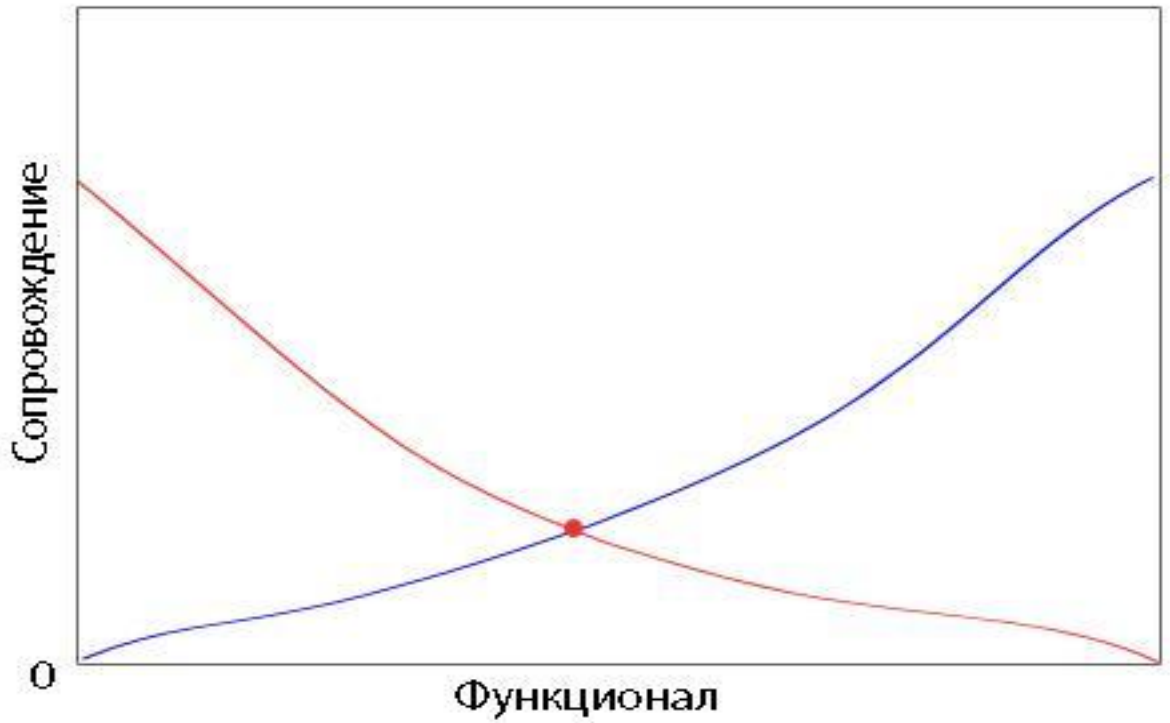
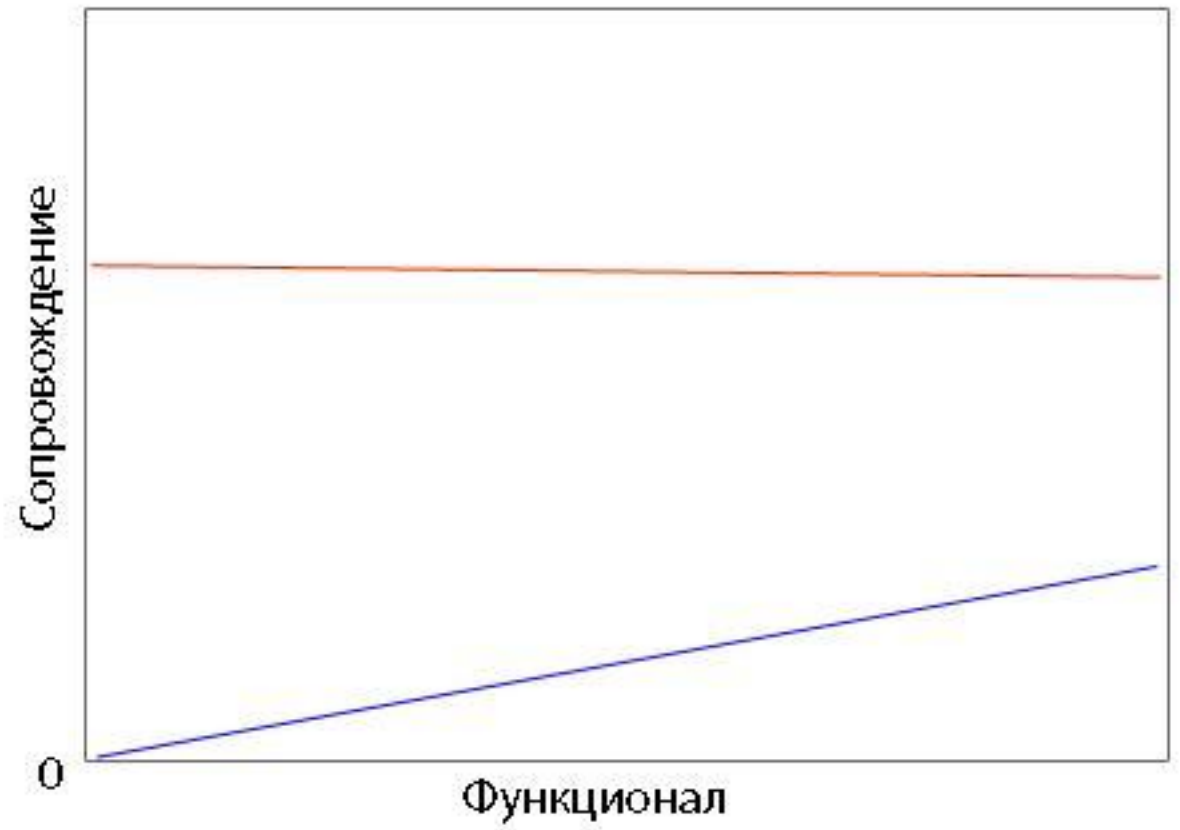
MVC, PSR

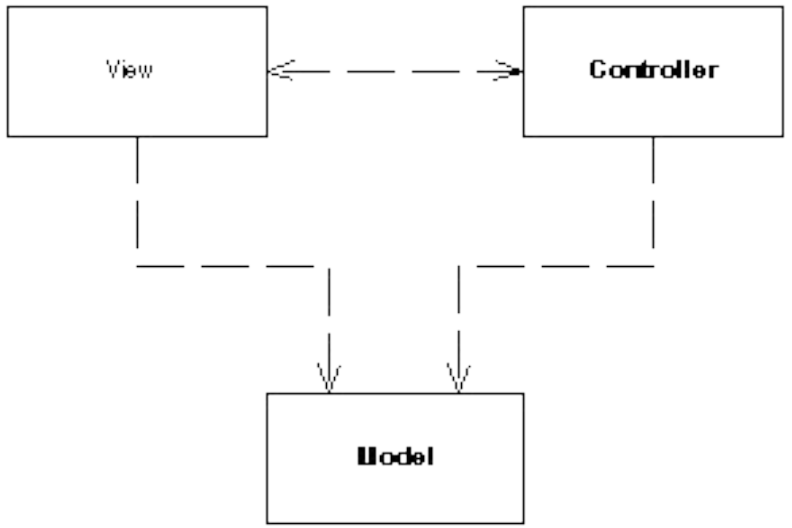
Плохая архитектура - затратна в сопровождении, монолитна, плохо поддается тестированию, хрупка к изменениям и имеет неоправданную сложность.



Хорошая архитектура делает систему легкой в освоении, простой в разработке, сопровождении и развертывании. Конечная ее цель – минимизировать затраты на протяжения всего срока службы системы и максимизировать продуктивность программиста.



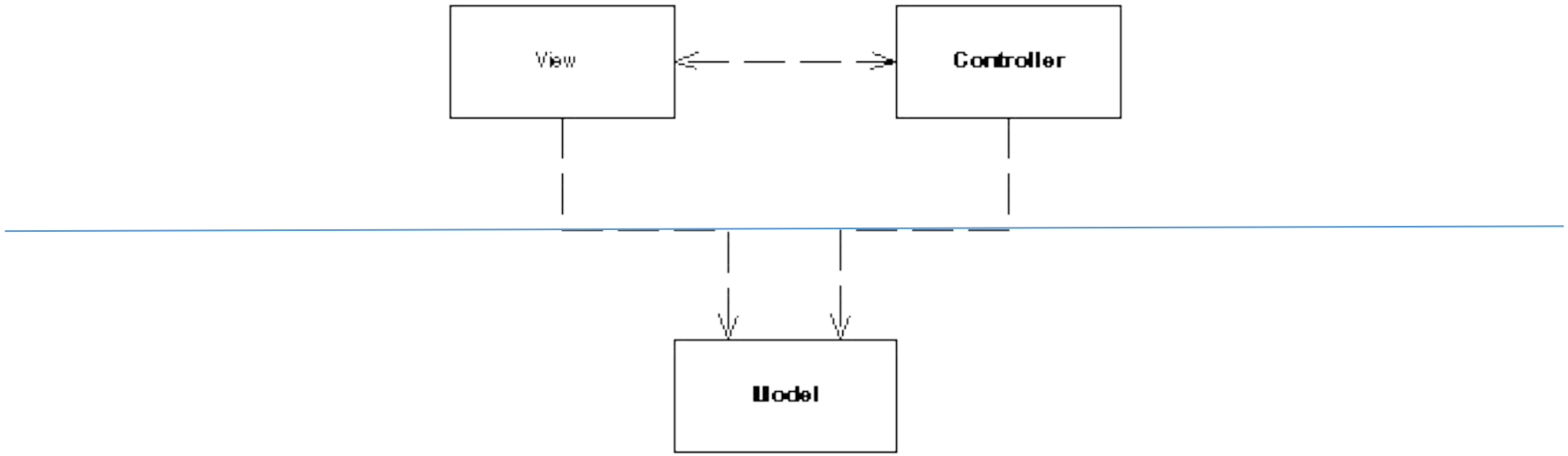
Классический MVC



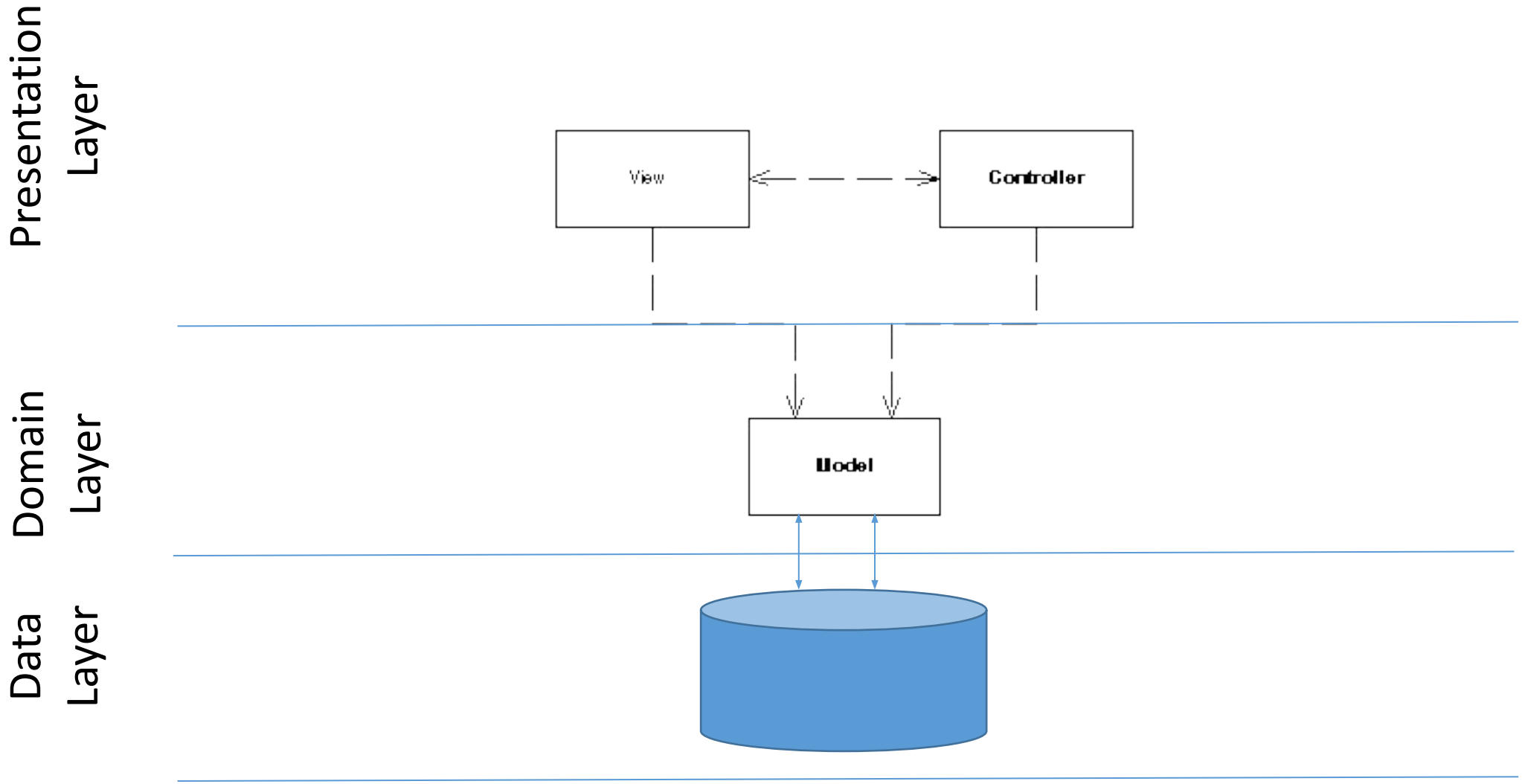
Разделение на слои

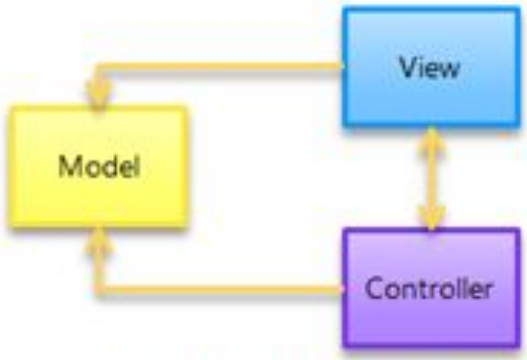
Presentation Layer

Domain Layer

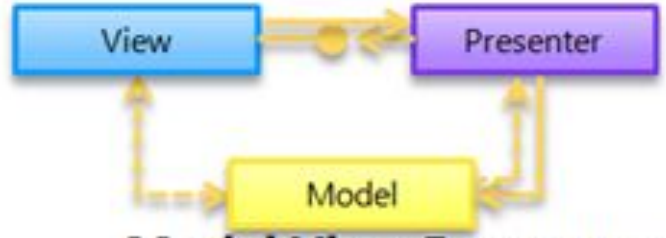


С разделением по слоям

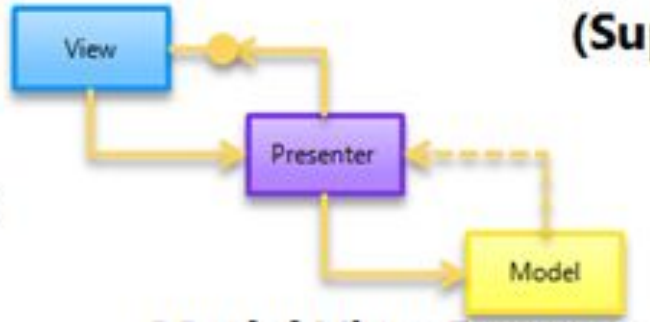




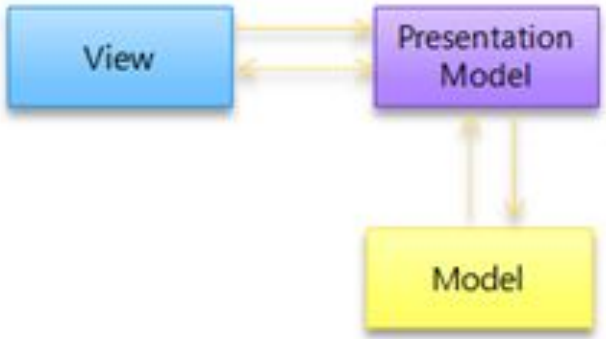
Model-View-Controller



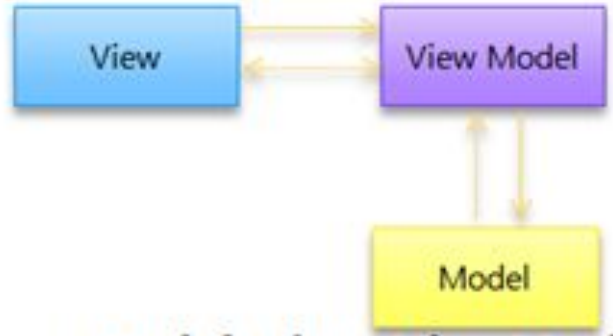
Model View Presenter
(Supervising Controller)



Model View Presenter
(Passive View)



Presentation Model



Model-View-ViewModel

PSR

PHP-FIG (PHP Framework Interop Group) — организованная группа разработчиков, цель которой находить способы совместной работы нескольких фрейморков.

PSR (PHP Standarts Recomendations) — стандартные рекомендации, результат работы PHP-FIG. Одни члены Группы предлагают правила для каждого PSR, другие голосуют в поддержку этих правил или за их отмену. Обсуждение проходит в Google Groups, а наборы PSR доступны на официальном сайте PHP-FIG.

Стандарты

0. Autoloading Standard
1. Basic Coding Standard
2. Coding Style Guide
3. Logger Interface
4. Autoloading Standard
5. PHPDoc Standard
6. Caching Interface
7. HTTP Message Interface
8. Huggable Interface
9. Security Advisories
10. Security Reporting Process
11. Container Interface
12. Extended Coding Style Guide
13. Hypermedia Links
14. Event Dispatcher
15. HTTP Handlers
16. Simple Cache
17. HTTP Factories
18. HTTP Client
19. PHPDoc tags

PSR-1 — Basic Coding Standart

Эти PSR регулируют основные стандарты, главная идея которых — если все разработчики используют одни стандарты, то перенос кода можно производить без всяких проблем.

Правила:

1. В файлах должны использоваться только теги <?php и <?.
2. В файлах должна использоваться только кодировка UTF-8 without BOM.
3. Имена пространств и классы должны следовать PSR-0.
4. Имена классов должны быть объявлены в нотации StudlyCaps.
5. Константы класса должны быть объявлены в верхнем регистре, разделенные подчеркиваниями.
6. Методы должны быть объявлены в нотации camelCase.

PSR-2 — Coding Style Guide

Это расширенные инструкции для PSR-1, описывающие правила форматирования кода.

Правила:

1. Код должен соответствовать PSR-1.
2. Вместо табуляции должны использоваться 4 пробела.
3. Не должно быть строгого ограничения на длину строки, рекомендуемая длина — до 80 символов.
4. Должна быть одна пустая строка после объявления пространства имен.
5. Скобки для классов должны открываться на следующей строке после объявления и закрываться после тела класса (то же самое для методов).
6. Видимость методов и свойств должна быть обязательно определена (public, private).
7. Открывающие скобки для управляющих структур должны находиться на той же строке, закрывающие скобки должны быть на следующей строке после тела структуры.
8. Пробелы не ставятся после открывающихся круглых скобок методов управляющих структур и перед закрывающимися скобками.

PCR-3 — Logger Interface

В PCR-3 регулируется логгинг, в частности основные девять методов.

Методы:

1. LoggerInterface предоставляет 8 методов для логирования восьми RFC 5424 уровней (debug, notice, warning, error, critical, alert, emergency).
2. Девятый метод log() принимает на вход уровень предупреждения первым параметром. Вызов метода с параметром уровня предупреждения должен возвращать такой же результат, как и вызов метода определенного уровня лога (log(ALERT) == alert()). Вызов метода с неопределённым уровнем предупреждения должен генерировать Psr\Log\InvalidArgumentException.

PCR-4 — Improved Autoloading

Так же как и PSR-0, PSR-4 предоставляет улучшенные методы автозагрузки

Правила:

1. Термин «класс» относится к классам, интерфейсам, трейтам и другим похожим структурам
2. Полностью определённое имя класса имеет следующую форму: \<NamespaceName>(\<SubNamespaceNames>)*\<ClassName>
3. При загрузке файла, соответствующему полностью определённому имени класса:
   - Непрерывная серия одного или более ведущих пространств имен, не считая ведущего разделителя пространства имен, в полностью определенном имени класса соответствует по крайней мере одной «корневой директории».
   - Имена директорий и поддиректорий должны соответствовать регистру пространства имен.
   - Окончание полного имени класса соответствует имени файла с окончанием .php. Регистр имени файла обязан соответствовать регистру окончания полного имени класса.
   - Реализация автозагрузчика не должна бросать исключения, генерировать ошибки любого уровня и не обязана возвращать значение.

Полезное

- Паттерны в картинках https://github.com/domnikl/DesignPatternsPHP
- Петтерны от дядюшки Боба (М. Фаулер) https://martinfowler.com
- Материал лекции https://github.com/dimaxz/Lectures/tree/master/Лекция%202.%20Паттерны
- https://www.php-fig.org/psr/