| 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 | Тестування ПЗ без запуску коду, для пошуку потенційних дефектів в ПЗ | Тестування поведінки програми, коли виконується код, тобто під час роботи. | | Перевага №1 | Займає менше часу | Високий рівень перевірки системи | | Перевага №2 | Проводиться на ранніх етапах | Можна автоматизувати | | Перевага №3 | Нижча вартість виправлення дефектів | Фіксація багів, які могли бути не поміченими на етап перевірки коду | | Обмеження №1 | Не перевіряє поведінку системи | Займає багато часу | | Обмеження №2 | Заважає знаходити вразливості у середі виконання | Більш дорогий процес перевірки | | Обмеження №3 | Більшість тестів неможливо автоматизувати | Дефекти знаходяться вже після розробки системи | | Висновок | Верифікація ПЗ  Надає можливість передбачити появу дефектів до розробки ПЗ | Валідація ПЗ  Надає можливість перевірити функціональність продукту на якість та ефективність. | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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.  1. 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  1. Скільки тестів для перевірки тверджень коду:      * 1. 2   2. 1   3. 3   4. 4 |
| 1. 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   [**https://miro.com/app/board/uXjVPDg6aU4=/?share\_link\_id=194833868203**](https://miro.com/app/board/uXjVPDg6aU4=/?share_link_id=194833868203)   1. Який потрібен мінімальний набір тест-кейсів, щоб переконатися, що всі запитання були поставлені, всі комбінації були пройдені та всі відповіді були отримані? **- 4 рази** |