| Must have рівень:   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 | Технінка тестування при якій не порібен запуск коду. По суті техніка праює для того,щоб перебдачити дефекти у прогрманому забезпеченні на стадіі розробки. | Техніка тесування яка потребує запуску коду і вона перевіряє функціональність програми,тобто техніка застосовується для того,щоб перевірити продукт на те чи функіонує він так як від нього очікується | | Перевага №1 | Не потрібен готовий продукт для виконання тестів | Програма тестується з точки зору користувача, тому тестування дозволяє покращити функціонал | | Перевага №2 | Допомагає швидше виявити дефетк і тим самим здешевити його усунення та затрати по часі на його усунення | Знаходить дефекти,які впливають безпосередньо на функціонал програми | | Перевага №3 (і т.д.) | Допомагає запобігти появленню дефектів | Виявляє складні помилки у пз, які під час статичного тесування важче виявити | | Обмеження №1 | Складно виявити дефекти які зявляються безпосередньо у середовищі вкионання | Потрібен готовий продукт,щоб виконати тестування | | Обмеження №2 | Займає багато часу | Тестування лише знаходить дефетки,проте ніяк не запобігає їх появленню | | Обмеження №3 (і т.д.) | Не надає можливості покращити функцується безпосередньо перед запуском готвого продукту | Вартість виявлення та усунення дефектів значно дорожча ніж при статичному | | Висновок | Допомагає виявити та запобігти дефетам на ранній стадії, Краще пропраювати саме прогрмане забезпечення. Але має малий вплив на функціонал продукту після його запуску, | Не дозволяє запобігти появі дефектів при розробці, проте довзоляє набагато якісніше пропраюцвати функціонал продукту. | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Середній рівень:   1. Виконай завдання попереднього рівня. 2. 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 потрібно тестів для перевірки тверджень коду:     1. 2    2. 1    3. 3    4. 4 |
| Програма максимум:   1. Виконай завдання двох попередніх рівнів. 2. 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, [fig jam](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? |

Потрібна кількість тест-кейсів - 4

