Must have рівень:

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 | є методикою тестування Пз *без запуску коду*. Це також процес виявлення і усунення помилок і дефектів ***в супровідних документах***, таких як специфікації вимог до програмного забезпечення і т. ін. | тип тестування, який перевіряє функціональність програми, *коли код виконується*. Включає в себе тестування ПЗ в режимі реального часу шляхом надання вхідних даних і вивчення результату поведінки програми |
| Перевага №1 | Виконується на ранніх етапах розробки програмного забезпечення | Це ретельне дослідження, яке розглядає всю функціональність програми |
| Перевага №2 | Забезпечує більш ширше охоплення,аніж динамічне, за коротший проміжок часу. | додаток тестується з точки зору користувача, що підвищує якість ПЗ |
| Перевага №3 | Комплексне тестування коду, яке допомагає знайти більше дефектів у системі. | Виявлення складних помилок, які могли вислизнути на етапі код рев'ю |
| Обмеження №1 | Перешкоджає виявленню вразливостей, представлених в середовищі виконання | займає багато часу = багато грошей |
| Обмеження №2 | Не кожний тестувальник може знати нюанси коду для його перегляду | Висока вартість проведення тестування |
| Обмеження №3 | Не перевіряє роботу Пз | зазвичай виконується після завершення кодування, і знайдені баги виявляються пізніше в життєвому циклі розробки. |
| Висновок | статичне тестування вимагає багато часу на бурхливі дискусії та зустрічі, все ж варто витратити час на запобігання появи дефектів на останніх етапах розробки продукту. Вважається важливим кроком *на шляху до розробки ПЗ без помилок.* | Завдяки безпосередньому виконанню тестів програмного забезпечення команда може *перевірити і підтвердити якість і ефективність ПЗ* |

Середній рівень:

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?

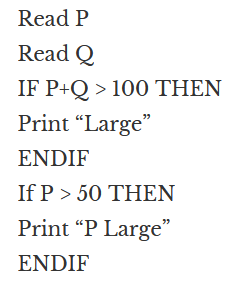
* 1. Коректно. Будь-який тест кейс надає 100% покриття тверджень, таким чином покриває 50% рішень.
  2. Коректно. Результат будь-якого тесту умови IF буде або правдими, або ні.
  3. Некоректно. Один тест може гарантувати 25% перевірки рішень в цьому випадку.
  4. Некоректно, бо занадто загальне твердження.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

2. 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

3. Скільки потрібно тестів для перевірки тверджень коду:

* 1. 2
  2. 1
  3. 3
  4. **4**

Програма максимум:

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*

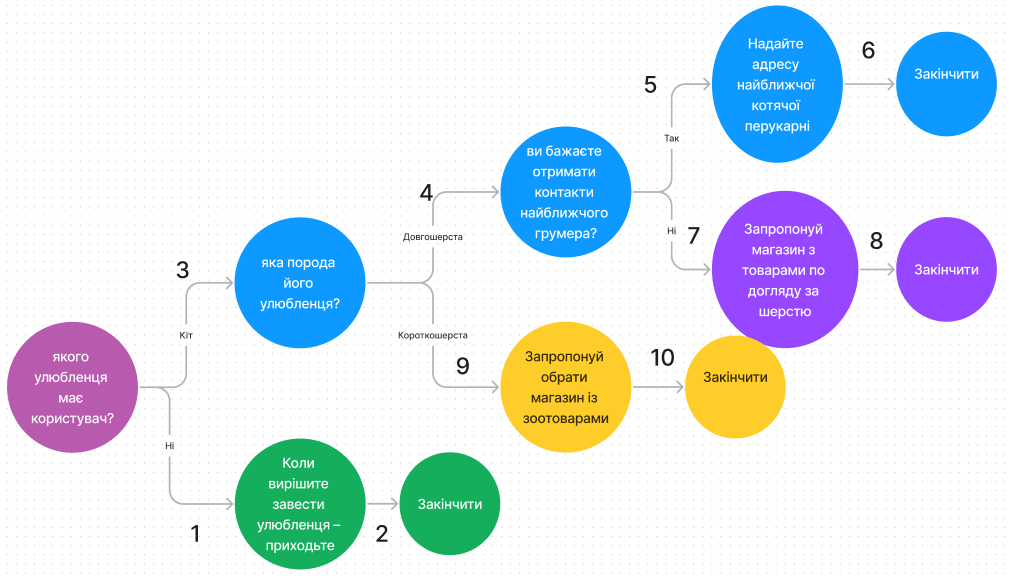
*Скажи “Коли вирішите завести улюбленця – приходьте”*

*Закінчити*

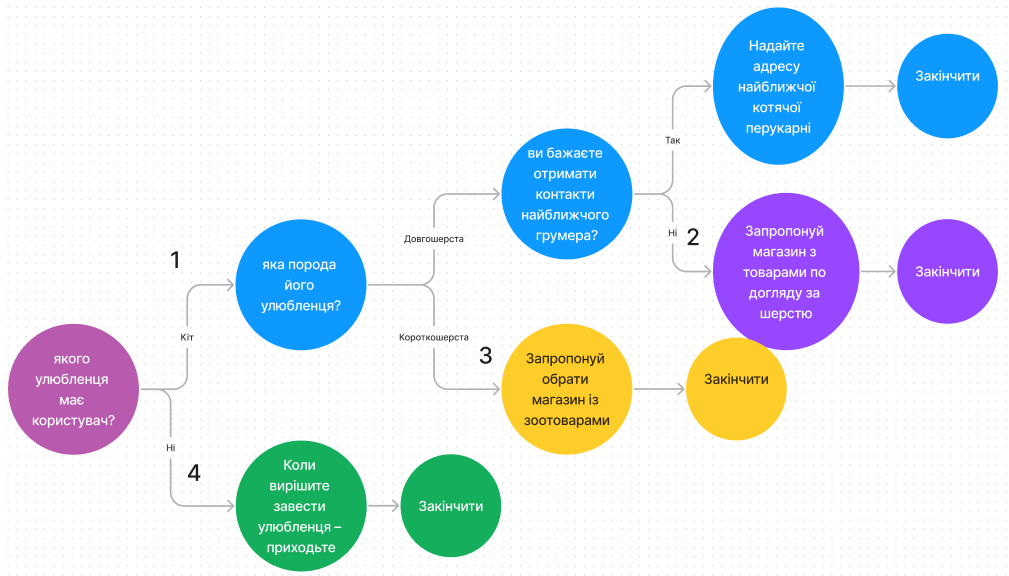
Завдання:

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?

**10 тест-кейсів для покриття операторів - буде достатньо для перевірки всіх комбінацій**



**4 тест-кейси для покриття шляху (можна пройти додатково)**

****

**6 тест-кейсів для покриття варіантів (так/ні) (можна пройти додатково)**

