Nesne Tabanlı Programlama

Dr. Öğr. Üyesi Emre Bendeş

Nesne Tabanlı Programlama - NTP (object oriented programming - OOP)

- Bir programlama paradigması
- Problemin tasarlanması için bir yaklaşım
  - Olgulardan sınıflar tasarlanması
  - Sınıflardan oluşturulacak nesneler amaca hizmet etmeli
- NTP Özellikleri
  - Çokbiçimlilik - polimorphizm
  - Miras – inheritance
  - Soyutlama – abstraction
  - Kapsülleme – encapsulation
- Tasarım kuralları
  - Bir sınıfta neler olmalı, nasıl tanımlanmalı

Ne için tasarım ?

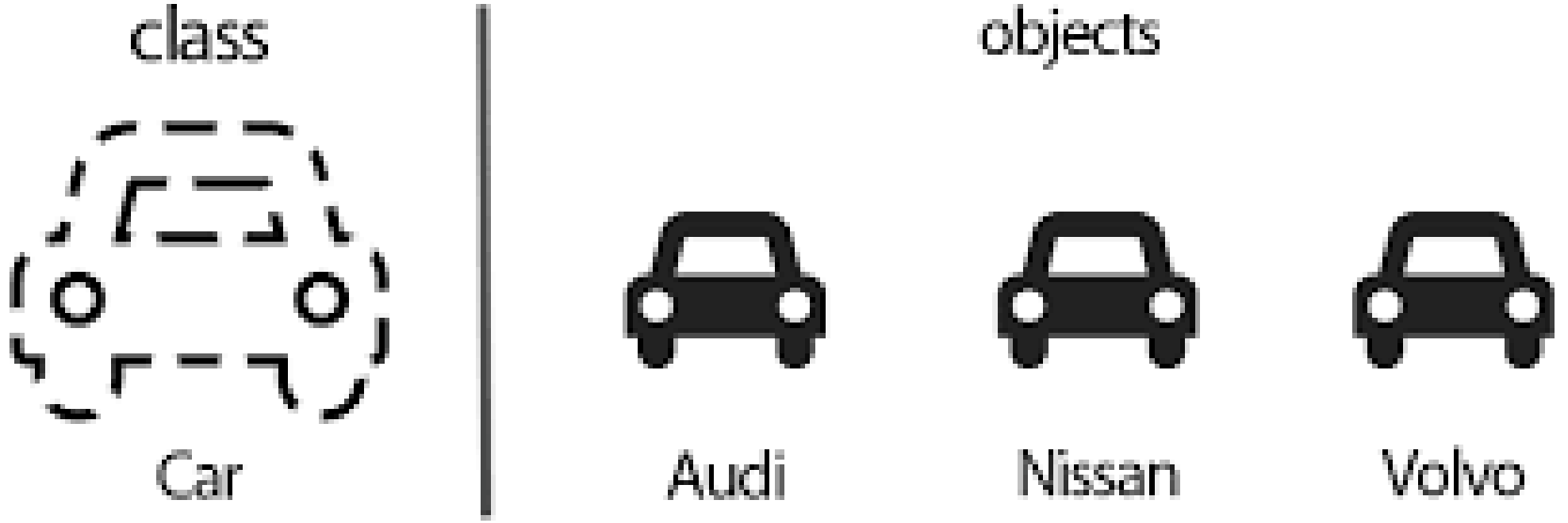
Bireysel projede

- Kod bakımı kolaylığı
- Unutulan kodların hatırlanmasına gerek yok, tasarımın bilinmesi yeterli

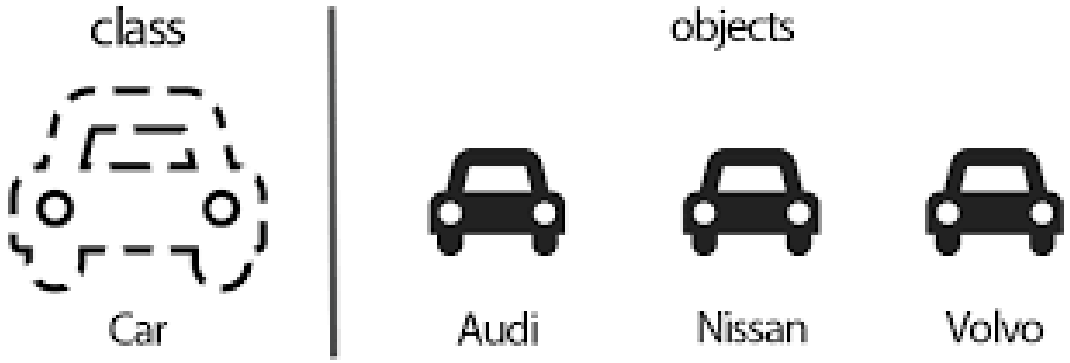
Grup projelerinde

- İş birliğinin kolaylaştırılması
- Modülaritenin sağlanması

Sınıf ve Nesne Ayrımı (class - object)



Sınıf ve Nesne Ayrımı (class - object)



- SINIF (CLASS)
  - Şablondur
    - İçeriği belirler
    - Tanımlamaları yapar
    - Fonksiyonların gövdeleri tanımlanır.
  - Kuralları koyar
  - Bellekte 1 adet bulunur.

- NESNE (OBJECT)
  - Şablondan oluşturulmuş gerçek varlıktır.
    - İçeriğin değerleri nesnelerde tutulur.
    - Program akışında değişebilen durumları vardır.
    - Tanımlanmış fonksiyonları icra edebilir.
  - Kuralları uygular
  - Bellek müsaade ettiği ölçüde oluşturulabilir.

Kazançlar

Modülarite

-Bir Nesnenin kaynak kodu diğer nesnelerden bağımsız olarak yazılabilir ve bakımı yapılabilir.

-Bir kez oluşturulduktan sonra sistem içerisinde kolayca kullanılabilir.

Bilgi koruma

-Sadece nesnenin metotları ile etkileşim sağlanacağından, diğer nesneler, iç işleyiş hakkında bilgi sahibi olamazlar.

Kodun yeniden kullanımı

-Eğer nesne zaten mevcut ise kendi kodunuzda kullanabilirsiniz.

-Bu, işin uzmanlarının karmaşık, görev özelindeki nesneleri derlemesi/test etmesi/ uygulamasına olanak sağlar.

Değiştirilebilirlik ve hata ayıklama kolaylığı

-Belirli bir nesnenin sorunlu olduğu ortaya çıkarsa, onu uygulamanızdan kaldırabilir ve onun yerine farklı bir nesne kullanabilirsiniz.

-Bu, gerçek dünyadaki mekanik sorunları düzeltmeye benzer. Bir cıvata kırılırsa, makinenin tamamını değil, onu değiştirirsiniz.

Java'da Sınıf Tanımlama

```
Public class SınıfIsmi
extends BaseClass
implements IClass1,IClass2
{
.
.
.

}
```

Erişim belirticisi (modifier)	• public, private (private sadece Nested Classes için kullanılabilir)
Sınıf ismi	• Geleneksel olarak ilk karakteri büyük harfdir.
Üst sınıf ismi (superclass),	• Eğer bir sınıftan miras alınacaksa **extents** ifadesi ile üst sınıfın ismi belirtilir.
Uygulanan interfacelerin virgüller ile ayrılmış listesi	• Eğer bir interface uygulanacak ise bu kısım **implements** ifadesi ile bulunur.
Sınıfın gövdesi,	• {} ile sınırları belirlenir.

Bir Java Sınıfında Neler Bulunabilir

- Üye değişkenler-alanlar (member variables - fields)
  - Primitive
  - Referans değişkenleri (diğer nesneler)
- Başlangıç blokları
  - Normal
  - Static
- Yapıcı Metodlar (consructor)
- Metodlar
  - Sınıfa ait işlemleri yerine getiren fonksiyonlar
- İç sınıflar

Paket, Arayüz, Miras kavramları

- Paket
  - Aynı amaca hizmet eden sınıfları bir araya toplayan yapıdır
  - İşletim sistemindeki dosya yapısına benzer
  - `package paketİsmi;`
- Miras
  - Alt sınıf- üst sınıf(ebebeyn- çocuk sınıf)
  - Her sınıf Object sınıfını miras alır
- Arayüz (interface)
  - Gövdesi olmayan, yapılması gereken işlemleri tanımlayan sınıf yapısı
  - `implements arayüzIsmi`

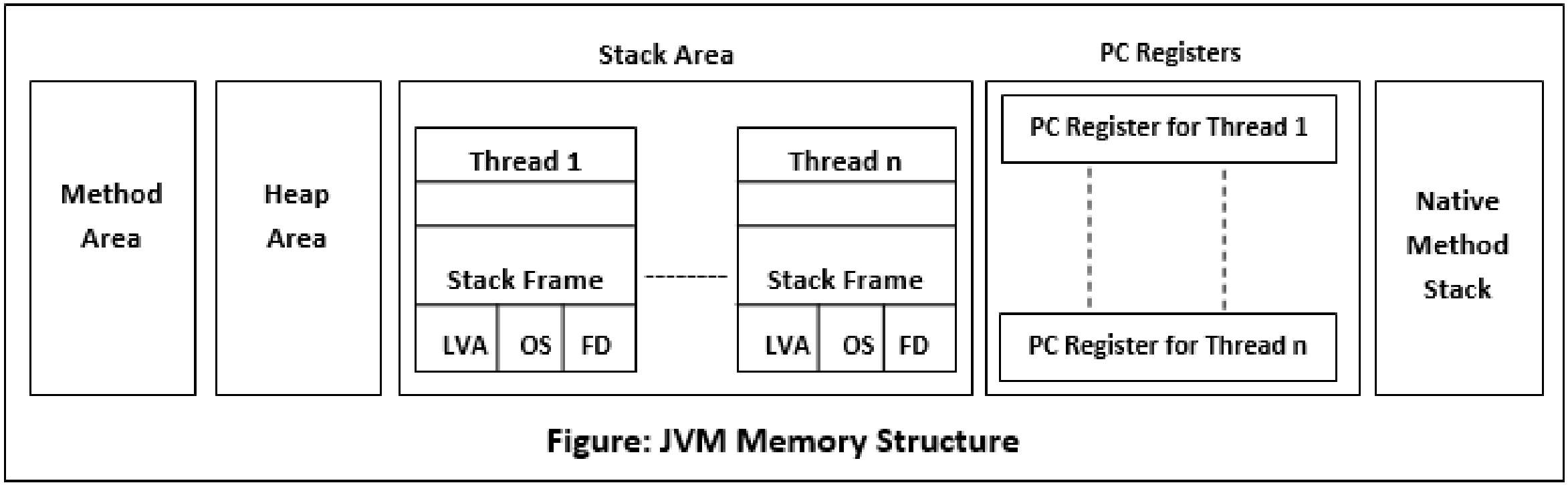


Figure: JVM Memory Structure