

## MVC چیست!؟

MVC مخفف سه کلمه Model (مدل) و View (نمایشگر) و controller (کنترلگر) است.

برخی از برنامه‌نویسان، همچنان از ASP.NET که بر مبنای فرم‌های وب و Postback است، استفاده می‌کنند، برخی از ویژگی‌های MVC سود می‌برند و بعضی‌ها هم دو پلتفرم را ترکیب می‌کنند و این موضوع بیانگر این است که هیچکدام از پلتفرم‌ها ناقص یکدیگر نیستند. در واقع MVC بر روی معماری‌های چند لایه‌ای جهت تفکیک بخش‌های مختلف برنامه (بخش‌های منطقی برنامه مانند داده‌ها، مجوزها، کنترل صحت داده‌ها و لایه‌های مرتبط با کاربر نهایی) قرار می‌گیرد. مفهوم تازه‌ای نیست، خیلی وقت است که در جاوا، PHP و بسیاری پلت فرم‌های دیگر از این الگوها برای طراحی نرم افزار استفاده می‌شود.

### اجزای تشکیل دهنده MVC

1- Model (مدل): قسمتی از برنامه کاربردی است که مسئول بازیابی داده از بانک اطلاعاتی، ذخیره آن، تبدیل آن به شی یا شیها و پیاده سازی منطق برنامه برای داده‌های دامنه‌ای مسئله است. در حقیقت بار اصلی معماری MVC بر عهده‌ی این بخش است. مثلاً یک شی Product ممکن است اطلاعات را از بانک اطلاعاتی بازیابی کرده، بر روی آنها عملیاتی را انجام دهد و سرانجام نتیجه را در بانک اطلاعاتی یا در جدول Products ذخیره کند.

۲- view (نمایشگر): اجزایی از برنامه است که واسط کاربری برنامه (UI) را می‌سازد. معمولاً این UI از داده‌های مدل ساخته می‌شود. در واقع نقطه پایان برنامه کاربردی است. به کاربر نتایج عملیات و بازیابی و نمایش داده از طریق برقراری ارتباط با دو بخش دیگر؛ یعنی مدل و کنترلگر را نشان می‌دهد. برای مثال، هنگامی که کاربر در فرم ورود به سیستم رمز عبور خود را وارد می‌کند، اکثر برنامه‌نویسان در همان فرم اقدام به چک کردن رمز عبور می‌کنند که این عمل مغایر با قوانین MVC است. در MVC هنگامی که کاربر رمز عبور را کرد، رمز عبور بدون هیچگونه اعمالی به بخش‌های دیگر فرستاده می‌شود و فقط یک نتیجه ساده یا خبر از بخش‌های دیگر دریافت می‌کند که از طریق آن اجازه‌ی ورود به برنامه داده می‌شود.

۳- Controller (کنترلگر): اجزایی از برنامه هستند که مدیریت تعامل با کاربر را بر عهده دارند. می‌توان گفت که واسط بین مدل و نمایشگر می‌باشند؛ یعنی با مدل کار می‌کند و در انتها نمایشگری را برای نشان دادن واسط کاربری انتخاب می‌کند. ورودی کاربر را مدیریت کرده و به آنها پاسخ می‌دهد و با کاربر تعامل می‌کند. برای مثال، کنترلگر عبارت‌های پرس و جوی بانک اطلاعاتی را مدیریت کرده و آن‌ها را به مدل ارسال می‌کند، وظیفه اجرای پرسوجوها با مدل است.

### مزایای برنامه‌های مبتنی بر MVC

- با تقسیم یک برنامه به سه قسمت مدل، نمایشگر و کنترلگر، مدیریت برنامه یا پروژه را ساده‌تر می‌کند.
- از الگوی کنترلگر جلو استفاده می‌کند که درخواست‌های برنامه را توسط یک کنترلگر پردازش می‌کند. این مسئله باعث می‌شود تا بتوانیم برنامه‌هایی را طراحی کنیم که از زیر ساخت‌های غنی مسیر یابی پشتیبانی می‌کند.
- پشتیبانی بهتری از طراحی و توسعه آزمون محور دارد.
- برای برنامه‌های پشتیبانی شده توسط تیم‌های بزرگ برنامه‌نویسان که کنترل بسیار بر رفتار برنامه را می‌خواهند، بهتر کار می‌کنند.

## ActiveRecord چیست؟!؟

Active record pattern الگوی طراحی‌ای است که معمولاً در نرم‌افزارهایی که از پایگاه‌های داده رابطه‌ای برای ذخیره داده‌های خود بهره می‌برند به کار می‌رود. نخستین بار این نام توسط مارتین فاولر در کتاب «الگوهای معماری کاربردی تجاری» یا Patterns of Enterprise Application Architecture به کار رفت.

ثبت‌کنشور روشی برای دستیابی داده‌ها از درون پایگاه داده است. یک جدول پایگاه داده یا نما توسط یک کلاس همپوشانی می‌شود، بنابراین یک نمونه از شی برای یک سطر یکتا در جدول گره می‌خورد. پس از ساخت یک شی، سطر جدیدی به جدول افزوده می‌شود. هر شی‌ای که بارگذاری می‌شود اطلاعات خود را از پایگاه داده می‌گیرد. هنگامی که شی‌ای به‌روز می‌شود، سطر متناظر آن جدول نیز به‌روز می‌شود. کلاس همپوشان متدها یا خواص دستیابی برای هر ستون در جدول را پیاده‌سازی می‌کند. این الگو معمولاً توسط ابزارهای پایدار شی یا object persistence tools و در نگاشت رابطه‌ای شی یا object-relational mapping به‌کار می‌رود. معمولاً روابط کلیدخارجی همچون نمونه شی‌ای که دارای نوع مناسب است توسط یک خاصیت نشان داده می‌شوند. پیاده‌سازی ثبت‌کنشور را می‌توان در فریم‌ورک‌های گوناگون و محیط‌های برنامه‌نویسی بسیاری یافت. برای نمونه، اگر در یک پایگاه داده جدولی با نام parts با ستون‌های name (از نوع رشته‌ای) و price (از نوع صحیح) وجود داشته باشد. الگوی ثبت‌کنشور در کلاسی با نام part پیاده‌سازی می‌شود. این پیاده‌سازی در شبکه‌کد زیر چنین است:

```
part = new Part();
part.name = «Sample part»;
part.price = 123.45;
part.save();
```

شبکه‌کد بالا سطر جدیدی را در جدول parts با مقادیر داده شده می‌سازد و تقریباً معادل دستور SQL زیر است:

```
INSERT INTO `parts` (`name`, `price`) VALUES ('Sample part', 123.45);
```

همچنین، می‌توان از این الگو برای پرس‌وجو از پایگاه داده نیز سود برد.

```
b = Part.find_first_part("name", "gearbox");
```

این شبکه‌کد شی تازه‌ای با نام part می‌سازد که اولین سطر مطابقت شده‌ای که ستون name آن برابر gearbox باشد مقادیر این شی را تشکیل می‌دهند. دستور SQL معادل این شبکه‌کد چنین است:

```
SELECT * FROM `parts` WHERE `name` = 'gearbox' LIMIT 1;
```

## مدل موجودیت-رابطه (Entity Diagram) برنامه

