**# Задание для самостоятельной разработки**

Необходимо разработать программу, используя подход Test Driven Development:

1) Выделить аспекты функциональности, для каждого из которых решение будет состоять из серии коммитов.

2) Создать гит-репозиторий для задания и коммитить в него изменения по следующей схеме:

2.1) Коммит с тестами, которые фиксируют требования к реализации. Подразумевается, что в этом коммите неформально сформулированы требования или ожидания к реализации.

2.2) Коммиты с кодом, которые, в конечном итоге, реализуют требования и соответствуют тестам.

3) Как только реализована основная функциональность, нужно создать запрос на слияние изменений в основную ветку (Merge Request), в которой преподаватель делает замечания. Далее - продолжать процесс итеративно.

**Литература:**

1. TDD

\* Kent Beck. Test-Driven Development By Example

\* David Astels. Test-Driven Development: A Practical Guide

2. Рефакторинг

\* Мартин Фаулер. Рефакторинг.

**## Continuous Integration (CI)**

Язык тестов и реализации можно выбрать по своему усмотрению.

Необходимо, чтобы была возможность вести TDD разработку и встраивать вспомогательные инструменты в CI.

1. наличие фреймворка для написания и запуска unit тестов;

2. наличие инструментов для сбора покрытия.

В проекте необходимо использовать CI, в который должно быть встроено не меньше одного инструмента по каждому пункту:

\* автоматического unit тестирования и сбора покрытия (может использоваться два инструмента),

\* статического анализа,

\* динамического анализа (например, для профилирования).

В части покрытия необходимо достигнуть 100% покрытия:

1. по строчкам кода,

2. по ветвлениям в коде,

3. еще одного критерия по собственному выбору.

**## Собственно задание**

Книжный магазин.

Покупатель

Книги

- название

- автор

- год выпуска

- цена

- издательство

- жанр

Покупатель:

- Создать корзину

Cart = newCart(id)

- Покупатель может добавлять книги в корзину

Book {}

cart.AddBookToCart(book\_id)

- Может оформить доставку

- адрес

- время

- выбрать способ оплаты при заказе/при получении

- Уточнить статус

- Оформить возврат

- курьером

- самостоятельно

- не доставленного товара

Магазин

- создать

- добавить книгу

- доставить

- вернуть книгу

1. **Основные классы**

Сначала очевидные: Book (отдельная книга), Client (авторизованный покупатель), Shop (магазин).

Четвертый, тоже б.-м. очевидный класс - Order (заказ), клиент и магазин обмениваются ими, статус заказа меняется, итд.

Еще один, менее очевидный, класс возникает из-за необходимости учитывать, какие id мы уже использовали – для клиентов, книг и заказов. Назовем его Id.