Міністерство освіти й науки України

Івано-Франківський національний технічний

університет нафти і газу

Кафедри ІПЗ

Лабораторна робота №3

# Тема: «Написання модульних тестів»

Варінт №10

Виконав студент

Групи ІП-21-2

Кавецький О.О

Перевірив

Храбатин Р. І.

м. Івано-Франківськ

2025р.

**Мета:** навчитися створювати модульні тести, розібратися в існуючих інструментах для створення модульних тестів та навчитися їх обирати.

**Короткі теоретичні відомості:**

Модульне тестування (unit testing) – це метод тестування, що дозволяє перевіряти окремі модулі або функції програмного забезпечення на коректність. Воно проводиться ізольовано від інших частин системи, що допомагає швидко знаходити та виправляти помилки.

Популярні інструменти для модульного тестування JavaScript:

* **Jest** – популярний тестовий фреймворк із простим синтаксисом.
* **Mocha** – забезпечує гнучкість у виборі бібліотек перевірок (Chai, Expect тощо).
* **Jasmine** – підтримує специфікацію BDD (Behavior Driven Development).
* **Node.js test-runner** – вбудований інструмент для тестування у Node.js.

Бібліотеки перевірок (assertion libraries):

* **Assert** – вбудована бібліотека Node.js.
* **Chai** – забезпечує гнучкі методи перевірки.
* **Expect** – використовується в Jest для декларативного стилю тестування.

**Виконання:**

Для виконання завдання використовуємо **Jest**:

Відкриваємо термінал і створюємо папку проєкту:

mkdir unit-testing && cd unit-testing

Ініціалізуємо npm-проєкт:

npm init -y

Встановлюємо Jest:

npm install --save-dev jest

Додаємо скрипт для запуску тестів у package.json:

"scripts": {

"test": "jest"

}

**3. Код функцій для тестування**

У файлі mathFunctions.js реалізуємо наступні функції:

export const sinh = a => Math.sinh(a);

export const cosh = a => Math.cosh(a);

export const tanh = a => Math.tanh(a);

**4. Написання модульних тестів**

Створюємо файл mathFunctions.test.js та додаємо тести:

const { sinh, cosh, tanh } = require("./mathFunctions");

describe("sinh function", () => {

  test("sinh(0) should be 0", () => {

    expect(sinh(0)).toBe(0);

  });

  test("sinh(1) should be approximately 1.175", () => {

    expect(sinh(1)).toBeCloseTo(1.175, 3);

  });

  test("sinh(-1) should be approximately -1.175", () => {

    expect(sinh(-1)).toBeCloseTo(-1.175, 3);

  });

  test("sinh(2) should be approximately 3.626", () => {

    expect(sinh(2)).toBeCloseTo(Math.sinh(2), 3);

  });

  test("sinh(-2) should be approximately -3.626", () => {

    expect(sinh(-2)).toBeCloseTo(Math.sinh(-2), 3);

  });

});

describe("cosh function", () => {

  test("cosh(0) should be 1", () => {

    expect(cosh(0)).toBe(1);

  });

  test("cosh(1) should be approximately 1.543", () => {

    expect(cosh(1)).toBeCloseTo(1.543, 3);

  });

  test("cosh(-1) should be approximately 1.543", () => {

    expect(cosh(-1)).toBeCloseTo(1.543, 3);

  });

  test("cosh(2) should be approximately 3.762", () => {

    expect(cosh(2)).toBeCloseTo(3.762, 3);

  });

  test("cosh(-2) should be approximately 3.762", () => {

    expect(cosh(-2)).toBeCloseTo(3.762, 3);

  });

});

describe("tanh function", () => {

  test("tanh(0) should be 0", () => {

    expect(tanh(0)).toBe(0);

  });

  test("tanh(1) should be approximately 0.761", () => {

    expect(tanh(1)).toBeCloseTo(0.761, 2);

  });

  test("tanh(-1) should be approximately -0.761", () => {

    expect(tanh(-1)).toBeCloseTo(-0.761, 2);

  });

  test("tanh(2) should be approximately 0.964", () => {

    expect(tanh(2)).toBeCloseTo(0.964, 3);

  });

  test("tanh(-2) should be approximately -0.964", () => {

    expect(tanh(-2)).toBeCloseTo(-0.964, 3);

  });

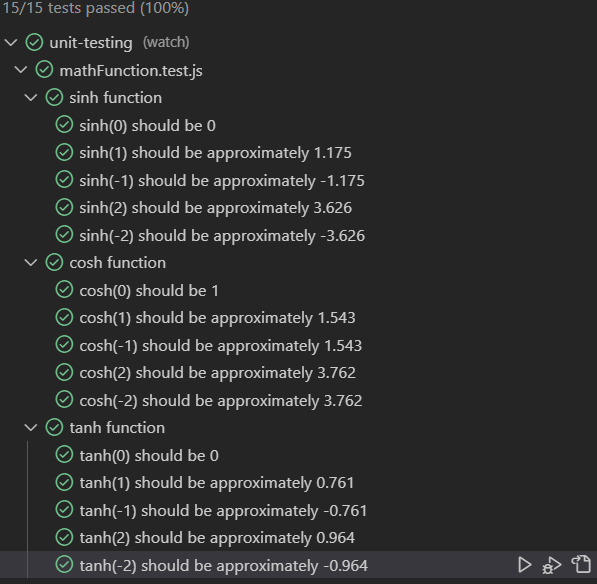
});

**5. Запуск тестів**

Запускаємо тести за допомогою команди:

npm test

Очікуваний результат: всі тести проходять успішно.



**Висновок:**

В ході лабораторної роботи було:

* Ознайомлено з принципами модульного тестування.
* Налаштовано середовище для написання тестів.
* Реалізовано функції та відповідні тести.
* Виконано тести за допомогою Jest.

Отримані знання дозволяють ефективно перевіряти коректність роботи модулів та виявляти помилки на ранніх етапах розробки.

Посилання на репозиторій - <https://github.com/olehkavetskyi/FAT-Labs>