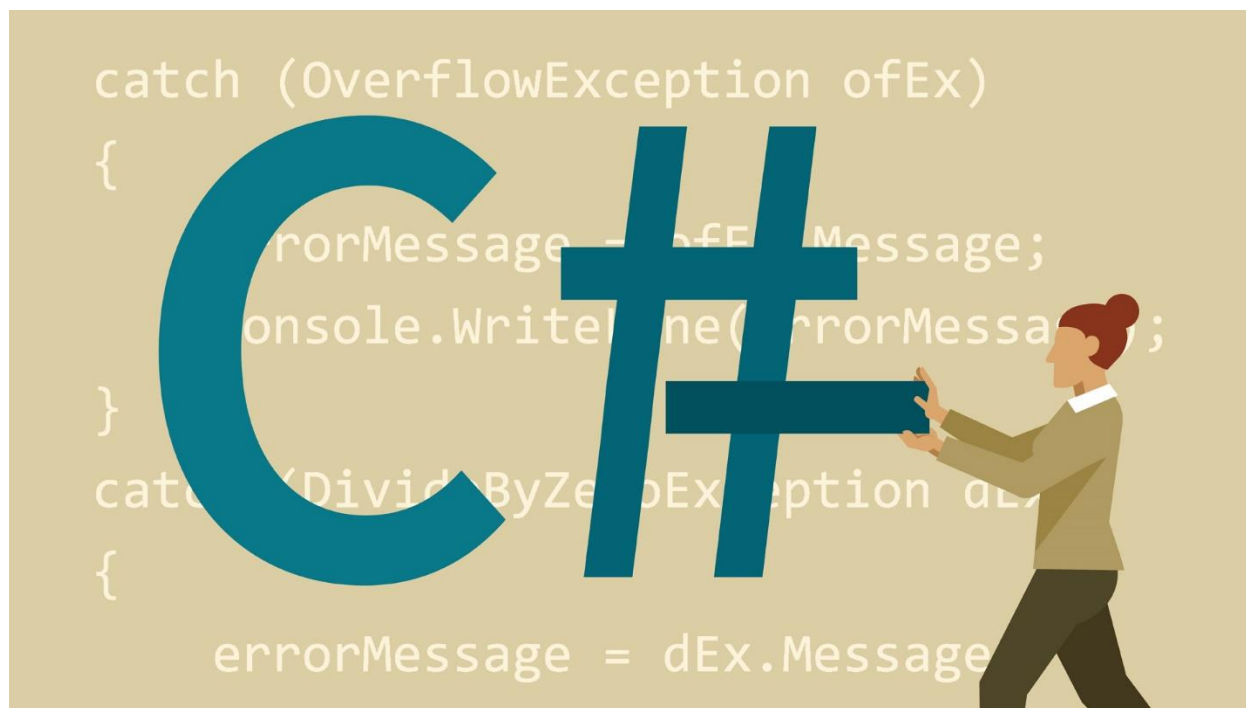


Override چیست؟

Override کردن متدها یک مفهوم کلیدی در برنامه‌نویسی شی‌گرا است. این کار به شما اجازه می‌دهد تا یک متد از کلاس پایه را با یک پیاده‌سازی جدید در کلاس فرزند جایگزین کنید. این امکان را فراهم می‌کند که رفتار یک متد را در کلاس‌های مختلف به وفور تغییر دهید بدون اینکه کدهای قبلی را تغییر دهید.

وقتی یک کلاس از یک کلاس دیگر ارث‌بری می‌کند، ممکن است متدهایی که در کلاس پایه وجود دارند، در کلاس فرزند نیاز به تغییر داشته باشند. به طور مثال، شما می‌توانید یک متدی را که در کلاس پایه تعریف شده است را با یک پیاده‌سازی متفاوت در کلاس فرزند جایگزین کنید. این کار با استفاده از کلیدواژه **override** در زبان‌های برنامه‌نویسی شی‌گرا مانند **Java** و **Python** انجام می‌شود. وقتی شما یک متد را **override** می‌کنید، شما در واقع به سیستم می‌گویید که از پیاده‌سازی جدید متد در کلاس فرزند استفاده کند، نه پیاده‌سازی اصلی آن در کلاس پایه. این به شما اجازه می‌دهد تا رفتار یک متد را در کلاس فرزند به نحوی که نیاز است، تغییر دهید بدون تغییر در کلاس پایه و اثرگذاری بر دیگر بخش‌های کد.



وقتی که یک کلاس فرزند یک متد را **override** می‌کند، این کار به ویژگی پلی‌مورفیسم معروف است. این به معنی این است که شما می‌توانید از یک نوع متد با نام و خواص یکسان در کلاس‌های مختلف استفاده کنید، اما پیاده‌سازی‌های متفاوتی داشته باشند. این ویژگی مهم از این جهت است که به شما اجازه می‌دهد که کدهای خود را ماژولارتر و قابل تعمیرتر کنید. به طور مثال، اگر یک کلاس را در زمینه پردازش تصاویر داشته باشیم و

می‌خواهیم یک متد به نام "resize" را override کنیم، می‌توانیم در کلاس فرزندی از آن کلاس برای اعمال الگوریتم resize متفاوتی استفاده کنیم، بدون اینکه کد اصلی کلاس پایه را تغییر دهیم. این به ما امکان می‌دهد که از کد پایه خود، که ممکن است در انواع مختلفی از برنامه‌ها استفاده شود، استفاده کنیم و باز هم امکان اعمال تغییرات محلی در رفتار را داشته باشیم.

نحوه استفاده از "Override"

برای مثال، فرض کنید یک کلاس Animal دارید که دارای متد Speak() است. اکنون فرض کنید که شما یک کلاس فرزند با نام Dog از کلاس Animal می‌سازید. اگر می‌خواهید که کلاس Dog به‌صورت خاص خود حرکت کند، می‌توانید متد Speak() را "Override" کنید.

```
public class Animal
{
    public virtual void Speak()
    {
        Console.WriteLine("The animal speaks.");
    }
}

public class Dog : Animal
{
    public override void Speak()
    {
        Console.WriteLine("The dog barks.");
    }
}
```

در اینجا، ما متد Speak() را در کلاس Animal با کلمه کلیدی virtual تعریف کردیم. این یعنی این متد می‌تواند در کلاس‌های فرزند "Override" شود. سپس ما در کلاس Dog متد Speak() را با کلمه کلیدی override تغییر دادیم.

چرا باید "Override" را یاد بگیریم؟

توجه داشته باشید که در دنیای برنامه‌نویسی، زمان بسیار ارزشمند است. هرچقدر بتوانید کد خود را به‌صورت کارآمدتر بنویسید، زمان بیشتری برای حل مشکلات دیگر خواهید داشت. "Override" یکی از این ابزارهای کارآمد در سی شارپ است.

کاربردهای override در برنامه‌نویسی :

1. **پلی‌مورفیسم**: این قابلیت از `override` می‌تواند به شما کمک کند تا از یک متد با نام و خواص یکسان در کلاس‌های مختلف استفاده کنید، اما پیاده‌سازی‌های متفاوتی داشته باشید، به این ترتیب که بسته به نوع شیء مراجعه‌کننده، پیاده‌سازی مناسب اجرا می‌شود

2. **گسترش و ارتقاء قابلیت‌ها**: با استفاده از `override`، می‌توانید عملکرد یک متد را در یک کلاس فرزند بهبود دهید یا آن را به روز رسانی کنید بدون اینکه نیاز به تغییر کد در کلاس پایه باشد. این امر امکان پذیر است زیرا شما قادرید تنها پیاده‌سازی متد در کلاس فرزند را تغییر دهید

3. **سفارشی‌سازی**: با `override`، می‌توانید رفتار یک متد را در کلاس فرزند به نحوی که نیاز است، تغییر دهید. این امکان به شما می‌دهد تا رفتار کلاس‌ها را در هر سناریوی خاص سفارشی‌سازی کنید بدون تغییر در کلاس‌های پایه.

جمع‌بندی:

با یادگیری "Override" در سی‌شارپ، شما می‌توانید برنامه‌نویسی کلاس‌بندی شده خود را به سطح بعدی برسانید. این فقط یکی از تکنیک‌های بسیاری است که می‌تواند به شما کمک کند تا به یک برنامه‌نویس سی‌شارپ حرفه‌ای تبدیل شوید. با مطالعه بیشتر و تمرین، شما قادر خواهید بود که این مهارت را به سطح بالاتری برسانید.

منابع: [chatgpt](https://chatgpt.com) و bugeto.net