

INTERFACE(ARAYÜZ)

Interface kelimesi ile tanımlanması yapılır. Arayüz iki şeyin birbiri ile temas ettiği, karşı karşıya geldiği, iletişimde bulunduğu ortama interface(arayüz) denir. [1]

Dili düşünelim. İnsanların arayüzüdür. Dil, mimik, beden dili insanlar için bir arayüzdür. Yazılım nesneleri birbirleri ile arayüz üzerinden haberleşme yaparlar. [1]

3 farklı arayüz java da kullanımı vardır:

- i) Method arayüz:İsmi,parameter listesi,return tipi, exceptionlar, method tipi vardı. [1]
- ii) Sınıf Arayüzü: Enum ve sınıf tipleri de vardır. Sahip olduğu methodların bütününe sınıf arayüzü denir. [1]
- iii) Interface ile tanımlanır. [1]

Method arayüzü, o methodun ne yaptığını ifade eden bilgisidir. Arayüz methodun nasıl yaptığını gerçekleştirme ile birlikte iki parçasından birdir. Sınıfın sahip olduğu method arayüzlerin tamamına o sınıfın arayüzü denir. Bir sınıfın arayüzü aynı zamanda o sınıftan üretilen nesnelerinin de arayüzüdür. [1]

Interface: Tüm methodları soyut olan sınıflardır. Abstract sınıflardan 1 tanesi abstract ise bu sınıflar abstract sınıf olmak zorundadır. İçinde hiç abstract method olmadan da abstract sınıf olarak niteleyebiliriz. Bunun nedeni nesne oluşturulmasını engellemektir. Hepsini soyut yaparsak interface olur. Abstract sınıfların hepsi soyut method olmayabilir. Bir kaç tanesi implement edilmiş method olabilir. [1]

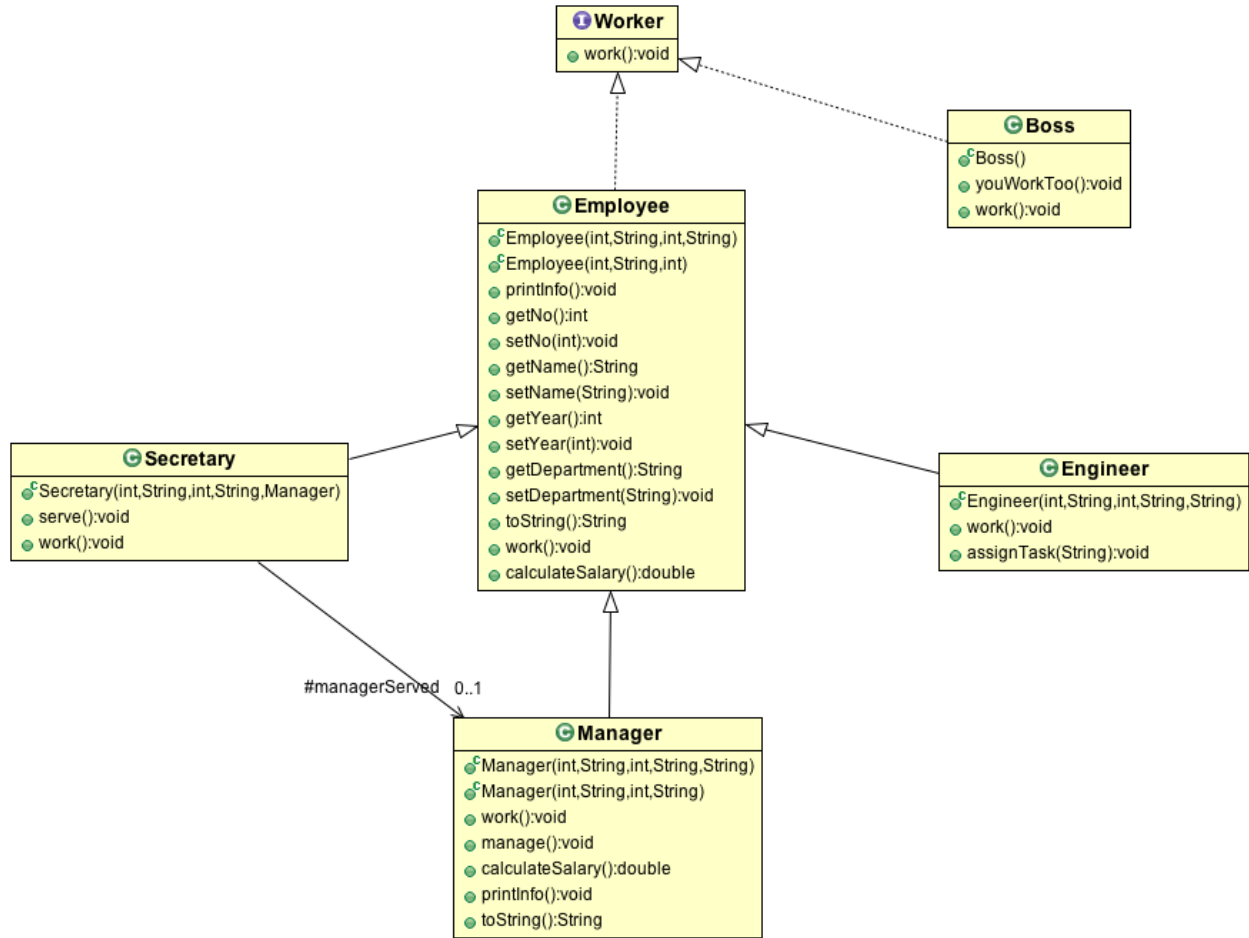
Alt sınıflar, o somut sınıfları da devralıp, ister devralıp ister devralmazlar. Ama alt sınıflar soyut methodların arayüzler soyut methodların arayüzlerini devralıp onları da implement etmek zorundadır. O yüzden arayüz diyoruz. Varsayılan olarak arayüzler public ve abstract olurlar. Default yapıda bu şekilde gelirler. [1]

Arayüz sadece arayüz sağlarlar. Sınıflar durum ve gerçekleştirme sağlarlar. Abstract sınıflara geçtiğimizde bazı durumlarda state ve field sağlarken bazen sağlamayabilir. Bu durum Java 8 kadar bu durum(interface) arayüzlerde gerçekleştirme tanımlayamazken, Java 8 ile birlikte radikal bir değişiklik ile tanımlama yapabiliyoruz. Araları yakınsasa da arayüzler arayüz sağlar, hiç bir şekilde gerçekleştirme ve durum sağlamazlar. Biz şu durumda SE 7 JAVA göre bu durum bu yöndeydi. Java SE 7 sonra Java 8 ileri de değişeceğiz. [1]

Arayüzdeki tüm methodlar soyut olmak zorundadır. Küme işareti kullanılmaz {} kullanılmaz. Arayüzlerin kurucusu yoktur. Nesnesi de oluşturulamazlar.Biz bunlara arayüz demektediriz. Bir şekildir. Bir nelik sağlar. İçerik(nasıllık) sağlamazlar. [1]

Arayüz de gerçekleştirme:

Arayüzler, soyuttur bütün methodlardır. Bir arayüzü yerine getirecek bir yapı olmalıdır. Arayüzler miras alınacak bir yapıdır. Arayüzler implement(yerine) getirme yapılır. Worker diye bir interface varsa ve Employee ise bir sınıftır. Employee bir Worker implement ederler. Implement etmek yerine getirmektir. Sınıflar bir birini extends (genişletmek) ederken interface ise implementation(gerçekleştirme) ile yerine getiririz. O yüzden implement kelimesi kullanılır. Ona bir implementation verir. Bir sınıf bir arayüzü yerine getirir. Sözünü yerine getirir. Biz sınıf ile sözümüzü, tahhütümüzü yerine getiririz. Böylece override eder sınıflar methodları. [1]



Resim -1 Akın Kaldıroğlu [1]

Eğer Employee Worker implement ediyorsa override etmek zorunadır. Arayüzünü aynısını devralıp implement ederler. Burada Boss da implement etmek zorundadır. Diğer sınıflarda da aynısını yapabilir. Final değilde method ama orada biz hem implementation hem de arayüzünü de devralıyorduk. Arayüzler öyle değildir amacı implementationdır. [1]

Extentionlar (extends) düz çizgi ile ifade edilirken, implementation kesik çizgiler ile ifade edilmektedir. Alt sınıflar da (Manager...vs) Worker implement etmek zorunda değildirler ama override edebilirler.. Interface private yapamayız. [1]

Arayüzler, sorumluluklara(methodlara) gerçekleştirme sağlamadıklarından, kendilerinden bu arayüzleri devralan alt sınıfların, arayüzdeki tüm metotlara gerçekleştirme vermeleri gereklidir. Çünkü arayüzdeki tüm metotlar soyutturular. Aksi taktirde, yani alt sınıf, arayüzden devraldığı bir metoda bile gerçekleştirme vermezse, alt sınıfın soyut (abstract) tanımlanması gereklidir. Çünkü sınıf hala soyut metoda sahiptir.[1]

Bir bir interface üst sınıf olarak kabul edersek aynı soyut sınıfı üst tip kabul etmemiz gibi devraldığımız bütün methodları implementation vermek zorundayız. Abstract methodlarda bütün methodlara implementation vermek zorunda değildir.

Arayüzler sorumlulukların(methodların) sıralandığı bir soyut yapıdır. Anayasa gibi düşünelim. Kanunlar daha çok uygulamaya yakınken yönetmelikler tamamen uygulamaya yönelik ülkemizde çıkartılmaktadırlar. [1]

Bir arayüz gerçekleştiren(yerine getiren) sınıf, arayüzdeki herbir methoda birer gerçekleştirme sağlamak zorundadır. [1]

Arayüzlerin is-a hiyerarşisinde bir üst tip olarak görülmesi yanında, yetkinlik kazandırması da söz konusudur. Nesneler, sınıflarının yerine getirdikleri arayüzler ile pek çok yetkinlik kazanır. [1]

Parent genişletmiyoruz, gerçekleştirme yaparız. Implementation kelimesini kullanıyoruz.

Method kastımız daha çok nasıllığı vurgulamaktayız. Sorumluluk ifadesi ise daha çok methodlara gerçekleştirmektir.

Arayüz bir sözleşme, protocol veya contact ve gerçekleştiren sınıflar da birere yüklenici (implementor) olarak nitelendirilmesine de izin verir.

Biz protokoller yazarız ve onları implement edenler yerine getirir. Arayüzlere uymalarını bekleriz. Handshake olarak da ifade edilen el sıkışma ifadesini de kullananlar vardır.

Aynı arayüzü gerçekleştiren arayüzler aynı sözleşmeyi yerine getiren yüklenicilerdir. Arayüz ile onu gerçekleştiren sınıflar arasında bir is-a ilişkisi vardır bundan dolayı polymorphism, upcasting, dowcasting gibi konularda sınıflardan bir farkı yoktur.

Bir arayüz, kendisini gerçekleştiren bir üst tiptir. Arasında temel fark soyutluk-somutluk spektrumundaki yerlerdir. Arayüzler soyuttur ve sınıflar concrete yani somuttur.

- 1- Akın Kaldıroğlu – www.github.com sitesinde yer alan, “03 - Arayüzler.pdf” alınmıştır ve eğitim, öğretim amaçlı kullanılmıştır. Kendisine teşekkürü borç bilirim. Önemli bir vurgulama yapmadan geçmeyim. [1] şeklinde verilen kaynakça bütün paragrafı niteleyecek şekilde anlaşılması önem arz etmektedir.Sadece [1] önceki cümleyi yerine getirmke için değildirler.
- 2- Google.com Aramalarım sonucu çeşitli makaleler.