**Reflection Nedir?**

Reflection, bir sınıfın elemanlarına referans, nesne ya da sınıf ismi üzerinden erişmektense programın çalışma anında (runtime) nesnelerin özelliklerine, tür bilgisine, metodlarına ve diğer özelliklerine erişimemizi sağlar. Nesneleri manipüle edebilmemize ve dinamik olarak yönlendirebilmemize olanak tanır.  Yani, bir program çalışırken, kodu yazdıktan sonra bile o kodun kendisini analiz edebiliriz.

**Reflection Ne İşe Yarar?**

1. Tip bilgisine erişim (Type info)  
   Bir nesnenin hangi sınıfa ait olduğunu, hangi metodları, property’leri, field’ları olduğunu çalışma zamanında öğrenmeni sağlar.
2. Çalışma zamanında nesne oluşturma  
   Kodda sınıfın adını string olarak vererek o sınıftan nesne oluşturabilirsin.
3. Metot çağırma  
   Bir nesne üzerinde, metot ismini string olarak verip o metodu çalıştırabilirsin.
4. Özellik ya da alanlara erişme  
   Bir nesnenin içindeki property veya field'ların değerini alabilir ya da değiştirebilirsin.
5. Özellikle plugin yapılarında kullanılır  
   Hangi sınıfların eklendiği bilinmeyen durumlarda, uygulamaya sonradan eklenen kodlarla etkileşim kurmak için çok kullanışlıdır.

**Ne zaman kullanılır?**

* Dinamik olarak nesnelerle işlem yapmak istendiğinde
* Bilinmeyen türlerde (örneğin plugin'ler) işlem yapılmak istendiğinde
* Json veya XML gibi veri kaynaklarından gelen bilgilerin nesnelere map'lenmesi gibi işlemlerde
* Unit test ve otomasyon sistemlerinde

| **Reflection Kullanımı** | **Açıklama** |
| --- | --- |
| GetType() | Nesnenin türünü çalışma zamanında almak |
| GetProperties() | Sınıfın özelliklerini listelemek |
| prop.GetValue(obj) | Özelliğin değerini nesne üzerinden okumak |
|  |  |

**Örnek Kod :**

using System;

using System.Reflection;

class Ogrenci

{

public string Ad { get; set; }

public string Soyad { get; set; }

public int Yas { get; set; }

}

class Program

{

static void Main()

{

Ogrenci ogrenci = new Ogrenci

{

Ad = "Şevval",

Soyad = "Bakırhan",

Yas = 21

};

// Tip bilgisini al

Type tip = ogrenci.GetType();

// Özellikleri döngüyle yazdır

foreach (PropertyInfo prop in tip.GetProperties())

{

// Özelliğin adını ve değerini al

string isim = prop.Name;

object deger = prop.GetValue(ogrenci);

Console.WriteLine($"{isim}: {deger}");

}

}

}