

## کلاس‌های انتزاعی (Abstract Classes)

کلاس انتزاعی به عنوان یک الگو یا طرح کلی برای کلاس‌های مشتق شده عمل می‌کند. این کلاس‌ها نمی‌توانند نمونه‌سازی شوند و حداقل یک متد انتزاعی (بدون پیاده‌سازی) دارند. متدهای انتزاعی در کلاس انتزاعی تعریف می‌شوند اما پیاده‌سازی آن‌ها در کلاس‌های مشتق شده الزامی است.

مثال:

```
public abstract class Animal  
  
    }  
  
    public abstract void MakeSound()  
  
    {
```

کلاس‌های مهر و موم شده (Sealed Classes)

کلاس مهر و موم شده کلاسی است که نمی‌توان از آن کلاس دیگری مشتق کرد. این ویژگی برای جلوگیری از تغییرات ناخواسته در رفتار یک کلاس استفاده می‌شود.

مثال:

```
public sealed class Dog : Animal  
  
}
```

```
()public override void MakeSound  
  
}
```

```
;Console.WriteLine("Woof!")
```

```
{
```

```
{
```

## کلاس‌های جزئی (Partial Classes)

کلاس جزئی کلاسی است که تعریف آن در چندین فایل مختلف تقسیم می‌شود. این ویژگی برای مدیریت بهتر کدهای بزرگ و پیچیده و همچنین برای ایجاد کتابخانه‌های قابل گسترش استفاده می‌شود.

مثال:

Part1.cs //

```
public partial class Person  
  
}
```

```
public string FirstName { get; set; }
```

```
public string LastName { get; set; }
```

```
{
```

Part2.cs //

public partial class Person

}

public int Age { get; set; }

{

چندریختی (Polymorphism)

چندریختی به توانایی یک متد یا عملگر برای رفتار متفاوت بر اساس نوع داده‌ای که روی آن اعمال می‌شود، گفته می‌شود.

\* چندریختی زمانی: هنگامی رخ می‌دهد که یک متد در کلاس‌های مشتق شده بازنویسی شود.

\* چندریختی چند شکلی: هنگامی رخ می‌دهد که یک متغیر از نوع کلاس پایه بتواند اشیاء از انواع مشتق شده را نیز نگه دارد.

مثال:

```
;()Animal animal = new Dog
```

```
!Woof ;()animal.MakeSound // خروجی:
```

بازنویسی (Overriding)

بازنویسی به معنای پیاده‌سازی مجدد یک متد در کلاس مشتق شده با همان نام و پارامترها است.

مثال:

```
public class Cat : Animal
```

```
()public override void MakeSound  
  
}
```

```
;Console.WriteLine("Meow!")
```

```
{
```

```
{
```

آرایه‌ها (Arrays)

آرایه مجموعه‌ای از متغیرهای هم‌نوع است که با استفاده از یک اندیس به عناصر آن دسترسی پیدا می‌کنیم.

مثال:

```
int[] numbers = { 1, 2, 3, 4, 5 };
```