

***Задание 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.

Решение

Юнит-тесты	
Название теста	Описание
testAddContact()	Проверяет, что метод для добавления нового контакта работает корректно.
testEditContact()	Проверяет, что метод для редактирования существующего контакта корректно изменяет данные контакта.
testDeleteContact()	Проверяет, что метод для удаления контакта удаляет контакт из записной книжки.
testGetAllContacts()	Проверяет, что метод для получения всех контактов возвращает список всех контактов из записной книжки.
Интеграционные тесты:	
testAddContactToEmptyBook()	Проверяет, что контакт успешно добавляется в пустую записную книжку.
testEditContactDetails()	Проверяет, что контакт можно успешно отредактировать с обновлением деталей.
testDeleteContactFromList()	Проверяет, что контакт можно успешно удалить из списка контактов.
Сквозные тесты	
testAddContactThenRetrieve()	Проверяет, что после добавления контакта он может быть успешно получен
testAddMultipleContactsThenSearch()	Проверяет, что после добавления нескольких контактов, поиск контактов по имени или другим критериям возвращает правильные результаты.
testAddContactWithInvalidData()	Проверяет, что при попытке добавить контакт с недопустимыми данными, приложение обрабатывает ошибку корректно.
testDeleteContactThenConfirm()	Проверяет, что после удаления контакта, пользователь должен подтвердить действие, чтобы избежать случайного удаления
testDataPersistence()	Проверяет, что данные в записной книжке сохраняются после перезапуска приложения.

***Задание 2. ***

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

1. Проверка того, что функция `addContact` корректно добавляет новый контакт в список контактов.
2. Проверка того, что при добавлении контакта через пользовательский интерфейс, контакт корректно отображается в списке контактов.
3. Проверка полного цикла работы с контактом: создание контакта, его редактирование и последующее удаление.

Решение

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

Тип теста: Юнит-тест.

Этот тест проверяет функциональность конкретного метода `addContact`. Он изолирован от других компонентов и сосредотачивается на тестировании отдельного модуля.

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

Тип теста: Интеграционный тест.

Этот тест проверяет взаимодействие между компонентами - пользовательским интерфейсом и логикой добавления контакта. Он тестирует, что приложение корректно обрабатывает пользовательский ввод и обновляет список контактов в интерфейсе. Это уже не изолированный юнит-тест, так как он включает в себя взаимодействие нескольких компонентов.

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

Тип теста: Сквозной тест.

Этот тест проверяет весь жизненный цикл контакта, начиная с его создания, затем редактирования и заканчивая удалением. Он моделирует полное взаимодействие пользователя с приложением, включая как пользовательский интерфейс, так и логику бизнес-процессов.