# 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 فرم های وب وPostback است، استفاده می کنند، برخی از ویژگی های MVC سود می برند و بعضی ها هم دو پلتفرم را ترکیب می کنند و این موضوع بیانگر این است که هیچکدام از پلتفرم ها ناقص یکدیگر نیستند. در واقع MVC بر روی معماری های چند لایه ای جهت تفکیک بخش های مختلف برنامه (بخش های منطقی برنامه مانند داده ها، مجوزها، کنترل صحت داده ها و لایه های مرتبط با کاربر نهایی) قرار می گیرد. مفهوم تازه ای نیست، خیلی وقت است که در جاوا، PHP و بسیاری پلت فرم های دیگر از این الگوها برای طراحی نرم افزار استفاده می شود.

**اجزای تشکیل دهنده ی 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 داده های دامنه ی مسئله است. در حقیقت بار اصلی معماری MVC بر عهده ی 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 می سازد. معمولاً این UI از داده های مدل ساخته می شود. 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 می دهد. برای مثال، هنگامی که کاربر در فرم ورود به سیستم رمز عبور خود را وارد می کند، اکثر برنامهنویسان در همان فرم اقدام به چک کردن رمز عبور می کنند که این عمل مغایر با قوانین MVC است. در MVCهنگامی که کاربر رمز عبور را کرد، رمز عبور بدون هیچگونه اعمالی به بخش های دیگر فرستاده می شود و فقط یک نتیجه ساده یا خبر از بخش های دیگر دریافت می کند که از طریق آن اجازه ی ورود به برنامه داده می شود.

۳- Controller(کنترلگر): اجزایی از برنامه هستند که مدیریت تعامل با کاربر را بر عهده دارند. می توان گفت که واسط بین مدل و نمایشگر می باشند؛ یعنی با مدل کار می کند و در انتها نمایشگری را برای نشان دادن واسط کاربری انتخاب می کند. ورودی کاربر را مدیریت کرده و به آنها پاسخ می دهد و با کاربر تعامل می کند. برای مثال، کنترلگر عبارت های پرس و جوی بانک اطلاعاتی را مدیریت کرده و آن ها را به مدل ارسال می کند، وظیفه اجرای پرسوجوها با مدل است.

**مزایای برنامه های مبتنی بر MVC**

- با تقسیم یک برنامه به سه قسمت مدل، نمایشگر و کنترلگر، مدیریت برنامه یا پروژه را ساده تر می کند.

- از الگوی کنترلگر جلو استفاده می کند که درخواست های برنامه را توسط یک کنترلگر پردازش می کند. این مسئله باعث می شود تا بتوانیم برنامه هایی را طراحی کنیم که از زیر ساخت های غنی مسیر یابی پشتیبانی می کند.

- پشتیبانی بهتری از طراحی و توسعه آزمون محور دارد.

- برای برنامه های پشتیبانی شده توسط تیم های بزرگ برنامهنویسان که کنترل بسیار بر رفتار برنامه را می خواهند، بهتر کار می کنند.

# ActiveRecord چیست!؟

Active record pattern الگوی طراحی‌ای است که معمولاً در نرم‌افزارهایی که از پایگاه‌های دادهٔ 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.

ثبت کنشور 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 پایگاه‌داده می‌گیرد. هنگامی که شی‌ای به‌روز می‌شود، سطر متناظرآن 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 ثبت کنشور را می‌توان در فریم‌ورک‌های گوناگون و محیط‌های برنامه‌نویسی بسیاری یافت. برای نمونه، اگر در یک پایگاه‌داده جدولی با نام parts با ستون‌های name (از نوع رشته‌ای) و price (از نوع صحیح) وجود داشته باشد. الگوی ثبت کُنشور در کلاسی با نام part پیاده‌سازی می‌شود. این پیاده‌سازی در شبه‌کد زیر چنین است:

part = new Part();

part.name = «Sample part»;

part.price = 123.45;

part.save();

شبه‌کد 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 پایگاه‌داده نیز سود برد.

b = Part.find\_first\_part("name", "gearbox");

این شبه‌کد 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 شبه‌کد چنین است:

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

# مدل موجودیت-رابطه(Entity Diagram) برنامه

