

# به نام خدا

## چند مورد از مفاهیم برنامه نویسی

### 1. Abstract Class :

کلاس انتزاعی کلاسی است که نمی توان از آن شیء ساخت و معمولاً شامل متدهای انتزاعی (بدون پیاده سازی) است که در کلاس های فرزند پیاده سازی می شوند.

مثال:

```
abstract class Animal {  
|   public abstract void Sound();  
|}  
  
class Dog : Animal {  
|   public override void Sound() {  
|       Console.WriteLine("Woof Woof");  
|   }  
|}  
  
class Cat : Animal {  
|   public override void Sound() {  
|       Console.WriteLine("Meow Meow");  
|   }  
|}
```

در این مثال، کلاس Animal یک کلاس انتزاعی است که متد Sound را تعریف می کند. هر کدام از کلاس های فرزند (Dog) و (Cat) این متد را به شکل خاص خودشان پیاده سازی می کنند.

### 2.Sealed Class :

کلاس مهر و موم شده کلاسی است که نمی‌توان از آن ارث‌بری کرد. این کار برای جلوگیری از توسعه بیشتر کلاس و تغییر رفتار آن استفاده می‌شود.

مثال:

```
sealed class Car {  
    public void Drive() {  
        Console.WriteLine("Driving a car");  
    }  
}  
  
// این کد خطا می‌دهد:  
// class SportsCar : Car { }
```

در این مثال، Car یک کلاس مهر و موم شده است و اجازه نمی‌دهد که کلاس دیگری از آن ارث‌بری کند.

### 3. Partial Class :

کلاس جزئی به برنامه‌نویسان این امکان را می‌دهد که یک کلاس را در چند فایل تقسیم کنند. این ویژگی در شرایطی که کلاس بزرگ است یا چندین نفر روی آن کار می‌کنند مفید است.

مثال:

```
// File1.cs  
partial class Person {  
    public string Name { get; set; }  
}  
  
// File2.cs  
partial class Person {  
    public int Age { get; set; }  
}
```

در این مثال، کلاس Person در دو فایل تعریف شده است و ویژگی‌های Name و Age در فایل‌های جداگانه قرار دارند.

## 4. Polymorphism :

چندریختی به مفهوم توانایی یک متد برای ارائه رفتارهای مختلف در کلاس‌های فرزند اشاره دارد. متد می‌تواند با توجه به نوع شیء رفتار متفاوتی داشته باشد.

مثال:

```
class Shape {  
    public virtual void Draw() {  
        Console.WriteLine("Drawing a shape");  
    }  
}  
  
class Circle : Shape {  
    public override void Draw() {  
        Console.WriteLine("Drawing a circle");  
    }  
}  
  
class Square : Shape {  
    public override void Draw() {  
        Console.WriteLine("Drawing a square");  
    }  
}
```

در این مثال، هر کدام از کلاس‌های Circle و Square متد Draw را به شکل خاص خودشان پیاده‌سازی کرده‌اند.

## 5. Overriding :

بازنویسی به معنی ارائه پیاده‌سازی جدیدی برای متدی است که از کلاس پایه به ارث برده شده است. این کار برای تغییر رفتار متد در کلاس فرزند استفاده می‌شود.

مثال:

```

class Parent {
    public virtual void Greet() {
        Console.WriteLine("Hello from Parent");
    }
}

class Child : Parent {
    public override void Greet() {
        Console.WriteLine("Hello from Child");
    }
}

```

در این مثال، کلاس Child متد Greet را از کلاس Parent بازنویسی کرده است.

## 6. Array :

آرایه یک ساختار داده است که مجموعه‌ای از عناصر با نوع داده مشابه را نگهداری می‌کند و هر عنصر آن با یک ایندکس دسترسی پذیر است.

مثال:

```

int[] numbers = {1, 2, 3, 4, 5};
for (int i = 0; i < numbers.Length; i++) {
    Console.WriteLine(numbers[i]);
}

```

در این مثال، یک آرایه از اعداد صحیح تعریف شده است که با استفاده از حلقه for به عناصر آن دسترسی پیدا می‌کنیم.