# Code Review of MVCApp

# Ревью собственного проекта.

# Харламов Александр Анатольевич

В начале рассмотрим вкратце, какие из заданий были реализованы, а затем остановимся на каждом из них поподробнее.

В ходе выполнения задания были реализованы следующие шаблоны:

1. Контроллер.
2. Метка синхронизации.
3. XSLT-преобразователь.
4. Класс-хелпер.
5. Фильтры.
6. Unit of Work.

Разберем подробнее каждый из этих пунктов.

***Контроллер:***

Отвечает основным пунктам, за которые отвечает контроллер:

* реагирует на события пользователя
* и отображает необходимый вид.

В принципе, механизм реакции на действия пользователя реализован правильно: на каждое различное событие пользователя есть отдельный метод, который обрабатывает это событие.

Механизм же отображения видов в результате реакции на событие может быть улучшен: один метод, который принимает как входную строку название вида, для отображения. Тем самым мы избегаем ситуации, когда плодятся методы с одинаковой функциональностью.

***Метка синхронизации:***

В контроллере реализована метка синхронизации, в ней хранится количество просмотренных страниц, если при каком-то событии на соответствующей странице оказалось число большее, чем определено, то выводится сообщение об ошибке.

Главный недостаток моей реализации метки синхронизации в том, что в данный момент на сайте может находится лишь один пользователь, так как метка сделана как статическая переменная. Это связано с тем, что было замечено, что количество экземпляров класса контроллера более чем один даже для одного пользователя, в виду этого, метка не всегда была корректна для определенных страниц, т.е. мы ее изменили, но при следующем действии она все равно имело нулевое значение. Поэтому в дальнейшем стоит развить функционал на произвольное количество пользователей.

***XSLT-преобразователь.***

Реализован в соответствии с шаблоном: по заданным файлам с описанием генерируются виды, в данном случае генерируются html-страницы.

***Класс-хелпер.***

Присутствует один класс-помощник, который занимается преобразованием одной модели в другую, а именно xslt в удобное для нас представление, в данном случае в веб-страницу. Фактически, отвечает за вывод корректной информации в вид, а именно сам вид.

***Фильтры.***

Класс FilterManager: реализует шаблон синглтон, в следствие этого в приложении лишь один экземпляр данного класса, это сделано для того, чтобы можно было четко отслеживать, какие цепи создаются, сколько их(данного функционала нету, но подразумевается). Этот класс является своеобразной фабрикой для создания различных цепей.

Класс FilterChain является собственно цепью, которая содержит последовательность действий, которые необходимо выполнить над некоторыми данными.

Интерфейс для одного элемента в цепи и его реализация также присутствуют, заметим что в реализации элемента цепи присутствует метод для определения идентификатора пользователя, этот шаг является ошибочным и требует исправления в следующей версии, так как данный функционал должен быть вынесен в класс, который взаимодействует с базой данных, фактически мы навязали нашему элементу дополнительный функционал, который он не должен выполнять.

***Unit of Work.***

Реализован в соответствии с требованиями, является неким промежуточным слоем между приложением и базой данных.

Недостатки:

* Вычитывает в память всех пользователей из базы данных, что хорошо лишь для малых объемов информации, но неприемлемо, для больших объемов.
* С другой стороны не совсем корректно повторение функционала в двух различных интерфейсах, фактически они перекрывают друг друга, но данный подход был выбран из-за удобности использования, так как мы можем реализовывать различные юниты и использовать лишь указатели на них, поэтому с этой точки зрения как раз по коду интерфейса понятно, какие функции выполняет должен выполнять данный класс.

В остальном, реализация приближена к реальности.

Подводя итог, хочется отметить, что основные функции, которые требовались от системы, были реализованы в полном объеме, хотя и присутствуют некоторые огрехи и неточности в исполнении, которые вызваны либо неполным знанием платформы, либо сознательным выбором в сторону простоты и понимания реализации.