

Задание на каникулы

Общее описание

Наша задача – закрепить некоторые навыки:

- Проектирование и разработку классов
- Получить опыт работы с модульными тестами
- Опробовать работу с системами контроля версий

Кроме этого мы закрепляем ряд теоретических навыков:

- Объектно-ориентированные механизмы в .Net/C# (объекты, интерфейсы, методы, ...)
- Использование механизма Generics или обобщенное программирование (внимание! Этот материал не давался на лекциях, его нужно проработать самостоятельно по методичке или иным источникам)
- Организация структур данных в .Net, в частности:
 - Основные интерфейсы: ICollection, IEnumerable, ...

Задание

Необходимо разработать библиотеку основных структур данных:

- Список
- Очередь
- Стек

Для каждой структуры необходимо:

- Создать 2 реализации: на массиве и на ссылках – но при этом добиться единообразия интерфейсов (т.е. программист, использующий вашу библиотеку должен иметь возможность перейти с одной реализации на другую, не изменяя кода – только указав в одном месте конкретную реализацию).
Принимается любая реализация – с использованием интерфейсов или общих абстрактных классов-предков.
- Реализовать для каждой коллекции подходящие интерфейсы из стандартных интерфейсов .Net. Как минимум, обязательно должны быть реализованы:
 - IEnumerable
 - ICloneable
 - IDisposable
- Обеспечить покрытие тестами всех публичных методов. В составе тестов должны присутствовать:
 - Проверки обычных положительных сценариев (т.е. по сути – сценарии нормальной работы)
 - Проверка граничных значений (например, извлечение последнего элемента, или занесение в последнюю доступную ячейку)
 - Проверка исключительных ситуаций (например, попытка получения элементов из пустой структуры)

- Все разработанные вами структуры должны быть Generic-типами (как уже писалось выше – Generic вам необходимо изучить самостоятельно)

Оформление задания

Ваша работа должна быть разбита на 2 проекта (в одном solution):

- Разрабатываемая вами библиотека
- Проект с тестами

Что касается стиля кодирования: рекомендуется следовать правилам, описанным в методическом пособии в главе **Стиль кодирования**.

Вся разработка и конечный результат должны быть выложены в вашем GitHub репозитории (я рекомендую создать репозиторий для всех заданий, которые мы будем в дальнейшем выполнять и каждое задание размещать в отдельной папке в его корне).

После окончания разработки вы просто вышлите мне ссылку на ваш репозиторий.

Примечание 1: если есть желание, вы можете освоить более сложную технику работы с репозиторием, чем просто коммит в основную ветку (например, создание ветки под каждое задание). Но конечный результат всегда должен быть в основной ветке.

Примечание 2: вы можете использовать любую библиотеку модульного тестирования. Можно – встроенную MS Test, можно – NUnit (с ним чуть сложнее стартовать, т.к. нужно установить плагин для VS, а также подключать саму библиотеку).

Материалы и ссылки

За каникулы вам потребуется не только сделать практическое задание, но и изучить/повторить ряд теоретических моментов.

Ниже приводится ряд ссылок, которые могут вам помочь:

- Методичка, которая была выложена ранее, там в частности, рекомендуется проработать главы:
 - Часть 11. Стиль кодирования
 - Часть 12. Классы и объекты
 - Часть 13. Иерархия классов
 - Часть 14. Интерфейсы
 - Часть 15. Исключения
 - Часть 16. Структуры данных
 - Часть 17. Коллекции
- В качестве введения в Git (обратите внимание – там обсуждаются консольные утилиты, а не реализация Git интерфейса в Visual Studio, которым мы будем преимущественно пользоваться) можно использовать свободно доступную книгу **Pro Git (версия 2)**.
 - Её русская версия доступна здесь: <http://git-scm.com/book/ru/v2>, там же есть ссылки на offline-варианты книги. Например, [PDF](#) вариант
 - В большей степени нам потребуется глава [Основы Git](#) – остальное, по желанию.
- В качестве примера готовых сторонних библиотек для работы со структурами можно посмотреть:

- [Power Collections](#)
 - [Iesi.Collections](#)
- По модульному тестированию можно посмотреть, например:
 - [Введение в Юнит Тестирование](#)
 - [NUnit: знакомство](#)