Polimorfizm (Çok biçimlilik) Nedir?(C#) Örnekli Açıklama

Süleyman Meral
3 min read · Just now

Polimorfizm , nesne yönelimli programlamada (OOP) bir nesnenin farklı şekillerde davranabilmesini sağlayan bir özelliktir. Polimorfizm, genellikle **metot geçersiz kılma (override)** ve **metot aşırı yükleme (overload)** gibi tekniklerle uygulanır.

Bu 2 tekniği örneklerle uygulayalım. Önceki yazılarımda olduğu gibi Animal yapıs üzerinden devam edeceğim. Öncellikle Animal adında düz bir class oluşturuyoruz.

```
public class Animal
{
    public virtual void AnimalSound()
    {
        Console.WriteLine("Animal makes sound");

    }
}
```

İçerisinde virtual türünde AnimalSound adında bir metot tanımladık.

Virtual keywordu ile tanımlamamızın sebebi metotun Animal classından türetilebilecek olan classlar tarafından override edilmesini sağlamaktır.

```
public class Wolf :Animal
{
    public override void AnimalSound()
```

```
        {
            Console.WriteLine("Auu");
        }
    }
```

Yukarıdaki kodda gördüğümüz üzere AnimalSound metodunun gövdesini değiştirip override ettik. Wolf classı Animal classından türetilmiş. Fakat burada Abstract class veya interface yapısı gibi bir implement zorunluluğu bulunmuyor. İlgili class dilerse metodu override edebilir.

```
using polimorfizm;

Animal animal = new Animal();
Wolf wolf = new Wolf();
animal.AnimalSound();
wolf.AnimalSound();
```

Gördüğümüz gibi animal classı abstract yapıda olmadığı için burada obje türetme işlemi yapabiliyoruz. Şimdi metotların çıktılarını inceleyelim.

```
Animal makes sound
Auu
```

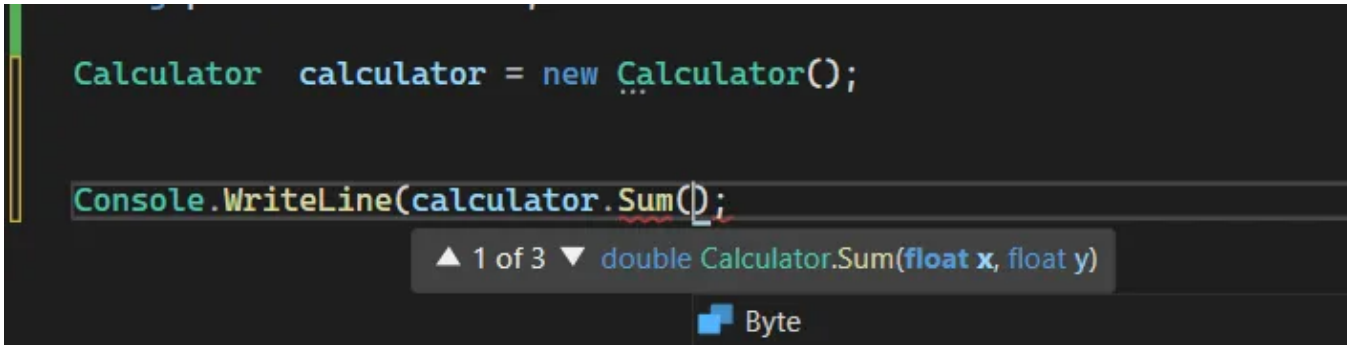
Wolf classından metodumuzu override etmeseydik Animal classında olduğu gibi aynı çıktıyı alabilirdik. Virtual keywordu ile metodu isteğe bağlı override edilmesini sağladık.

Aynı zamanda polimorfizm aynı isimde fakat farklı türde parametre alan metodları tanımamıza olanak sağlar.

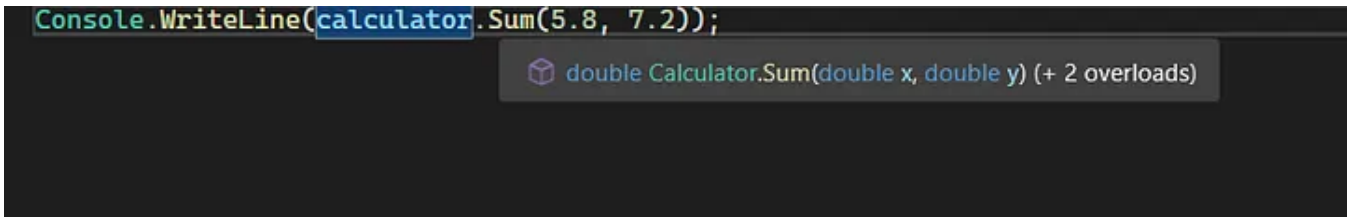
```
public class Calculator
{
    public int Sum(int x, int y) { return x + y; }
    public int Sum(int x, int y,int z) { return x + y + z; }

    public double Sum(float x, float y) { return x + y; }
```

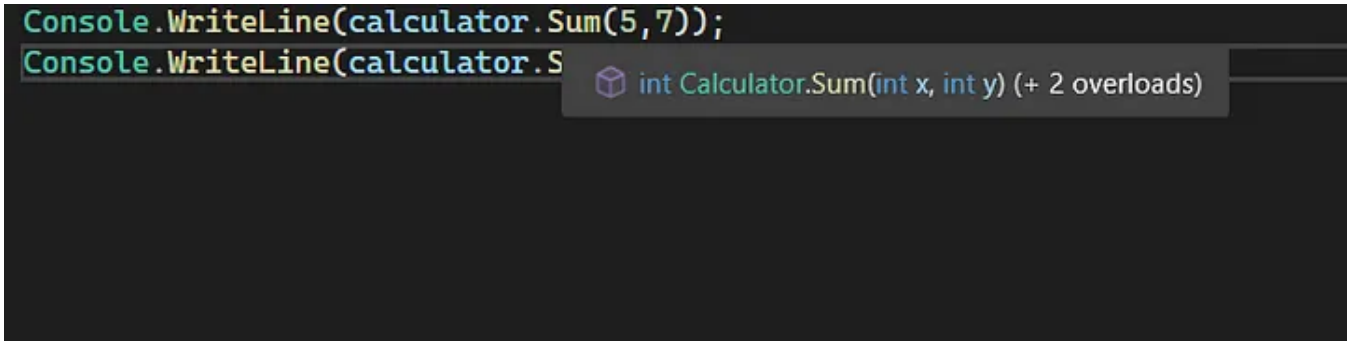
```
    }
```

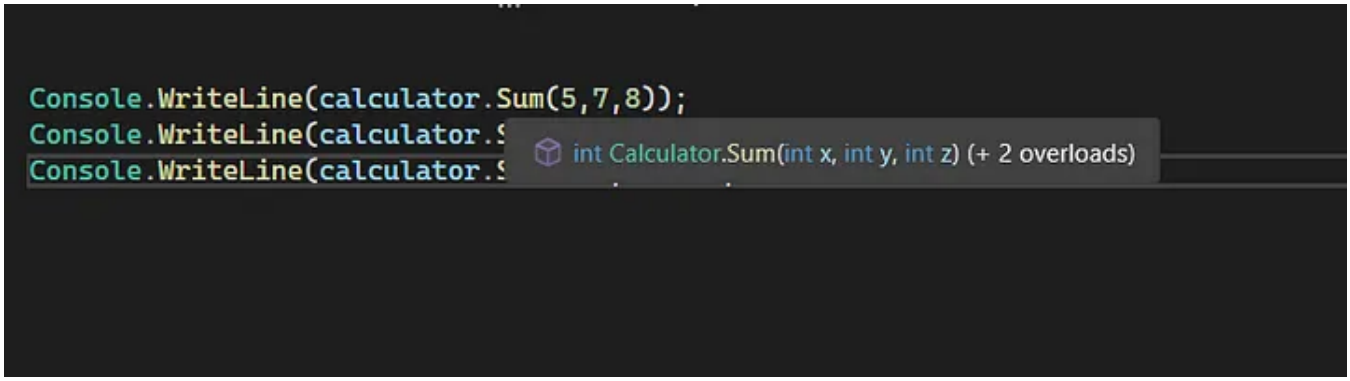
Aynı isimde fakat parametre sayısı veya parametre türü farklı olan metotlar tanımladık.



Görüldüğü üzere Sum metodunu çağırdığımızda 3 ayrı tür gözüküyor. İlgili parametrelere göre ilgili metodumuz çalışacak.







Girilen parametrelere göre otomatik olarak o parametreleri içeren fonksiyonlar çalışmakta.

Bu sayede kod tekrarını azalttık, yeni türde metot tanımlamayı kolaylaştırdık ve yönetilmesi daha kolay bir kod yazmış olduk.

Polimorfizm, nesne yönelimli programlamada önemli bir prensiptir ve birçok dilde kullanılır (C#, Java). Özellikle **bağımlılığı azaltmak ve genişletilebilir kod yazmak** için sıkça tercih edilir.

Software Development | Software Engineering | Oop

Edit profile

Written by Süleyman Meral

0 Followers · 1 Following

Software Engineering Student/ .Net Developer

No responses yet

Süleyman Meral

What are your thoughts?