# Шаблонът MVC

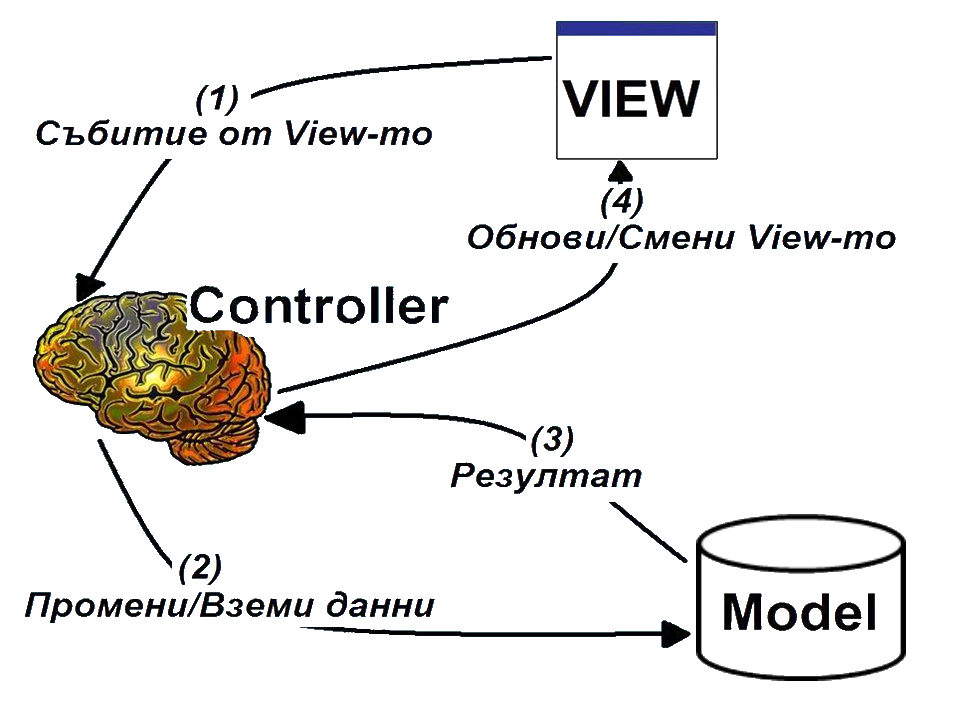
## Кога се използва

* Честа смяна на потребителския интерфейс.
* Различно представяне на едни и същи данни.
* Работа върху различни устройства.
* Улеснено модулно тестване на системата.

## Какво се постига

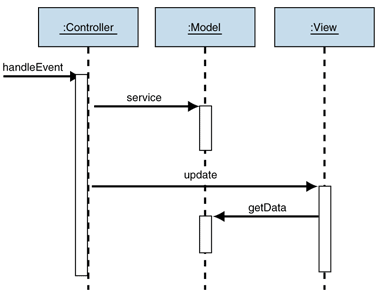
* Независимост на отделните модули.
* Преизползваемост на модулите.
* Добра структурна организация на системата.

## Дефиниция



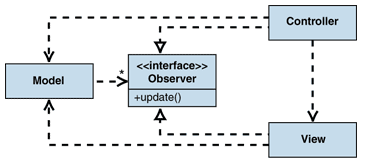
* Model – представя данните и бизнес правила за тяхното манипулиране.
* View – потребителски интерфейс.
* Controller – извършва промени по модела предизвикани от потребителя.
* MVC Control Flow:
  + Потребителят взаимодейства с интерфейса (например натиска бутон).
  + Controller-ът обработва възникналото събитие.
  + Controller-ът уведомява Model-a за действието и евентуалната промяна.
  + View-то използва Model-a (индиректно), за да генерира подходящия потребителски интерфейс.
  + Чака се ново действие от страна на потребителя.
* Особености
  + Model-ът НЕ трябва да зависи от Controller-а или View-то.
  + Model-ът може да предоставя методи на Controller-a и View-то.
  + Дизайнът на Controller-а зависи от Model-a.
  + Дизайнът на View-то зависи от Model-a.

## Пасивен вариант



* Model-ът се променя от намеса на Controller-a.
* Controller-ът информира View-то за промяната в Model-a.
* Пълна независимост на Model-a от Controller-a и View-то.
* Пример: HTTP.

## Активен вариант

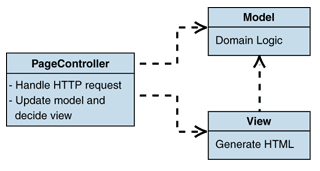


* Model-ът може да бъде променян БЕЗ намеса на Controller-a.
* Тук постигаме независимост на Model-a чрез Observer Pattern.
  + View-то и Controller-ът се „абонират“ за промени в Model-a.
  + Model-ът не се интересува кои са „абонатите“ за неговите промени.

## Свързани шаблони

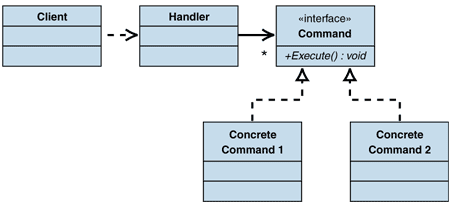
* Strategy
* Observer
* Composite
* Command
* Page Controller
* Front Controller

## Page Controller



* Обработва потребителски заявки.
* Извиква операции от Model-a.
* Решава кое е следващото View.
* Използва базов клас, за да се избегнат повторения на код.
* View-то чете директно от Model-a.

## Front Controller



* Състои се от две части:
  + Handler;
  + Йерархия от „команди“.
* Handler-ът получава параметри от заявката и избира команда.
* Всяка команда е конкретно действие.

## Приложения

* Smalltalk
* PHP – php.MVC, Symfony
* Ruby On Rails
* .NET – ASP.NET MVC
* Java – Struts, Spring, Stripes

## Предимства

* По-добра поддръжка и разширяемост.
* Тестването на отделни модули е по-лесно.
* Различни View-та към едни и същи данни.
* Изолация на промените в рамките на модула.

## Недостатъци

* Сложен за реализация.
* Трудно се прави Debug.
* Риск от чести подновявания на View-то.
* Намалена производителност.

## Как се стартира примерното web приложение:

## Като обикновено приложение – Ctrl + F5.

* Към URL-а в браузъра се добавя /Movies/ (напр. <http://localhost:12345/Movies/>).
* Приложението поддържа функционалност за създаване, редакция и изтриване на филми от локална база от данни.