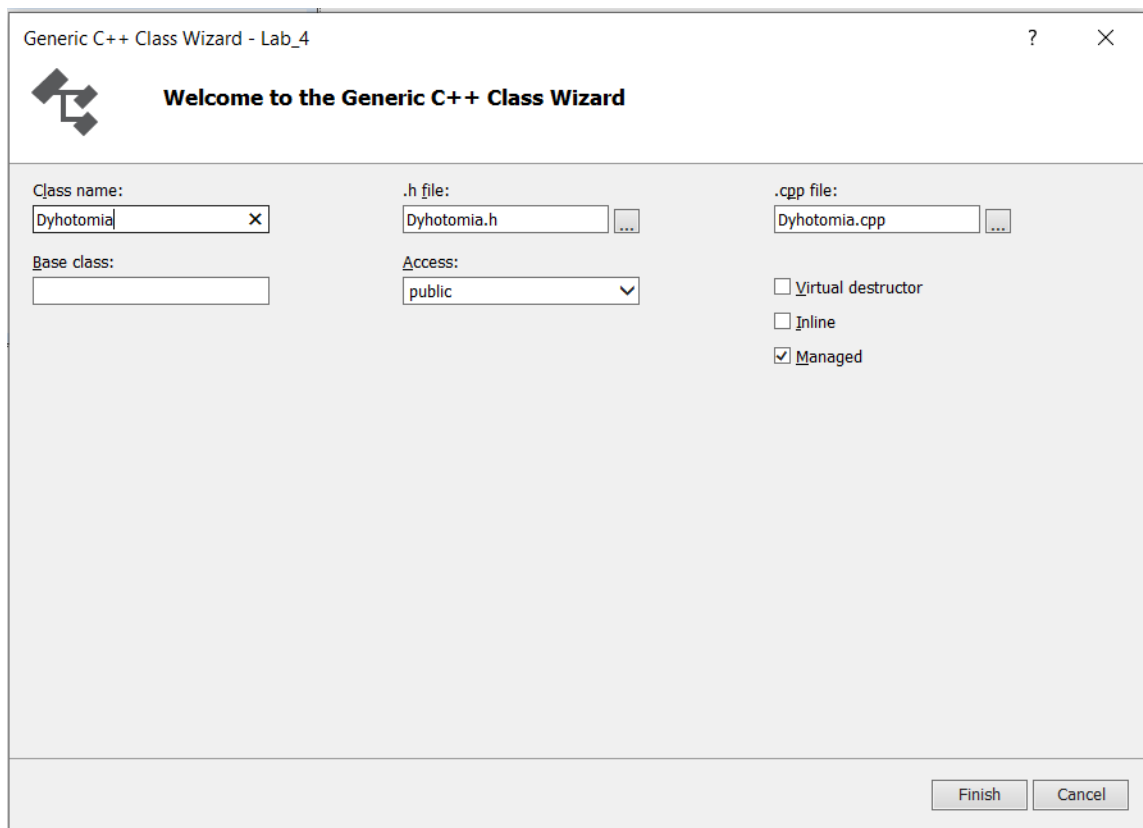
Для створення свого класу у Visual Studio виберемо Додати клас у меню Проект:



Отримаємо наступний екран, на якому оберемо Клас C++ та натиснемо Додати:

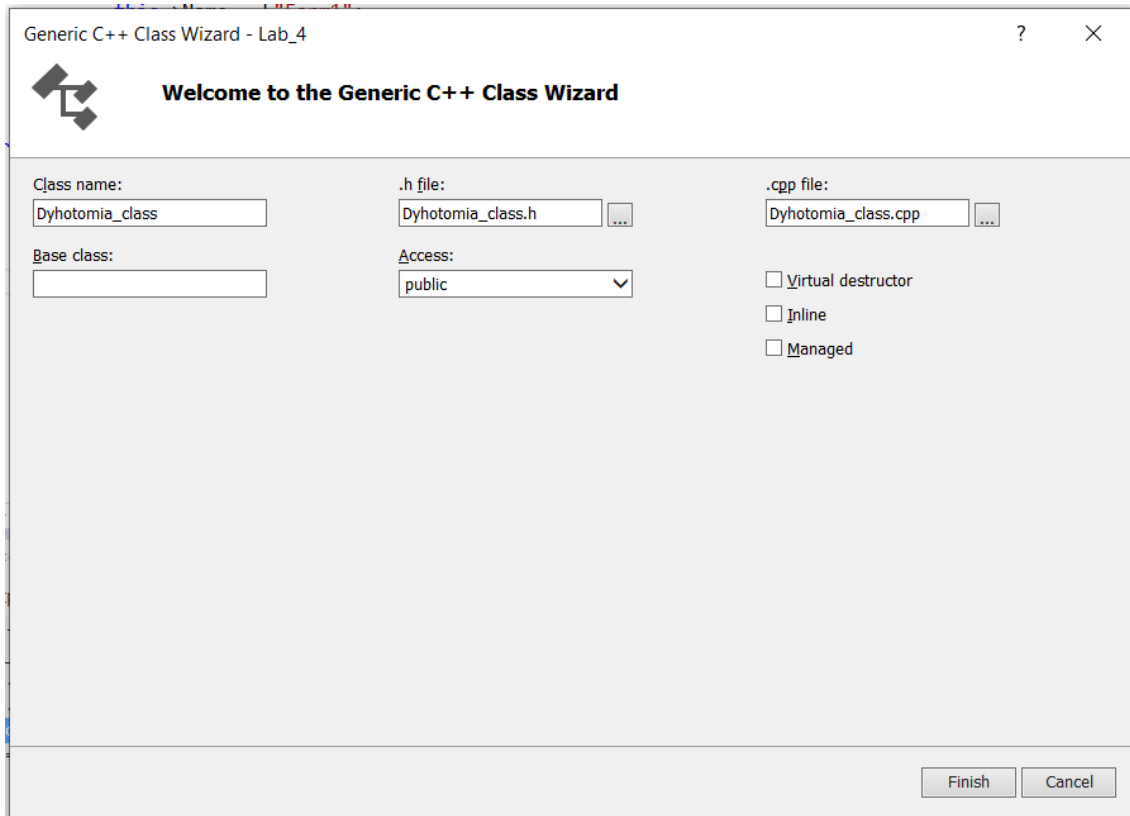


На наступному екрані налаштовуємо ім'я класу та інші налаштування.



За замовчуванням стоїть позначка Керований. У цьому випадку буде створений ref class.

Щоб створити звичайний класичний клас, позначку треба зняти.



У обидвох випадках система згенерує два файли із такою ж назвою, як і в класу з розширеннями .h та .cpp.

У першому представлений заголовок класу із конструктором (для керованого класу):

```
#pragma once
ref class Dyhotomia
{
public:
    Dyhotomia(void);
};
```

Та з конструктором та деструктором (для звичайного класу)

```
#pragma once
class Dyhotomia_class
{
public:
    Dyhotomia_class(void);
    ~Dyhotomia_class(void);
};
```

У другому файлі – реалізації – підключені бібліотеки та реалізація методів класу: поки що лише конструктора по замовченню

```
#include "stdafx.h"
#include "Dyhotomia.h"

Dyhotomia::Dyhotomia(void)
{
}
```

та реалізації конструктора та деструктора у випадку звичайного класу

```
#include "stdafx.h"
#include "Dyhotomia_class.h"

Dyhotomia_class::Dyhotomia_class(void)
{
}

Dyhotomia_class::~Dyhotomia_class(void)
{
}
```

У файлі заголовків треба описати поля, з які будуть параметрами методів класу. Вартує надати їм модифікатор доступу private

```
private:
    double a;
    double b;
    double eps;
```

Для зміни цих параметрів треба описати (у файлі заголовків) з модифікатором public

```
void setVolumes (double vol_a, double vol_b);
void setTolerance (double vol_eps);
```

Нагадаємо, що для того, щоб параметр методу чи простої функції можна було змінювати, його треба передати за посиланням

```
int count (double &x);
```

Усі методи треба реалізувати (у файлі реалізації). Приклад реалізації одного з методів може бути таким

```
void Dyhotomia::setVolumes (double vol_a, double vol_b)
{
    a = vol_a;
    b = vol_b;
}
```

У основній програмі треба підключити бібліотеку з нашим класом. Для цього у описаному випадку треба прописати на початку файлу основної програми треба прописати

```
#include "Dyhotomia.h"

#include "Dyhotomia_class.h"
```

Щоб використовувати функціонал нашого класу, треба створити змінну типу нашого класу – екземпляр класу. У випадку керованого класу це робиться так само, як було показано у минулій лабораторній:

```
Dyhotomia^ dyh = gcnew Dyhotomia ();
```

Для звичайного класу використовується класичний оператор new:

```
Dyhotomia_class * dyh = new Dyhotomia_class();
```

Звернення до методів класу буде однаковим.

```
int err = dyh ->count(x);
```

Нагадуємо, що об'єкт, створений оператором new треба знищити оператором delete, інакше він залишатись у пам'яті

```
delete dyh;
```

Саме ця команда запустить деструктор нашого класу.

2. Метод дихотомії для розв'язання нелінійних рівнянь

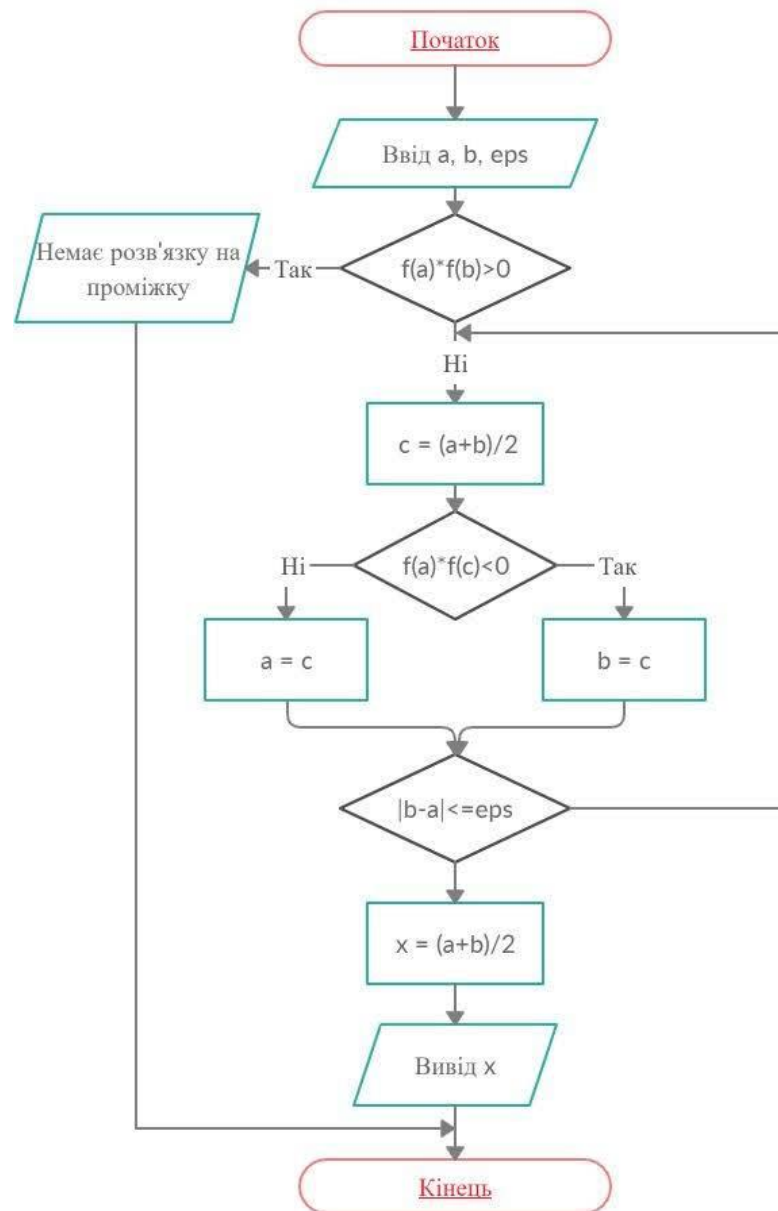
Одним із найпростіших методів розв'язання нелінійних рівнянь на визначеному проміжку є метод дихотомії також відомий як метод поділу відрізка навпіл.

Запишемо алгоритм методу дихотомії покроково:

1) Задати значення параметрів: кінців відрізка, на якому треба визначити розв'язок a та b , точність ϵ .

- 2) Для початку варто перевірити, чи є розв'язок рівняння на проміжку. Зрозуміло, що якщо графік функції перетинає вісь абсцис (тобто на проміжку є розв'язок), добуток значень функції на кінцях відрізка буде від'ємним.
- 3) Інакше вивести повідомлення про відсутність розв'язку на проміжку.
- 4) Розрахувати значення центру відрізка c .
- 5) Якщо розв'язок на проміжку ac , змінній b присвоїти значення c , інакше – присвоїти значення c змінній a .
- 6) Поки довжина відрізка більше заданого значення точності, повторювати кроки 4-5.
- 7) Обрахувати значення x як середини останнього відрізка.
- 8) Вивести значення x .

Представимо цей алгоритм у вигляді блок-схеми:



Варіанти завдань:

№ варіанту	функція $f(x)$	проміжок
1	$3\sin\sqrt{x} + 0.35x - 3.8$	$[2; 3]$
2	$x^2/4 + x - 1.2502$	$[0; 2]$
3	$x - 1/(3 - \sin(3.6x))$	$[0; 0.8]$
4	$0.1x^2 - 5x\ln(x) - 1$	$[1; 2]$
5	$\cos(2/x) - 2\sin(1/x) + 1/x$	$[1; 2]$
6	$3x - 4\ln(x) - 5$	$[2; 4]$
7	$x - 2 + \sin(1/x)$	$[0.9; 2]$
8	$e^x - e^{-x} - 2$	$[0; 1.5]$
9	$x + x^{1/2} + x^{1/3} - 2.5$	$[0.4; 1]$
10	$\operatorname{tg}(x) - (\operatorname{tg}^3(x) + 1)/3 + 0.2\operatorname{tg}^5(x)$	$[0; 0.8]$
11	$e^x + \ln(x) - 10x$	$[3; 4]$
12	$\cos(x) - \exp(-x^2/2) + x - 1$	$[1; 2]$
13	$\arccos(x) - (1 - 0.3x^3)^{1/2}$	$[0; 1]$
14	$(1 - 0.4x^2)^{1/2} - \arcsin(x)$	$[0; 1]$

Додаткове завдання

1. Побудувати графік функції з використанням функціоналу, вивченого при виконанні попередньої лабораторної роботи. Для цього залежно від розмірів зображення, на якому будемо відображати графік, розрахувати положення точок, які будуть відповідати графіку цільової функції та осям координат. Нагадаємо, що координати точок зображення у програмі починаються з верхнього лівого кута.