МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Дисципліна «Автоматизоване тестування»

Лабораторна робота № 3

Виконав:

студент гр. 6.1213

Явдощен Яків Валерійович

Перевірив:

Викладач

Кудін О.В.

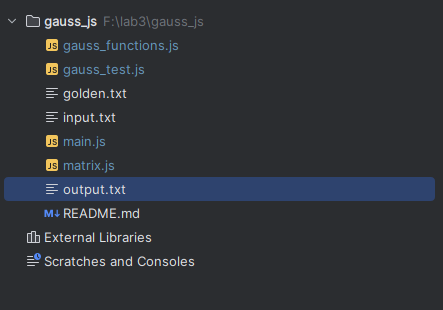
Запоріжжя

2023

Завдання для самостійної роботи

1. Ознайомитись з програмною реалізацією алгоритму розв’язання СЛАР <https://github.com/LiannaAsatryan/gauss_js>
2. Реалізувати mock методи класу Matrix.





1. Створити github репозиторій з тестами.

**https://github.com/Kallaban/automated-testing/tree/main/lab3/gauss\_js**

Контрольні запитання

1. Приклади використання Fake, Stub та Mock:

Fake: Фейк (Fake) - це об'єкт, який імітує деяку реальну функціональність, але робить це дуже спрощено. Наприклад, ви можете створити фейк для бази даних, який зберігає дані в оперативній пам'яті замість реальної бази даних. Видалення, оновлення та отримання даних будуть відбуватися в імітованому середовищі без реального взаємодії з базою даних.

Stub: Запуск (Stub) - це об'єкт, який замінює певну функціональність об'єкта або модуля для тестування. Наприклад, якщо у вас є функція, яка взаємодіє з зовнішнім API, ви можете використовувати запуск для заміни викликів до API на піддавання під контроль сценаріїв для тестування.

Mock: Мок (Mock) - це об'єкт, який записує і перевіряє виклики до об'єкта або функції під час виконання тестів. Ви можете використовувати мок для перевірки того, що певні функції були викликані з необхідними аргументами та в необхідній кількості разів.

1. Особливості модульного та приймального тестування:

Модульне тестування: Модульне тестування спрямоване на тестування окремих компонентів або модулів програми, незалежно від інших частин системи. Це дозволяє визначити правильність роботи окремих функцій або класів. Модульні тести легко підтримувати і виконувати швидко, але вони можуть не виявити проблеми, які виникають при взаємодії компонентів.

Приймальне тестування: Приймальне тестування (іноді відоме як інтеграційне тестування) спрямоване на перевірку правильної взаємодії між компонентами системи. Це включає тестування взаємодії між модулями, різними сервісами, базами даних тощо. Приймальне тестування може виявити проблеми, які не були б помічені на рівні модульного тестування.

1. Призначення бібліотек Sinon та Sinon-Chai:

Sinon: Sinon - це бібліотека для створення фейків, запусків і моків в JavaScript. Вона дозволяє вам контролювати поведінку функцій і об'єктів під час модульного тестування. Sinon допомагає створювати і перевіряти виклики функцій, створювати фейкові таймери та події, і багато іншого.

Sinon-Chai: Sinon-Chai - це розширення бібліотеки Chai для тестування з використанням Sinon. Воно надає зручний і зрозумілий синтаксис для перевірки викликів і інших аспектів Sinon об'єктів у ваших тестах.

Ці бібліотеки допомагають вам створювати ефективні модульні тести, які перевіряють, чи працюють ваші функції і компоненти правильно, та чи взаємодіють вони між собою належним чином.