**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** | тип тестування, який не вимагає запускати програму чи додаток, дає змогу знайти найбільш очевидні помилки ще на ранніх етапах створення продукту та включає рев’ю (перевірку робочих продуктів, як-от документація, користувацькі історії, діаграми, інструкції та інші документи, що використовуються під час розробки ПЗ) і статичний аналіз (автоматизоване тестування коду та документації для пошуку очевидних помилок та хиб); передбачає верифікацію. | тип тестування, який перевіряє функціональність програми, коли код виконується; динамічне тестування виконується шляхом фактичного використання програми і визначення того, чи працює функціональність так, як очікується; передбачає верифікацію і валідацію |
| Перевага №1 | Виконується на ранніх етапах розробки, а тому знижує вартість виправлення дефектів. | Розглядає всю функціональність, тому є ретельним. |
| Перевага №2 | Результати тестування допомагають покращити функціонал та уникнути подібних дефектів. | Добре налагоджений, додаток тестується з точки зору користувача, що підвищує якість ПЗ. |
| Перевага №3 | Підвищує поінформованість про дефекти та покращує обмін інформацією між членами команди. | Виявлення складних помилок, які могли бути не помічені на етапі статичного тестування. |
| Перевага №4 | Ідентифікуються дефекти, які складно виявити при динамічному тестуванні. | Може бути автоматизовано. |
| Обмеження №1 | Процес статичного тестування може займати багато часу, так як в основному він виконується вручну. | Складний процес, а тому займає багато часу. |
| Обмеження №2 | Не дозволяє виявити дефекти, які можуть бути виявлені в середовищі виконання. | Висока вартість проведення тестування. |
| Обмеження №3 | Не дозволяє виявити дефекти в самій логіці ПЗ. | Зазвичай виконується після завершення кодування, і знайдені баги виявляються не на ранніх етапах, а пізніше після розробки. |
| **Висновок** | Пошук та усунення дефектів вважається більш ефективним до проведення динамічного тестування, оскільки є перевіркою на ранніх етапах. В результаті такої перевірки можна попередити дефекти дизайну або кодування, а також заощадити кошти на виправленні дефектів. | Дозволяє провести тестування в середовищі, з точки зору кінцевого користувача. Є складним, але детальним тестуванням, яке дозволяє знайти складні дефекти. |

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).*

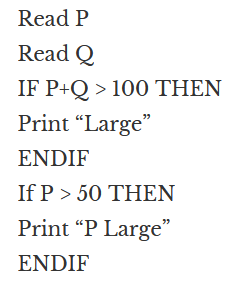
Яке твердження є коректним?

* 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? **6 тест-кейсів**

