عنوان: **5 دلیل برای استفاده از یک ابزار ORM**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۳/۰۱ ۱۴:۱۳:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: Tips

چرا باید از ابزارهای Object relational Mapper یا به اختصار ORM استفاده کرد؟ در اینجا سخن در مورد ORM خاصی نیست. هدف تبلیغ یک محصول ویژه هم نمی‌باشد و یک بحث کلی مد نظر است.
کار ابزارهای ORM خواندن ساختار دیتابیس شما بوده و سپس ایجاد کلاس‌هایی بر اساس این ساختار ، برقراری ارتباط بین اشیاء ایجاد شده و جداول، ویووها، رویه‌های ذخیره شده و غیره می‌باشد. همچنین این ابزارها امکان تعریف روابط one-to-one, one-to-many, many-to-one و many-to-many بین اشیاء را نیز بر اساس ساختار دیتابیس شما فراهم می‌کنند.
در ادامه به فواید استفاده از ORM ها خواهیم پرداخت:

الف) یک ابزار ORM زمان تحویل پروژه را کاهش می‌دهد

اولین و مهم‌ترین دلیلی که بر اساس آن در یک پروژه، استفاده از ORM حائز اهمیت می‌شود، بحث بالا بردن سرعت برنامه نویسی و کاهش زمان تحویل پروژه به مشتری است. این کاهش زمان بسته به نوع پروژه بین 20 تا 50 درصد می‌تواند خود را بروز دهد.
بدیهی است ابزارهای ORM کار شگفت انگیزی را قرار نیست انجام دهند و شما می‌توانید تمام آن عملیات را دستی هم به پایان رسانید؛ اما اجازه دهید یک مثال کوتاه را با هم مرور کنیم.
برای پیاده سازی یک برنامه متداول با حدود 15 تا 20 جدول، حدودا به 30 شیء برای مدل سازی سیستم نیاز خواهد بود و برنامه نویسی این مجموعه بین 5000 تا 10000 سطر کد را به خود اختصاص خواهد داد. بدیهی است برنامه نویسی و آزمایش این سیستم چندین هفته یا ماه به طول خواهد انجامید.
اما با استفاده از یک ORM ، عمده وقت شما به طراحی سیستم و ایجاد ارتباطات بین اشیاء و دیتابیس در طی یک تا دو روز صرف خواهد شد. ایجاد کد بر اساس این مجموعه و با کمک ابزارهای ORM ، آنی است و با چند کلیک صورت می‌گیرد.

ب) یک ابزار ORM کدی با طراحی بهتر را تولید می‌کند

ممکن است شما بگوئید که کد نویسی من بی‌نظیر است و از من بهتر کسی را نمی‌توانید پیدا کنید! به تمامی زوایای کار خود مسلطم و نیازی هم به این‌گونه ابزارها ندارم!
عده‌ای از شما به طور قطع این‌گونه‌اید؛ اما نه همه. در یک تیم متوسط، همه نوع برنامه نویس با سطوح مختلفی را می‌توانید پیدا کنید و تمامی آن‌ها برنامه نویس‌ها و یا طراح‌های آنچنان قابلی هم نیستند. بنابراین امکان رسیدن به کدهایی که مطابق اصول دقیق برنامه نویسی شیء گرا نیستند و در آن‌ها الگوهای طراحی به خوبی رعایت نشده، بسیار محتمل است. همچنین در یک تیم زمانیکه از یک الگوی یکسان پیروی نمی‌شود، نتایج نهایی بسیار ناهماهنگ خواهند بود.
در مقابل استفاده از ORM های طراحی شده توسط برنامه نویس‌های قابل (senior (architect level) engineers) ، کدهایی را بر اساس الگوهای استاندارد و پذیرفته شده‌ی شیءگرا تولید می‌کنند و همواره یک روند کاری مشخص و هماهنگ را در یک مجموعه به ارمغان خواهند آورد.

ج) نیازی نیست تا حتما یک متخصص دات نت فریم ورک باشید تا از یک ORM استفاده کنید

قسمت دسترسی به داده‌ها یکی از اجزای کلیدی کارآیی برنامه شما است. اگر طراحی و پیاده سازی آن ضعیف باشد، کل برنامه را زیر سؤال خواهد برد. برای طراحی و پیاده سازی دستی این قسمت از کار باید به قسمت‌های بسیاری از مجموعه‌ی دات نت فریم ورک مسلط بود. اما هنگام استفاده از یک ORM مهمترین موردی را که باید به آن تمرکز نمائید بحث طراحی منطقی کار است و ایجاد روابط بین اشیاء و دیتابیس و امثال آن. مابقی موارد توسط ORM انجام خواهد شد و همچنین می‌توان مطمئن بود که پیاده سازی خودکار انجام شده این قسمت‌ها، بر اساس الگوهای طراحی شیءگرا است.

د) هنگام استفاده از یک ابزار ORM ، مدت زمان آزمایش برنامه نیز کاهش می‌یابد

بدیهی است اگر قسمت دسترسی به داده‌ها را خودتان طراحی و پیاده سازی کرده باشید، زمان قابل توجهی را نیز باید به بررسی و آزمایش صحت عملکرد آن بپردازید و الزامی هم ندارد که این پیاده سازی مطابق بهترین تجربیات کاری موجود بوده باشد. اما هنگام استفاده از کدهای تولید شده توسط یک ابزار ORM می‌توان مطمئن بود که کدهای تولیدی آن که بر اساس یک سری الگوی ویژه تولید می‌شوند، کاملا آزمایش شده هستند و همچنین صدها و یا هزارها نفر در دنیا هم اکنون دارند از این پایه در پروژه‌های موفق خود استفاده می‌کنند و همچنین بازخوردهای خود را نیز به تیم برنامه نویسی آن ابزار ORM ارائه می‌دهند و این مجموعه مرتبا در حال بهبود و به روز شدن است.

ه) استفاده از یک ابزار ORM ، کار برنامه نویسی شما را ساده‌تر می‌کند

توضیح این قسمت نیاز به ذکر یک مثال دارد. لطفا به مثال زیر دقت بفرمائید:

```
try {
       Employees objInfo = new Employees();
       EmployeesFactory objFactory = new EmployeesFactory();

       objInfo.EmployeeID = EmployeeID;
       objFactory.Load(objInfo);

       // code here to use the "objInfo" object
}
catch(Exception ex) {
      // code here to handle the exception
}
```

به نظر شما کار کردن با یک یا چند شیء تولید شده که نمایانگر ساختار دیتابیس شما هستند و با استفاده از اینترفیس عمومی آن‌ها می‌توان تمامی اعمال بارگذاری، درج و حذف و غیره را انجام داد، ساده‌تر است یا کار کردن با کوهی از دستورات ADO.Net ؟

برداشتی آزاد از Five Reasons for using an ORM Tool

نظرات خوانندگان

نویسنده: صدای سکوت
تاریخ: ۱۳۸۸/۰۳/۰۲ ۱۰:۵۳:۴۷

سلام
از مطالب خوبتون ممنون
من با اجازه شما رو به بلاگ خودم لینک کردم.
موفق باشید

نویسنده: افشار محبی
تاریخ: ۱۳۸۸/۰۳/۰۳ ۰۸:۴۴:۳۶

این را هم اضافه کنید: با استفاده از ORM نگهداری کدها و فهم آنها آسان‌تر می‌شود.
ما این موضوع در استفاده از NHibernate به خوبی درک کرده‌ایم.

نویسنده: Anonymous
تاریخ: ۱۳۸۸/۰۳/۰۳ ۰۸:۵۰:۲۱

باسلام و تشکر بایت این پست
دوست جناب نصیری بهترین ORM tool موجود مال کدوم شرکته
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۳/۰۳ ۱۰:۰۵:۱۴

NHibernate در حال حاضر پخته‌ترین ORM موجود برای دات نت فریم ورک است.

نویسنده: Anonymous
تاریخ: ۱۳۸۸/۰۳/۰۵ ۰۷:۴۵:۲۶

آیا typed dataset ها در .net هم یک orm هستند؟

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۳/۰۵ ۱۰:۲۴:۰۵

سؤال شما به موضوع بحث مرتبط نیست. آیا جایی در مقاله به typed dataset اشاره شده یا کار نقد orm های مختلف صورت گرفته؟ ... بگذریم.

در مورد typed dataset ها ، بله نسبتا تا حدودی و تا حد نازلی بله! شبه ORM هستند که این مشکلات را دارند:
- مشکل Synchronization بین آن‌ها و دیتابیس مساله ساز است که در یک ORM خوب باید این مساله حل شده باشد.
- join table queries در طراح آن کار نمی‌کند!
- query syntax استانداردی نداشته و هنگام کار با دیتابیس‌های مختلف (نوع‌های مختلف) این مورد مساله ساز می‌شود.
- typed dataset کمتر حال و هوای یک ORM واقعی و دنیای شیء گرایی با اشیایی که وابستگی کمی به دیتابیس دارند را ارائه می‌دهد.
- کلا dataset اشیایی با سربار بالا در دات نت فریم ورک مطرح هستند و زمانیکه کارآیی مطرح هست سعی می‌شود به روش‌هایی دیگری کوچ شود.
همچنین از لحاظ مباحث serialization هم بسیار ضعیف و کند عمل می‌کنند.
- زمانیکه از typed dataset استفاده می‌کنید عملا مدل رابطه‌ای دیتابیس خود را با business layer مخلوط کرده‌اید.

نویسنده: محمد امین شریفی
تاریخ: ۱۳۸۸/۰۷/۱۸ ۲۰:۰۴:۱۰

strongType ها و entity ها را می توان هم پای فناوری های مطرح ORM قرار داد؟

نویسنده: شاهین کیاست
تاریخ: ۱۳۸۹/۱۱/۰۸ ۱۵:۴۵:۳۷

سلام جناب نصیری.
آیا تجربه ای در ضمینه Telerik ORM دارا هستید؟
قابل رقابت با llblgen , nhibernate , EF هست؟
ایا شما nettiers که یک Template برای Codesmith هست را یک ORM به حساب می آورید؟
با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۱/۰۸ ۱۷:۰۹:۲۷

سلام، به نظر من عمر یک برنامه نویس کوتاه است و بهتر است روی موردی وقت بگذارد که :
- امیدی به ادامه‌ی آن باشد (نگن امروز به به! فردا ... خوب دیگه ... تموم شد! صرف نمی‌کنه، دیگه توسعه نمی‌دیم! خیلی از سیاست‌های مایکروسافت همینطوری است. مثلا همون کاری که با LINQ to SQL کرد)
- چند هزار نفری پیرو و دنبال مباحث آن باشند (حداقل 2 تا فوروم رفع اشکال بتونید پیدا کنید)
- دو تا کتاب در موردش باشه
- 4 تا وبلاگ در موردش مطلب بنویسند.
و مسایلی از این دست.

به همین جهت یا روی EF یا NH سرمایه گذاری کنید.
به شخصه NH رو ترجیح می‌دم چون سورس باز است، به همین جهت مرگ برای آن معنی ندارد (این گروه نخواست ادامه بده ... گروه‌های دیگر هستند)، رایگان است، مجوزش اجازه استفاده در کارهای تجاری سورس بسته را می‌دهد. چندتا کتاب در موردش هست و ...
به EF شک دارم. نمی‌دونم مایکروسافت مثلا 4 سال دیگه آیا این را هم بازنشسته اعلام می‌کند یا نه.

نویسنده: mojtaba amrollahi
تاریخ: ۱۳۹۰/۰۲/۰۲ ۲۳:۰۸:۴۶

درود

آقای نصیری یه سئوال از خدمتتون داشتم:
ما این روزا می خایم استارت یه نرم افزار انترپرایس رو بزنیم ولی یکی از چالش های اساسی ما جنگ بر سر استفاده از EF یا NH است.
می خواستم ببینم می تونید اینجانب رو یه راهنمایی با دلایل منطقی بکنید.
ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۲/۰۳ ۰۰:۱۱:۳۶

نگارش جدید EF (نگارش 4 و یک) یک ماهی هست که منتشر شده. مقایسه‌ای از اون رو با NH می‌تونید اینجا مطالعه کنید:
http://jfromaniello.blogspot.com/2011/03/entityframework-41-rc-code-first-review.html

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۲/۰۳ ۰۰:۱۴:۱۱

یک مقایسه دیگر هم اینجا است

/http://farasun.wordpress.com/2011/01/26/entity-framework-vs-nhibernate

کامنت‌ها رو هم حتما بخونید