ЛАБОРАТОРНАЯ РАБОТА №8

Модульные тесты (Unit tests).

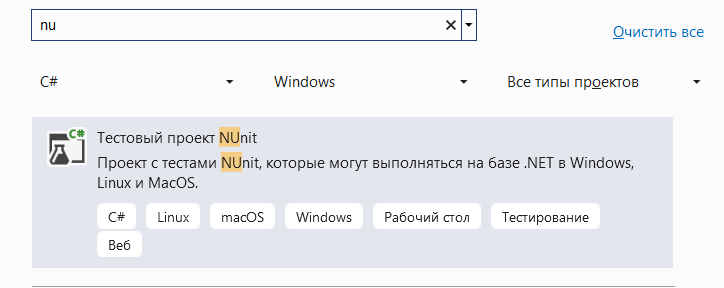
Создание модульных тестов для бизнес-логики приложения с использованием библиотеки NUnit.

Содержание работы:

1. Создать в решении новый проект NUnit.
2. Создать тесты для сервиса бизнес-логики и отчета.
3. Продемонстрировать выполнение тестов.

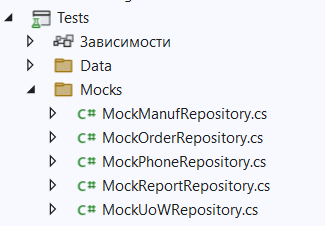
Основные этапы работы.

1. Создадим проект для тестов.



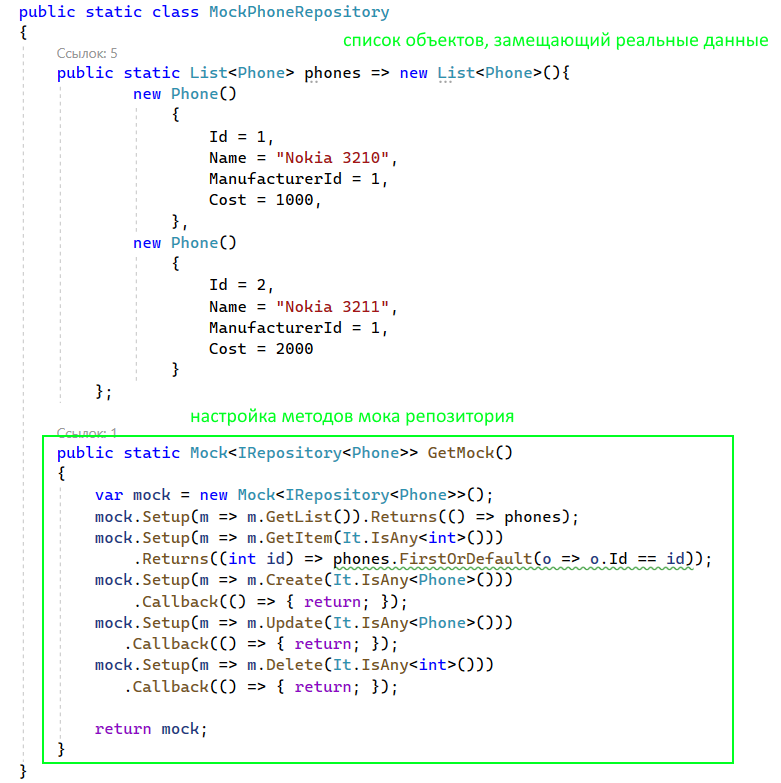
1. Создадим моки для тестирования сервисов.

Вместо реальных компонентов системы используются их имитации, называемые моками (mocks).



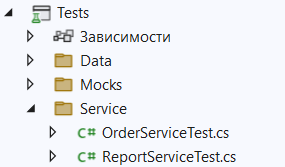
Для сервисов требуются uow и репозитории. Настроим моки репозиториев сущностей, чтобы они могли возвращать подменный список и искать объект из этого списка. Создание, изменение и удаление не тестируем – т.к. никакой логики на стороне приложения в них не реализуется, а сама передача данных - это интеграционное тестирование базы данных.

В моке репозитория отчетов настроим отчет, использующий Linq: он будет возвращать объекты dto , сконвертированные из подменного списка.

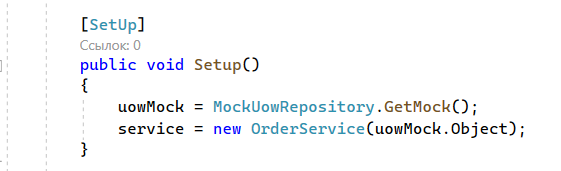
 

1. Создадим тесты сервисов.

Ограничимся тестами основной бизнес-функции и отчета.

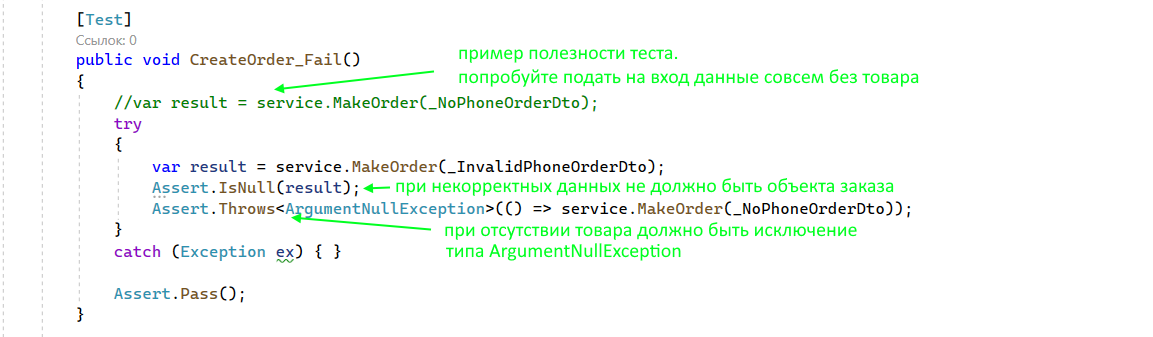


Общие объекты для тестов выносятся в метод Setup:

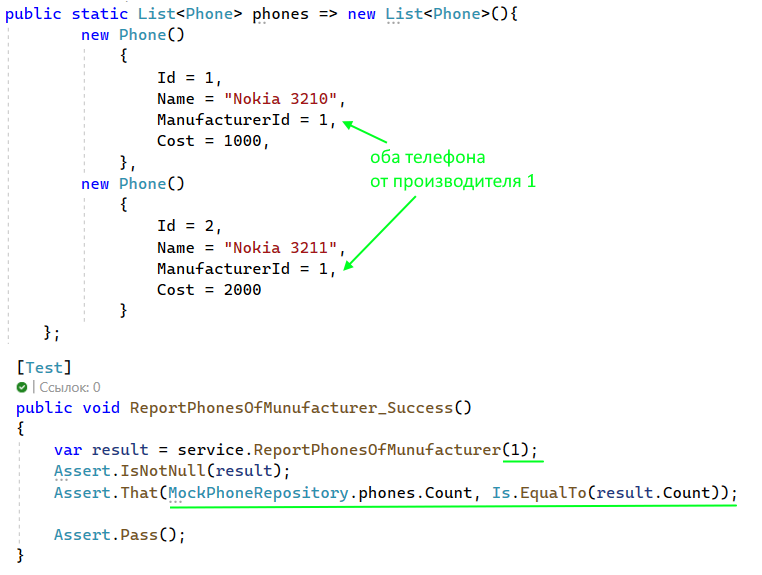


Необходимо написать тесты как для успешного выполнения функции, так и для неуспешного. При подаче невалидных данных в тесте необходимо проверять, что сервис корректно их обработает. Например, подаем в создание заказа товары, которых заведомо нет. В сервисе заказов в этом случае должно выбрасываться исключение. Если подать модель заказа, не содержащую ни одного телефона (закомментированный код), то тест провалится. Таким образом, вы можете найти места вашего кода, потенциально вызывающие сбой приложения.

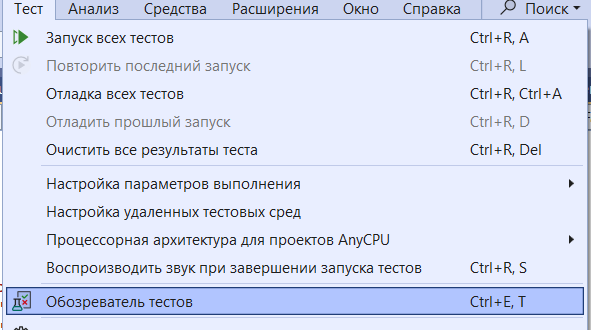
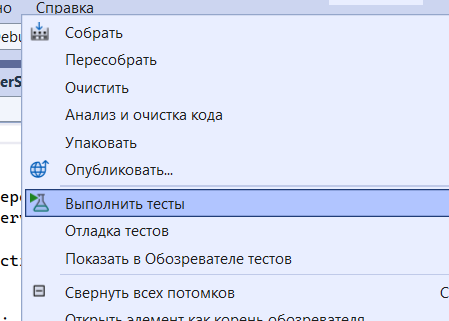
Для успешного создания заказа проверяются данные с одним телефоном и с несколькими.



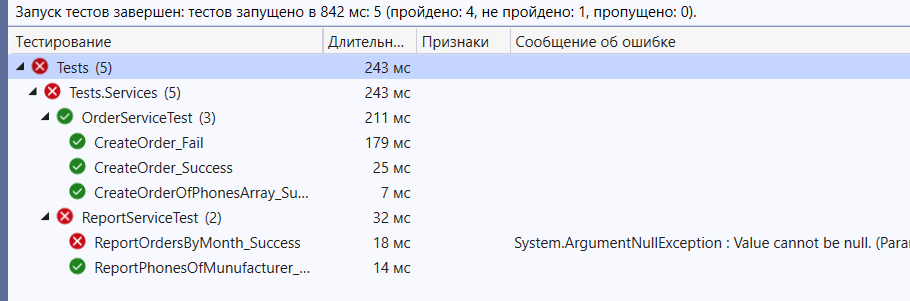
Так как один из отчетов использует ХП БД, мы его не тестируем (эта логика работы заложена на стороне БД, а не приложения). В тесте отчета «список телефонов производителя» проверяем что сервис возвращает количество записей, соответствующее подменным данным.



1. Запуск тестов. Из обозревателя тестов или через контекстное меню проекта с тестами.

Заваленные тесты отображают ошибку и их можно отлаживать.



В итоге все тесты должны проходить успешно:

