

Цели:

- Разработать модуль для выполнения основных операций с матрицами: сложение, вычитание, умножение на число, умножение матриц и вычисление детерминанта.
- Написать набор юнит-тестов для проверки его корректности.

Задачи:

#### 1. Разработка модуля:

- Создайте модуль `matrix_operations.py`.
- В модуле определите класс `Matrix`, содержащий следующие методы:
  - `__init__(self, matrix)` конструктор, принимающий двумерный массив (список списков) в качестве матрицы.
  - `__str__(self)` метод для печати матрицы в удобочитаемом виде.
  - `add(self, other)` метод для сложения двух матриц.
  - `subtract(self, other)` метод для вычитания двух матриц.
  - `multiply_by_number(self, number)` метод для умножения матрицы на число.
  - `elementwise_multiply(self, other)` метод для умножения двух матриц.
  - `determinant(self)` метод для вычисления детерминанта матрицы.

#### 2. Написание юниттестов:

- Создайте отдельный файл `test_matrix_operations.py` для юнит-тестов.
- Используйте библиотеку `unittest` Python для написания тестов для каждого метода в классе `Matrix`.
- Протестируйте метод `add` на матрицах разных размеров, убедитесь, что он возвращает корректный результат или ошибку при попытке сложения матриц разных размеров.
- Аналогично протестируйте методы `subtract`, `multiply_by_number`, `elementwise_multiply`, убедитесь, что они корректно обрабатывают граничные случаи.
- Для метода `determinant` протестируйте вычисление детерминантов для матриц.

Помните о проверке корректности входных данных для каждого метода. Методы должны генерировать исключения или возвращать значимые сообщения об ошибках при некорректных входных данных.

Каждый юнит-тест должен покрывать несколько случаев использования метода, включая граничные условия.