

## به نام خدا

برنامه نویسی شی گرا

مرتضی صداقت یار

استاد: میثاق یاریان

شماره دانشجویی: ۰۱۲۰۰۳۳۷۲۰۰۲۳

### بازنویسی یا Override کردن متدها در برنامه نویسی شیءگرا

فرض کنید در برنامه نویسی شی گرا، کلاسی به نام Animal داریم که متدی به نام makeSound() برای تولید صدای هر حیوان تعریف می کند.

حالا فرض کنید دو کلاس Dog و Cat از کلاس Animal ارث بری می کنند. هر کدام از این حیوانات صدای مخصوص خود را دارند، بنابراین می خواهیم متد makeSound() را در هر کدام از کلاس های Dog و Cat به گونه ای بازنویسی (Override) کنیم که صدای مخصوص آن حیوان را تولید کند.

به این کار، بازنویسی متد یا Override کردن متد می گویند.

"در واقع، در بازنویسی متد، ما متدی با نام و امضای مشابه متدی که در کلاس پایه تعریف شده است، در کلاس مشتق (فرزندی) تعریف می کنیم و عملکرد آن را مطابق با نیاز خودمان تغییر می دهیم"

### مثال:

فرض کنید در یک بازی، شخصیت های مختلفی مانند انسان، حیوان و ... وجود دارند. همه این شخصیت ها می توانند صحبت کنند، اما نوع صحبت کردن آنها (مثلا گفتن کلمات، غرش کردن، و...) با هم فرق دارد.

با استفاده از بازنویسی متد، می توانیم برای هر شخصیت، متد speak() را به گونه ای بازنویسی کنیم که نوع صحبت مخصوص آن شخصیت را پیاده سازی کند.

### نکات مهم در مورد بازنویسی متدها:

۱. نام و امضای متد در کلاس پایه و کلاس مشتق باید دقیقاً یکسان باشد.
۲. نوع بازگشتی متد در کلاس مشتق می‌تواند با نوع بازگشتی متد در کلاس پایه مشابه یا مشتق شده از آن باشد.
۳. سطوح دسترسی متد در کلاس مشتق می‌تواند وسیع‌تر از سطح دسترسی متد در کلاس پایه باشد، اما نمی‌تواند محدودتر باشد.
۴. متدهای static و final قابل بازنویسی نیستند.
۵. اگر در کلاسی متدی را به صورت abstract تعریف کنیم، حتماً باید در یکی از کلاس‌های مشتق آن را بازنویسی کنیم.

### مزایای بازنویسی متدها:

۱. ایجاد انعطاف‌پذیری در برنامه: با بازنویسی متدها، می‌توانیم رفتار کلاس‌های مشتق را بدون نیاز به تغییر در کلاس پایه، سفارشی‌سازی کنیم.
۲. ترویج چندریختی: بازنویسی متدها یکی از پایه‌های اصلی چندریختی در برنامه‌نویسی شیء‌گرا است. به این معنی که می‌توانیم از اشیاء کلاس‌های مختلف، بدون دانستن نوع دقیق آنها، استفاده کنیم و متدهای آنها را فراخوانی کنیم.