Перший рівень

Зроби порівняння статичних та динамічних технік тестування. Наведи переваги та можливі обмеження при використанні кожної з них.

|  | Статистична техніка тестування | Динамічна техніка тестування |
| --- | --- | --- |
| Основна інформація | Не вимагає запускати програму чи додаток, дає можливість знайти найбільш очевидні помилки на ранніх етапах, включає рев’ю та статичний аналіз. | Передбачає запуск програмного коду, виконується шляхом фактичного використання програми. |
| Перевага №1 | Знижує вартість виправлень, бо знаходить дефекти на ранніх етапах | Виявляє помилки, які могли бути пропущені |
| Перевага №2 | Попередження дефектів дизайну та кодування | Може бути автоматизоване |
| Обмеження №1 | Не виявляє дефектів, які є в середовищі виконання | Дефекти виявляються після написання коду |
| Обмеження №2 | Займає багато часу | Висока вартість виправлень |
| Висновок | Проводиться на ранніх етапах, спрямоване на попередження дефектів, тому виправлення є менш кошторисними, не потребує коду, забезпечує більш широке покриття | Проводиться на пізніх етапах, забезпечує високу якість функціонування продукту, але вартість виправлень є високою, спрямоване на пошук та усунення дефектів, потребує написаного та запущеного коду |

Другий рівень

1. Наступне твердження стосується покриття рішень:

*Коли код має одну ‘IF” умову, не має циклів (LOOP) або перемикачів (CASE), будь-який тест, який ми виконаємо, дасть результат 50% покриття рішень (decision coverage).*

Яке твердження є коректним?

a. Коректно. Будь-який тест кейс надає 100% покриття тверджень, таким чином покриває 50% рішень.

b. Коректно. Результат будь-якого тесту умови IF буде або правдими, або ні.

c. Некоректно. Один тест може гарантувати 25% перевірки рішень в цьому випадку.

d. Некоректно, бо занадто загальне твердження. Ми не можемо знати, чи є воно коректним, бо це залежить від тестованого ПЗ.

2. Є псевдокод: Switch PC on -> Start MS Word -> IF MS Word starts THEN -> Write a poem -> Close MS Word.

Скільки тест кейсів знадобиться, щоб перевірити його функціонал?

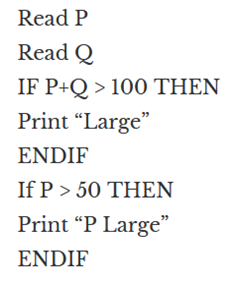
a. 1 – для покриття операторів, 2 – для покриття рішень

b. 1 – для покриття операторів, 1 – для покриття рішень

c. 2 – для покриття операторів, 2 – для покриття рішень

d. 2 – для покриття операторів, 1 – для покриття рішень

3. Скільки потрібно тестів для перевірки тверджень коду:



a. 2

b. 1

c. 3

d. 4

Третій рівень

Продовжуємо розвивати стартап для застосунку, який дозволяє . обмінюватися фотографіями котиків.

Є алгоритм:

*Запитай, якого улюбленця має користувач.*

*Якщо користувач відповість, що має кота, то запитай, яка порода його улюбленця: «короткошерста чи довгошерста?»*

*Якщо клієнт відповість «довгошерста», то запитай: «ви бажаєте отримати контакти найближчого грумера?»*

*Якщо клієнт відповість «так», то скажи: «Надайте адресу найближчої котячої перукарні»*

*Інакше*

*Скажи: «Запропонуй магазин з товарами по догляду за шерстю»*

*Закінчити*

*Інакше*

*Скажи «Запропонуй обрати магазин із зоотоварами»*

*Закінчити*

*Якщо клієнт не має кота*

*Скажи “Коли вирішите завести улюбленця – приходьте”*

*Закінчити*

Завдання:

1. Намалюй схему алгоритму (в інструменті на вибір, наприклад, у вбудованому Google Docs редакторі, [figjam](https://www.figma.com/figjam/) чи [miro](https://miro.com/))

2. Який потрібен мінімальний набір тест-кейсів, щоб переконатися, що всі запитання були поставлені, всі комбінації були пройдені та всі відповіді були отримані?

**Мінімальний набір тест-кейсів – 4. (схема нижче)**

