МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

Протокол практичної роботи №1

з дисципліни об'єктно-орієнтоване програмування

на тему: «**Тестування та налагодження програмного забезпечення**»

Виконав студент групи

ІПЗ-2 курс

Дибчук О. М.

Прийняв

Рудніченко М.Д.

Одеса, 2023

**ЗМІСТ**

[**ВСТУП 3**](#_Toc132390065)

[**ПРАКТИЧНА ЧАСТИНА 4**](#_Toc132390066)

[**ВИСНОВОК 11**](#_Toc132390067)

[**ЛІТЕРАТУРА 12**](#_Toc132390068)

# 

# ВСТУП

**МЕТА ЛАБОРАТОРНОЇ РОБОТИ:**

Мета даної лабораторної роботи полягає у навчанні студентів основних принципів тестування та відлагодження програмного забезпечення. Під час виконання роботи студенти будуть знайомитися з основними методами тестування, використанням юніт-тестів та іншими інструментами, що допомагають знайти та виправити помилки в програмному коді.

# **ПРАКТИЧНА ЧАСТИНА**

Завдання

1 Відкрийте обрану IDE і створіть проєкт для виконання операцій з фінансами (Додаток

А).

2. Підключіть до проекту бібліотеку JUnit.

3. Створіть каркас для юніт-тестів (Наприклад, в IDE Eclipse можна вибрати потрібний клас, відкрити

контекстне меню, і вибрати New->Junit test case, у діалоговому вікні, що з'явилося, вибрати методи,

для яких знадобляться юніт-тести).

4. Створіть юніт-тест (Додаток А).

5. Налагодьте та запустіть юніт-тест.

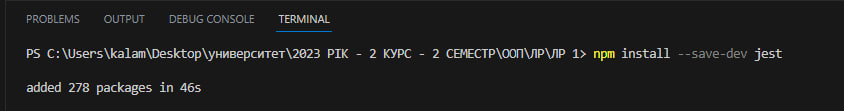
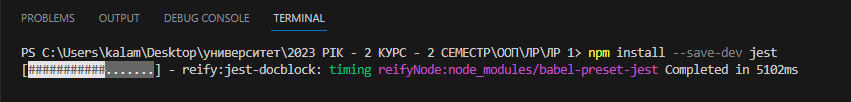
6. Оцініть результати виконання юніт-тестування та зробіть відповідні висновки.

7. Необхідно подати вихідний код, опис і результати роботи юніт-тесту на перевірку

правильності інтеграції модулів системи.

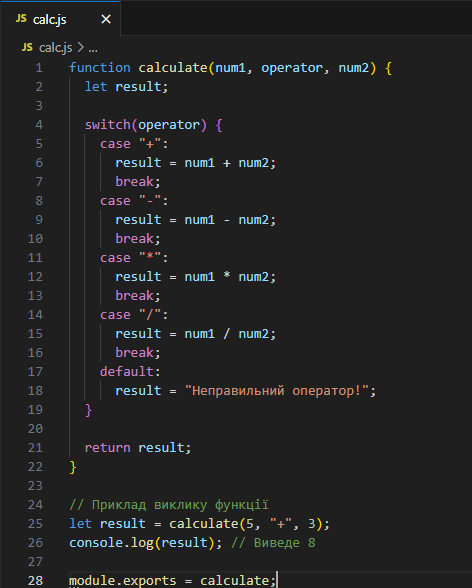
8. Викласти висновки за результатами тестування та виконаної лабораторної роботи у звіті.

Я вирішив виконати завдання вище за допомогою JS, Node JS та його фреймоворку для юніт-тестів – Jest. Для початку завантажимо цей фреймворк через командний рядок Ноди наступним чином: (рис. 1, рис. 2):



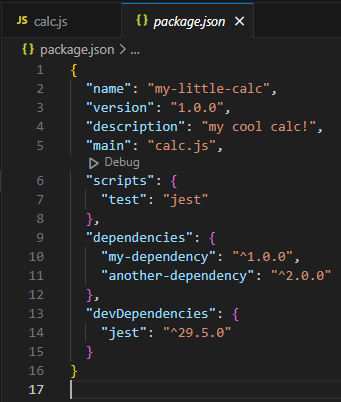
Риcунок 1, 2 – завантаження фреймворку у VS Code.

Після успіху, ми напишемо у файлі calc.js звичайний калькулятор на 4 дії, і експортуємо його у модуль «calculate» (рис. 3):

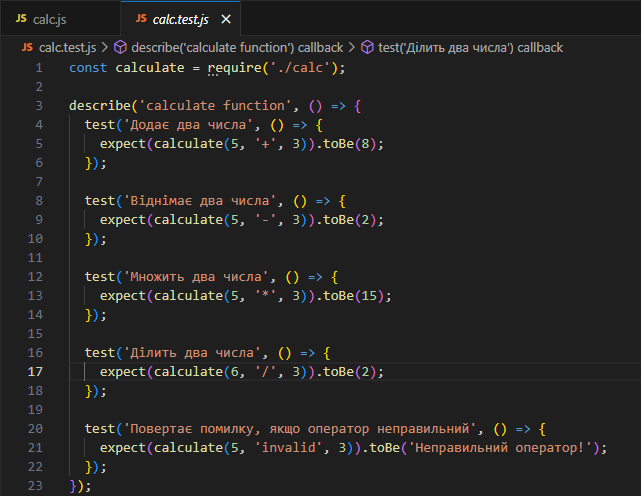


Риcунок 3 калькулятор на чотири дії у VS Code.

Щоб провести наше юніт-тестування, напишемо у новий файл саме для цього код, який зможе довести, що наш калькулятор працює коректно, з синтаксисом, який застосує наш фреймворк Jest. Перед цим, пропишемо у наш «package.json» назву нашого проекту та скрипт, яким ми будемо запускати наше тестування через фреймворк (рис. 4, рис. 5):

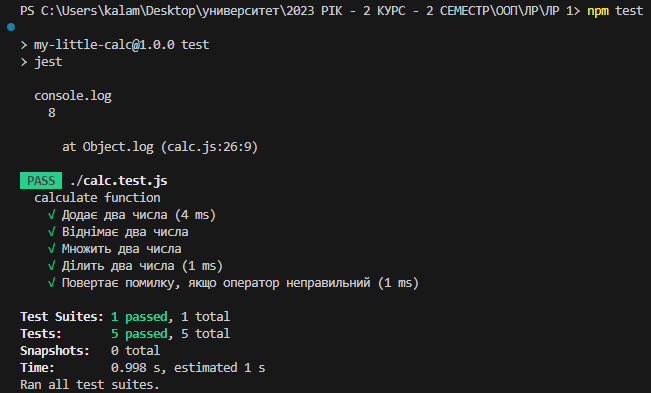


Риcунок 4 «package.json» у VS Code.



Риcунок 5 файл з кодом для тестування у VS Code.

Виконаймо тестування, прописавши у термінал – «npm test» (рисунок. 6):

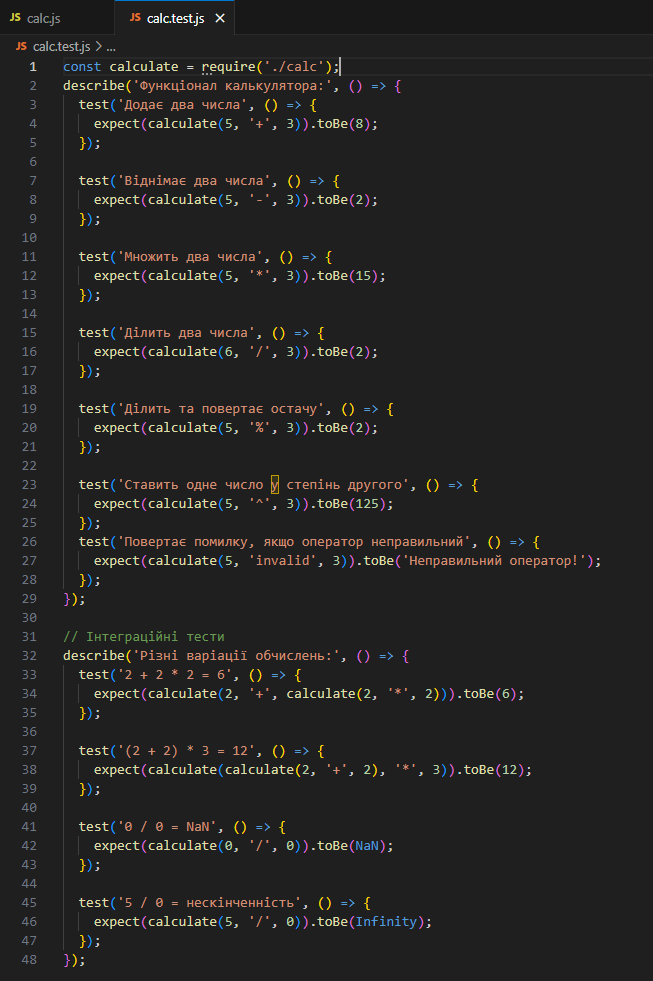


Риcунок 6 результати тестування у VS Code.

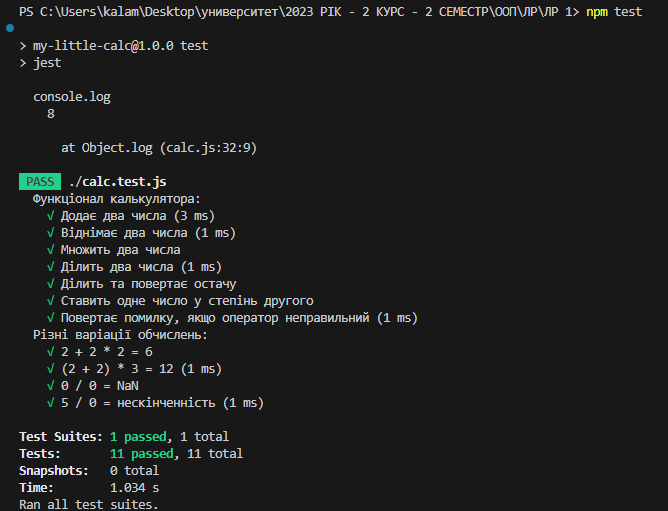
Як бачимо, результати задовільні. Додамо до калькулятору дві нових дії та проведемо модульні та інтеграційні тести (рис. 7, 8, 9):



Риcунок 7. Оновлений калькулятор у VS Code.



Риcунок 8. Оновлені модульні та інтеграційні тести у VS Code.



Риcунок 9. Результати нових тестів у VS Code.

Як бачимо, результати – задовільні.

# **ВИСНОВОК**

Після виконання лабораторної роботи я ознайомився з основами модульного тестування в JavaScript з фреймворком Jest. Я створив простий калькулятор з декількома операціями і написав для нього модульні тести, перевіривши роботу кожної операції окремо. Крім того, я написав інтеграційний тест, щоб перевірити роботу всього калькулятора як єдиного цілого.

Під час виконання роботи я дізнався про те, що модульне тестування є важливою складовою процесу розробки програмного забезпечення, оскільки воно дозволяє виявити та виправити помилки на ранніх етапах розробки, що збільшує якість та надійність програми. Також я зрозумів, що інтеграційні тести допомагають перевірити взаємодію різних компонентів програми, що забезпечує її коректну роботу в цілому.

У результаті виконаної роботи я здобув нові знання та практичні навички з модульного тестування на прикладі фреймворку Jest, що дозволить мені створювати якісніші та надійні програмні продукти.

# 

# ЛІТЕРАТУРА

Посилання на **git**: https://github.com/coltnavydaddy/OOP\_Lab\_3\_Dybchuk.git