# Лабораторна робота 7 «Тестування програмного забезпечення»

## Мета роботи

Метою даної лабораторної роботи є навчити студентів розробці та використанню автоматизованих тестів для забезпечення високої якості програмного забезпечення. Студенти використають бібліотеку xUnit.net, або аналогічну, для створення обширного набору тестів, які покривають максимально можливу кількість коду у своєму проекті. Через тестування студенти навчаться ідентифікувати та виправляти помилки, оптимізувати код та забезпечувати його надійність.

Лабораторна робота зосереджена на важливості тестування в процесі розробки програмного забезпечення. Студенти дізнаються про принципи та методики тестування, такі як модульне тестування, інтеграційне тестування, та тестування інтерфейсу. Розуміння цих концептів допоможе студентам створювати більш надійні та ефективні програмні рішення.

Основна увага буде приділена не тільки написанню тестів, але й аналізу покриття коду тестами, що дозволяє оцінити якість тестування та ідентифікувати частини коду, які потребують додаткової уваги. Це допоможе студентам зрозуміти, як забезпечити комплексне тестування своїх проектів, що є ключовим для розробки надійного програмного забезпечення.

## Опис завдання

У цій лабораторній роботі студенти мають завдання максимально покрити тестами свій проект за допомогою бібліотеки xUnit або іншої аналогічної бібліотеки тестування. Робота над цим завданням включає наступні кроки:

1. **Вивчення бібліотеки xUnit.net**: Студенти повинні ознайомитися з основами бібліотеки xUnit.net або іншої обраної бібліотеки тестування. Це включає розуміння основних концепцій та вивчення синтаксису для написання тестів, перевірок та використання анотацій.
2. **Аналіз існуючого коду проекту**: Перед написанням тестів, студенти повинні аналізувати свій проект, щоб визначити ключові компоненти та функції, які потребують тестування. Це допоможе в плануванні тестів та розумінні, які частини коду є критичними для проекту.
3. **Розробка тестових сценаріїв**: Студенти мають розробити тестові сценарії для кожної функції або модуля, які планують тестувати. Це включає написання як позитивних, так і негативних тестових випадків, щоб перевірити різні аспекти функціональності.
4. **Імплементація тестів**: Після розробки сценаріїв, студенти мають імплементувати тести, використовуючи бібліотеку тестування. Це включає написання конкретних тестових методів, які виконують перевірки на відповідність очікуваним результатам.
5. **Виконання тестів і аналіз результатів**: Студенти повинні виконати тести та проаналізувати результати. Усі помилки чи невдачі в тестах повинні бути детально досліджені і виправлені.
6. **Оцінка покриття коду тестами**: Важливо також оцінити покриття коду тестами. Студенти повинні прагнути до максимального покриття, звертаючи увагу на важливі частини програми.
7. **Документування та вивантаження результатів**: Результати тестування, включно з документацією про написані тести та аналіз покриття, повинні бути вивантажені в репозиторій GitHub.

Це завдання не тільки забезпечує високу якість програмного продукту, але й розвиває навички студентів у написанні надійних тестів та використанні сучасних інструментів автоматизації тестування.

## Звіт про виконану роботу

Звіт про виконану роботу має включати опис процесу розробки та тестування програмного проекту, використовуючи бібліотеку xUnit.net або аналогічну для автоматизації тестів. Студенти повинні детально описати, як вони реалізували новий функціонал або вдосконалили існуючий в рамках свого проекту та як ці зміни були покриті тестами.

Опис має включати інформацію про те, як було вивчено та застосовано основи бібліотеки тестування, включно з написанням тестових сценаріїв для різних компонентів програми. Студенти повинні вказати, як вони структурували тести для покриття різних аспектів програми, від простого перевірки поведінки до більш складних інтеграційних тестів.

Важливою частиною звіту є аналіз результатів тестування. Студенти мають описати, як тести були виконані, які проблеми були виявлені під час тестування та як вони були виправлені. Також має бути надана інформація про використання інструментів для вимірювання покриття коду тестами, з результатами, які показують, яку частину коду покривають тести, і де потрібно покращення.