

NESNE TABANLI PROGRAMLAMA - BIL207



Sunum Ana Hatları

BUGÜNÜN BAŞLIKLARI

- Nesne Tabanlı Programlama Giriş Ortamı
- Temel Kavramlar
- Nesne ve Sınıf Kavramı
- Python Programlama Diline Giriş

Ders Hakkında

Bu ders nesneye yönelik programlamada Python programlama dili kullanarak kavramsal ve uygulama tabanlı bir giriş niteliğindedir. Nesneye yönelik programlama ile ilişkilendirilen temel kavramlar (örnek olarak nesneler, sınıflar, protokol, sıradüzen, kalıtım, çok biçimlilik, kodun tekrar kullanımı, beraber çalışma vb.) kullanılarak tanıtılacak ve örneklerle açıklanacaktır.



“

DÜŞÜNÜLECEK ŞEY

Oturduğumuz sürece korkular yaratırız. Harekete geçtiğimizde ise korkularımızın üstesinden geliriz.

DR HENRY LINK

”

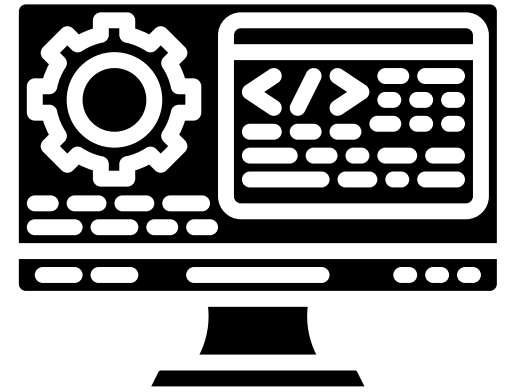
Ders Hakkında

Bu ders nesneye yönelik programlamada Python programlama dili kullanarak kavramsal ve uygulama tabanlı bir giriş niteliğindedir. Nesneye yönelik programlama ile ilişkilendirilen temel kavramlar (örnek olarak nesneler, sınıflar, protokol, sıradüzen, kalıtım, çok biçimlilik, kodun tekrar kullanımı, beraber çalışma vb.) kullanılarak tanıtılacak ve örneklerle açıklanacaktır.



Ders Hakkında

Nesne yönelimli diller 1960'lardan beri var olmasına rağmen, son 10 yılda yazılım endüstrisinde nesne teknolojilerinin kullanımı ve kabulünde benzersiz bir büyüme görüldü. Başarısız olarak başlamış olmasına rağmen, Java, CORBA ve C++ gibi son başarılar, nesne yönelimli (OO) teknikleri yeni kabul seviyelerine taşıdı. Bu bir tesadüf değil. **Yıllarca akademide sıkışıp kaldıktan ve yerleşik uygulamalara karşı yokuş yukarı bir savaş vermek zorunda kaldıktan sonra, nesne yönelimli programlama (OOP), insanların sonunda tekniğin vaatlerini gerçekleştirebilecekleri bir noktaya geldi.** Geçmişte, patronunuzu nesne yönelimli bir dil kullanmanıza izin vermesi için ikna etmeniz gerekirdi. Bugün, birçok şirket kullanımını zorunlu kılıyor. İnsanların nihayet dinlediğini söylemek güvenlidir.



Ders Hakkında

OOP'nin mevcut durumunu anlamak için programlama tarihi hakkında biraz bilgi sahibi olmalısınız. Hiç kimse bir gecede OOP'yi düşünmedi. Bunun yerine, OOP, yazılım geliştirmenin doğal evriminde sadece başka bir aşamadır. Zamanla, işe yarayan ve başarısız olduklarını kanıtlayan uygulamaları seçmek daha kolay hale gelir. **OOP, kanıtlanmış, zamanla test edilmiş uygulamaları mümkün olduğunca verimli bir şekilde birleştirir.**



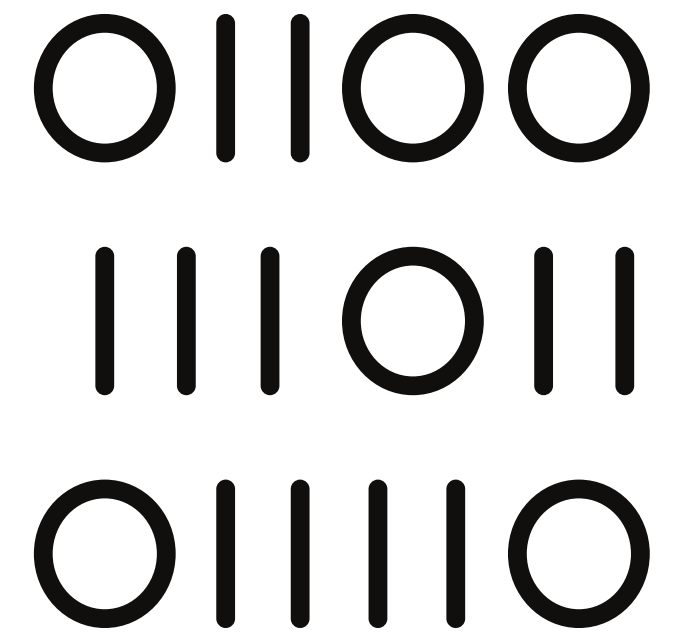
OOP Çalışma Ortamı

Program, herhangi bir elektronik cihaza bir işlem yaptırmak için yazılan komutlar dizisidir. Programlar; bilgisayar, cep telefonu, tablet, elektronik ev eşyaları, araba ve daha birçok yerde kullanılır. Elektronik cihazlara bilgisayar, cep telefonu, akıllı saat, akıllı televizyon vb. örnek olarak verilebilir.

Program yazmak denilince akla ilk gelen bilgisayarlar olsa da günümüzde pek çok elektronik cihaza kod yazılabilir. Bilgisayar ve diğer elektronik cihazlar çalışma prensibi olarak programları kullanır. Bu cihazlar önce girilen bilgiyi alır, ardından bu bilgiyi işler ve en sonunda ortaya bir sonuç çıkarır. Bu sonuç bazen bir mesaj bazen de bir işlemi gerçekleştirme şeklindedir. Bu durum, insanların birbiriyle iletişimi gibi düşünülebilir. İnsanlar da konuşmalarında önce karşı tarafın ifadelerini algılar, algıladıklarını işler ve son olarak karşı tarafa bir cevap verir.

OOP Çalışma Ortamı

Programlar önceden “1” ve “0”lardan oluşan kod blokları ile makine dilinde yazılırdı. Makine dilinde kod yazılması ve yazılan kodun anlaşılması oldukça zordu. Bu nedenle daha okunaklı ve kolay kod yazılabilen Assembly dilleri geliştirildi. İlerleyen zamanlarda daha çok anlaşılır, konuşma diline daha yakın ifadelerle kod yazma imkânı sağlayan C, C++, Turbo Pascal, Visual Basic, C#, Python gibi programlama dilleri kullanıldı.

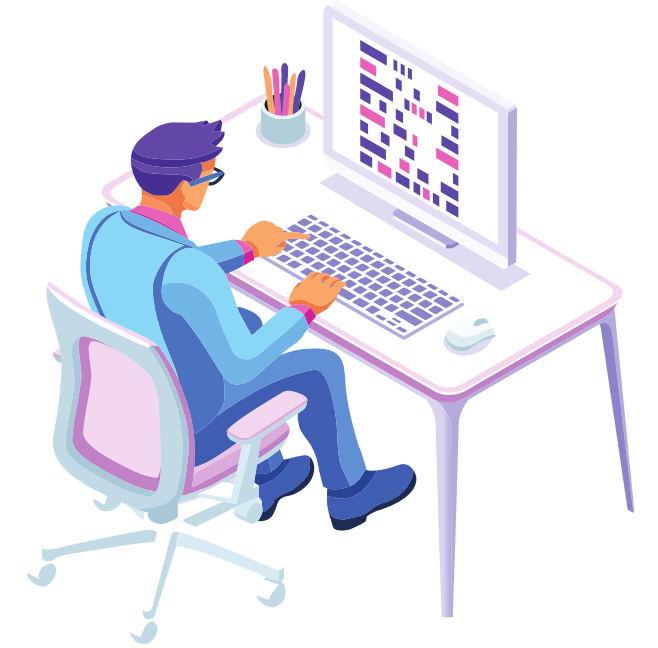


```
011000  
111011  
011110
```

Programlama Dili Seçimi

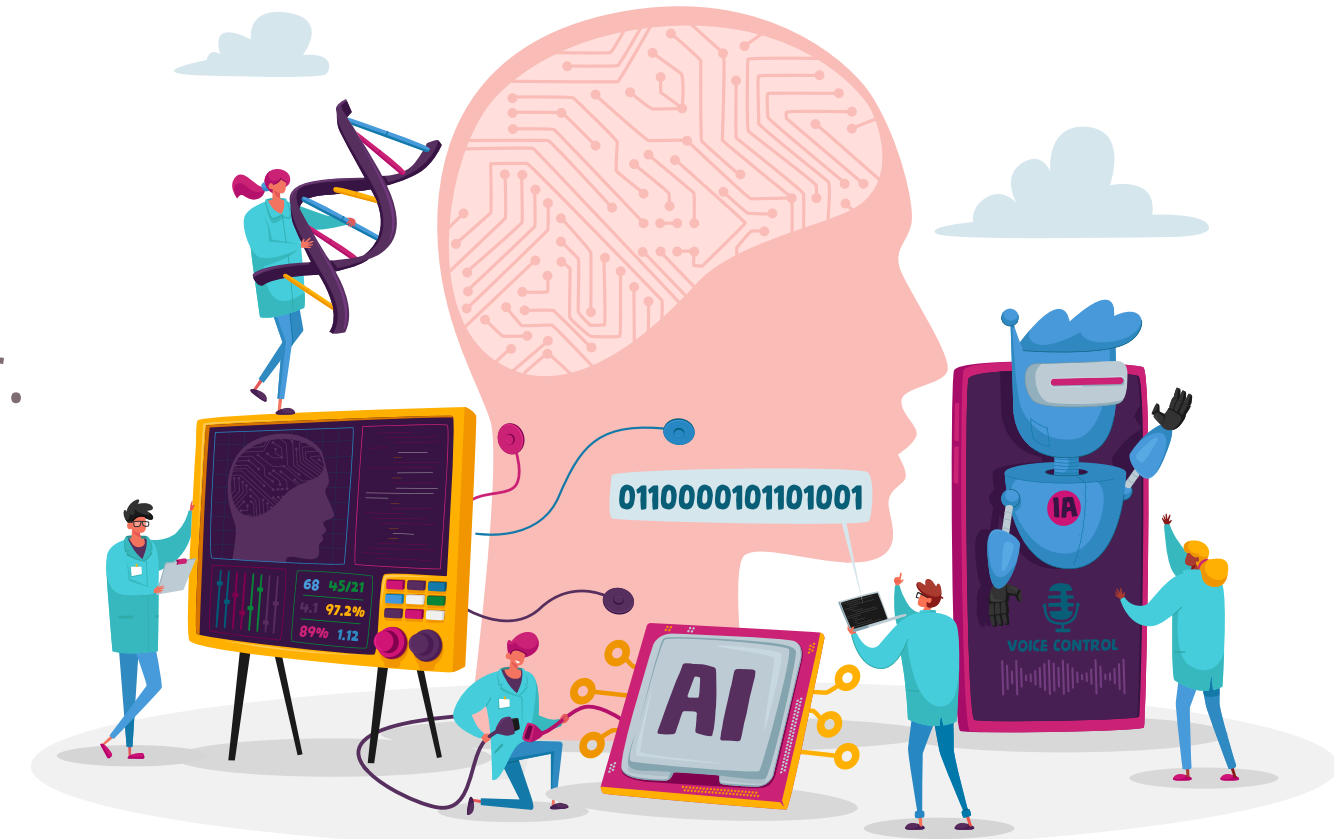
Programlama; bilgisayar programlarının yazılması, test edilmesi ve bakımının yapılması sürecine verilen isimdir. Bu sürecin daha verimli geçmesi için amaca yönelik bir programlama dili seçilmelidir.

Bu ders içeriğinde Python programlama dilinin kullanımı anlatılacaktır.



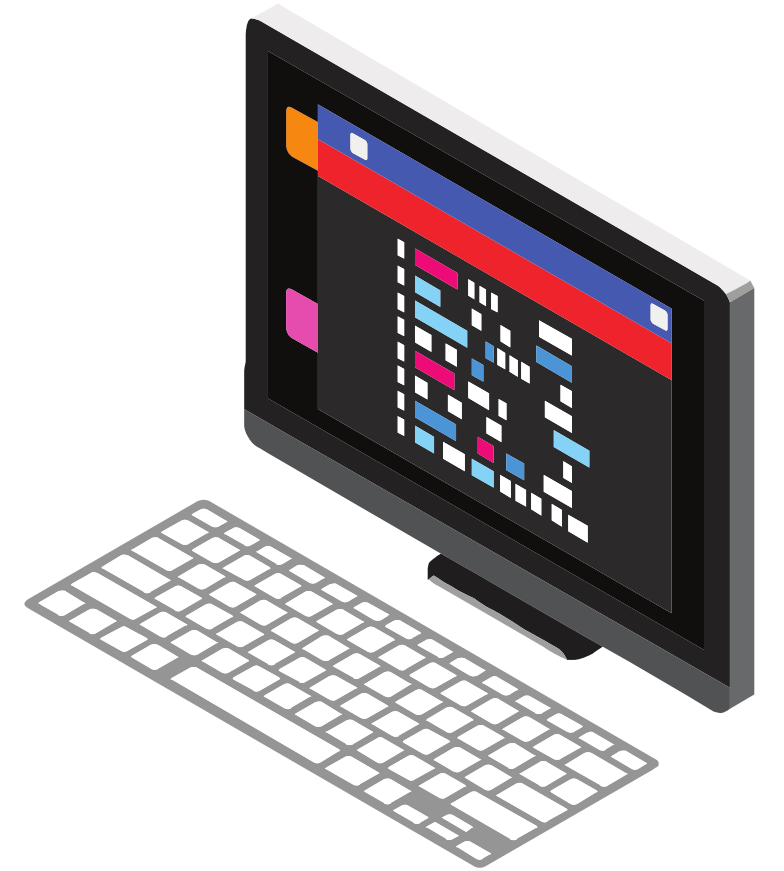
Python Programlama Dili

- Web uygulamaları geliştirme
- OOP geliştirme
- Python bilimsel ve numerik hesaplamalarda,
- Yapay zekâ uygulamalarında,
- Nesnelerin İnterneti,
- Veri madenciliği (Data Mining) uygulamaları,
- Görselleştirme uygulamaları,
- Masaüstü uygulamaları vb. yönelik geliştirmeler yapılabilir.



Python Programlama Dili

- Programlamada yeniyseniz hızlıca öğrenilebilir.
- Python kolay okunur, kolay yazılır.
- Python web geliştirme için kullanılabilir.
- Tekrar tekrar kullanılabilen tasarıma sahiptir.
- Çeşitlilik ve esneklik sunuyor.

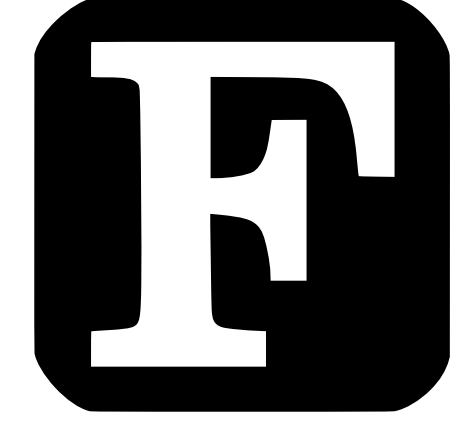


Kitap Önerisi



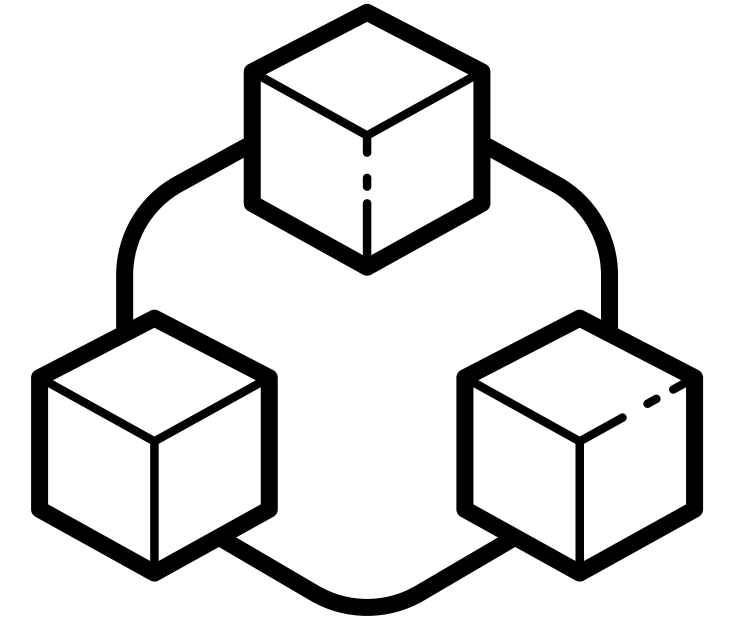
OOP için Öncüler

Bilgisayarlar daha yaygın hale geldikçe, daha üst düzey, prosedürel diller ortaya çıkmaya başladı; ilki FORTRAN'dı. Ancak, ALGOL gibi daha sonraki prosedürel diller OO üzerinde daha fazla etkiye sahipti. **Prosedürel diller, programcının bir programı verileri işlemek için bir dizi ince taneli prosedüre indirgemesine izin verir.** Bu ayrıntılı prosedürler, programın genel yapısını tanımlar. Bu prosedürlere ardışık çağrılar, prosedürel bir programın yürütülmesini sağlar. Program, prosedür listesini çağırmayı bitirdiğinde sona erer.



OOP için Öncüler

Modüler programlama, programları bir dizi bileşen veya modüle ayırır. **Verileri ve prosedürleri ayıran prosedürel programlamanın aksine, modüller ikisini birleştirir.** Bir modül, verilerden ve bu verileri işlemek için prosedürlerden oluşur. Programın diğer bölümlerinin bir modül kullanması gerektiğinde, sadece modülün arayüzünü kullanırlar. Modüller, programın geri kalanından tüm dahili verileri gizlediğinden, durum fikrini tanıtmak kolaydır: Bir modül, zaman içinde değişebilecek durum bilgisini tutar.



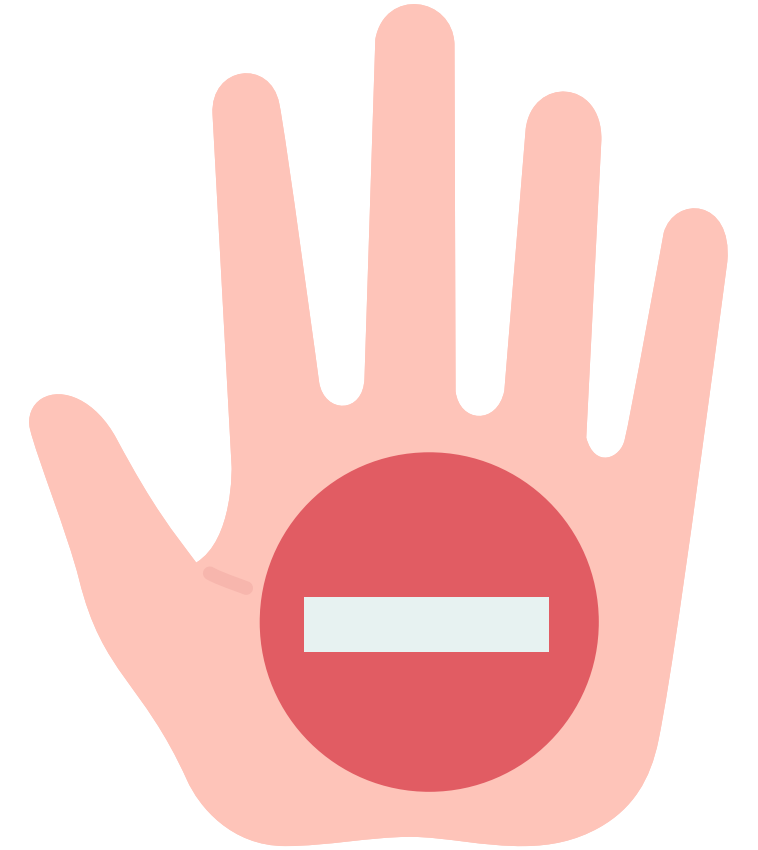
OOP için Öncüler

Modüler programlama, programları bir dizi bileşen veya modüle ayırır. **Verileri ve prosedürleri ayıran prosedürel programlamanın aksine, modüller ikisini birleştirir.** Bir modül, verilerden ve bu verileri işlemek için prosedürlerden oluşur. Programın diğer bölümlerinin bir modül kullanması gerektiğinde, sadece modülün arayüzünü kullanırlar. Modüller, programın geri kalanından tüm dahili verileri gizlediğinden, durum fikrini tanıtmak kolaydır: Bir modül, zaman içinde değişebilecek durum bilgisini tutar.

Yeni Terim: Dahili değişken, bir nesne içinde tutulan bir değerdir.

OOP için Öncüler

Ancak modüler programlama, kendi büyük eksikliklerinden muzdariptir. Modüller genişletilebilir değildir, yani kodu kırmadan ve doğrudan değişiklik yapmadan bir modülde artımlı değişiklikler yapamazsınız. **Ayrıca, yetkilendirme dışında bir modülü diğerine dayandıramazsınız. Ve bir modül bir tip tanımlasa da, bir modül başka bir modülün tipini paylaşamaz.**



OOP için Öncüler

Modüler ve prosedürel dillerde, yapılandırılmış ve yapılandırılmamış verilerin bir "tür"ü vardır. Tür, verilerin bellek içi biçimi olarak en kolay şekilde düşünülür. Kesin olarak yazılan diller, her nesnenin belirli ve tanımlanmış bir türü olmasını gerektirir. Ancak türler, "toplama" adı verilen bir stil aracılığıyla başka bir tür oluşturmak için genişletilemez. Örneğin, C'de birbiriyle ilişkili iki veri türümüz olabilir:

```
typedef struct
{
    int a;
    int b;
} aBaseType;

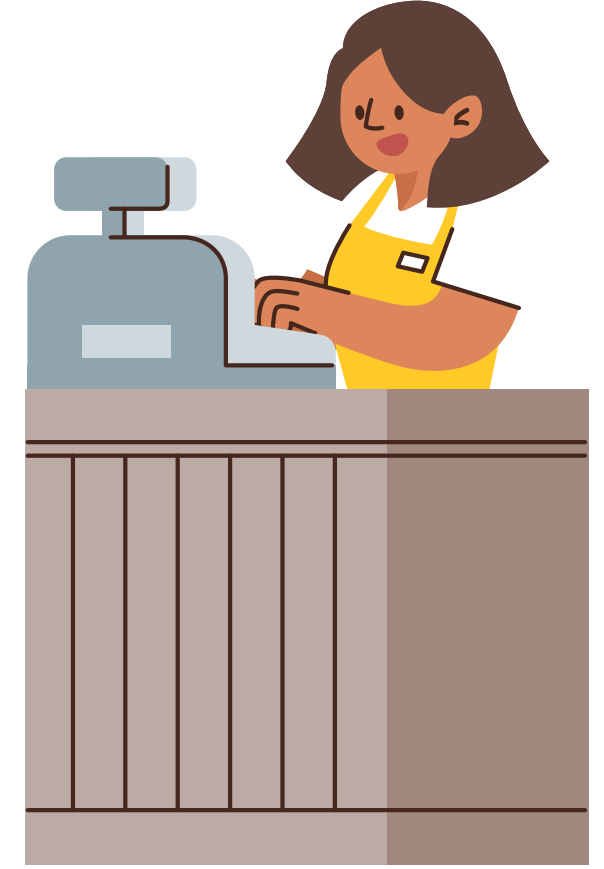
typedef struct
{
    aBaseType Base;
    int c;
} aDerivedType;
```

Nesne Tabanlı Programlama

OOP, modüle kalıtım ve polimorfizm ekleyerek modüler programlamadan sonraki mantıksal adımı atar. OOP, bir programı bir dizi üst düzey nesneye bölerek yapılandırır. Her nesne, çözmeye çalıştığınız problemin bazı yönlerini modeller. Program akışını yönlendirmek için sıralı bir prosedür çağrıları listesi yazmak artık OO altında programlamanın odak noktası değildir. **Bunun yerine, genel program akışını yönlendirmek için nesneler birbirleriyle etkileşime girer. Bir bakıma, bir OOP programı, çözmeye çalıştığınız problemin canlı bir simülasyonu haline gelir.**

Nesneleri Kullanan Yazılımlara OOP Yaklaşımı

Bir çevrimiçi alışveriş sepeti veya bir satış noktası terminali uygulamak için bir OOP programı geliştirmeniz gerektiğini düşünün. Bir OOP programı, ürün, alışveriş sepeti, kupon ve kasiyer nesneleri içerecektir. Bu nesnelerin her biri, programı yürütmek için birbirleriyle etkileşime girecektir. Örneğin, kasiyer bir siparişi toplarken, her bir öğenin fiyatını kontrol edecektir.



Nesneleri Kullanan Yazılımlara OOP Yaklaşımı

Bir programı nesneler açısından tanımlamak, yazılımı görüntülemenin kapsamlı bir yoludur. **Nesneler, sizi her şeyi bir nesnenin ne yaptığıının kavramsal düzeyinde görmeye zorlar: davranışları.** Bir nesneye kavramsal düzeyden bakmak, bir şeyin nasıl yapıldığına bakmaktan bir ayrılımadır: uygulama. Bu zihniyet, sizi programlarınızı doğal, gerçek dünya terimleriyle düşünmeye zorlar. Programınızı bir dizi ayrı prosedür ve veri (bilgisayar dünyasının terimleri) olarak modellemek yerine, programınızı nesnelerde modellersiniz. **Nesneler, programlarınızı problem alanınızdaki isimler, fiiller ve sıfatlarda modellemenize izin verir.**



Nesneleri Kullanan Yazılımlara OOP Yaklaşımı

Geri çekilip çözdüğünüz problemin terimlerini düşündüğünüzde, uygulama detaylarında çıkmaza girmekten kaçınırsınız. Elbette, yüksek seviyeli nesnelerinizden bazılarının bilgisayarla bazı düşük seviyeli, bilgisayar odaklı yollarla etkileşime girmesi gerekecektir. Ancak, nesne bu etkileşimleri sistemin geri kalanından izole edecektir.



Nesneleri Kullanan Yazılımlara OOP Yaklaşımı

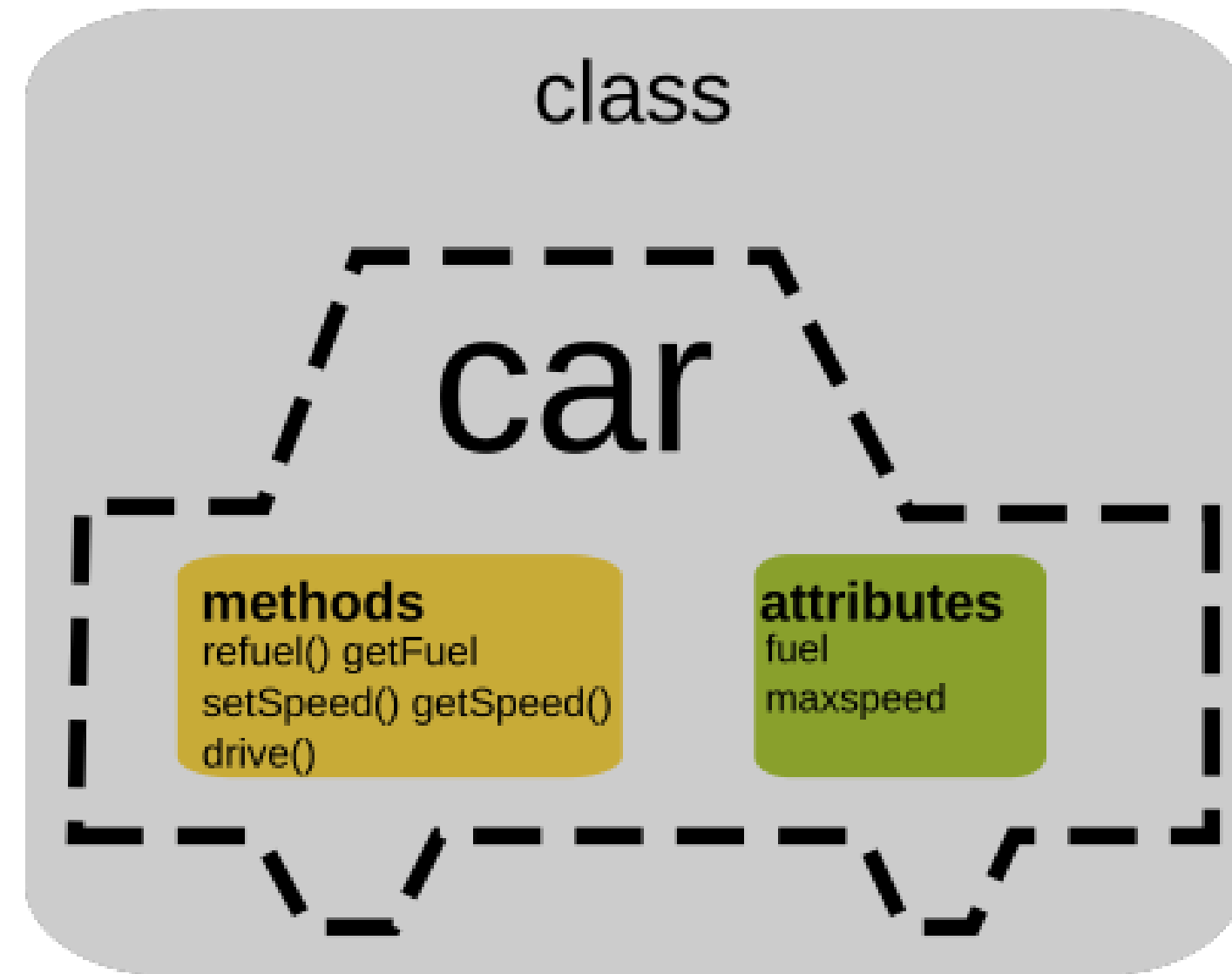
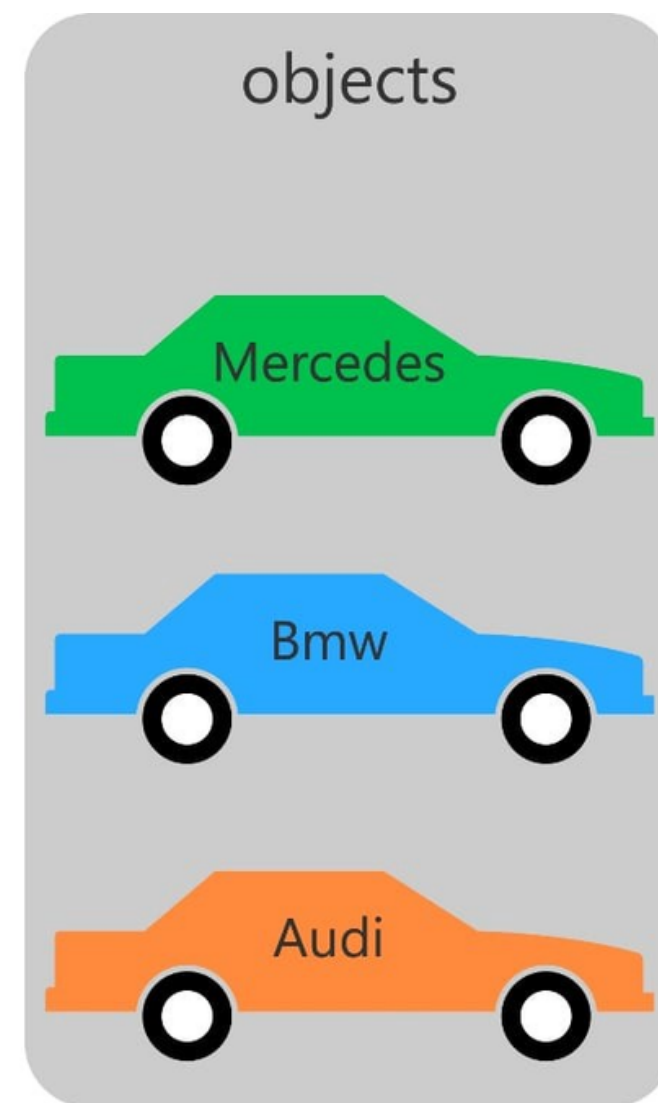
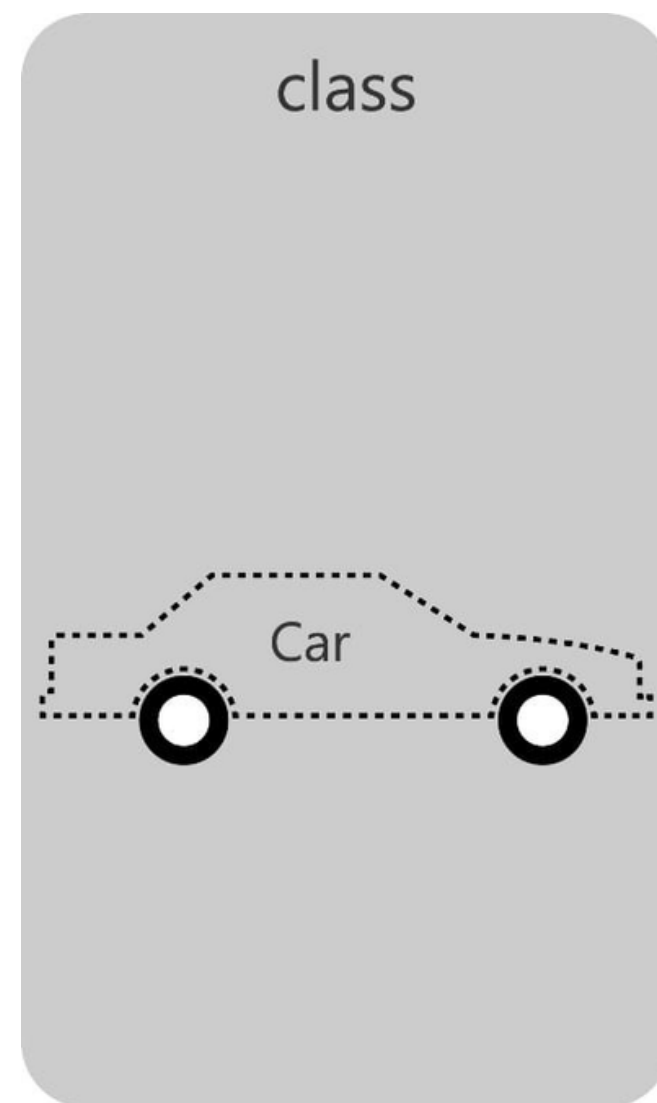
Bu noktada nesneyi resmi olarak tanımlayabilirsiniz:

Nesne, durumu ve davranışı kapsayan bir yazılım yapısıdır. **Nesneler, yazılımınızı gerçek dünya terimleri ve soyutlamalarla modellemenize olanak tanır.**

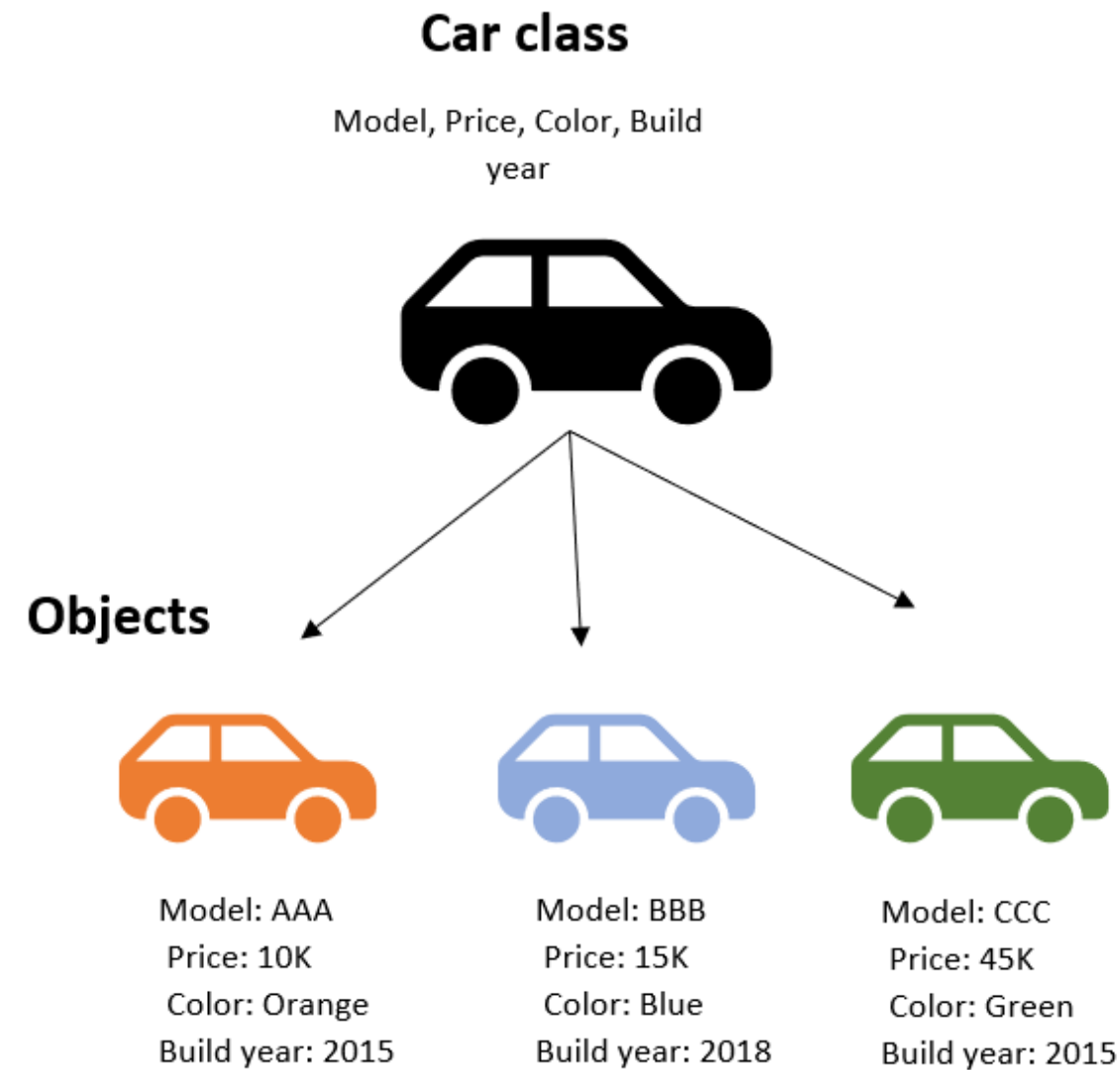
Açıkçası, bir nesne bir sınıfın bir örneğidir.



Nesneleri Kullanan Yazılımlara OOP Yaklaşımı



Nesneleri Kullanan Yazılımlara OOP Yaklaşımı



Gerçek hayattaki bir nesneyi yazılım dünyasında temsil edecek olan yapıya **class** denir. Tabi ki bu terim nesne tabanlı programlamayı daha somut bir şekilde anlamamız için söyleyebileceğimiz bir açıklamadır.

Nesneleri Kullanan Yazılımlara OOP Yaklaşımı

Gerçek dünya nesnelerden oluştuğu gibi, nesne yönelimli yazılım da öyle. Saf bir OO programlama dilinde, tamsayılar ve Booleanlar gibi en temel türlerden en karmaşık sınıf örneklerine kadar her şey bir nesnedir; tüm nesne yönelimli diller şimdiye kadar gitmez. Bazılarında (Java gibi), *int* ve *float* gibi basit temel öğeler nesne olarak ele alınmaz.



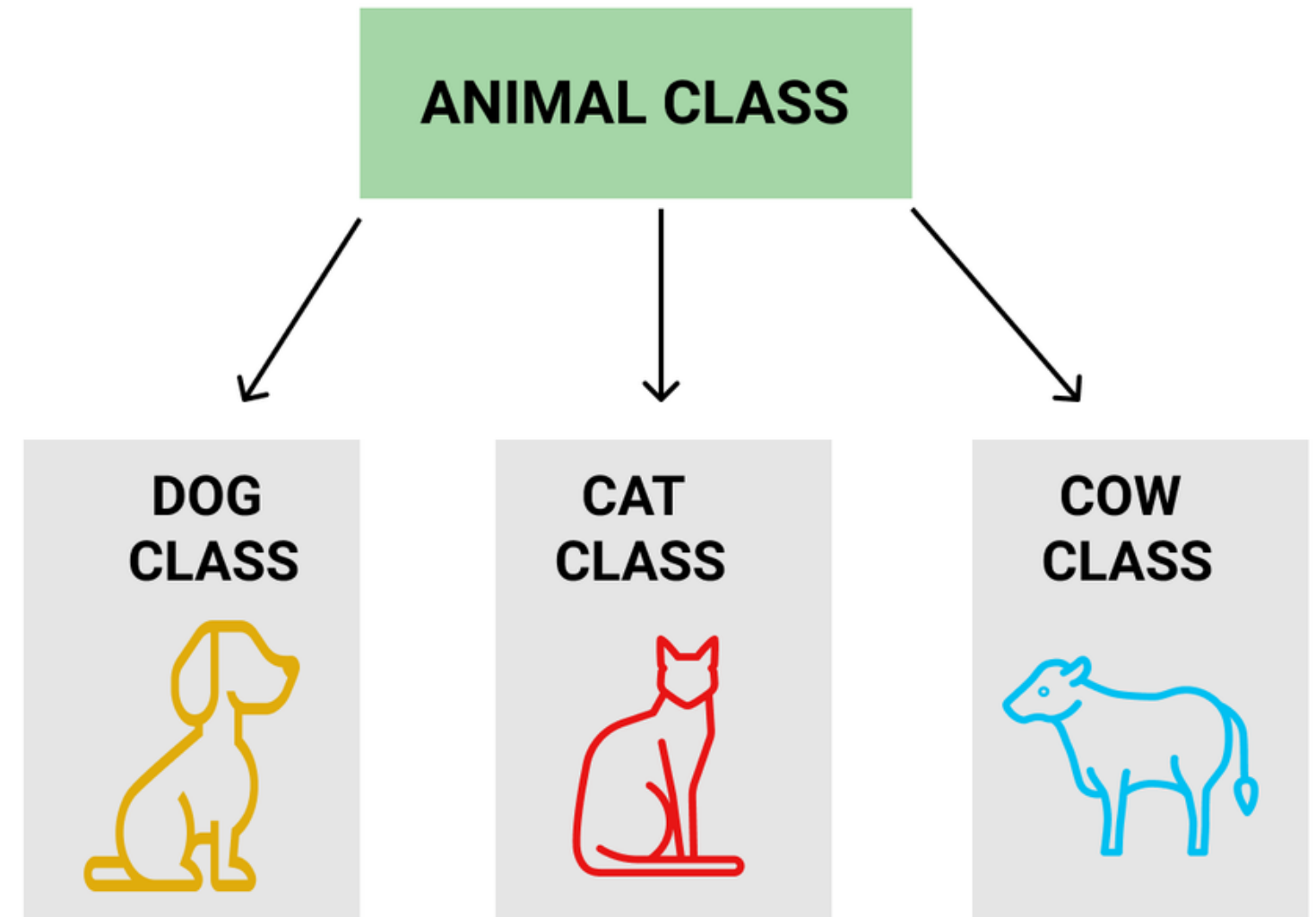
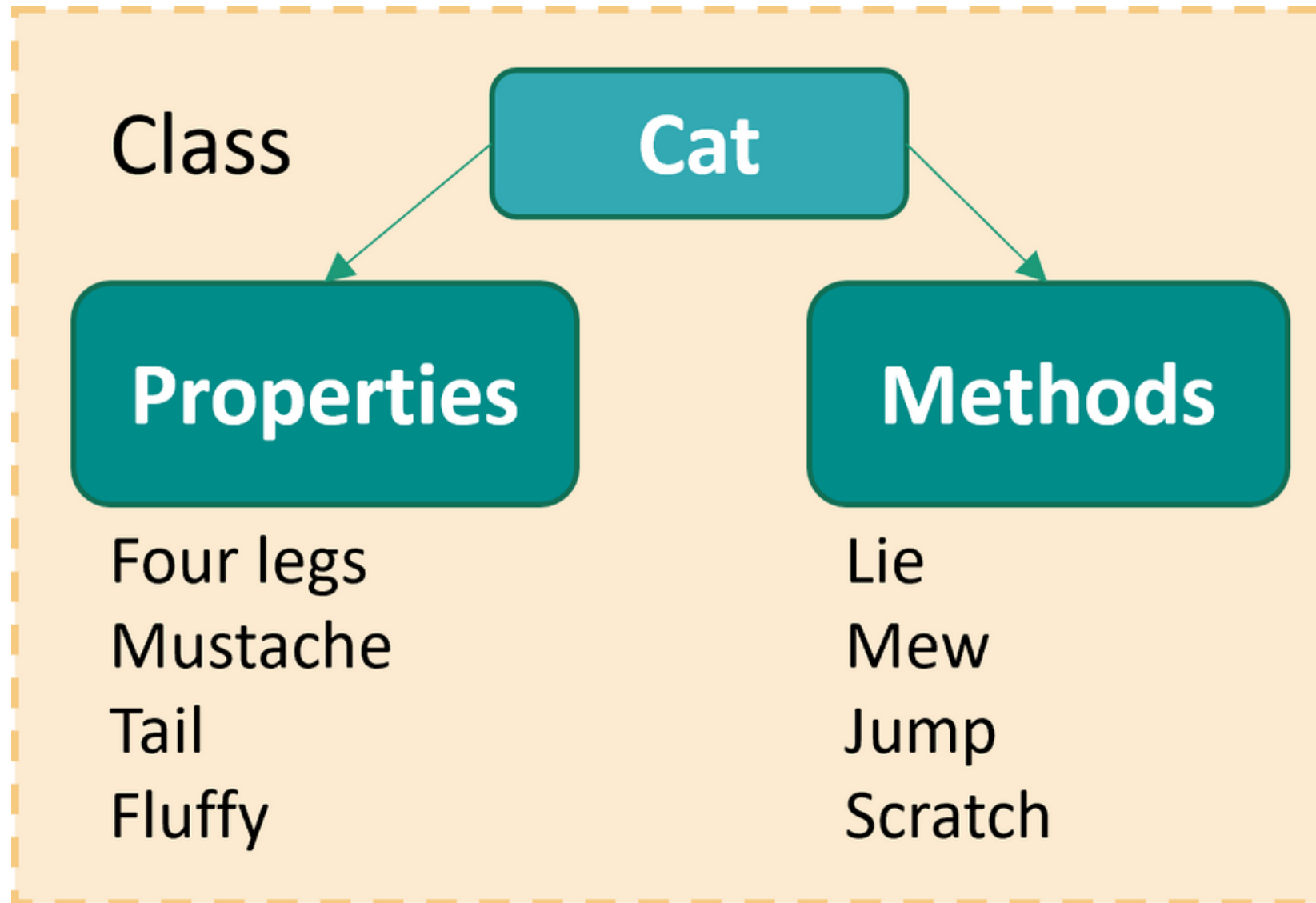
Sınıf Nedir? (Class)

Gerçek dünyadaki nesneler gibi, OOP dünyası da nesneleri ortak davranışlarına ve özelliklerine göre gruplandırır.

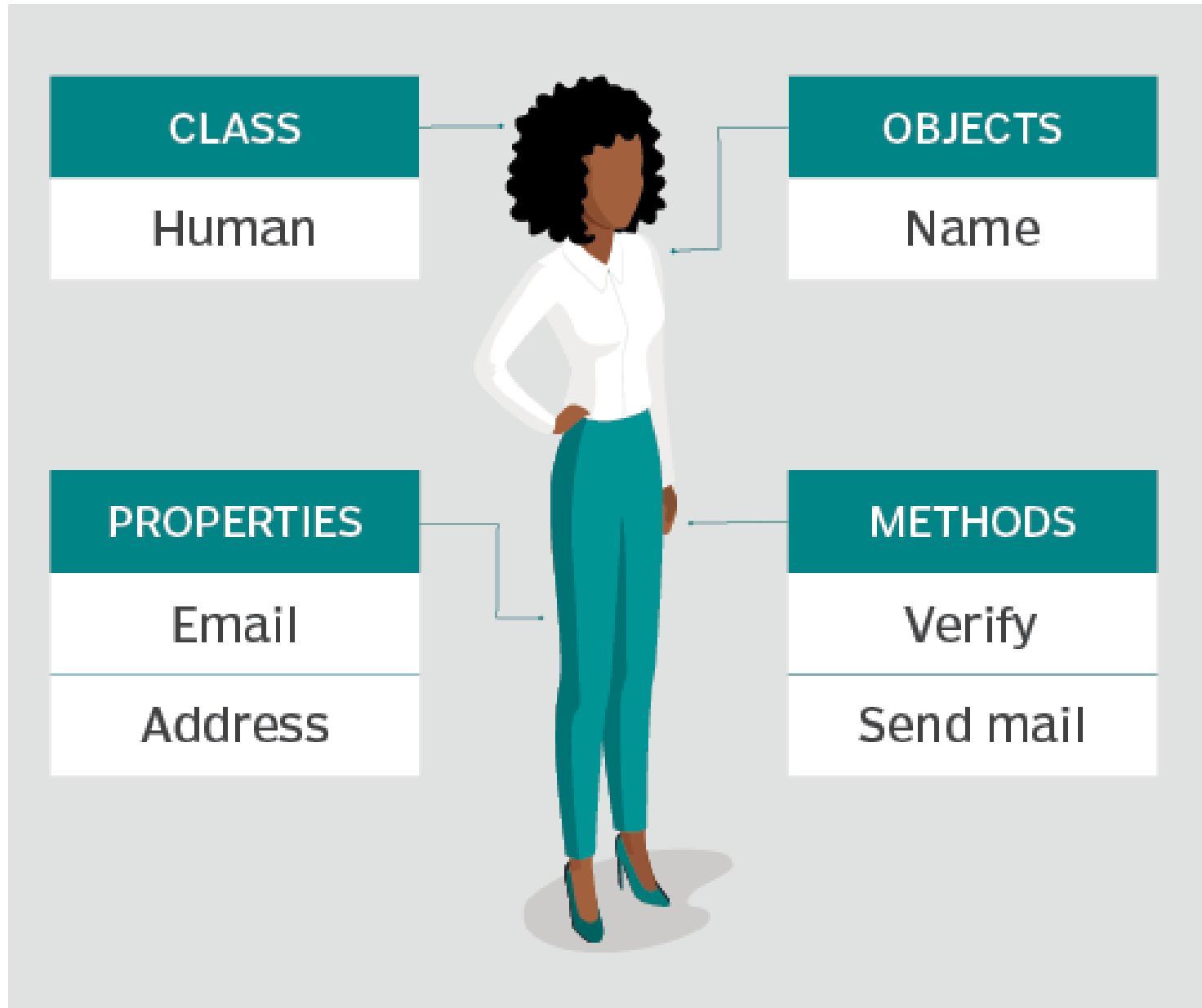
Biyoloji tüm köpekleri, kedileri, filleri ve insanları memeliler olarak sınıflandırır. Paylaşılan özellikler, bu ayrı yaratıklara ortak özellikler verir. Yazılım dünyasında, sınıflar ilgili nesneleri aynı şekilde gruplandırır.

class

Sınıf Nedir? (Class)



Sınıf Nedir? (Class)



İnsan | **Sınıf**

boy

kilo

saç tipi

cinsiyet

yemekYe()

nefesAlVer()

...

...

...

öznitelikler → özellikler

Nesne | Berkecan

boy : 180 cm

kilo : 79 kg

saç tipi : uzun dalgalı

cinsiyet : erkek

yemekYe()

nefesAlVer()

...

...

...



Uygulamalı Ders

İçeriğın devamı için lütfen dersi takip ediniz.



Referanslar

- Sams Teach Yourself Object Oriented Programming in 21 Days, Second Edition
 - Mesleki ve Teknik A. L., Bilişim Teknolojileri, Nesne Tabanlı Programlama
 - Nesne Tabanlı Programlama I, Kitap
-

İletişim Bilgileri

ÖĞR. GÖR. BUSE YAREN TEKİN



bytekin@kastamonu.edu.tr

