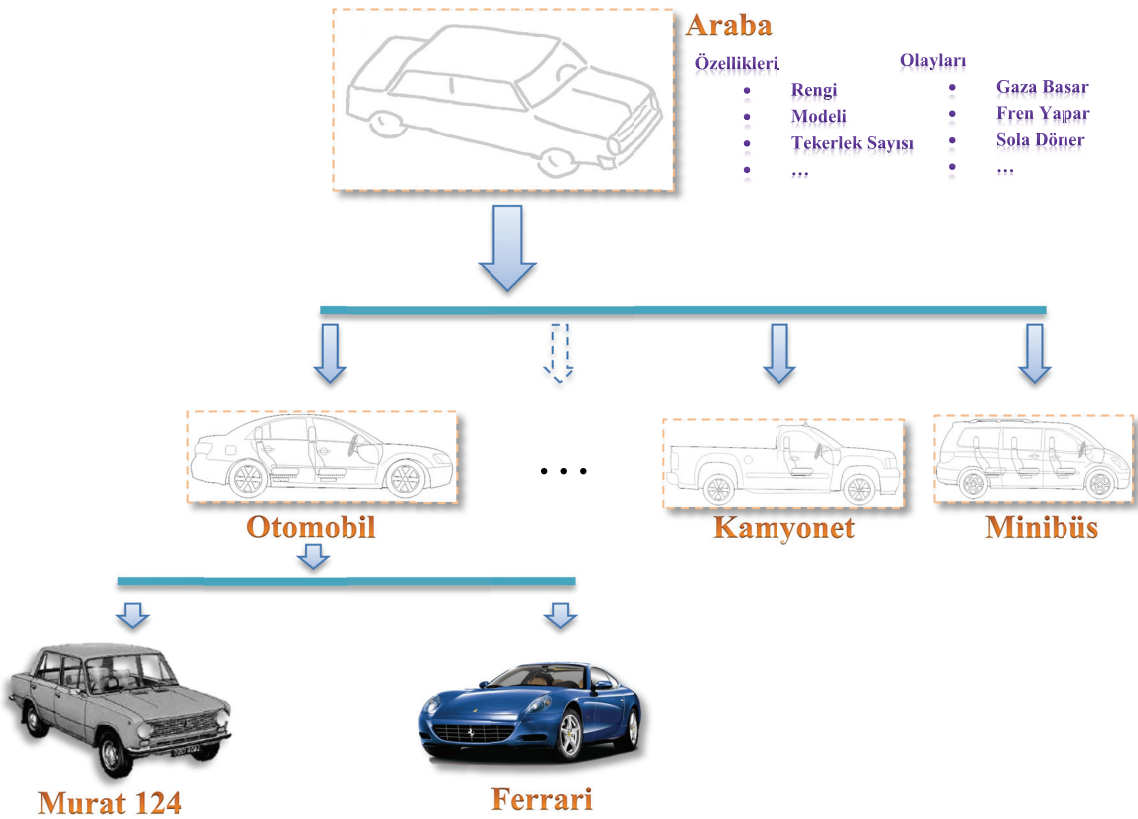


4.2.3. Nesnelerin Çok Biçimlilik Özellikleri

Yine NYP tekniğinde uygulamalarımızın esnekliğine ve onların hep dediğimiz gibi daha müdahale edilebilir bir yapıya bürünmelerine hizmet edebilecek özelliklerinden bir tanesi de nesnelerin *çok biçimlilik* özellikleridir.

Bu özelliği de kısaca özetlemek gerekirse, nesneler çok biçimlilik özelliklerinden yararlanarak, türlerine göre aynı komutu (aslında metotları) farklı şekillerde yorumlayabilmektedir.

Örneğin kendinize bir araba nesnesi oluşturduğunuzu düşünün ve artık biliyoruz ki nesnelerin kalıtım özellikleri sayesinde bu nesneyi temel alan bir sürü türevlerini –tür olarak çeşitlerini- de üretebiliriz.



Araba nesnesi diğer nesnelere ebeveynlik yapacağı için, diğer tüm türlerin, araba olmak adına olan ortak özelliklerini üzerinde barındırmalıdır.

Örneğin bir arabanın nesnesi vardır, yani ne gibi özellikleri bulunur:

- ☞ Rengi,
- ☞ Modeli,
- ☞ Tekerlek Sayısı,
- ☞ ...

gibi şu anda hepsini sayamayacağımız daha bir sürü özellik. Yalnız dikkat edin bu özellikler tüm arabalarda olur. Fakat *kasa hacminden* bahseden bir özellik olsaydı, onu araba

nesnesine yerleřtirmemiz doęru olmazdı. ünkü tm arabaların kasası bulunmamaktadır. Belki sadece kamyonetlerin kasası olabilir.

Dięer taraftan bir araba ile neler yapılır:

- ☞ Gaza basılır,
- ☞ Frene basılır,
- ☞ Sola dnlr,
- ☞ Saęa dnlr... gibi daha pek ok eylem yapılabilir.

Bunlar da genel olarak arabalarla yapılabilecek eylemlerdir. Bu noktada řimdiye kadar ğrendiklerimize gre bu zelliklerin ve olayların ortak zellik ve olaylar olduęundan dolayı alt basamaklarda ki trlerde de tekrar tekrar tanımlanmaması gerektięini dřnebilirsiniz.

! Yalnız bazen nesnelere yle zellikleri veya olayları olur ki ilgili zellięin veya olayın, *temel trde de bulunması gerekirken tr deęiřince alt trlerde de yeniden tanımlanması* gerekebilir.

rneęin yukarıdaki řekillerde de gsterildięi gibi arabaların gaza basması. Tm arabalarda olması gereken bir zelliktir aslında (zellikten ziyade bir olaydır, fakat bazen zellik kelimesi daha geniř anlamda tanımlayıcı kelimesinin yerine kullanılmaktadır). Yalnız arabalardan treyen yeni arabalarda da gaza basma iřlemi giderek deęiřmektedir. Zira ikisi de birer otomobil olan Murat 124 ile Ferrari'nin gaza basma iřlemi sonunda verdikleri tepkiler takdir edersiniz ki farklı olmaktadır 



Murat124.GazaBas;

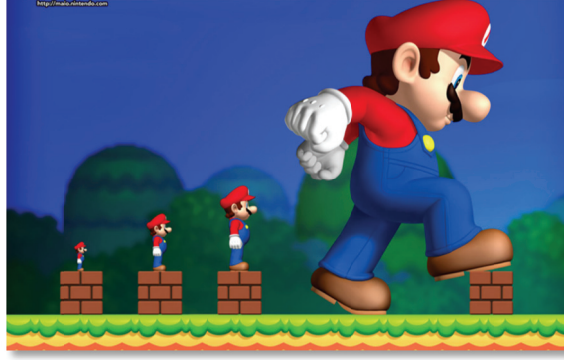


Ferrari.GazaBas;



Bir araba gaza basınca n'apar? Cevap ok basit: gider! demi  Yalnız bizim *Hacı Murat*'ın gidiřiyle *Ferrari*'nin gidiři farklı olacaktır.

Tam bu noktada nesnelere ok biimlilik zellikleri devreye girmektedir. Bazen olur ki bir zellięi hem temel nesneye de oluřturmanız gerekir hem de temel nesneden treyen yeni nesnelere oluřturmak zorunda kalırsınız. İřte byle bir durumda bu nesnelere ok biimlilik zelliklerinden dolayı bu iřlem mmkn olabilmektedir.



Bir başka örnekte de, Mario benzeri küçük bir oyun programı yazdığımızı düşünelim. Bu oyunda kahramanımız Mario yediği her bir mantarla birlikte giderek büyümektedir. Dolayısıyla da tehlikelere karşı giderek daha dirençli olmaktadır.

Kötü niyetli katil kaplumbağalar ve ekip arkadaşları ise kahramanımız Mario'yu sürekli takip etmekte ve sıkıştırdıkları anda da eğer ki kaçamazsa onu vurmaktadırlar☺

Bu oyunda kahramanımızı temsil eden temel bir nesne oluşturarak Mario'nun özelliklerini bu nesneye yükleyebiliriz. Kahramanımız büyüdükçe de bu temel nesneden yeni özelliklere sahip daha gelişmiş nesnelere türeterek Mario'nun büyüdükçe kazanması gereken özellikleri de bu yeni nesnelere yükleyebiliriz.

Oyunda gerektiğinde, Mario'nun da vurulması gerekiyordu ve vurulma özelliğinin Mario'nun tüm büyüklüklerinde olması gerekiyor. Yani Mario büyük de olsa küçük de olsa yeri geldiğinde bir şekilde vurulması gerekiyor.

İşte bu vurulma özelliğini gerçekten Mario'nun tüm büyüklüklerinde yeniden kodlamak gerekmektedir. Çünkü Mario'nun değişik büyüklüklerinde ki vurulmalar da farklı olacaktır.

Mario'nun en büyük halinin vurulduğunu düşünün, bu onu fazla etkilemezken (belki sadece sendeleyecek veya sadece bir canı eksilecek) Mario'nun en küçük halinin vurulduğunu düşünün, bu da onun aktif futbol hayatının bitmesine ve onun yeşil sahalardan silinmesine yol açacaktır☺

Oyun boyunca Mario'muzu temsil eden ise tek bir değişkenimiz olacaktır. Biz gerektiğinde Mario'muzun vurulması için kodun herhangi bir yerinde;

☞ *eğer (şunlar şunlar olduysa) mario.Vurul*

şeklinde bir kod çalıştıracakız ve gerisini nesnelere çok biçimlilik özelliği halledecektir. Mario'muzu temsil eden değişkenimize o anda Mario'nun hangi türü yüklendi ise vurulma olayı da Mario'nun o türüne göre gerçekleşecektir.

4.3. Sınıf Nedir?

NYP tekniğini ne kadar da kısaltırsak kısaltalım bahsetmeden geçemeyeceğimiz diğer bir yapıda sınıflardır. Zira sınıflarımız olmadan maalesef nesnelimizde olamamaktadır. Aslında yukarıda ki özellikleri anlatırken çok defa klavyemizin (!) ucuna gelip geri döndüğü