

Interfaceler

<https://medium.com/@m.baybars.keskin.0/interfaceler-f72c3083116e>

Class nedir ve ne işe yarar?

Class OOP temel taşı olup tip (type) tanımlamak için kullanılır. Class, oluşturulacak objelerin hangi özelliklere ve hangi metotlara sahip olduğunu ve bu metotların nasıl çalışacaklarını tanımladığımız yerdir. Bir class sadece ve sadece bir class'a extends edebilir. (extends kelimesi çok pahalı bir kelimedir. Sadece bir class için kullanılabilir.) Bir class birçok interface'i implement edebilir.

Interface

Interface'ler: uygulamalarda bağımlılığı yönettiğimiz yapılardan bir tanesidir. Nesne tabanlı programlamanın yapı taşları: Class, Abstract Class ve Interface'lerden meydana gelmektedir. Interface'ler bir class kabul edilmez ancak onlarda normal class veya abstract class gibi referans tutmaktadırlar.

Interface, Türkçe anlamı “Arayüz” olan ve diğer class'lara rehberlik görevi üstlenen bir yapıdır. Burada rehberlik kavramına incelemek gerekirse, yani gerçekten yaptığı iş sadece diğer class'lara yön göstermektir. Bu yüzden Interface tanımları içerisinde kesinlikle kod bloğu bulunmaz.

Polimorfizm

Polimorfizm bir objenin birden fazla tipte(type) bulunabilmesidir. Tip dönüşümleri casting ile yapılır. Bir obje o anda bulunduğu tipin özelliklerini ve metotlarını kullanabilir. Tipler (type) class ve interface kullanılarak tanımlanır.

Interface'lerin Özellikleri:

- Interface'lerde abstract metotlar gibi bir durum söz konusudur. Tamamlanmamış operasyonlar içerirler. Class'larda metotlar tamamlanmış olmalıdır. Abstract Class'larda metotlar boş olabilirdi onu inherit eden (kullanmak isteyen) içini doldurmak zorundaydı.
- Interface'ler inherit edilemez implemente edilir.
- Bir class birden çok interface'i implemente edebilir.
- Bir interface'in üyeleri default (varsayılan) olarak public'tir
- Interface'ler new'lenemez. İmza taşırlar.
- Interfaceler referans tutar bunu asla unutmuyoruz!

Eğer hem inherit hem de impelente varsa syntax (dilin yazım kuralları) şu şekilde olur:

```
public class ClassAdi1 extends ClassAdi2 implements InterfaceAdi
```

dal = data access layer --> veri erişimi ile ilgili kodlar buraya yazılır

Katmanlar

[Data Access - Dal]

[İş - Manager]

[Arayüz - Swing, JavaFx, HTML]

Katmanların geçişleri genellikle interface'ler ile yapılır, bunun nedeni bağımlılıkları önlemek içindir.

Önemli Not: Bu yazıdaki örnekler ve anlatımlar **Engin Demiroğ'a** aittir, vermiş olduğu eğitimler ile biz öğrencilere tüm katkılarından dolayı minnettarız. Dilerseniz https://www.youtube.com/watch?v=Cn2P_r_oLmk adresindeki videodan, öğretmenin kendi anlatımı ile konuyu dinleyebilir ve pekiştirebilirsiniz.

Kaynaklar:

<https://www.mshowto.org/c-ile-interface-kullanimi-arayuz-nedir-nasil-kullanilir-ornek-anlatimli.html>

<https://medium.com/codable/interfacelerin-mantigi-nedir-1-hikayeli-f9b960228328>