1.

| | Статистична техніка тестування | Динамічна техніка тестування |
|---|---|---|
| Основна інформація | Статичні техніки тестування передбачають перевірку коду та інших артефактів програми без її виконання | Динамічні техніки тестування передбачають виконання програмного забезпечення з метою виявлення помилок. Ці техніки включають в себе автоматизовані та ручні тести, валідацію та верифікацію. |
| Перевага №1 | Ефективний при пошуку деяких типів помилок (наприклад, синтаксичні помилки, помилки в форматуванні, інші помилки, пов'язані зі стилем програмування тощо) | Можна автоматизувати, що дозволяє ефективно виконувати тести та виявляти помилки швидше та з меншими зусиллями. |
| Перевага №2 | Забезпечує можливість знайти помилки на ранніх етапах розробки | Динамічні техніки тестування можуть бути легкі в використанні, особливо для тестування інтерфейсу користувача та інших елементів, які можна перевірити шляхом взаємодії з програмою. |
| Перевага №3 (і т.д.) | Ефективний при роботі з великими проектами | Динамічні техніки тестування можуть надати детальну інформацію про помилки |
| Обмеження №1 | Не може знайти деякі типи помилок, які виникають тільки при | Динамічні техніки тестування можуть вимагати великої |

| | | |
|---|---|---|
| | виконанні програми | кількості тестів |
| Обмеження №2 | Потребує багато часу та зусиль для ручної перевірки коду | Висока ціна: Автоматизовані динамічні техніки тестування можуть бути дорогими для розробників програмного забезпечення. |
| Обмеження №3 (і т.д.) | Потребує фахівців, які можуть проводити аналіз коду | Недостатньо широке покриття: Динамічні техніки тестування можуть не знайти деякі помилки |
| Висновок | Статичні техніки тестування виконуються без виконання програми, що тестується. Ці техніки оцінюють якість програмного забезпечення, аналізуючи його код, документацію, вимоги та інші артефакти, що використовуються під час розробки програмного забезпечення. | Динамічні техніки тестування виконуються з виконанням програми, що тестується. Ці техніки тестують функціональність, продуктивність, стабільність та інші аспекти програмного забезпечення шляхом запуску тестів на реальних чи штучних вхідних даних та аналізу результатів. |

2.Некоректно, оскільки твердження є неправильним. Навіть якщо у коді є тільки одна умова IF і немає циклів або перемикачів, це не гарантує, що будь-який тест забезпечить 50% покриття рішень. Є можливість, що код містить складні вирази або оператори, які можуть виконуватись по-різному в залежності від вхідних даних. Тому, для досягнення повного покриття рішень, потрібно виконати тести для всіх можливих шляхів виконання коду.

2.1 – для покриття операторів, 1 – для покриття рішень

3.4

Максимум:

2.https://www.figma.com/file/fFpczof8klJxKUpsy8x9Wv/Untitled?node-id=0%3A1&t=F3BU5TrcQLvNRSXs-1

Тест-кейси №1, 2 та 3 перевіряють всі можливі комбінації відповідей на запитання про породу та бажання отримати контакти грумера. Тест-кейс №4 перевіряє сценарій, коли користувач не має кота. Загальна кількість тест-кейсів - 4.