آموزش برنامه نویسی شی گرا در (C#) قسمت ۸ : abstract و sealed (نسخه چاپی)

در قسـمت قبلی آموزش با مفهوم polymorphism آشـنا شـدیم. در ادامه قصد داریم با کلاس های abstract و sealed آشنا شویم. زمانی که شروع به نوشـتن برنامه ای می کنیم، بعد از مشـخص کردن موجودیت های برنامه و طراحی کلاس های مربوطه، باید یکسری محدودیت ها برای اسـتفاده از کلاس ها وضع کرد. برای مثال، کلاس پایه ای داریم که این کلاس نبایـد مورد اسـتفاده قرار بگیرد و تنها باید از کلاس های فرزنـد قابلیت ایجاد شـئ وجود داشـته باشـد یا کلاسـی نوشـته ایم و نباید اجازه ایجاد کلاس فرزند از روی آن کلاس داده شود. این قابلیت ها بخصوص در مواقعی که در تیم شـما، افرادی از کدهای نوشته شده توسط شما استفاده می کنند یا کدی را برای استفاده از سایر برنامه نویس ها بر روی اینترنت منتشـر می کنیـد کاربرد دارنـد. راه حل برای این شـرایط اسـتفاده از کلاس های abstract و کلاس های sealed می باشد.

کلاس ها و اعضاء abstract

به بخش قبل و کلاس Shape بر میگردیم که سه کلاس فرزند از روی آن ها ساخته بودیم:

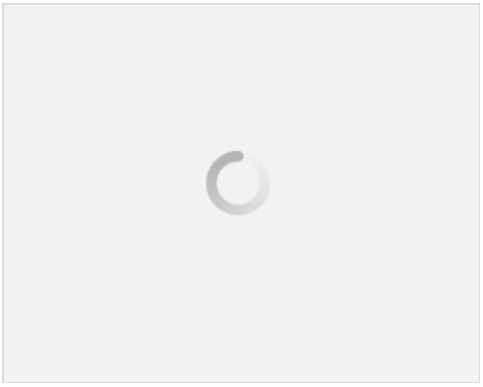
```
public class Shape
{
    public void Draw()
    {
        Console.WriteLine("Drawing the shape!");
    }
}

public class Rectangle : Shape
{

}

public class Triangle : Shape
{

}

public class Circle : Shape
{

}
```

کلاس Shape به تنهایی برای ما کاربردی نداشـته و تنها داخل کـد باید از کلاس های فرزند اسـتفاده کرد، یعنی نباید از روی کلاس Shape شئ ایجاد شود. برای اینکار کافیست کلاس Shape را از نوع abstract تعریف کنیم:

```
public abstract class Shape
{
    public virtual void Draw()
```

```
    {
        Console.WriteLine("Drawing the shape!");
    }
}
```

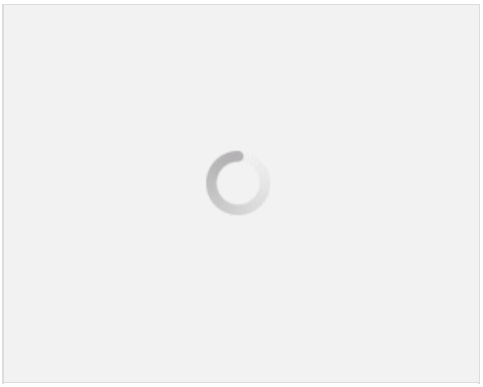
حال اگر بخواهیم از روی کلاس Shape یک شئ ایجاد کنیم با پیغام خطا مواجه خواهیم شد:



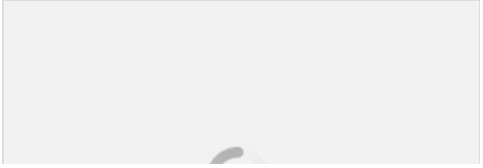
اما کاربرد کلاس های abstract به همین جا ختم نمی شود، در قسمت قبل متد Draw را در کلاس Shape به صورت virtual تعریف کرده و در کلاس های فرزند آن را override کردیم. یکی از قابلیت های زبان سی شارپ، تعریف متدها به صورت abstract می باشد. متدهای abstract تنها شامل signature که در قسمت های اولیه با آن آشنا شدیم می باشد و بدنه ندارد، علاوه بر آن تمامی کلاس هایی که از یک کلاس abstract مشتق می شوند، در صورتی که کلاس abstract رفتار یا خاصیتی از نوع abstract داشته باشد، باید حتماً آن رفتار یا خاصیت را override کنند. برای مثال کلاس Shape را به صورت زیر تغییر می دهیم:

```
public abstract class Shape
{
    public abstract void Draw();
}
```

همانطور که مشاهده می کنید، متد Draw تنها تعریف شده و بدنه ای ندارد. حال اگر کلاس فرزندی از کلاس Shape مشتق شود باید متد Draw داخل آن Override شود. در غیر اینصورت با پیغام خطا مواجه خواهیم شد:



می توانیم به صورت دستی متد Draw را تعریف کرده یا از قابلیت Resharper برای پیاده سازی اعضای abstract به صورت خودکار استفاده کنیم. برای انکار مکان نما را بر روی نام کلاس قرار داده و کلید های Alt+Enter را فشار می دهیم. با این کار منویی با نام Action Context نمایش داده می شود:



از منوی ظاهر شده، گزینه Implement missing members را انتخاب می کنیم تا اعضای abstract پیاده سازی شوند. بعد از اینکار کد کلاس Rectangle به صورت زیر خواهد بود:

```
public class Rectangle : Shape
{
    public override void Draw()
    {
        throw new NotImplementedException();
    }
}
```

کدی که به صورت خودکار داخل بدنه متد Draw قرار گرفته مربوط به یکی از ویژگی های زبان سی شارپ با نام استثناها یا Exceptions می باشد که در بخش های بعدی با آن آشنا می شویم. حال کد مورد نظر را داخل متد Draw می نویسیم:

```
public class Rectangle : Shape
{
    public override void Draw()
    {
        Console.WriteLine("Drawing rectangle!");
    }
}
```

در صورتی که Resharper را نصب ندارید، برای پیاده سازی خودکار اعضاء abstract کلاس می توانید از قابلیت خود Visual Studio استفاده کنیـد. برای اینکـار، مکان نما را به انتهای اول تعریف کلاس برده و کلیـد های Alt+. را فشـار دهیـد. منویی به صورت زیر نمایش داده می شود:

با انتخاب گزینه Implement abstract class Shape اعضاء ای که از نوع abstract تعریف شده اند در کلاس پیاده سازی می شوند.

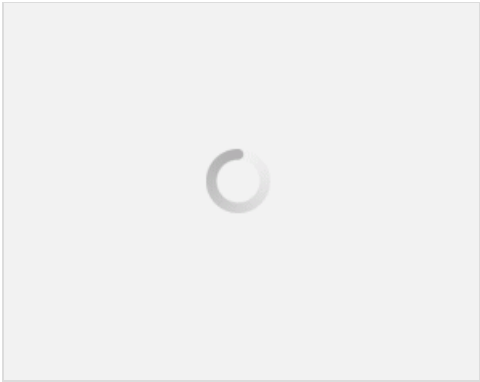
این قابلیت دقیقاً کار مشابهی با متدهای virtual انجام می دهـد، با این تفاوت که متدهای abstract بدنه ای ندارند، اما اعضاء virtual می توانند بدنه ای داشـته باشند که بوسیله کلمه کلیدی base می توان به آنها در کلاس فرزند دسترسـی داشت. همچنین توجه کنید که اعضاء abstract تنها داخل کلاس های abstract قابل تعریف هستند.

کلاس ها و اعضاء sealed

در قسـمت وراثت گفتیم که می توان زنجیره وراثت داشت. یعنی کلاس B از کلاس A و کلاس C از کلاس B مشتق شوند. اما فرض کنید بخواهیم زنجیره وراثت را در یک کلاس قطع کنیم. یعنی کلاس دیگری نتوانـد از کلاس ما ارث بری کنـد. برای اینکار کافیست کلاس مورد نظر را از نوع sealed تعریف کرد:

```
public class A
{

}

public sealed class B : A
{

}
```

در کد بالا کلاس B از نوع sealed تعریف شده، بدین معنا که هیچ کلاس دیگری نمی تواند از این کلاس مشتق شود. در تصویر زیر، کلاس C از کلاس B مشتق شده و پیغام خطا دریافت کردیم:



یکی دیگر از کاربردهـای کلمـه کلیـدی sealed جلـوگیری از override کردن یـک متـد است. کلاـس Shape و Rectangle را مثـال میزنیم. میخواهیم اگر کلاسـی از کلاس Rectangle مشـتق شد قابلیت ovverride ردن متـد Draw را نداشـته باشـد. کافیست متد Draw را از نـوع sealed تعریف کنیم:

```
public abstract class Shape
{
    public virtual void Draw()
    {
        Console.WriteLine("Drawing the shape!");
    }
}

public class Rectangle : Shape
{
    public sealed override void Draw()
    {
        Console.WriteLine("Drawing rectangle!");
```

```
    }
}
```

همانطور که در کد بالا مشاهده می کنید، متد Draw در کلاس Rectangle از نوع sealed تعریف شده. حالا اگر کلاسی تعریف کنیم و آن را کلاس Rectangle مشتق کنیم، در کلاس فرزند قابلیت تعریف مجدد متد Draw را نخواهیم داشت.تا این قسمت از آموزش با کلیات برنامه نویسی شئ گرا، مفاهیم وراثت و Polymorphism و همچنین کلاس های abstract و sealed آشنا شدیم. در بخش های بعدی با تکمیلی برنامه نویسی شئ گرا آشنا شده و مباحث Casting را با هم بررسی خواهیم کرد. **ITPRO باشید**

نویسنده : حسین احمدی

منبع : جزیره برنامه نویسی وب سایت توسینسو

سلمان بدرقه

متشکرم اقای مهندس راستی چند قسمت هست اموزش سی شارپ

حسین احمدی

تو قسمت مقدمه کلیت دوره رو نوشتم، تصمیم دارم بعد از اتمام دوره Database، آموزش ASP.NET MVC به همراه WebAPI و AngularJS رو شروع کنم. اما تا انتهای دوره بانک اطلاعاتی، فکر کنم بیشتر از ۷۰ قسمت بشه که تو سه سری جداگانه تو سایت قرار میدم:

۱. آموزش مقدماتی سی شارپ که در حال نگارش هست
۲. دوره پیشرفته
۳. برنامه نویسی بانک های اطلاعاتی

علیرضا موسوی

سلام آقای مهندس میخواستم بپرسم چرا در مورد اینترفیس ها که خیلی مهم هستند چیزی گفته نشده....

چون میبایستی مبحث اینترفیس ها بعد از وراثت گفته شود.

با عرض معذرت

حسین احمدی

دوست عزیز، فعلاً در حال به روز رسانی نرم افزار انجمن هستم، به زودی بخش های بعدی که interface ها هم جزوش هست رو شروع میکنم.

جلال مقدم

سوال : برای توسعه برنامه های سی شارپ، محیطی غیر از ویژوال استودیو وجود دارد؟ مثل نت بینز در جاوا...

محیطی سبک تر ولی با قابلیت ادیتور و کامپایل ویژوال

مطلب اصلی