| Must have рівень:   1. Зроби порівняння статичних та динамічних технік тестування. Наведи переваги та можливі обмеження при використанні кожної з них.  |  | Статична техніка тестування | Динамічна техніка тестування | | --- | --- | --- | | Основна інформація | Виконується з метою запобігання появі можливих багів та є етапом верифікації ПЗ | Виконується з метою пошуку і усуненню багів та є етапом валідації ПЗ | | Перевага №1 | Тестування вимог щодо недоліків неясностей, суперечностей | Перевірка функціональності в середовищі використання | | Перевага №2 | Можливість тестування коду на помилки на ранньому етапі (до динамічного тестування) | Перевіряються критичні межі ПЗ | | Перевага №3 (і т.д.) | Підвищує ймовірність, що після використання цієї техніки тестування кількість багів буде меншою | Тестуються функціональні та нефункціональні вимоги | | Обмеження №1 | Займає багато часу | Займає багато часу | | Обмеження №2 | Тестування відбувається без запуску коду - тому не можна перевірити функціональні та нефункціональні вимоги | Для тестування необхідно завершений код (або якийсь окремий модуль) | | Обмеження №3 (і т.д.) | Виконується здебільшого вручну |  | | Висновок | Часомісткий метод, проте допомагає виявляти баги на ранніх етапах, що зменшує вартість виправлення помилок | Метод виконується на пізніх етапах розробки ПЗ та дозволяє забезпечити контроль якості 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, [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?   <https://webwhiteboard.com/board/EZN580iO1p5sd38SnX6SpDhfSG2efz9b/>  Щоб переконатися що:   * всі відповіді були отримані знадобиться 4 тест-кейси * всі запитання були поставлені знадобиться 1 тест кейс * всі комбінації були пройдені 4 тест-кейси   (1. False; 2. True False; 3. True True False; 4. True True True) |



**Щоб покрити всі оператори знадобиться один тест-кейс (червоний).**