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

Звіт
про виконання лабораторної роботи №6
«Лямбда вирази у C++»

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

Викладач:
Кушнір Олексій
Олександрович

Львів-2025

Мета роботи: Ознайомитися з методами чисельного розв’язання нелінійних рівнянь — методом дихотомії (бісекції) та методом Ньютона (дотичних). Навчитися реалізовувати ці методи мовою C++ з використанням лямбда-виразів. Розвинути навички передачі функцій як параметрів та оптимізації коду шляхом скорочення структури програми без втрати її логіки.

Виконання роботи

Посилання на GitHub: https://github.com/D-Kachm/OOP/tree/main/Lab6

function.cpp > ...

```cpp
#include "function.h"
#include <cmath>

// Метод бісекції
double bisection(std::function<double(double)> f, double a, double b, double eps) {
    double c;
    while ((b - a) / 2 > eps) {
        c = (a + b) / 2;
        if (f(c) == 0.0)
            break;
        if (f(a) * f(c) < 0)
            b = c;
        else
            a = c;
    }
    return (a + b) / 2;
}

// Метод Ньютона
double newton(std::function<double(double)> f, std::function<double(double)> df, double x0, double eps) {
    double x = x0;
    while (std::abs(f(x)) > eps) {
        x = x - f(x) / df(x);
    }
    return x;
}

```

function.h > ...

```cpp
#ifndef FUNCTION_H
#define FUNCTION_H

#include <functional>

// Метод бісекції (дихотомії)
double bisection(std::function<double(double)> f, double a, double b, double eps);

// Метод Ньютона (дотичних)
double newton(std::function<double(double)> f, std::function<double(double)> df, double x0, double eps);

#endif

```

main.cpp > ...

```cpp
#include <iostream>
#include <functional>
#include <cmath>
#include "function.h"

using namespace std;

int main() {
    // Лямбда для f(x)
    auto f = [](double x) {
        return x + sqrt(x) + cbrt(x) - 2.5;
    };

    // Лямбда для f'(x)
    auto df = [](double x) {
        return 1 + 0.5 / sqrt(x) + 1.0 / (3 * cbrt(x * x));
    };

    double a = 0.4, b = 1.0, eps = 1e-6;
    double x0 = 1.0;

    // Виклик методу бісекції
    double root_bis = bisection(f, a, b, eps);
    cout << "Root found using Bisection Method: " << root_bis << endl;

    // Виклик методу Ньютона
    double root_newton = newton(f, df, x0, eps);
    cout << "Root found using Newton's Method: " << root_newton << endl;

    return 0;
}
```

Результати

```
PS D:\GitHub\OOP\Lab6> 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\Lab6>
```

Висновок: У цій лабораторній роботі я реалізував метод бісекції та метод Ньютона для розв’язання нелінійного рівняння. Для цього я використав лямбда-вирази замість окремого класу з функціями, що дозволило спростити структуру програми. Завдяки лямбдам код вийшов коротшим, зручнішим для розуміння та редагування. Я також краще зрозумів, як працює передача функцій як параметрів і наскільки це корисно при реалізації чисельних методів. У результаті роботи я навчився ефективніше використовувати можливості сучасного C++.