

با توجه به اینکه الگوهای طراحی زیادی وجود دارند، چگونه می-توانید مناسب-ترین الگوی طراحی را برای حل مسئله خود انتخاب کنید و مهم-تر اینکه چگونه آن را اعمال نمایید؟ برای پاسخ به این سوال، رهنمودهای زیر را همیشه در نظر داشته باشید:

- شما نمی-توانید یک الگو را به کار بگیرید مگر آنکه آن را به خوبی فهمیده باشید. بنابراین در اولین گام باید اصول و الگوهای طراحی را هم به شکل انتزاعی و هم به شکل واقعی خوانده و تمرین کنید. دقت کنید که یک الگو را به شکل-های مختلفی می-توان پیاده سازی کرد. هر چه پیاده سازی-های بیشتری ببینید، به هدف و چگونگی استفاده از آن بهتر مسلط می-شوید.

- آیا می-خواهید با استفاده از یک الگوی طراحی، برنامه خود را پیچیده-تر کنید؟ این معمول است که توسعه دهندگان برای حل هر مسئله از الگوهای طراحی استفاده می-کنند. ابتدا هزینه و فایده پیاده سازی یک الگو را ارزیابی کرده، سپس اقدام به استفاده کنید. همیشه قاعده [KISS](#) را به خاطر داشته باشید.

- مسئله خود را تعمیم دهید. پیامدهای مسئله خود را با دید انتزاعی و سطح بالا بررسی کنید. به یاد داشته باشید که الگوهای طراحی، راه حل-های سطح بالا برای مسائل سطح بالا هستند. بنابراین روی پیامدهای جزئی یا وابسته به دامین مسئله خود تمرکز نکنید.

- به الگوهای مشابه و هم گروه نگاه کنید. اگر قبلا از یک الگو استفاده کرده-اید بدین معنی نیست برای هر مسئله-ای آن الگو درست است.

- هر چیزی که تغییر می-کند را بسته بندی کنید. ببینید که چه چیزی در برنامه کاربردی شما ممکن است تغییر کند. اگر شما می-دانید که یک الگوریتم اعمال تخفیف ممکن است به مرور زمان تغییر کند، به دنبال الگویی باشید که تغییرات در آن الگوریتم را بدون تاثیر بر سایر قسمت-های برنامه کاربردی انجام دهید.

- وقتی که یک الگو را انتخاب کردید، از زبان الگو در کنار زبان دامین برای نام گذاری کلاس-ها استفاده کنید. برای مثال اگر از [الگوی Strategy](#) استفاده می-کنید تا هزینه حمل و نقل کالا توسط شرکت FedEx را محاسبه کند، از نام `FedExShippingCostStrategy` استفاده کنید. با استفاده از زبان مشترک بین الگو طراحی و مدل دامین، کد برنامه برای شما و دیگران خواناتر و قابل فهم-تر می-گردد.

- همیشه منظور هر الگو را در ذهن خود مرور کنید و هنگام برخورد با یک مسئله به دنبال مناسب-ترین الگو بگردید. یک تمرین یادگیری عالی شناسایی الگوهای طراحی در فریم ورک .Net است. برای مثال، `ASP.Net Cache` از الگوی `Singleton` استفاده می-کند و کلاس `Guid` از الگوی `Factory` بهره می-برد.

تا به حال شما باید هدف و الگوریتم استفاده از الگوهای طراحی را درک کرده باشید. در [ادامه](#) با لایه بندی برنامه کاربردی آشنا می-شوید و سپس نحوه استفاده از این الگوها در لایه-های مختلف را فرا خواهید گرفت.

نظرات خوانندگان

نویسنده: سید مهدی فاطمی
تاریخ: ۱۹:۱۵ ۱۳۹۲/۰۹/۱۲

تشکر

مطالبی که گفتید رو من به عینه باهاش درگیر بودم و هستم اما تا حالا راه حلی براش پیدا نکردم مثلاً الگویی که من استفاده می‌کنم به این صورت هست که برا هر موجودیتی یک فرم در نظر می‌گیرم و در این فرم 4 عمل (جستجو - اضافه - حذف - ویرایش) رو در اون تعبیه می‌کنم که در بعضی مواقع این احساس بهم دست می‌ده که کدهام دارای پیچیدگی شده .

این مطالب شما به صورت نظری هستن اگه امکانش هست مثال هایی به صورت ملموس‌تری بزنید ممنون میشم.

نویسنده: محسن خان
تاریخ: ۲۲:۳ ۱۳۹۲/۰۹/۱۲

مثال اگر نیاز دارید سری بحث‌های [معماری لایه بندی نرم افزار](#) در سایت مفید است. همچنین اگر تمام متدها رو داخل یک فرم قرار دادید بهتره از مطلب [آشنایی با Refactoring - قسمت 1](#) شروع کنید.