



دانشگاه پیام نور استان تهران

مرکز تهران شمال

مهندسی کامپیوتر گرایش نرم افزار

شریعتی ، پایین تر از پل صدر ، بن بست شمشاد

شرکت شاتل

برنامه نویسی وب

نام استاد کارآموزی : جناب آقای سید علی رضوی ابراهیمی

نام دانشجو : زینب جلیوند

شماره دانشجویی : 935121562

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

موضوع کارآموزی : برنامه نویسی- back – end

آموزش **Html و java script**

محل کارآموزی : شرکت شاتل

نام استاد کارآموزی در شرکت : جناب آقای مهندس حسین ربیعی

نام کارآموز: زینب جلیلوند

شماره دانشجویی : 935121562

نیم سال اول اول. ۱۴۰۰ – ۱۳۹۹

آدرس : خیابان شریعتی – پایین تر از پل صدر – بن بست شمشاد – پالک 5

ساختمان نرم افزار شرکت شاتل

تاریخ شروع و پایان کارآموزی : 99/7/27 تا 99/10/20

چکیده:

یادگیری اصول اولیه برنامه نویسی در محیط عملیاتی واقعی و در کنار برنامه نویسان با تجربه این شرکت ،
و همچنین در ابتدا نحوه استفاده صحیح از مفاهیم شی گرای با مثال هایی ساده در برنامه های ساده تر.
سپس یادگیری کد زدن با استفاده از معماری سه لایه و معماری چند لایه و معماری mvc و دلایل استفاده
از این معماری ها.

یادگیری HTML و CSS و Bootstrap در حد نیاز برای درک مفاهیم front-end

سپس یادگیری جاوا اسکریپت و jquery

کار با ORM ها برای ارتباط با دیتابیس در لایه DA

کار با Linq در سی شارپ

استفاده از بعضی design pattern

ها کار با git مطالعه کتاب clean

code

فعالیت های انجام شده :

- 1- ابتدا کد زدن در پروژه های ساده و تعریف شده توسط استاد کارآموزی برای جا افتادن مفاهیم شی گرای و معماری سه لایه و....
- 2- کار با ابزار git هنگام کار تیمی
- 3- مشاهده پروژه های واقعی در محیط عملیاتی واقعی کنار برنامه نویسان
- 4- مقدار کمی کد زدن در پروژه های اصلی شرکت با نظارت استاد کارآموزی

فهرست

فصل اول : برنامه نویسی

6.....مفاهیم ابتدایی برنامه نویسی

6.....آشنایی با ابزار git

7.....معماری سه لایه و دلایل استفاده از آن

8..... orm

..... entity framework کار با

9..... مطلب پایانی

مفاهیم ابتدایی برنامه نویسی:

در ابتدا برای یادگیری بیشتر مفاهیم پایه شی گرایی پروژه ساده ای در محیط console توسط استاد کارآموزی تعریف شد ، پروژه ساده درباره موجودات زنده و ویژگی ها و رفتار های آن ها بود که برای آشنایی با مفاهیم interface و مفاهیم پایه شی گرایی که شامل encapsulation و abstraction و inheritance و polymorphism می باشد ، بود.

به این شکل که مثلا ویژگی ها و رفتارهای کلاس پایه موجودات زنده را تعریف کرده ، سپس در کلاس های مثلا انسان ها یا حشرات یا آبزیان از این کلاس موجودات زنده ارث بری کرده و متد های آنرا override کرده و همچنین interface های تعریف میکنیم.

این مثال خیلی خوبی برای درک شی گرایی و مفاهیم اصلی آن بود.

سپس هر روز این برنامه ساده و پروژه را بهبود داده و توسعه دادیم که در نهایت به یک مثال کامل خوب تبدیل شده است.

آشنایی با ابزار git:

هنگام کار تیمی روی یک پروژه نیاز به یک ابزاری داریم که برنامه نویسان امکان این را داشته باشند که همزمان بتوانند روی یک پروژه کار کنند و در نهایت تغییرات خود را روی پروژه با یکدیگر ادغام کنند زیرا نمیشود که مثلا بقیه اعضای تیم بیکار باشند تا یک نفر کار خود را انجام دهد سپس نفر بعدی از

پروژه استفاده کند ، در این حال پروژه روی یک سروری قرار میگیرد و همه اعضای تیم با استفاده از ابزار Git یک کپی از پروژه را از سرور روی سیستم خود تهیه میکنند ، سپس مشغول کد زدن و توسعه قسمتی از پروژه که وظیفه اوست مشغول میشوند و در نهایت تغییرات خود را به سرور با استفاده از ابزار Git ارسال میکنند ، این ابزار همه تغییرات برنامه نویسان را جمع میکند و سپس آنها را با هم merge میکند و سپس روی پروژه اصلی اعمال میکند ، که برای این کار هم همه اعضای تیم باید از یک سری قوانین کد نویسی که در تیم تعریف شده پیروی کنند به عنوان مثال باز کردن کروش به بعد از نام متد باید جلوی نام متد صورت گیرد ، اگر عضوی از تیم کروش را در خط بعدی در برنامه خود باز کند سپس تغییرات را به سرور ارسال کند موقع merge کردن Git این کروش را به عنوان یک تغییر اضافی در نظر میگیرد ، یا یک سری قوانین نامگذاری باید رعایت شود که همه برنامه نویسان تیم و آن پروژه از آن قوانین نامگذاری پیروی کنند ، مثلاً یکی از این قوانین نامگذاری در این شرکت شروع نام متد های private با حرف کوچک است.

معماری سه لایه و دلایل استفاده از آن:

بعد از یادگیری مفاهیم شی گرای و کار کردن چند پروژه سمپل و تست ، سراغ مفاهیم معماری سه لایه رفتیم. ابتدا معماری سه لایه شرح داده شد که پروژه را در سه لایه پیاده سازی میکنیم که در آن بالاترین لایه را UI یا رابط کاربری یا presentation layer مینامیم که وظیفه ارتباط با کاربر را دارد و از کاربر مقادیر را دریافت میکند و یا مقادیر درخواست شده توسط کاربر را نشان میدهد که فقط وظیفه این لایه ارتباط با کاربر است نه کار دیگر.

لایه بعدی لایه کسب و کار است که به اختصار BL نامیده میشود ، که در این لایه همه منطق کسب و کار وجود دارد و همه محاسبات و همه بیسینس ما به عبارت دیگر در این لایه پیاده سازی میشود.

لایه بعدی لایه دسترسی به داده است که به اختصار DAL نامیده میشود که وظیفه ارتباط با دیتابیس را دارد و کلاس های اصلی موجودیت ها در این لایه تعریف میشود و وظیفه اضافه کردن و آپدیت کردن و حذف داده در دیتابیس و همچنین واکشی داده از دیتابیس را برعهده دارد.

نکته دیگری که در معماری سه لایه وجود دارد این است که هر لایه فقط لایه پایین تر از خود را میبیند ، یعنی لایه UI در reference هایش فقط لایه BL اضافه میشود و همچنین لایه BL در reference هایش فقط لایه DAL اضافه میشود ، پس برای انتقال داده بین لایه ها باید در لایه های دیگر کلاس هایی داشت که بتوان مقادیر و درخواست های وارد شده توسط کاربر را به DAL رساند ، برای این منظور و برای انتقال موجودیت اصلی بین لایه ها در لایه BL کلاسی با همان اسم موجودیت و با پسوند (data transfer object) DTO میسازیم و در UI کلاسی با نام همان موجودیت و با پسوند ViewModel میسازیم که بتوان مقادیر را بین لایه ها انتقال داد و این کلاس ها را به یکدیگر Map یا نگاشت کرد.

روال کار به این صورت هست که ابتدا دیتایی که از دیتابیس میاید به کلاس موجودیت اصلی map میشود ، سپس این مقدار به لایه بالایی خود میاید و به کلاس DTO در لایه بالایی map میشود ، سپس از این لایه به لایه UI میرود و در آن لایه به کلاس ViewModel در لایه بالایی map میشود ، هنگام ارسال داده از UI به دیتابیس نیز این روال به صورت برعکس تکرار میشود.

دلایل استفاده از این معماری سه لایه و معماری چند لایه در برنامه نویسی کاهش Dependency یا وابستگی لایه ها به یکدیگر تا حد صفر میباشد ، زیرا بعد ها در صورت نیاز اگر تکنولوژی جدیدتری آمد یا خواستیم UI خود را مثلا از WinForm به Web تغییر دهیم به کمترین حد کد نویسی نیاز داشته باشیم و فقط لایه بالایی خود را تغییر دهیم بدون اینکه نیاز باشد به لایه های پایین تر کاری داشته باشیم ، یا اگر تکنولوژی جدیدتری آمد برای ارتباط با دیتابیس فقط لایه DAL خود را تغییر دهیم که همین اتفاق برای من در یکی از پروژه های سمپل افتاد ، که ابتدا ارتباط با دیتابیس را با استفاده از تکنولوژی Ado.Net انجام داده بودم که بعد از یادگیری Entity Framework هنگام تغییر ارتباط دیتابیس فقط لایه DAL خود را تغییر دادم بدون نیاز به تغییر بخش های دیگر برنامه.

کار با orm entity framework

برای ارتباط با دیتابیس میتوان از orm های مختلفی استفاده کرد که یکی از محبوبترین های آن Entity Framework میباشد که سه مدل برای پیاده سازی دارد که یکی از مدل های آن به نام Model first منسوخ شده است تقریبا و استفاده خیلی کمتری دارد.

مدل دیگر آن Db first (Data Base First) می باشد که ابتدا شما دیتابیس خود را طراحی میکنید سپس EF از روی دیتابیس شما کلاس های شما را ایجاد میکند موقع ذخیره سازی و یا واکنشی اطلاعات نیز کار راحت تری دارید.

اما مدل سوم و پر استفاده ترین مدل آن در این روز ها مدل Code First میباشد که در آن شما کلاس های موجودیت خود را ایجاد میکنید ، سپس یک کلاس دیگر که از کلاس DbContext ارث بری میکند ایجاد میکنید و در آن مقادیری را اضافه میکنید ، سپس EF از روی کد های شما دیتابیس شما را طراحی میکند.

مطلب پایانی:

بعد از یادگیری مفاهیم و انجام تمرین های بسیار پروژه هایی در asp.net mvc انجام دادم و در کنار آن ها مشغول کد زنی هستم با نظارت استاد کارآموزی ، و فکر میکنم این مدت 2 ، 3 ماهه در اینجا خیلی زیاد بار علمی برای من داشت و باعث یادگیری مباحث زیادی برای من شد.