Кукульчук Владислав i-01

Docker та розгортання програмного забезпечення:

Завдяки контейнеризації, Docker спрощує процес розгортання програмного забезпечення. Використання контейнерів дозволяє упаковувати програмне забезпечення та всі його залежності в ізольований контейнер. Цей контейнер легко переноситься між різними середовищами, що робить розгортання передбачуваним та простим. Оскільки весь стек програм та його конфігурація розташовані в одному контейнері, використання Docker є популярним завдяки його простоті, швидкості розгортання та здатності працювати в різних середовищах.

Управління командою в розробці програмного забезпечення:

Ефективне управління командою включає розподіл завдань, забезпечення ефективної комунікації, мотивацію, планування та визначення ролей і відповідальностей. Спільна відповідальність за код, ефективний обмін інформацією та чітке визначення цілей є важливими аспектами успішної розробки. Використання методів Agile та інструментів для спільної роботи, таких як системи управління завданнями, може підвищити продуктивність команди.

Git та управління змінами в коді:

Git, як система контролю версій, дозволяє відстежувати зміни в програмному коді. Це особливо важливо для колективної роботи, де розробники можуть одночасно працювати над різними гілками коду та об'єднувати їх. Git забезпечує історію змін, що полегшує відкат до попередніх версій і вирішення конфліктів.

Стратегії деплойменту програмного забезпечення:

Ключові стратегії деплойменту включають канаринські релізи (послідовне впровадження нових функцій для обмеженої аудиторії), блакитно-зелені релізи (перемикання між двома версіями), плаваючі релізи (поступове впровадження частин функціоналу) та інші. Вони є важливими для мінімізації впливу помилок та забезпечення стабільності системи.

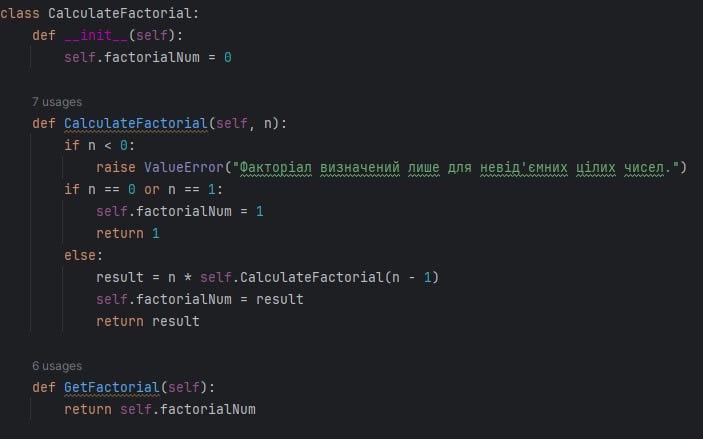
Перевірка та валідація в розробці програмного забезпечення:

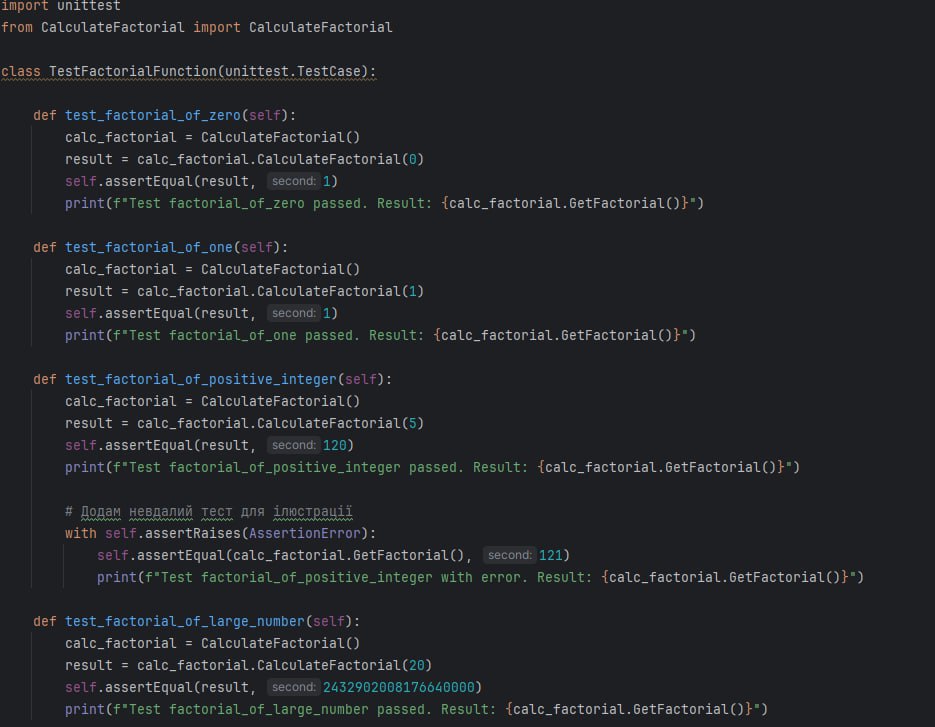
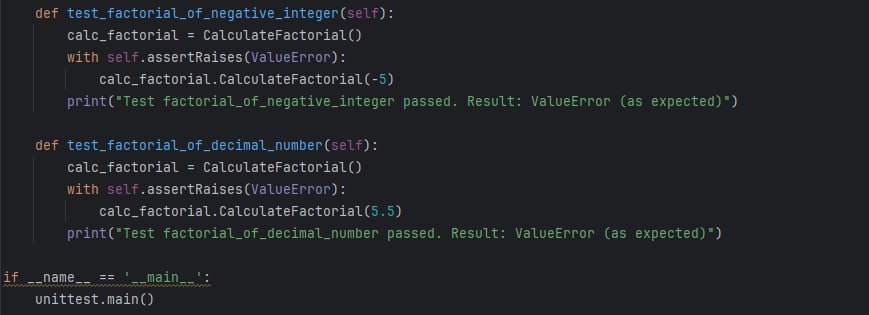
Процес перевірки та валідації включає тестування програмного коду для виявлення помилок та переконання в тому, що програмне забезпечення відповідає вимогам. Це важливо для забезпечення якості продукту та виявлення можливих проблем до випуску продукту. Валідація переконується, що система відповідає функціональним вимогам, тоді як перевірка забезпечує відповідність коду стандартам і вимогам програмування.

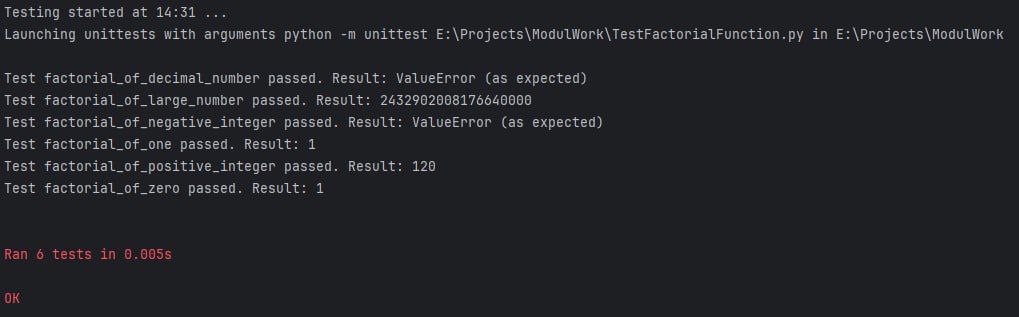
**Практичне завдання:**

В результаті тестів було зазначено, що функція коректно обчислює факторіали для різних вхідних значень та правильно обробляє помилки для недопустимих вхідних даних.

Код:



Результат:

Git:  
  
https://github.com/Fols133/ModulWork