**Огляд технік тест-дизайну. White Box техніки**

1. 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 | Тестується без запуску коду. З її допомогою можна виявити дефекти в документації. | Тестується із запуском коду. З її допомогою можна перевіряти програму на функціональність. |
| Перевага №1 | Зниження вартості виправлення дефекту в майбутньому, так як виправлення відбувається на етапі аналізу документації. | Виявляє складні помилки, які не можуть бути помічені під час статичного тестування. |
| Перевага №2 | Скорочення ресурсів на виправлення дефектів, сприяє продуктивності розробки. | Може бути автоматизованим, що дозволяє скоротити час на тестування. |
| Перевага №3 | Покращує обмін критичною та важливою інформацією між учасниками команди. | Тестування з точки зору користувача, через що покращується якість системи. |
| Обмеження №1 | Займає багато часу через ручну роботу всіх операцій. | Через складність займає багато часу. |
| Обмеження №2 | Перешкоджає виявленню вразливостей, представлених в середовищі виконання. | Висока вартість виправлення дефектів. |
| Обмеження №3 |  | Виконується тільки після закінчення написання коду. |
| Висновок | Орієнтоване на пошук дефектів в системі та їх виправлення ще до розгортання продукту. | Орієнтоване на виправлення дефектів в системі, знайдених під час функціонального та нефункціонального тестування. |

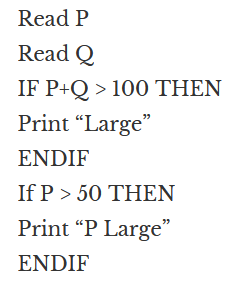
1. Наступне твердження стосується покриття рішень:

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?

1. Коректно. Будь-який тест кейс надає 100% покриття тверджень, таким чином покриває 50% рішень.
2. **Коректно. Результат будь-якого тесту умови IF буде або правдими, або ні.**
3. Некоректно. Один тест може гарантувати 25% перевірки рішень в цьому випадку.
4. Некоректно, бо занадто загальне твердження. Ми не можемо знати, чи є воно коректним, бо це залежить від тестованого ПЗ.
5. 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?

1. **1 – для покриття операторів, 2 – для покриття рішень**
2. 1 – для покриття операторів, 1 – для покриття рішень
3. 2 – для покриття операторів, 2 – для покриття рішень
4. 2 – для покриття операторів, 1 – для покриття рішень
5. Скільки потрібно тестів для перевірки тверджень коду:
   1. 2
   2. **1**
   3. 3
   4. 4
6. 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*

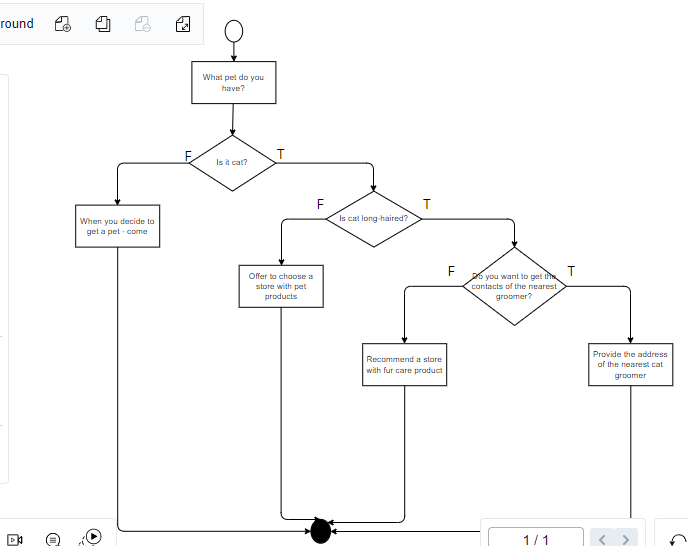
*Якщо клієнт не має кота*

*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?*
3. Алгоритм зображений нижче:



1. Мінімальний набір тестів – 4.
2. Створити алгоритм для завдання нижче:



