


Паттерны проектирования





... Разрабатываемые приложения становились больше и сложнее, из одной сплоченной команды разработчиков стало много разных команд разработчиков, архитекторов, юзабилитистов, дизайнеров и PMов. Такой подход к разработке требует наличие стойкой архитектуры, которая бы синхронизировала разные функциональные области продукта между собой.

Шаблонность действий

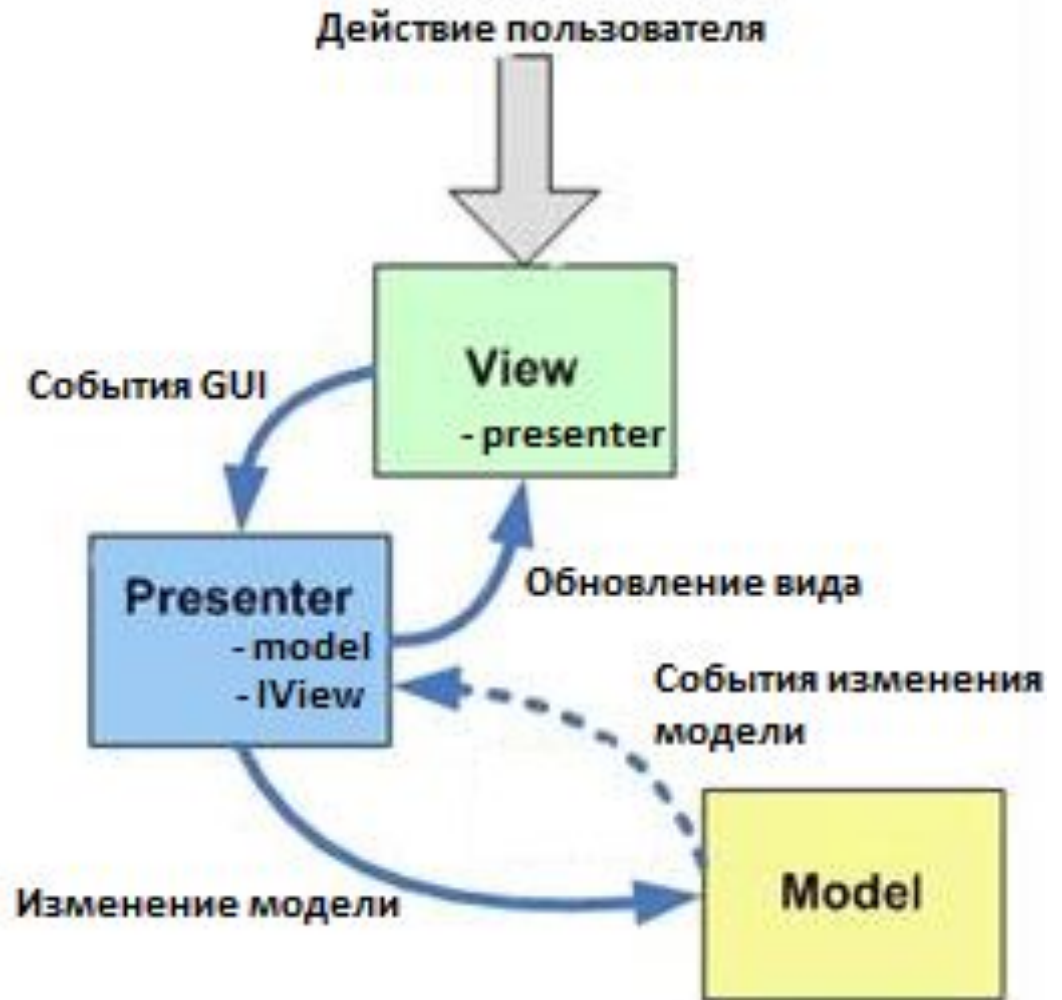
- облегчает коммуникацию между разработчиками;
- позволяет ссылаться на известные конструкции;
- снижает количество ошибок.

Паттерн

(англ. design pattern) - повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста.

Model-View-Controller (MVC)

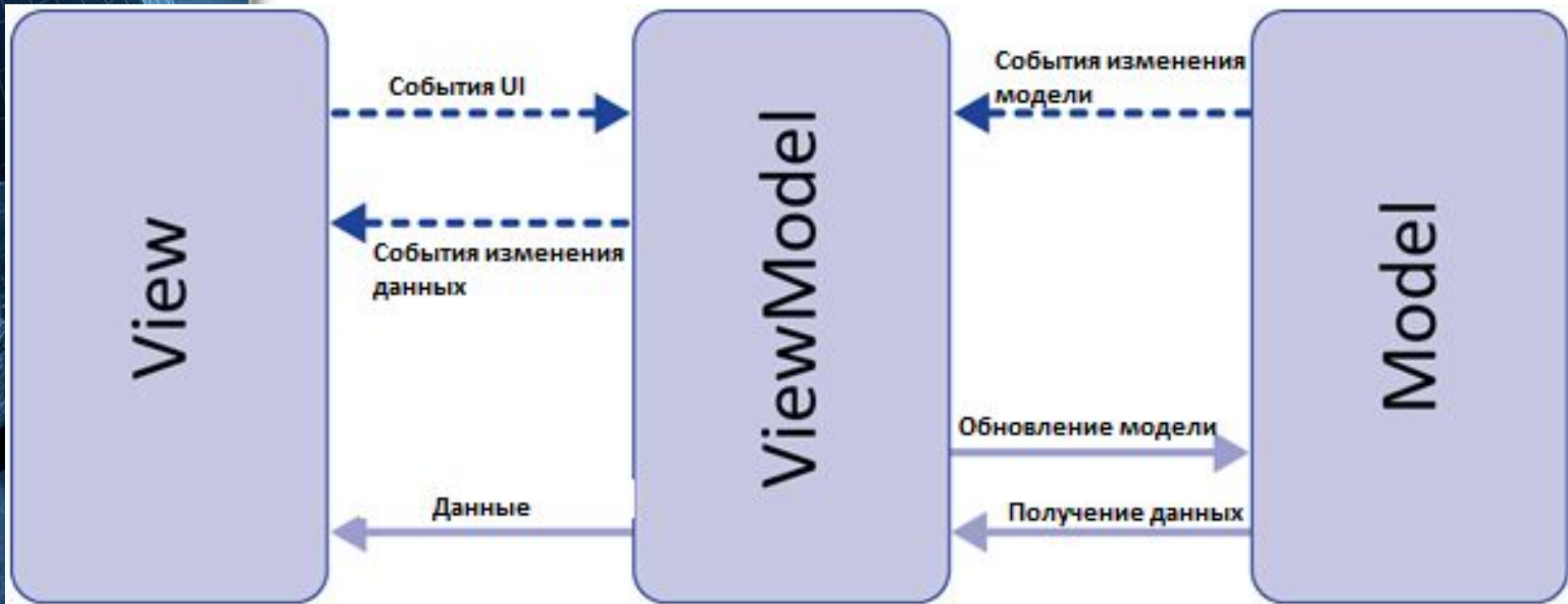
MVP



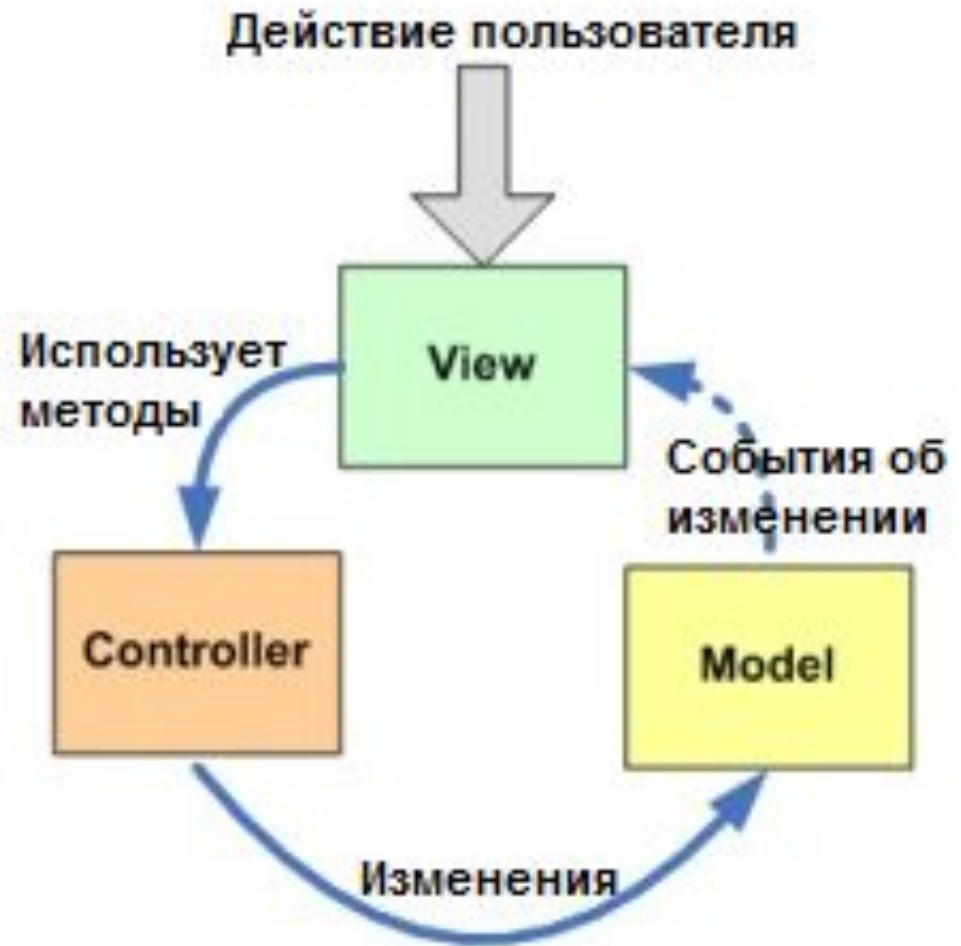
Model-View-Presenter

MVVM

(Model-View-ViewModel)



MVC



Model-View-Controller

Общие правила выбора паттерна

- MVVM

Используется в ситуации, когда возможно связывание данных без необходимости ввода специальных интерфейсов представления (т.е. отсутствует необходимость реализовывать IView);

Частым примером является технология WPF.

- MVP

Используется в ситуации, когда невозможно связывание данных (нельзя использовать Binding);

Частым примером может быть использование Windows Forms.

- MVC

Используется в ситуации, когда связь между представлением и другими частями приложения невозможна (и Вы не можете использовать MVVM или MVP);

Частым примером использования может служить ASP.NET MVC.

Что почитать

1. Паттерны для новичков: MVC vs MVP vs MVVM

<https://habrahabr.ru/post/215605/>

2. Паттерны разработки: MVC vs MVP vs MVVM vs MVI

<https://habrahabr.ru/post/344184/>

3. Различия между MVVM и остальными MV*-паттернами

<https://habrahabr.ru/company/mobileup/blog/313538/>