

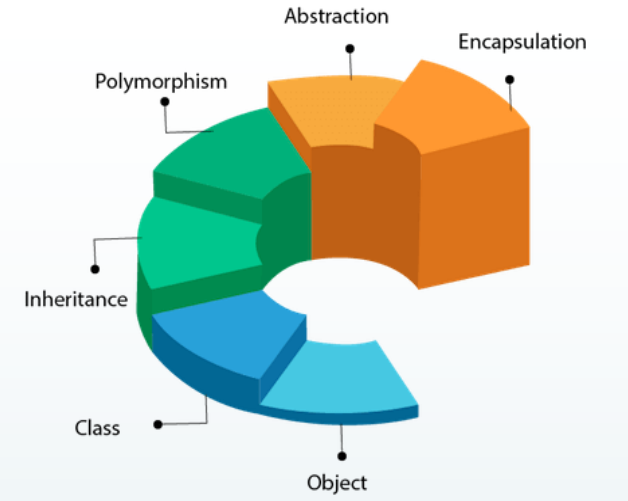


www.youtube.com/BMderslerim

MAKÜ

BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ

Nesneye Yönelik Programlama



C#'da Abstraction (Soyutlama)

1



Hazırlayan	: Erkut Uysal 2011404017
Tarih	: 31/05/2022
Sürüm	: v1
Ders Yürütücüsü	: Doç. Dr. İsmail KIRBAŞ

İÇİNDEKİLER

C# nesne tabanlı programlama nedir?

C# Abstract Class Nedir?

C# Abstract Metot Nedir?

Neden soyut sınıf(abstract class) ve metotlarını kullanırız?

C# Abstract Class Kullanımı



C# nesne tabanlı programlama nedir?

Nesne tabanlı programlama (OOP); yazılım tasarımını işlevler ve mantık yerine, veri veya **nesneler** etrafında düzenleyen bir **programlama** dili modelidir. Günümüzde yazılım geliştirme teknolojileri içerisinde büyük bir önem taşır.



C# Abstract Class Nedir?

Abstract Class, Nesneye yönelik programlamanın içerisindeki önemli kavramlardan bir tanesidir. Programlarımız içerisinde bulunan faaliyet öğelerini gizleyerek sadece sınıfın sahip olduğu işlevleri gösterir. Bu durum projemize esneklik, güvenlik ve hızlı kontrol yapısı sağlar.

C# Abstract Class Nedir sorusunu toparlayacak olursak, **Abstract (Soyut)** sınıfları kullanarak, **nesne** üretirken bir şablon oluşturmak istememizdir diyebiliriz. Yani bir sınıf üzerinde bulunan özellikleri kullanarak yeni bir sınıf türetmek istediğimiz zaman, içerisinde bulunan bazı temel yapıları yeniden oluşturmak için **override** (aşırı yükleme) işlemini uygular ve sınıf yapılarına göre özel bir hale getiririz.



C# Abstract Metot Nedir?

Sadece abstract class içerisinde kullanılır. Metot içerisine herhangi bir şey yazılmaz. Bir **abstract class** hem **abstract metotlara** hem de normal metotlara sahip olabilir.

Bir örnek senaryo üzerinden Abstract yapısını ele alırsak aklımızda daha sağlıklı ve sağlam bir şekilde yer edeceğini düşünmekteyim. Dünya'nın en büyük akıllı telefon üreten markalarını düşünelim. Hepsinin üretim aşamaları aynı şekilde ilerlemesine rağmen, telefon üzerinde kullandıkları mimari yapısı ve bileşenler aralarındaki farkları oluşturur. Bu senaryo içerisinde Akıllı telefonlar **abstract sınıfları**, akıllı telefon üretirken kullanılan yazılım, ekran, diğer mimari yapılar ise **abstract metotları** ve son olarak markalarda Abstract sınıfımızdan türetilen ana sınıflarımızı simgeler.



Neden soyut sınıf(**abstract class**) ve metotlarını kullanırız?

Soyut sınıf(abstract class**) ve metotları** işleri parçalara ayırıp öyle hareket etmemize olanak sağlayan bir teknolojidir. OOP özelliklerinden bir tanesi de defalarca kullanacağımız kod bloklarını bir kere yazıp, ihtiyacımız olduğunda tekrar tekrar çağırmaktır. Böylelikle kod ve zaman tasarrufu sağlanmış olur.



C# Abstract Class Kullanımı

C# Abstract Class Nedir sorusuna cevap bulduğumuza göre artık bu sınıf yapısını projelerimizde nasıl kullanacağımızı, kullanırken nelere dikkat etmemiz gerektiğini öğrenmeye başlayabiliriz.

Öncelikle bir **soyut sınıf** oluştururken dikkat etmemiz gereken noktalara birlikte göz atalım:



- Bir abstract (soyut) sınıf oluşturuyorsak sınıf içerisinde muhakkak bir abstract metot, eğer bir abstract metot tanımlıyorsak muhakkak sınıf yapısının da abstract class olması gerekmektedir.
- Abstract sınıf kullanılarak türetilen sınıf içerisindeki abstract metot yapısı override edilerek içerisinde doldurulması gerekir.

Dikkat etmemiz gereken ve uygulamamız gereken konular hakkında bir fikrimiz olduğuna göre bir soyut sınıf oluşturmamız için gereken kullanım şablonunu inceleyebiliriz.

```
1. namespace AbstractDemo
2. {
3.     abstract class AbstractClassManager //---> Abstract Class Tanımlaması.
4.     {
5.         public abstract void abstractMethod(); //---> Abstract Metot
        Tanımlaması.
6.     }
7. }
```



Abstract sınıfımızı şablondaki gibi tanımladıktan sonra bu sınıfımızdan bir alt sınıf türeterek, abstract metodumuzu yeniden oluşturmamız gerekiyor. Bunu aklımızda kalacak basit iki örnek üzerinde işleyerek göstermek istiyorum.



1. örneğimizde soyut bir **MobilePhone** sınıfı oluşturulmuş bulunmaktadır. Bu **abstract** sınıfa ait iki adet metod bulunmaktadır. Bunlar arama yapmak için kullanılan, **Calling()** ve mesaj gönderebilmek için kullanılan **SendSMS()** metodlarıdır. Ardından cep telefonu modellerini içeren sınıflar bu **abstract** sınıftan türetilmiş bulunmaktadır. Yeni türetilen modele ait sınıflar **abstract class**'ı kullanmaktadır. Çünkü tüm modeller için, **abstract** sınıftaki arama yapmak ve mesaj göndermek temel fonksiyonlardır.

```
1 abstract class MobilePhone {  
2     public void Calling();  
3     public void SendSMS();  
4 }  
5 public class Samsung: MobilePhone {}  
6 public class Nokia2700: MobilePhone {  
7     public void FMRadio();  
8     public void MP3();  
9     public void Camera();  
10 }  
11 public class BlackBerry: MobilePhone {  
12     public void FMRadio();  
13     public void MP3();  
14     public void Camera();  
15     public void Recording();  
16     public void ReadAndSendEmails();  
17 }
```



2. örneğimizde ise mobilya isimli bir abstract sınıf ve bu sınıfta ozellikyaz() isimli bir metod bulunmakta. Bu sınıftan kanepe ve masa isimli mirasçı sınıf oluşturarak override edelim.

Class yapımızı aşağıdaki şekilde oluşturuyoruz.



```
1
2 abstract class mobilya
3 {
4     public string renk;
5     abstract public void ozellikyaz();
6 }
7
8 class kanepe : mobilya
9 {
10     public string kumas;
11     public override void ozellikyaz()
12     {
13         Console.WriteLine("Kanepenin Özellikleri");
14         Console.WriteLine("Renk: {0}", renk);
15         Console.WriteLine("Kumaş: {0}", kumas);
16     }
17 }
18
19 class masa : mobilya
20 {
21     public string malzeme;
22     public override void ozellikyaz()
23     {
24         Console.WriteLine("Masanın Özellikleri");
25         Console.WriteLine("Renk: {0}", renk);
26         Console.WriteLine("Malzeme: {0}", malzeme);
27     }
28 }
29
```



Anaprogramımız ise aşağıdaki gibi olacaktır.

```
1
2 class Program
3     {
4         static void Main(string[] args)
5         {
6             kanepe knp1 = new kanepe();
7             masa calisma_masasi = new masa();
8             knp1.renk = "Siyah";
9             knp1.kumas = "Deri";
10            calisma_masasi.renk = "Sarı";
11            calisma_masasi.malzeme = "Ahşap";
12            knp1.ozellikyaz();
13            calisma_masasi.ozellikyaz();
14            Console.ReadKey();
15        }
16    }
17
```



Uygulamamızı çalıştırdığımızda ekran görüntüsü;

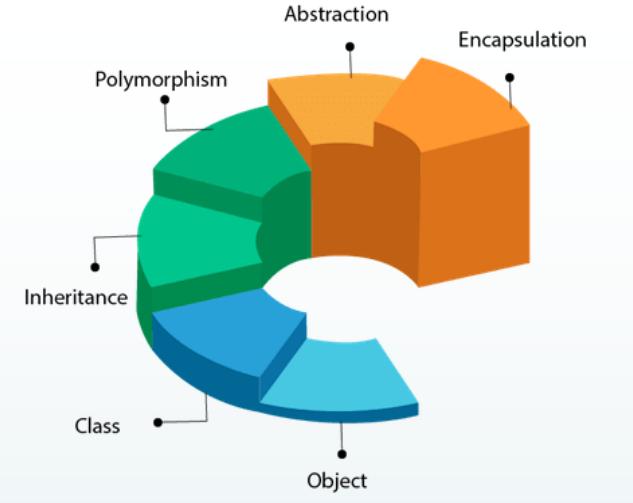
```
Kanepenin özellikleri  
Renk: Siyah  
Kumaş: Deri  
Masanın özellikleri  
Renk: Sarı  
Malzeme: Ahşap  
-
```



Yardımcı Kaynaklar

1. <https://emrecelen.com.tr/c-sharp-abstract-class-nedir/#close>
2. <https://www.yazilimkodlama.com/programlama/c-abstract-siniflar-ornek-uygulama/>
3. <https://cevapo.biz.tr/c-nesne-tabanl%C4%B1-programlama-nedir/>
4. <https://medium.com/@srhtayata/c-abstraction-soyutlama-bb1a8f3ddddd6>





İlginiz için teşekkürler...



Hazırlayan
E-posta

: **Erkut Uysal 2011404017**
: **erkut_uysal_2001@hotmail.com**

Tarih

: 31/05/2022

Ders Yürütücüsü

: Doç. Dr. İsmail KIRBAŞ