**Перший рівень** — відпрацюй навички на базовому рівні.

1. Склади порівняльну таблицю функціонального, нефункціонального і пов’язаного зі змінами видів тестування.

Порівняння має містити такі блоки:

* що перевіряється;
* коли застосовується;
* обмеження;
* особливості

| Види тестування | Що перевіряє | Коли застосовується | Обмеження | Особливості |
| --- | --- | --- | --- | --- |
| Функціональне | чи дає обробка правильні результати чи має якісь помилки. | ви шукаєте будь-які прогалини, помилки або будь-що, що не відповідає вимогам до програмного забезпечення чи програми. | можливість упущення логічних помилок у програмному забезпеченні;  ймовірність надмірного тестування. | Тестування функціональності може проводиться у двох аспектах:   * вимоги * бізнес-процеси |
| Нефункціональне | вимірює пз, наскільки добре воно працює за різних обставин. | В цілому, це тестування того, “Як” система працює. | цей вид тестування передбачає перевірку властивостей, які напряму не відносяться до функціональності системи, але складають її додаткову вигоду. | не перевіряє систему на виконання тих функцій, які вимагає від неї замовник, воно дозволяє контролювати більш глобальні властивості: безпеку, продуктивність, надійність та інше. |
| Пов'язане зі змінами | виявлення та усунення багів, будь-яких змін в технічному середовищі чи банально – в замовленні користувача | Після проведення необхідних змін, таких як виправлення бага / дефекту |  |  |

2. Поясни, в чому різниця між регресією та ретестингом (5 речень).

Регресія та ретестинг вони обидва пов’язані зі змінами, але це – абсолютно різні процеси. Регресійне тестування перевіряє наявність нових багів після змін у коді, а повторне тестування – чи нещодавно виявлені дефекти точно виправили.  
Ретест — це перевірка конкретних тестових сценаріїв, у яких нещодавно виявили баги. Завдання ретесту – підтвердити, що після змін у коді новий баг і справді більше не з’являється.

**Другий рівень** — детальніше заглибся в практику.

1. Виконай завдання попереднього рівня.

2. Як ти вважаєш, чи можливе для продукту проведення тільки функціонального тестування, без перевірки нефункціональних вимог?

* Якщо так – в яких випадках?
* Якщо ні – чому?
* Обґрунтуй свою відповідь.  
    
  Я вважаю,що можливо провести тільки функціональне тестування продукту. Тому що функціональне та нефункціональне тестування перевіряє різні аспекти програмного забезпечення. Функціональне тестування під час тестування програмного забезпечення пов’язане з тим, чи кожна функція програмного забезпечення чи програми відповідає необхідним специфікаціям, а нефункціональне тестування вимірює, наскільки добре працює програмне забезпечення чи програми, а   
  не те, чи вони взагалі працюють.

3. Як ти розумієш необхідність проведення smoke (димового) тестування? Чи завжди воно є доречним?

Вважаю дуже доречним використанням smoke тестування. Димові тести виконують щоб забезпечити стабільність і перевірити критичні функції. Випробування на дим перевіряє стабільність усієї системи, у [тестуванні програмного забезпечення](https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F) означає мінімальний набір тестів на явні [помилки](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D2%91). Цей тест зазвичай виконується самим програмістом. Програму, що не пройшла такий тест, немає сенсу передавати на глибше тестування.

**Третій рівень** — різнобічно опануй тематику уроку.

1. Виконай завдання двох попередніх рівнів.

2. Ти – засновник/ця стартапу і плануєш випустити на ринок мобільний застосунок для обміну світлинами котиків для iOS та Android пристроїв.

Користувачі можуть завантажувати фотографії котиків. Але не можуть завантажувати фотографії інших тварин/людей/об’єктів. Користувачі можуть додавати друзів, ставити “вподобайки”, залишати коментарі.

Завдння: Напиши 5 функціональних тест-кейсів, які перевіряли б роботу застосунку.

3. Напиши, які нефункціональні вимоги ти хотів/ла б застосувати для продукту твого стартапу.

Опиши перевірки, які б їх перевіряли (3-5 прикладів).