**1.** Представьте, что вы работаете над разработкой простого приложения для записной книжки, которое позволяет пользователям добавлять, редактировать и удалять контакты. Ваша задача - придумать как можно больше различных тестов (юнит-тесты, интеграционные тесты, сквозные тесты) для этого приложения. Напишите название каждого теста, его тип и краткое описание того, что этот тест проверяет.

**-Юнит-тесты.**

Проверка класса Контакты на создание объекта с правильными данными.

Проверка класса Репозиторий на добавление в него объекта класса Контакты. Вместо объекта класса Контакты можно использовать mock.

Проверка класса Репозиторий на удаление из него объекта класса Контакты. Вместо объекта класса Контакты можно использовать mock.

Проверка класса Репозиторий на редактирование в нем объекта класса Контакты. Вместо объекта класса Контакты можно использовать mock.

Проверка класса Сервис на вызов нужного метода в объекте класса Репозиторий при добавлении объекта класса Контакты. Вместо объекта класса Репозиторий можно использовать mock.

Проверка класса Сервис на вызов нужного метода в объекте класса Репозиторий при удалении объекта класса Контакты. Вместо объекта класса Репозиторий можно использовать mock.

Проверка класса Сервис на вызов нужного метода в объекте класса Репозиторий при редактировании объекта класса Контакты. Вместо объекта класса Репозиторий можно использовать mock.

**-Интеграционные тесты**

Проверка взаимодействия классов Контакты и Репозиторий. Можно использовать fake.

Проверка взаимодействия классов Репозиторий и Сервис.

Проверка взаимодействия всех трех репозиториев вместе. Можно использовать fake.

**-Сквозные тесты**

Проверить через интерфейс пользователя добавление Контакта в систему.

Проверить через интерфейс пользователя удаления Контакта из системы.

Проверить через интерфейс пользователя редактирования Контакта в системе.

**2.** Ниже список тестовых сценариев. Ваша задача - определить тип каждого теста (юнит-тест, интеграционный тест, сквозной тест) и объяснить, почему вы так решили.

Проверка того, что функция addContact корректно добавляет новый контакт в список контактов.

Проверка того, что при добавлении контакта через пользовательский интерфейс, контакт корректно отображается в списке контактов.

Проверка полного цикла работы с контактом: создание контакта, его редактирование и последующее удаление.

1 - Юнит-тест, так как данный функционал логично организовать в рамках одного модуля

2 – Сквозной тест, так как задействован пользовательский интерфейс и отображение в нем же результата добавления контакта

3 - Интеграционный тест, цикл CRUD для всей системы без использования пользовательского интерфейса, но с взаимодействием всех модулей можно проверить с помощью интеграционных тестов.