

Object Oriented Programming

+

Nesne Yönelimli/Tabanlı Programlama

Günümüzde nesne tabanlı
programlama birçok modern dil
tarafından desteklenir haldedir.

Nesne Tabanlı Programlama

+ Yazılım geliştirme süreçlerini oldukça kısaltan ve sistematik hale getiren bir tekniktir, Nesne Tabanlı Programlama...

Nesne Tabanlı Programlama

Nesne Tabanlı Programlama

+ Gerçek hayatı, programlama için simüle eden nesneleri baz alan bir programlama tekniğidir.

+ Herşey bir nesnedir.

- Sen
- Ben
- Personel
- Ürün
- Satış
- Araba

+ Gerçek bir sistem, nesnel parçalara ayrılır ve bu parçalar(nesneler) arasında ilişkiler kurulur.

+ Nesneler kendi aralarında haberleşebilirler.

Nesnenin Anatomisi

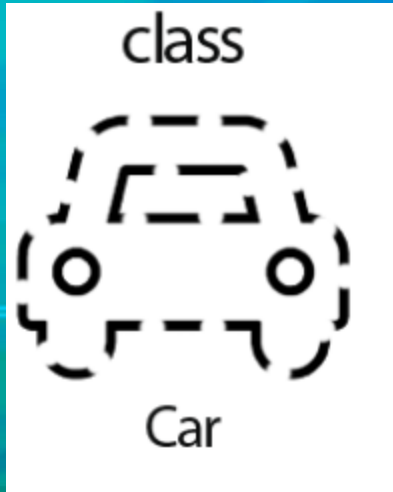


Nesne Kavramı

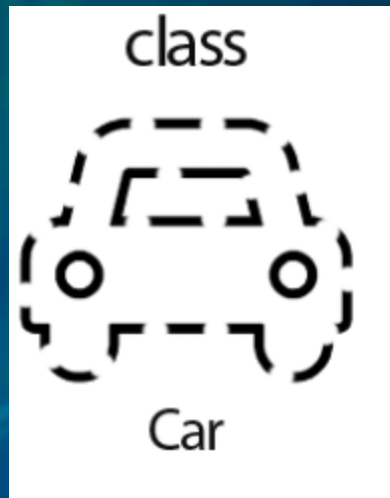
- + Nesne, nesnellik felsefesine dayanan bir kavramdır. Kainattaki her bir şeyi nesne olarak görmek ve o şekilde yorumlamak fikrine dayanır.
- + Nesne, gerçek hayatta elle tutulur, gözle görülür objelerdir. Dolayısıyla programlamada da nesnelerimiz günlük hayattaki nesnelerin muadilidir.
- + Gerçek hayattaki herhangi bir olguyu, nesneyi, objeyi, programlama dünyasında tarif ederken onu bir nesne olarak tarif edecek ve o şekilde modelleyeceğiz.

Nesne Modellemesi

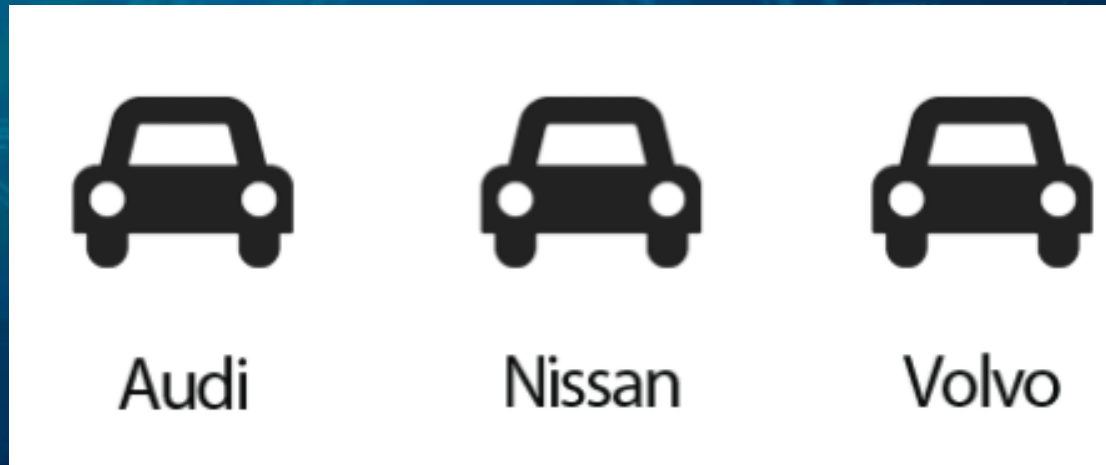
- + Nesnelerin oluşturulabilmesi için öncelikle modellenmesi gerekmektedir.
- + Nesne modeli class ile gerçekleştirilir.



Oluşturulan bu Car modeli sayesinde bizler istediğimiz araba verilerini taşıyan nesneler üretebiliriz...



→ Nesne Modeli



→ Nesne modelinden
üretilen nesneler



Nesne Modeli



+ Nesneler Referans türlü değerlerdir...

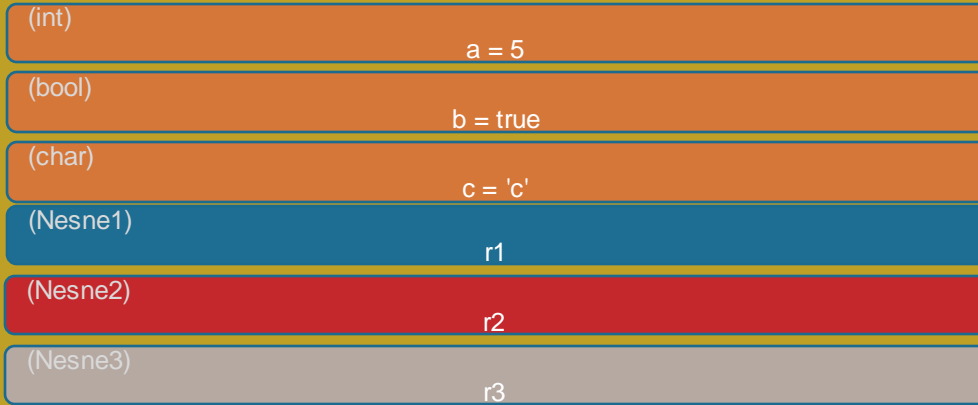
Nesneler Hangi Türdendir?

Peki referans türlü değerler
nelerdir?

Referans Türlü Değerler

RAM

STACK



Kod yazarken Stack'te ki bu değişkenlere isimleri üzerinden erişebilmekteyiz. Örneğin; a dediğimizde 5 değerini elde edebilmekteyiz.

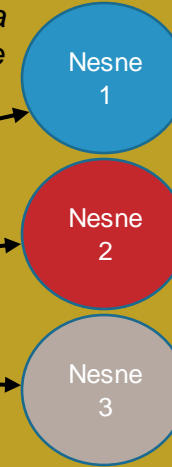
r1 referansı

r2 referansı

r3 referansı

*Nesnelere/Classlara referans
türlü değerler denmesinin
sebebi işte bundandır!*

HEAP



Peki biz Heap'te ki bu nesnelere nasıl erişeceğiz?

- Normal şartlarda biz developer'ların Stack'e direkt erişim hakkı vardır lakin Heap'e erişim hakkımız bulunmamaktadır.
- Heap'te ki nesnelere biz direkt erişemesekte Stack'te ki referanslar erişebilmektedir. Bizde, Stack'te Heap'teki nesneleri işaret eden referanslar tanımlarız. Haliyle Stack'te ki referans'a erişebilir ve dolaylı olarak o referans aracılığıyla Heap'te ki nesneye erişmiş oluruz.

- Değer türlü değişkenler ve değerleri tutulur.
- Referansları tutulur.

Sadece nesneler tutulur.

Sınıf(Class) Kavramı

+