

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Дисципліна «Автоматизоване тестування»

Лабораторна робота №2. Використання розробки на основі тестування.

Виконав:

студент гр. 7.F2.25-1
Бойчук Богдан Едуардович

Перевірів:

Викладач
Мильцев Олександр Михайлович

Запоріжжя
2025

Завдання до лабораторної роботи

1. Наведіть особливості TDD підходу.
2. Як модульне тестування використовується у методології TDD?
3. Як враховується специфікація програмного забезпечення та на якому етапі?

Скріншоти до лабораторної роботи

```
PS F:\Лабораторні роботи\Автоматизоване тестування\Лабораторна робота №2\lab 2> npm test

> lab-2@1.0.0 test
> mocha

Exception during run: file:///F:/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%B0%D1%96%20%D1%80%D0%BE%D0%B1%D0%BE%D1%82%D0%B8/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%BE%D0%B2%D0%B0%D0%B0%D0%B5%20%D1%82%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%B0%D0%B0%D1%8F/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%B0%20%D1%80%D0%BE%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%962/lab%202/test/matrix.test.js:2
import { transpose, add } from '../src/matrix.js';
            ^^^
SyntaxError: The requested module '../src/matrix.js' does not provide an export named 'add'
    at ModuleJob.instantiate (node:internal/modules/esm/module_job:124:21)
    at async ModuleJob.run (node:internal/modules/esm/module_job:190:5)
PS F:\Лабораторні роботи\Автоматизоване тестування\Лабораторна робота №2\lab 2>
```

Рис. 1. Результат червоної фази розробки.

```
PS F:\Лабораторні роботи\Автоматизоване тестування\Лабораторна робота №2\lab 2> npm test

> lab-2@1.0.0 test
> mocha

TDD Matrix Library
  transpose
    ✓ should transpose a 2x2 matrix
  add
    ✓ should correctly add two 2x2 matrices
    ✓ should throw an error when adding matrices of different sizes

3 passing (8ms)
PS F:\Лабораторні роботи\Автоматизоване тестування\Лабораторна робота №2\lab 2>
```

Рис. 2. Результат зеленої фази розробки.

```
PS F:\Лабораторні роботи\Автоматизоване тестування\Лабораторна робота №2\lab 2> npm test

> lab-2@1.0.0 test
> mocha

TDD Matrix Library
  transpose
    ✓ should transpose a 2x2 matrix
  add
    ✓ should correctly add two 2x2 matrices
    ✓ should throw an error when adding matrices of different sizes

3 passing (8ms)

PS F:\Лабораторні роботи\Автоматизоване тестування\Лабораторна робота №2\lab 2> █
```

Рис. 3. Результат фази рефактоингу.

Activity

🔍 All branches ▾	📁 All activity ▾	👤 All users ▾	🕒 All time ▾	Showing oldest first ▲
first commit				
LaughIngGas created main • e623a73 • 30 minutes ago				
first red test (transpose operation)				
LaughIngGas pushed 1 commit to main • e623a73...625f7b2 • 25 minutes ago				
first green test (added transpose operation to matrix library)				
LaughIngGas pushed 1 commit to main • 625f7b2...26609a4 • 23 minutes ago				
second red test (add operation)				
LaughIngGas pushed 1 commit to main • 26609a4...cc907a4 • 7 minutes ago				
second green test (added add operation to matrix library)				
LaughIngGas pushed 1 commit to main • cc907a4...84d35b9 • 7 minutes ago				
third red test (multiply operation)				
LaughIngGas pushed 1 commit to main • 84d35b9...01d0f84 • 6 minutes ago				
third green test (added multiply operation to matrix library)				
LaughIngGas pushed 1 commit to main • 01d0f84...736158d • 4 minutes ago				
refactored matrix library				
LaughIngGas pushed 1 commit to main • 736158d...14c3bda • 26 seconds ago				

Рис. 4. Відображення процесу TDD за допомогою репозиторію GitHub.

Посилання на репозиторій GitHub

<https://github.com/Laugh1ngGas/automatic-tests-lab2.git>

Контрольні запитання

1. Наведіть особливості TDD підходу.

Розробка на основі тестування (TDD) – це методологія, яка визначає пріоритетність створення та перевірки тестів на кожному етапі розробки.

Основні особливості TDD:

- Цикл "Червоно-зелений рефакторинг". TDD зазвичай дотримується цього безперервного циклу.
- Тести перед кодом. Наголошується на створенні модульних тестів перш ніж писати будь-який фактичний код.
- Написання коду лише при невдалому тесті. Методологія TDD рекомендує програмістам створювати новий код лише у випадку, якщо автоматизований тест не вдається.
- Запобігання дублюванню. Цей підхід запобігає дублюванню коду. Етап рефакторингу спеціально передбачає усунення дублювання.
- Рефакторинг. Після проходження тесту код переробляється відповідно до обраної архітектури програмного забезпечення.

2. Як модульне тестування використовується у методології TDD?

Модульне тестування є невід'ємною частиною TDD. Воно використовується для керування процесом розробки:

- Написання тесту (Червона фаза). Спочатку створюється модульний тест для функціональності, яка ще не реалізована. Цей тест використовує фрагмент коду так, ніби він вже існує, і очікувано завершується невдачею.
- Написання коду (Зелена фаза). Далі розробляється програмний код, достатній для того, щоб цей модульний тест пройшов успішно.
- Ітерація. Процес повторюється — створюється наступний тест, який буде невдалим, доки не буде додано новий код для його проходження.

3. Як враховується специфікація програмного забезпечення та на якому етапі?

Специфікація програмного забезпечення враховується на першому етапі розробки.

Розробники використовують специфікації вимог до програмного забезпечення для того, щоб створити тести, які демонструють, як повинен функціонувати код. По суті, перший крок у TDD – це

планування та створення тестів (на основі специфікацій) для кожного аспекту функціональності програми.