Контрольні запитання

1. Що таке модульне тестування?

2. Як використовуються модульні тести?

3. Назвіть вимоги до юніт-тестів.

4. Наведіть переваги використання юніт-тестів у розробці ПЗ.

5. Що таке рефакторінг?

6. Які метрики та засоби використовуються для оцінювання ефективності

застосування юніт-тестів у проекті?

7. Що таке TDD? Назвіть переваги застосування TDD.

8. Що таке принцип «Triple A»? Поясніть сутність його використання.

9. Як використовуються в юніт-тестах класи Assert?

10. Що таке Mock та Stub? З якою ціллю вони використовуються в юніт-

тестах?

Відповіді:   
1)Модульне тестування - це процес перевірки правильності роботи окремих модулів або компонентів програмного забезпечення. Кожен модуль тестується ізольовано від інших частин системи.

2)Модульні тести використовуються для перевірки правильності роботи конкретного модуля програми під час розробки. Вони дозволяють виявляти та виправляти помилки раніше, полегшуючи підтримку коду.

3)Тести повинні бути автоматизовані.

Тести повинні бути повторюваними та незалежними.

Тести повинні покривати всі можливі сценарії використання модуля.

Тести повинні бути швидкими та ефективними.

4)Раннє виявлення та виправлення помилок.

Забезпечення безпеки внесених змін.

Покращення розуміння коду через його вивчення під час написання тестів.

Забезпечення документації та прикладів використання коду.

5)Рефакторінг - це процес покращення якості коду шляхом його переписування без зміни його функціональності. Це робить код більш зрозумілим, зручним для підтримки та зменшує ймовірність помилок.

6)Покриття коду тестами (Code Coverage): Визначає відсоток коду, який був використаний під час виконання тестів.

Час виконання тестів: Швидке виконання тестів є важливим для зручності розробників та інтеграції тестів у CI/CD.

7)TDD (Test-Driven Development) - це методологія розробки програмного забезпечення, в якій тестування визначає специфікацію та контролює процес розробки. Розробка починається з написання тестів, які потім визначають функціональність.

8)Принцип "Triple A":

"Triple A" (Arrange, Act, Assert) - це підхід до написання тестів, де:

Arrange: Підготовка до тестування.

Act: Виклик методу чи функції, яку ви тестуєте.

Assert: Перевірка, чи отриманий результат відповідає очікуваному.

9) Класи Assert надають методи для перевірки різних умов у тестах. Наприклад, Assert.AreEqual(expected, actual) перевіряє, чи рівні очікуване та отримане значення.

10)Mock та Stub у юніт-тестах:

Mock: Об'єкт, що імітує реальний об'єкт у системі та дозволяє визначати, як він повинен взаємодіяти з іншими об'єктами.

Stub: Об'єкт, який надає фіксовану реалізацію для методів чи функцій та повертає заздалегідь визначені значення. Використовується для ізоляції тестованого коду від залежностей.