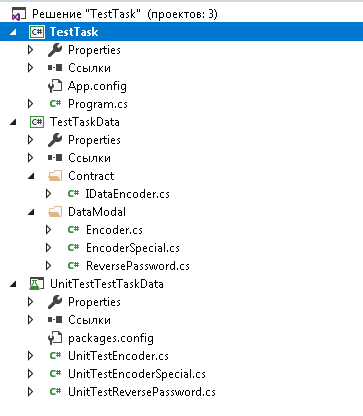
Коллеги, высылаю решённые задания - смотрите приложенный к письму архив:



Решение содержит 3 проекта:

* TestTask - консольное приложение, которое позволяет проверить необходимый функционал с данными, которые пользователь вводит с клавиатуры
* TestTaskData - реализация предложенных заданий
* UnitTestTestTaskData - модульные тесты при помощи которых я убеждался в работе своего кода для различных входных данных

Задание №1

> Оцените сложность данного алгоритма

Сложность представленного мной алгоритма - линейная

Задание №2

> Один из клиентов хочет использовать более надежное кодирование паролей – а именно хэш-функцию SHA-1. Опишите изменения – что необходимо изменить в коде, чтобы для одного клиента использовалась новая функция SHA-1, а для остальных клиентов – осталась существующая функция. Объясните Ваше решение.

Здесь можно предложить несколько вариантов:

#1. Простое решение

Мы заменяем реализацию действующей функции (без изменений в названии класса, namespace, и сигнатуры метода) и отдаём сборку клиенту (после предварительного тестирования), которому требуется новая реализация.

Плюсы: клиенту достаточно обновить только dll/exe с новой реализацией и всё начинает работать

Минусы: в репозитарии разработки возникает 2 ветки - одна со старой реализацией, вторая - с новой. Эти ветки придётся поддерживать параллельно

Если проект предполагает какое-то развитие, этот вариант лучше не рассматривать .

#2. Сложное решение

Необходимо внести изменения в техническую реализацию (рефакторинг) - добавить интерфейс типа IEncoder { string Encode(string value) }; и две его реализации - два класса с разной реализацией Encode (в моём проекте это Encoder и EncoderSpecial). Клиентское приложение должно выбирать (например, на основании данных в своей конфигурации \*.config) - какую из реализаций использовать. Я продемонстрировал эту идею (упрощённо, без концепции IoC и её реализаций, например, с AutoFac) в консольном приложении TestTask

Плюсы: у разработчиков остаётся только одна ветка разработки, а клиенты могут переключать реализацию кодирования на своё усмотрение через конфигурационный файл

Минусы: разработчикам потребуется немного больше усилий, чем в простом решении.