

بِه نام خدا

عرفان مالکی

درس برنامه نویسی شی گرا

استاد میثاق یاریان

دانشکده شمسی پور

تمرین 2: override کردن متدها

در زبان #C و فریمورک (.NET) MVC، override کردن متدها به شما این امکان را می‌دهد که یک متد یا عملکرد را که در کلاس والدین (base class) تعریف شده است، در یک کلاس فرزند (derived class) بازنویسی (override) کنید تا رفتار آن را تغییر دهید یا به‌روزرسانی کنید. این امر به طور گسترده در توسعه وب و MVC (Model-View-Controller) در محیط .NET استفاده می‌شود.

مفهوم Override کردن متدها

Override کردن متد به این معنی است که یک متد موجود در کلاس پایه (base class) توسط یک متد با همان نام و امضا (signature) در کلاس فرزند (derived class) بازنویسی می‌شود. با این کار، متد در کلاس فرزند می‌تواند عملکرد متفاوتی نسبت به متد اصلی داشته باشد. این اصطلاح به کار می‌رود زمانی که یک کلاس فرزند نیاز دارد که رفتار یک متد از کلاس پایه را به‌روزرسانی یا تغییر دهد.

استفاده از Override در .NET

در .NET، برای override کردن یک متد، از کلمه کلیدی **override** استفاده می‌شود. این کلمه کلیدی به کامپایلر نشان می‌دهد که متد مورد نظر در کلاس فرزند، یک بازنویسی از متد مشابه در کلاس پایه است. برای استفاده از **override** کافی است متد مورد نظر را با استفاده از کلمه کلیدی **override** و درست قبل از تعریف متد در کلاس فرزند، مشخص کنید.

مثال اول Override: کردن متد در #C

```
public class Animal
{
    public virtual void MakeSound()
    {
        Console.WriteLine("Animal makes a sound");
    }
}

public class Dog : Animal
{
    public override void MakeSound()
    {
        Console.WriteLine("Dog barks");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Animal myAnimal = new Animal();
        Animal myDog = new Dog();

        myAnimal.MakeSound(); // Output: Animal makes a sound
        myDog.MakeSound();    // Output: Dog barks
    }
}
```

در این مثال، کلاس Animal یک متد Make Sound را تعریف می‌کند که به صورت مجازی (virtual) است. کلاس Dog از Animal ارث‌بری می‌کند و متد Make Sound را با استفاده از کلمه کلیدی override بازنویسی می‌کند تا عملکرد آن را به "Dog barks" تغییر دهد.

مثال دوم Override: کردن متد در ASP.NET MVC

```
public class BaseController : Controller
{
    protected override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        // عملیات قبل از اجرای هر عملیات در کنترلر اصلی
        base.OnActionExecuting(filterContext);
    }
}

public class HomeController : BaseController
{
    protected override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        // قبل از اجرای عملیات Home عملیات خاص در کنترلر
        base.OnActionExecuting(filterContext);
    }
}
```

در این مثال، متد OnActionExecuting در کلاس Base Controller از کلاس Controller ارث‌بری شده است و سپس در کنترلر Home Controller با استفاده از override بازنویسی شده است تا عملکرد قبل از اجرای عملیات‌های کنترلر Home را مشخص کند.

نکات مهم

- برای استفاده از override، متد مورد نظر در کلاس پایه باید به صورت virtual تعریف شده باشد.
 - متد در کلاس فرزند که بازنویسی می‌شود، باید امضای مشابهی (نام، نوع و تعداد پارامترها) با متد اصلی داشته باشد.
 - استفاده از override به شما امکان ایجاد چابکی و انعطاف‌پذیری در برنامه‌نویسی شیء‌گرا (OOP) را می‌دهد.
- با استفاده از این مفهوم، می‌توانید رفتار متدها را در کلاس‌های فرزند به دلخواه خود تغییر دهید و به شیوه‌ای انعطاف‌پذیر و قابل توسعه برنامه‌نویسی کنید.