

Keza koltuklar, direksiyon sistemi ve hava yastıkları gibi pek çok sistem küçük rötuşlarla diğer uygulamalarda da kullanılabilir olmaktadır. Tüm sistemi oluşturan programcının artık koltuğun kumaşının terletip terletmeyeceğine kafa yorması gerekmez. O koltuğu programlayan programcının düşünmesi gereken bir konudur bu. Tüm sistemi programlayan programcı daha çok koltuğun bağlantı noktaları ve sisteme entegrasyonu üzerinde kafa yormaktadır.

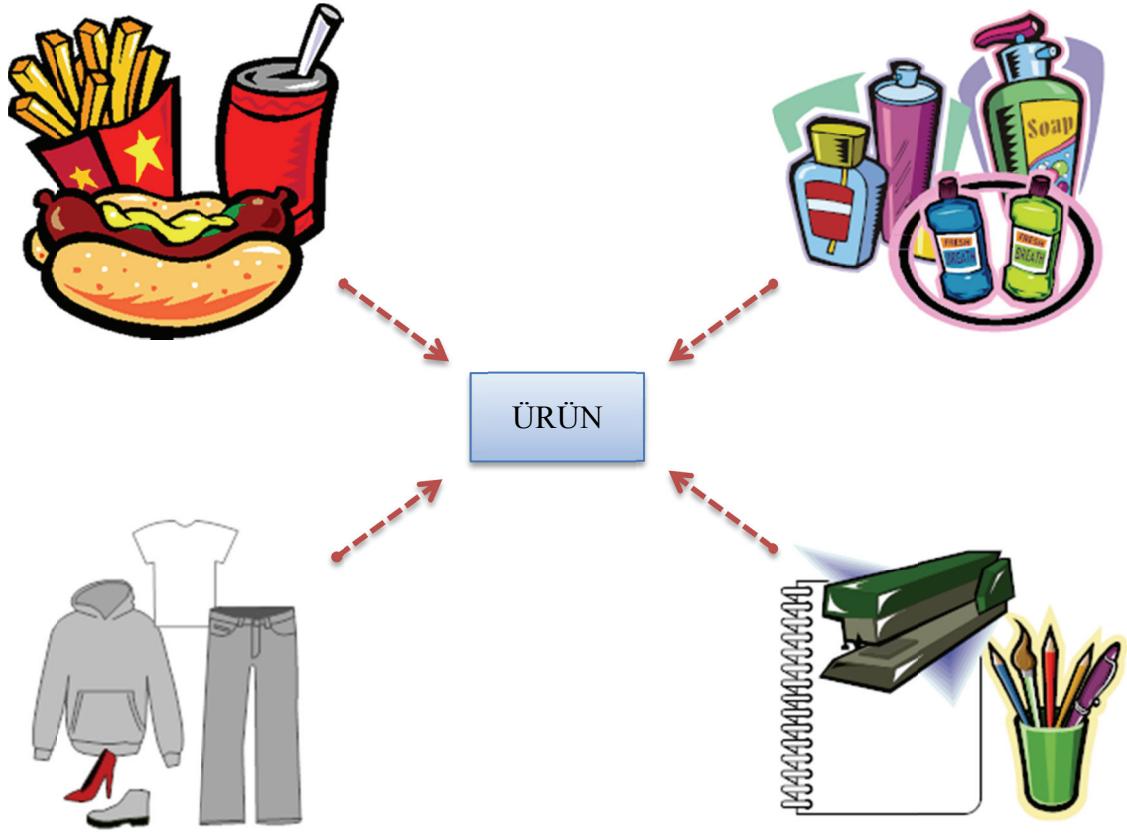
4.2.2. Nesnelerin Kalıtım Özelliği

NYP tekniğinin diğer önemli bir avantajı da nesnelerin sahip oldukları kalıtım özelliğinden ileri gelmektedir.

Nesneler, sahip oldukları bu kalıtım özellikleri sayesinde temel bazı özelliklerini ebeveynlerinden kalıtım yoluyla devralabilmektedirler.

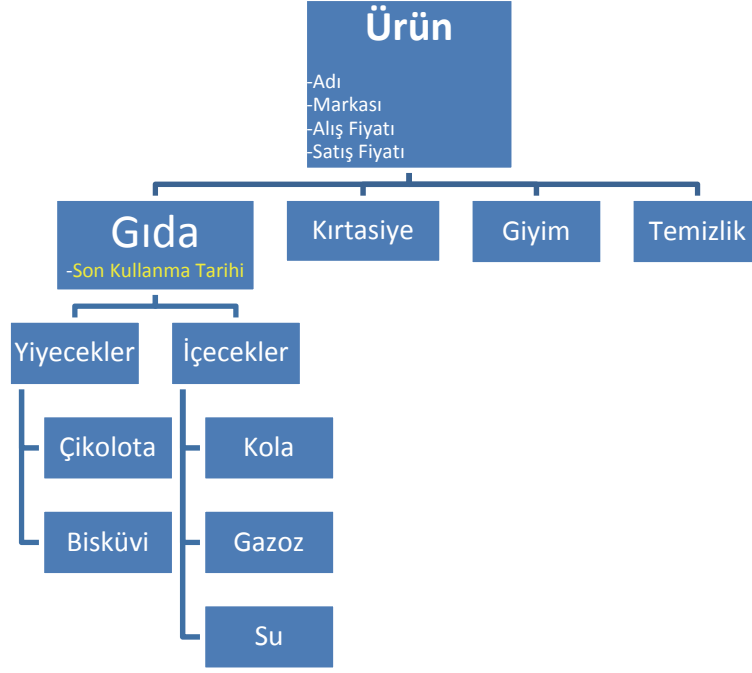
Bu sayede de uygulamalar daha kolay oluşturulabilmekte, düzeltilebilmekte ve daha kolay da genişletilebilmektedir. Tüm bunlar da zaten yine sistemlerin müdahale edilebilir olmalarına hizmet eden özelliklerdir.

Örneğin bir *süper market* için geliştireceğimiz *stok takip uygulamasında* NYP tekniğini tercih etmeniz durumunda süper marketteki *her bir ürünü bir nesne olarak oluşturmak* iyi bir çözüm olacaktır.



Bir süre sonra ise fark edeceksiniz ki oluşturduğunuz nesnelere bir sürü ortak noktası bulunmaktadır. Örneğin bir kırtasiye ürünü olan dolma kalemini düşünün: bir adı, markası, alış fiyatı, satış fiyatı gibi özellikleri bulunmaktadır. Benzer özellikler ele alacağınız bir çikolatada da karşımıza çıkabilmektedir. Yalnız dikkat edilecek olursa çikolatanın gıda ürünü olmasından kaynaklı olarak bir *son kullanma tarihi* de olmak zorundadır.

İşte yine de ortak özellikleri nesnelere tekrar tekrar kodlamak yerine bunları belli bir hiyerarşiyi takip edecek şekilde ebeveynlerinden devralabilmeleri sağlanabilmektedir.



Marketteki ürünlerimizi ilk etapta yukarıdaki gibi hiyerarşik bir düzene sokmak mümkündür. Tabii bu düzen çok daha iyileştirilebilir durumdadır. Şimdilik konumuzun mantığını açıklamaya yetecek kadarı ile yetinelim.

Bu hiyerarşik düzende dikkat edecek olursanız en tepeye *ürün* adlı nesnemiz oturmuştur. Bu nesnemizin de kendine özgü özellikleri bulunmaktadır. Aslına bakarsanız *ürün nesnesinin özellikleri çok da kendine özgü değildir*. Hatta hiç değildir! (Tamda bu noktaya dikkat çekmek için böyle bir geri dönüş manevrasında bulundum☺)

Zira ürün nesnesi hiyerarşik düzenin en tepesinde bulunduğundan dolayı taşıdığı bütün özellikler kendisinden türeyen tüm nesnelere de bulunmaktadır. Örneğin bu tabloda biraz daha detaylandırdığımız gıda ürünlerinden çikolatayı ele alırsak, yukarıya doğru çıktığımızda nihayetinde çikolatanın da bir ürün olduğunu söylemek yanlış olmaz. Bu durumda *çikolata nesnesinin* siz kodlamamış bile olsanız, bir;

- ☞ Adı,
- ☞ Markası,
- ☞ Alış Fiyatı ve
- ☞ Satış Fiyatı

Özellikleri otomatikman olmaktadır. *Çünkü o sonuçta bir üründür*. Bir *gıda ürünü* olmasından kaynaklı olarak da bir *son kullanma tarihine* sahiptir.

? Şimdi düşünün: bu örnekte *son kullanma tarihi* özelliğini *gıda ürünlerinden* alıp en tepedeki *ürün nesnesine* vermiş olsaydık neler olurdu?

Cevap çok basit, tüm ürünlerin birer son kullanma tarihi olmuş olurdu. Bu durumda giyim ürünleri de sonuçta bir ürün olduğu için onların da birer son kullanma tarihleri olmuş olacaktı. Bu kul yapısıdır, elbette bir son kullanma tarihi olmalıdır demeyin sakın, zira

anlatmaya çalıştığımız sadece gıda ürünlerinde bulunması gereken bozulmadan kalabilecekleri maksimum süreyi gösterir tarihtir.

Evet, NYP tekniğinde görüldüğü üzere nesnelimizin belli bir hiyerarşik düzene göre oluşturulmaları da mümkündür. Bu düzen içerisinde ki herhangi bir nesne özelliklerinin bazılarını üstündeki nesnelere kalıtım yoluyla devralabilmektedir.

! Bu hiyerarşik düzenin oluşturulmasında ki önemli kural ise şudur:

-dir, -dır, -dur, -dür... kuralı ☺ İngilizcesi daha kolay: “is a...” kuralı. Eğer ki herhangi bir nesne için şudur-budur şeklinde bir genelleme yapabiliyorsanız genelleme yaptığımız nesneyi hiyerarşik düzende bir üst basamağa yerleştirebilirsiniz demektir.

Örneğin çikolata, *tüm çikolatalar sonuçta bir yiyecektir* ifadesini söyleyebiliyorsanız, çikolataları *yiyecek* grubunun altına yerleştirebiliriz demektir. Aynı şekilde *tüm yiyeceklerde bir gıda ürünüdür* ve *tüm gıda ürünleri de marketteki herhangi bir ürünüdür* şeklinde genellemeler yapılabilmektedir. Bu genellemeler yardımıyla da hiyerarşik düzen oluşturulabilmektedir.

İşte bu hiyerarşik düzen çerçevesinde de herhangi bir nesne özelliklerinin bazılarını bir üst basamağından devralmaktadır. Bu da NYP tekniğinde *nesnelere kalıtım özellikleri* olarak bilinmektedir.

Nesnelere kalıtım özelliği nesnelere oluşturmamızda kolaylık sağlaması, kodlamayı kısaltması gibi avantajları yanında bir de uygulamaların kolaylıkla genişletilebilmelerine olanak sağlamaktadır.

Düşünün bir kere, herhangi bir süper market için bir uygulama geliştiriyorsunuz veya geliştirdiniz. Yukarıdaki gibi de bir hiyerarşik düzen içerisinde nesnelerinizi birbirlerinden türeterek kodlama yoluna gittiniz ve sistem tıkır tıkır işliyor. Sonra bir gün marketin patronu barkod sistemine geçmeye karar veriyor (bu arada da marketin patronu bizim müşteriniz olmaktadır, müşteri de veli-nimetimiz, ne derse yapmak durumundayız demi☺).

Bu sisteme göre de marketteki her bir ürünün barkod numarasının olması gerekmektedir. Haydaa!...-amiyane bir tabirle- *buyur burdan yak!* derler ya, tam öylesine bir durum demi ☺ Fakat, tabi n’apıyoruz, öncelikle panik yapmıyoruz!

Ekleme istediğimiz bu yeni özelliğin hangi tür nesnelere bulunması gerektiğini biliyor muyuz: biliyoruz, marketteki tüm ürünlerde barkod numarasının bulunması gerekmektedir. Bizim nesnelimiz temel özelliklerini bir üst basamağından devralarak geliyordu, bu durumda olması gereken en temel özelliklerini kimden devralır? Tabii ki en tepede ki ürün nesnesinden demi?

İşte sisteme yapmamız gereken müdahale de tam da bu noktada, *ürün nesnesine bir barkod numarası özelliğini eklemek* şeklinde olacaktır. Diğer tüm nesnelere bir şekilde bu nesneden türeyerek üretildikleri için otomatikman marketteki tüm nesnelere de bu özelliği eklemiş oluruz. Dolayısıyla çikolatada nihayetinde bir ürün olduğundan, bir barkod numarasına, herhangi bir giyim malzemesi de nihayetinde marketteki bir ürün olduğundan dolayı otomatik olarak bir barkod numarasına sahip olmaktadırlar. Görüldüğü üzere de bu sayede uygulamalarımız daha müdahale edilebilir bir yapıya bürünmektedir.