МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет електроніки і комп’ютерних технологій

**Звіт**

Про виконання лабораторної роботи № 6

## «**Лямбда вирази у С++**»

**Виконав:**

Студент групи ФЕП-14с

Грицюк Максим Олегович

**Перевірив:**

ас. Левуш П. Н.

Львів 2025

**Мета:** Вивчити синтаксис та дослідити можливості лямбда виразів у мові С++ останніх редакцій.

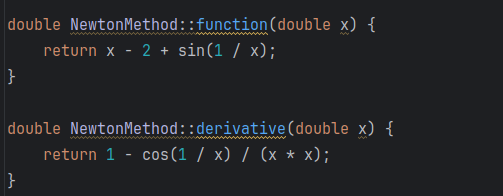
**Обладнання:**

Комп'ютер з встановленим програмним забезпеченням(Microsoft Visual Studio/Clion).

**Хід роботи**

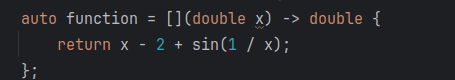
**Завдання:** Переписати код лабораторнох роботи, а саме №3 розв’язок нелінійного рівняння обидвома методами з використанням, де це доцільно, лямбда виразів.

1. Ось як виглядали функції function та derivative до внесення змін:



2. Я вирішив використати лямбда-функцію. Лямбда-функція дозволяє визначити функцію безпосередньо там, де вона буде використовуватися. Це зменшує кількість коду і робить його більш компактним, якщо функція потрібна лише в одному місці.

3. У початковій версії коду для методів дихотомії і Ньютона я використовував окремі функції function і derivative. Але тепер переписав під лямбда.



4. До змін у мене були окремі методи function і derivative для кожного класу (BisectionMethod і NewtonMethod). Ці методи використовувалися для обчислення значень функції та похідної. Вони були оголошені у заголовковому файлі і реалізовані у .cpp файлі.

Наприклад: метод function обчислював рівняння (f(x) = x - 2 + \sin(1/x)). Метод derivative обчислював похідну (f'(x) = 1 - \cos(1/x) / x^2). Це була класична реалізація, де кожна функція була чітко визначена, і кожен метод мав своє окреме місце в структурі класу.

Після змін:я переніс реалізації функцій function і derivative до лямбда-виразів безпосередньо в методах solve. Це зробило код компактнішим і зручнішим, оскільки ці функції використовуються лише в межах відповідних методів. Зберіг структуру SolverMethods.h - заголовковий файл майже не змінився, окрім того, що я видалив оголошення цих двух функцій. Лямбда-функції дозволили локально визначати логіку рівняння і його похідної, що зменшило залежність класу від зовнішніх функцій. Код став більш локалізованим: логіка функцій тепер знаходиться там, де вони використовуються.

**Висновок:**

Після аналізу початкового коду я вирішив переписати його, замінивши методи function і derivative на лямбда-функції. Це дозволило зробити логіку обчислення більш локалізованою. Лямбда-функції також спростили передачу логіки обчислення, оскільки більше не потрібно створювати окремі методи для одного завдання.

**p.s** Посилання на github: https://github.com/unknownpanic/OOP\_with\_cpp