

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій

Звіт
про виконання лабораторної роботи №3
«Створення власного класу. Методи розв'язку рівнянь»

Виконав:
студент групи ФЕП-11с
Качмар Д. Б

Викладач:
Кушнір Олексій
Олександрович

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

Виконання роботи

Посилання на GitHub: <https://github.com/D-Kachm/OOP/tree/main/Lab3>

$$9 \quad x + x^{1/2} + x^{1/3} - 2.5 \quad [0.4; 1]$$

```
C++ main.cpp > main()
1  #include <iostream>
2  #include <cmath>
3  #include "function.h"
4
5  using namespace std;
6
7  // Метод дихотомії (бісекції)
8  double bisection(MyFunction func, double a, double b, double eps) {
9      double c;
10     while ((b - a) / 2 > eps) {
11         c = (a + b) / 2;
12         if (func.f(c) == 0.0)
13             break;
14         if (func.f(a) * func.f(c) < 0)
15             b = c;
16         else
17             a = c;
18     }
19     return (a + b) / 2;
20 }
21
22 // Метод Ньютона (дотичних)
23 double newton(MyFunction func, double x0, double eps) {
24     double x = x0;
25     while (abs(func.f(x)) > eps) {
26         x = x - func.f(x) / func.df(x);
27     }
28     return x;
29 }
30
31 int main() {
32     MyFunction f; // Об'єкт класу з функцією
33
34     double a = 0.4, b = 1.0, eps = 1e-6;
35     double x0 = 1.0;
36
37     // Метод дихотомії
38     double root_bis = bisection(f, a, b, eps);
39     cout << "Root found using Bisection Method: " << root_bis << endl;
40
41     // Метод Ньютона
42     double root_newton = newton(f, x0, eps);
43     cout << "Root found using Newton's Method: " << root_newton << endl;
44
45     return 0;
46 }
```

```

function.cpp > ...
1  #include "function.h"
2  #include <cmath>
3
4  // Реалізація f(x)
5  double MyFunction::f(double x) {
6      return x + sqrt(x) + cbrt(x) - 2.5;
7  }
8
9  // Реалізація f'(x)
10 double MyFunction::df(double x) {
11     return 1 + 0.5 / sqrt(x) + 1.0 / (3 * cbrt(x * x));
12 }
13

function.h > MyFunction
1  #ifndef FUNCTION_H
2  #define FUNCTION_H
3
4  // Клас функції f(x) і її похідної f'(x)
5  class MyFunction {
6  public:
7      double f(double x);    // f(x) = x + sqrt(x) + cbrt(x) - 2.5
8      double df(double x);   // f'(x)
9  };
10
11 #endif
12

```

Результати

```

PS D:\Github\OOP\Lab3> g++ main.cpp function.cpp -o solver
>> ./solver
>>
Root found using Bisection Method: 0.73762
Root found using Newton's Method: 0.737619
PS D:\Github\OOP\Lab3>

```

Висновок: У результаті виконання завдання я закріпив знання про структуру класу в мові C++ та навчився реалізовувати основні принципи об'єктноорієнтованого програмування, зокрема інкапсуляцію. Створення власних класів для розв'язання нелінійних рівнянь методами дихотомії та дотичних допомогло мені зрозуміти, як організувати функціонал у вигляді методів класу і використовувати їх у головній програмі для ефективного та зручного розв'язання математичних задач. Це сприяло розвитку практичних навичок програмування і глибшому розумінню застосування ООП у розв'язанні інженерних задач.