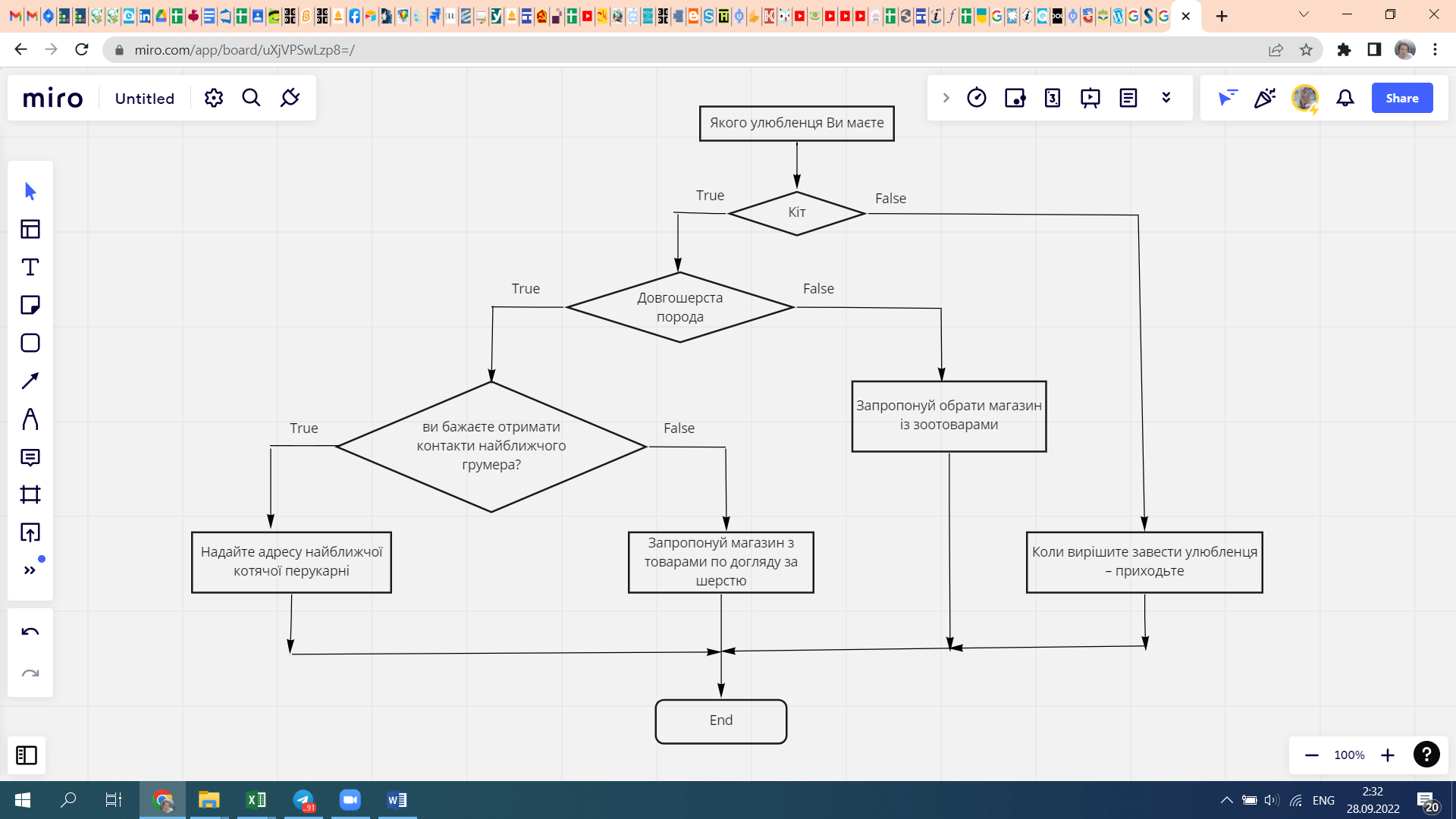
| Must have рівень:   1. Зроби порівняння статичних та динамічних технік тестування. Наведи переваги та можливі обмеження при використанні кожної з них.  |  | Статична техніка тестування | Динамічна техніка тестування | | --- | --- | --- | | Основна інформація | є частиною верифікації ПЗ, тестується проектна документація, специфікації, вимоги | є частиною валідації ПЗ, тестується поведінка ПЗ під час його роботи (продуктивність, безпека) | | Перевага №1 | займає менше часу, ніж динамічна | ретельно перевіряються всі функції системи включно з кодом | | Перевага №2 | є більш економічною, ніж динамічна | таблиці рішень, зрозумілі для всієї проектної команди | | Перевага №3 (і т.д.) | тестування можливе без запуску коду | помилка є констатацією факту її виникнення | | Обмеження №1 | для тестування використовується спеціальне ПЗ | потребує багато часу | | Обмеження №2 | для аналізу використовується розрахований вручну вихідний код | потребує знань програмування | | Обмеження №3 (і т.д.) | помилка є передбаченням після аналізу моделі | неможливе без запуску коду | | Висновок | використовується для виявлення багів та проблем з ПЗ на ранніх етапах розробки. | використовується на фінальних стадіях розробки, ПЗ тестується з позицій користувача на відповідність вимогам. | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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? |



**Потрібно 4 тест-кейси**