NESNEYE DAYALI PROGRAMLAMA
Şubat 2022
Dr.Öğr.Üyesi Çiğdem BAKIR

GENEL BİLGİLER

BAŞARIM DEĞERLENDİRME

- Vize
- 3 Ödev, Proje, Laboratuvar
- Final

KAYNAKLAR:

- Java Programlama (Nesne Yönelimli):
  - Java How to Program, Deitel & Deitel, Prentice-Hall. (≥ 9th ed. Early Objects Version)
  - Core Java 2 Vol. I, Horstmann & Cornell, Prentice-Hall. (≥ 7th ed.)
- Java Programlama (Yapısal):
  - Algoritma Geliştirme ve Programlamaya Giriş, Fahri Vatansever, Seçkin Y.
- UML:
  - UML Distilled, 3rd ed. (2003), Martin Fowler, Addison-Wesley.

KAYNAKLAR:

- Java Programlama (Nesne Yönelimli):
  - Java How to Program, Deitel & Deitel, Prentice-Hall. (≥ 9th ed. Early Objects Version)
  - Core Java 2 Vol. I, Horstmann & Cornell, Prentice-Hall. (≥ 7th ed.)
  - Java Tutorial (Java yla alakalı tüm dokümanlar mevcuttur)

BAŞARI DEĞERLENDİRME

Vize: %25
Ödev (3 adet): %10
Laboratuvar: %20
Proje: %10
Final: %35

GENEL BİLGİLER

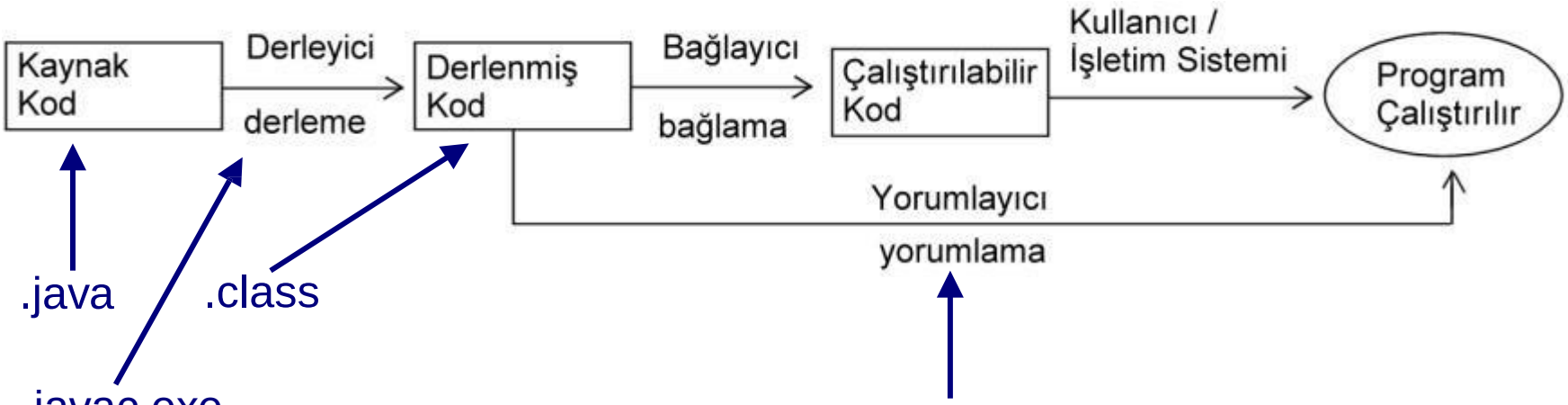
DERS İÇERİĞİ

- Temel içerik:
  - Java programlama diline genel bakış
  - Nesne ve Sınıf Kavramları
  - UML Sınıf Şemaları
  - Kurucular ve Sonlandırıcılar
  - Nesne Davranışı ve Metotlar
  - Nesne ve Sınıfların Etkileşimleri ve İlişkileri
  - UML Etkileşim (Sıralama) Şemaları
  - Kalıtım ve Soyut Sınıflar
  - Nesne Arayüzleri ve Çoklu Kalıtım
  - Çokbiçimlilik, Metotların Yeniden Tanımlanması ve Çoklu Tanımlanması

JAVA PROGRAMLAMA DİLİNE GENEL BAKIŞ

JAVA UYGULAMA ORTAMI

- Java yorumlanan bir dildir.



Java Sanal Makinesi
(java.exe, javaw.exe)
(JVM: Java Virtual Machine)
JRE ile gelir
(Java Runtime Environment)

JAVA PROGRAMLAMA DİLİNE GENEL BAKIŞ

JAVA UYGULAMA ORTAMI

- Standart Sürüm – Standard Edition:
  - Masaüstü ve sunucu bilgisayarlarda çalışabilecek uygulamalar geliştirmeye yönelik.
- Mikro Sürüm – Micro Edition:
  - Cep telefonu ve avuç içi bilgisayarları gibi taşınabilir cihazlara yönelik.
  - Standart sürümündeki bileşenlerin bir kısmını daha az işlevsellikle içerir.
- Şirket Sürümü – Enterprise Edition:
  - Çok katmanlı uygulamalar ile web hizmetleri uygulamalarını kullanıma açmak için gerekli hizmet yazılımını içerir.
    - Sun Java System Application Server
    - IBM Websphere
    - BEA WebLogic
    - Apache Tomcat
    - …

JAVA PROGRAMLAMA DİLİNE GENEL BAKIŞ

JAVA SÜRÜMLERİ

- Eski ve yeni sürümlendirme:

Eski Sürüm (Developer Version)	Yeni Sürüm (Product Version)
Java 1.0	
Java 1.1	
Java 1.2	Java 2 Platform
Java 1.3	Java 2 SE 3 (J2SE3)
Java 1.4	J2SE4
Java 1.5	J2SE5
Java 1.6 {Sun}	Java Platform Standard Edition, version 6 (JSE6)
Java 1.7 {Oracle}	Java Platform Standard Edition, version 7 (JSE7)
Java 1.8	Java Platform Standard Edition, version 8 (JSE8)

JAVA PROGRAMLAMA DİLİNE GENEL BAKIŞ

JAVA SÜRÜMLERİ

- Mevcut durum:

Yeni Sürüm	Açıklama
Java SE 8	Son halka açık güncelleme (eski sistemler için)
Java SE 9,10,12	Artık desteklenmiyor
Java SE 11	Uzun süreli destek (Eylül 2018)
Java SE 13	Güncel sürüm (Eylül 2019)

JAVA PROGRAMLAMA DİLİNE GENEL BAKIŞ

JAVA SÜRÜMLERİ

- Sürümlendirme ayrıntıları:
  - JDK 8.0.241:
    - Java Version 8.0, update 241.
  - JDK 13.0.2
    - Java Version 13.0, update 2.
    - Update:
      - Hata düzeltme, daha iyi başarım ve güvenlik nedenleriyle güncellemeler.
      - Birkaç aylık aralıklarla.
- Nereden indirmeli?
  - Oracle.com/java
  - Dokümantasyonu da ayrıca indirip açınız.

JAVA PROGRAMLAMA DİLİNE GENEL BAKIŞ

ÜCRETSİZ JAVA GELİŞTİRME ORTAMLARI

- Eclipse: http://www.eclipse.org
  - Ayrıca indirilir.
  - UML için eUML2 plug-in'i kurulmalı.
  - GUI için ayrı plug-in kurulmalı.
  - Yönetici olarak kurulum gerekmiyor, unzip yetiyor.
- NetBeans:
  - JSE dağıtımı ile birlikte (seçimlik)
  - UML için ayrı plug-in gerek.
    - Kuran bana da isim söylesin.
  - Dahili GUI editörü var.
  - Yönetici olarak kurulum gerektiriyor.

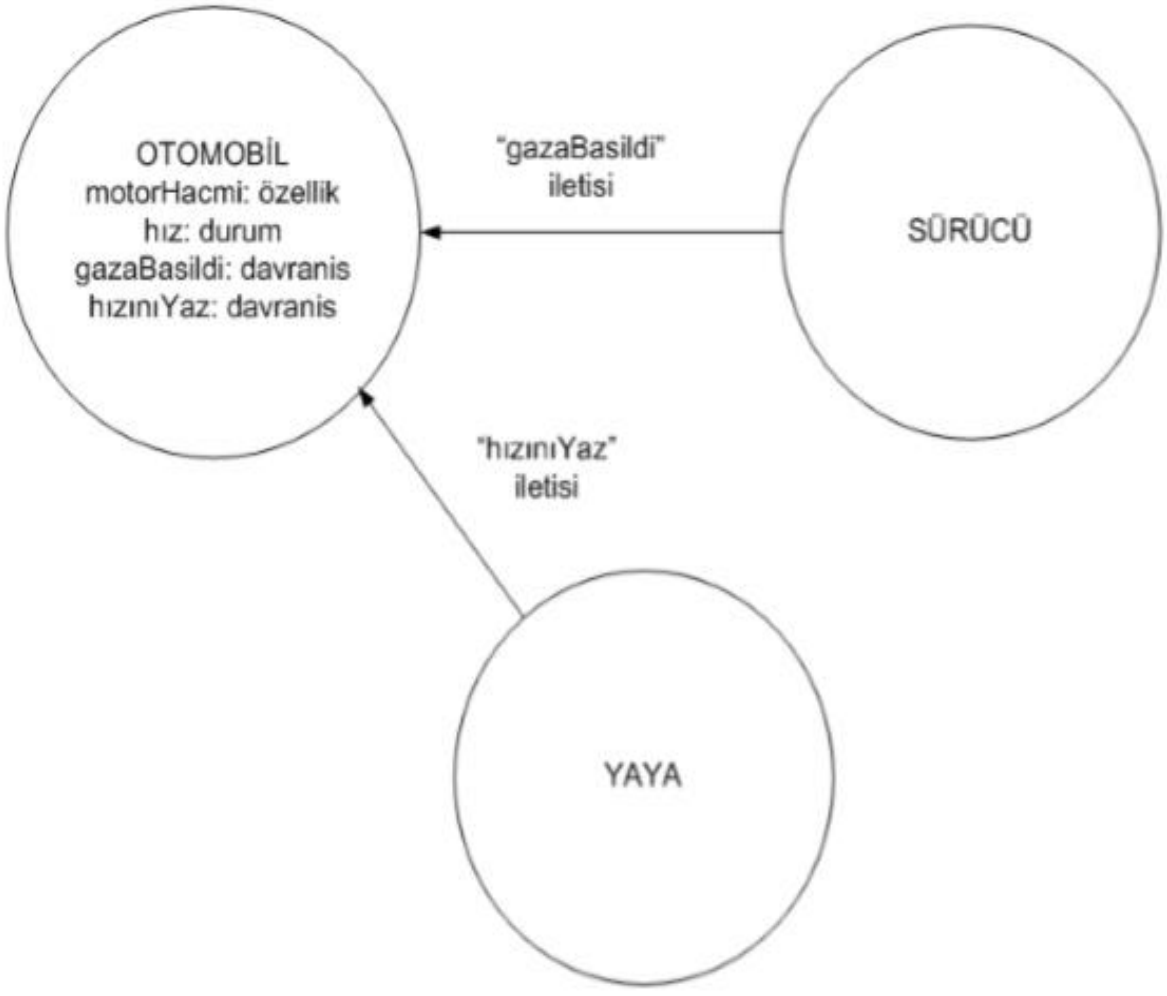
ÜCRETSİZ UML MODELLEME ORTAMLARI

- Violet UML: Hafif sıklet, bizim için yeterli
- Argo UML

NESNEYE DAYALI PROGRAMLAMA

Dr.Öğr.Üyesi Çiğdem BAKIR

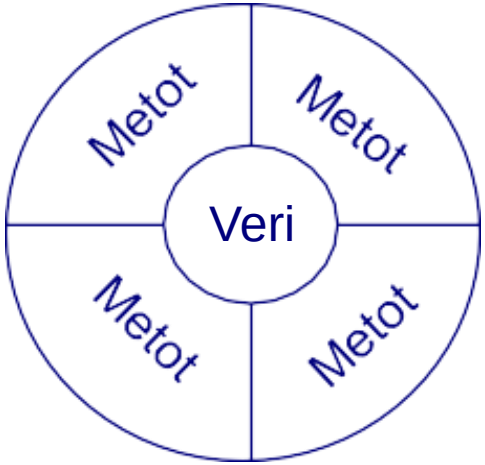
DERS NOTLARI:
A. NESNEYE YÖNELİMİN TEMELLERİ


OTOMOBİL
motorHacmi: özellik
hız: durum
gazaBasildi: davranis
hızınıYaz: davranis
"gazaBasildi"
iletisi
SÜRÜCÜ
"hızınıYaz"
iletisi
YAYA

SINIFLAR, NESNELER VE ÜYELER

NESNE

- Nesne: Nitelikler ve tanımlı eylemler içeren, temel programlama birimi.
  - Nesne ≈ bir gerçek dünya varlığına denk gelir.
    - Nesneler değişkenlere de benzetilebilir
    - Süpermen de sıradan insanlara benzetilebilir!
  - Nesnenin nitelikleri ≈ Nesne ile ilgili veriler.
  - Eylemler ≈ Nesnenin kendi verisi üzerinde(*) yapılan işlemler ≈ Nesnenin kendi verileri ile çalışan metotlar (fonksiyonlar).
    - * Çeşitli kurallar çerçevesinde aksi belirtilmedikçe.
    - Metotlara parametre(ler) de verilebilir.
  - Sarma (Encapsulation): Veri ve eylemlerin birlikteliği.
    - Verilere, eylemler üzerinden erişilir.



Özellik (property) NYP bağlamında, nesnenin durumunu oluşturan, nesnenin ait olduğu sınıfın içine kodlanmış herbir değişkene verilen isimdir. Özellik, Java bağlamında nitelik (attribute) olarak adlandırılır.

Davranış (behaviour) NYP bağlamında, nesnenin durumunda değişiklik yapabilen, nesnenin ait olduğu sınıfın içine kodlanmış herbir işleve verilen isimdir. Davranış, Java bağlamında yöntem (method) olarak adlandırılır.

Java'da Methodlar

Java'da methodlar aynen C'deki işlevler gibi yazılır. Kısaca hatırlayalım: Yöntemlerin belirli bir biçime uygun yazılmaları gerekir. Bu biçim şu şekildedir:

```
tür yöntemAdi(parametreListesi)
{

}
```

tür: Yöntemin döndüreceği değerin türüdür. Örneğin verilen gerçel sayının karekökünü bulup döndüren bir yöntem yazacaksak, yöntemden geriye dönecek değerin türü gerçel sayı tipinde olacaktır. O zaman bu alana gerçel sayı türünü ifade eden double yazarız. Eğer yöntemin geriye herhangi bir değer döndürmesi beklenmiyorsa, bu alana değer dönmeyeceğini belirtmek üzere void yazılır.
yöntemAdi: Yönteme verilen addır. Yöntem adı, yöntemin ne iş yaptığını ifade edecek şekilde özenle seçilmelidir. Böylece hem kodun okunurluğu sağlanabilir hem de daha sonra hangi yöntemin kullanılacağına kolaylıkla karar verilebilir. Örneğin karekök bulan yöntem için karekokBul adı kullanılabilir.
parametreListesi: Bir yöntem, kendisine verilen değerler üzerinde çalışabilir. Karekök bulan yöntemin karekökünü bulacağı değer o yöntemin parametresidir. Parametre listesi, sıfır ya da daha çok parametreden oluşur. Her parametrenin önce türü sonra adı yazılır ve parametrelerin arasına “,” (virgül) koyulur.

```
int obeb(int sayi1, int sayi1) {
 // buraya obeb bulma algoritması kodlanacak }
}
```

NESNE, SINIF ÖRNEĞİ

```
class Zaman {

 int saat;
int dakika;
int saniye;
  /**
  * Her çağrıldığında nesneyi bir saniye sonrasına götüren yöntem.   * saniye 59, dakika 59, saat ise 23'ten büyük olamaz.    */

            void ilerle() {
               saniye++;
               if (saniye == 60) {
                   saniye = 0;    dakika++;
                    if (dakika == 60) {
                        dakika = 0; saat++;
                         if (saat == 24) {
                               saat = 0; }
                    }
               }
         }
 }
```

Zaman nesnesinin durumunu soran yöntem

```
/**
 * Çağrıldığı zaman nesnenin o anki durumunu ekrana yazar.
*/
void zamaniYaz()
      {

        System.out.println("Zaman: " + saat + ":" + dakika + ":"
+ saniye);

     }
```

Zaman sınıfını kullanan Program sınıfı

```
class Program {
        public static void main(String[] args) {
            Zaman zamanNesnesi = new Zaman();
            zamanNesnesi.ilerle();
            zamanNesnesi.zamaniYaz();
            }
 }
```

SINIFLAR, NESNELER VE ÜYELER

SINIF

- Sınıf: Nesneleri tanımlayan şablonlar.
  - Şablon = Program kodu.

```
class myClass {
   /*
       program kodu
   */
}
```

SINIFLAR, NESNELER VE ÜYELER

NESNELER VE SINIFLAR

- Örnek nesne: Bir otomobil.
  - Nitelikler: Modeli, plaka numarası, rengi, vb.
    - Niteliklerden birinin tekil tanımlayıcı olması sorgulama işlerimizi kolaylaştıracaktır.
  - Eylemler: Hareket etmek, plaka numarasını öğrenmek, satmak, vb.
- Örnek sınıf: Taşıt aracı.
  - Nitelikleri ve eylemleri tanımlayan program kodu.
- Gerçek dünya benzetimi:
  - Nesne: Bir varlık olarak bir otomobil.
  - Sınıf: Dilbilgisi açısından bir genel isim olarak taşıt aracı.
- Bir nesneye yönelik program içerisinde, istenildiği zaman herhangi bir sınıftan olan bir nesne oluşturulabilir.
- Aynı anda aynı sınıftan birden fazla nesne etkin olabilir.

SINIFLAR, NESNELER VE ÜYELER

NESNELER VE SINIFLAR

- Bir nesnenin nitelikleri iki (!) çeşit olabilir:
  - Tamsayı, karakter gibi tek bir birim bilgi içeren 'ilkel' veriler,
  - Aynı veya başka sınıftan olan nesneler.
    - Sonsuz sayıda farklı sınıf oluşturulabileceği için, 'iki çeşit' deyimi çok da doğru değil aslında.
- Nesnenin niteliklerinin bir kısmı ilkel, bir kısmı da başka nesneler olabilir.

SINIFLAR, NESNELER VE ÜYELER

TERMİNOLOJİ VE GÖSTERİM

- NYP Terminolojisi:
  - Veri: Üye alan (Member field) = Nitelik (attribute)
  - Durum bilgisi: Nesnenin belli bir andaki niteliklerinin durumları
  - Eylem: Metot (Member method)
  - Nesnenin (Sınıfın) üyeleri = Üye alanlar + üye metotlar
  - Sınıf = tür = tip.
  - S sınıfından oluşturulan n nesnesi = n nesnesi S sınıfının bir örneğidir (instance)

- UML Gösterimi:

Sınıf: Sınıf şemasında

Nesne: Etkileşim şemasında

SINIFLAR, NESNELER VE ÜYELER

TERMİNOLOJİ VE GÖSTERİM

- İki tür UML etkileşim şeması (interaction diagram) vardır:
  1. Sıralama şeması (Sequence diagram)
  2. İşbirliği şeması (Collaboration diagrams)
- Bu derste sıralama şemaları çizeceğiz.
  - "Etkileşim" bu tür şemaların özünü çok iyi tarif ediyor. O yüzden "sıralama" ve "etkileşim" terimlerini birbirlerinin yerine kullanabilirim.

SINIFLAR, NESNELER VE ÜYELER

HER NESNE FARKLI BİR BİREYDİR!

- Aynı türden nesneler bile birbirinden farklıdır:
  - Aynı tür niteliklere sahip olsalar da, söz konusu nesnelerin nitelikleri birbirinden farklıdır = Durum bilgileri birbirinden farklıdır.
  - Durum bilgileri aynı olsa bile, bilgisayarın belleğinde bu iki nesne farklı nesneler olarak ele alınacaktır.
  - Bu farklılığı sağlamak üzere, her nesne programcının ulaşamadığı bir tekil tanımlayıcıya (UID: unique identifier) sahiptir.
    - Hiçbir nesnenin tanımlayıcısı birbiri ile aynı olmayacaktır.
    - UID'yi JVM kotarır.
- Örnek: Sokaktaki arabalar.
  - Örnek nitelikler: Modeli, rengi.
  - Modelleri ve renkleri farklı olacaktır.
  - Aynı renk ve model iki araba görseniz bile, plakaları farklı olacaktır.
  - Plaka sahteciliği sonucu aynı plakaya sahip olsalar bile, bu iki araba birbirlerinden farklı varlıklardır.

SINIFLAR, NESNELER VE ÜYELER

HER NESNE FARKLI BİR BİREYDİR! (devam)

- Aynı tür bile olsa, her nesnenin durum bilgisi farklı olduğu için, aynı türden iki nesne bile aynı mesaja farklı yanıt verebilir.
  - Örnek: Bana adımı sorsanız "Yunus" derim, sizin aranızda kaç Yunus var?
  - Kaldı ki, nesneye mesaj gönderirken farklı parametreler de verebilirsiniz. Aynı nesneye aynı mesaj farklı parametre ile giderse, geri dönen yanıtlar da farklı olacaktır.
- Terminoloji: Aynı türden iki nesne, aynı mesaja farklı yanıtlar verir.

SINIFLAR, NESNELER VE ÜYELER

NESNELERE MESAJ GÖNDERME

- Bir nesneye neden mesaj gönderilir?
  - Ona bir iş yaptırmak için
  - Bir üyesine erişmek için.

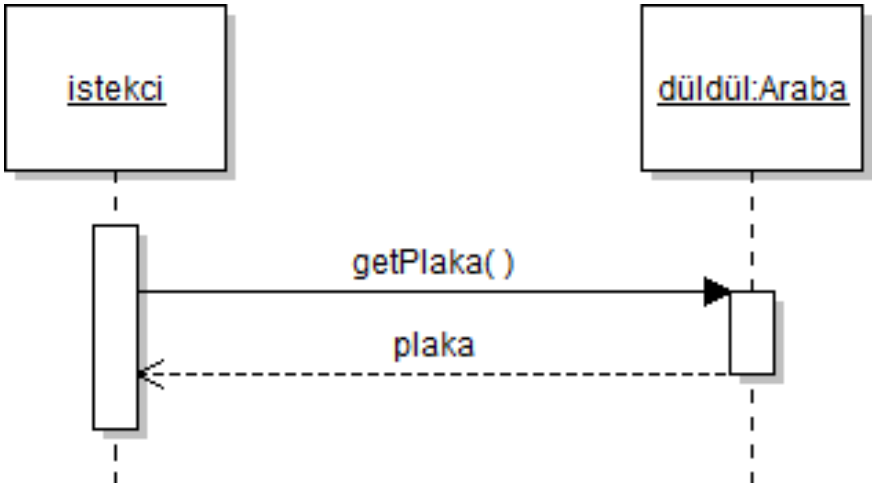
ÜYELERE ERİŞİM

- Üye alana erişim:
  - Üyenin değerini değiştirmek (setting)
  - Üyenin değerini okumak (getting)
    - (Değiştirmeden herhangi bir işlemde kullanmak)
- Üye metoda erişim:
  - Bir eylemler sürecini varsa kendine özgü çalışma parametreleri ile yürütmek.
  - Fonksiyon çağırmak gibi, ama unutmayın: Aksi belirtilmedikçe metot, üyesi olduğu nesnenin üyeleri ile çalışır.
    - Aksinin nasıl belirtileceğini ileride nesneler arasındaki ilişkileri öğrenince göreceksiniz.

SINIFLAR, NESNELER VE ÜYELER

TERMİNOLOJİ VE GÖSTERİM

- NYP Terminolojisi:
  - Bir nesnenin diğer bir nesnenin bir üyesine erişmesi, bir nesnenin diğerine bir mesaj göndermesi olarak da tanımlanır.
  - Bir nesneye yönelik program, nesneler arasındaki mesaj akışları şeklinde yürür.
- UML Gösterimi:



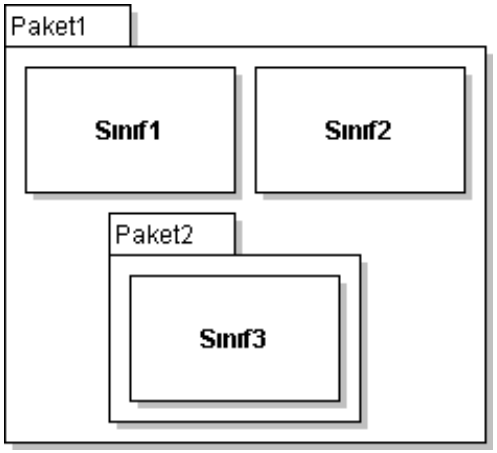
- Anlamı:
  - istekçi adlı bir nesne vardır.
  - istekçi nesnenin sınıfı belli değil.
  - düldül adlı bir nesne vardır.
  - düldül nesnesinin sınıfı Araba'dır.
  - Araba sınıfının getPlaka adlı bir metodu vardır.
  - istekçi nesne düldül nesnesine getPlaka mesajı gönderir.
  - düldül nesnesi bu mesaja yanıt olarak kendi plakasını döndürür.
- Kod Gösterimi:

```
duldul.getPlaka();
//Nesne adını Türkçe veremiyoruz.
```

SINIFLAR, NESNELER VE ÜYELER

PAKETLER

- Sınıflar paket (package) adı verilen mantıksal kümelere ayrılabilir.
- Amaç: Kendi içerisinde anlam bütünlüğü olan ve belli bir amaca yönelik olarak birlikte kullanılabilecek sınıfları bir araya toplamaktır.



- Bir paketteki sınıfları koda ekleme:

```
import paket1.Sınıf1;
import paket1.*;
import paket1.Paket2.*;
```

  - paket1 eklenince alt paketi olan paket2 içindeki sınıflar eklenmiş olmaz.
- Paketler, aynı adlı sınıfların birbirine karışmamasını da önler:
  - Sınıf adı aynı bile olsa farklı paketlerde bulunan sınıflar, belirsizlik oluşturmaz.

```
java.io.File
com.fileWizard.File
```

- Paket hiyerarşisi, aynı zamanda dosya hiyerarşisidir.

```
com.fileWizard.File -> com\fileWizard\File.java
```

SINIFLAR, NESNELER VE ÜYELER

GÖRÜNEBİLİRLİK KURALLARI VE VERİ GİZLEME

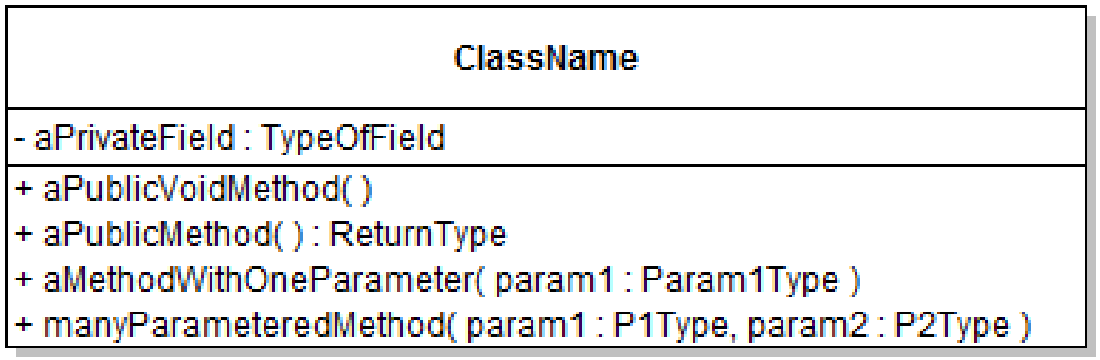
- Bir nesne, kendi sınıfından olan diğer nesnelerin ve bizzat kendisinin bütün üyelerine erişebilir,
- Ancak bir nesnenin üyelerine diğer sınıflardan olan nesnelerin erişmesi engellenebilir.
- Veri Gizleme İlkesi (information hiding):
  - İlke olarak, bir sınıfın içsel çalışması ile ilgili üyeler diğerlerinden gizlenir.
  - Böylece bir nesne diğerini kullanmak için, kullanacağı nesnenin ait olduğu sınıfın iç yapısını bilmek zorunda kalmaz.
- Örnek: TV çalıştırmak için uzaktan kumandalardaki ses ayarı, kanal değiştirme ve güç düğmelerinin evrensel işaretlerini tanımak yeterlidir;
  - Televizyonun içinde katot tüpü adlı bir cihaz olduğunu bilmek gerekmez.
  - Böylece LCD, plazma gibi yeni teknolojiler kullanıcıyı yeniden eğitmeye gerek kalmadan televizyonlarda kullanılabilir.

- Örnek: Arkadaşınızsizden borç para istedi.
  - Borç verirsiniz ya da vermezsiniz.
  - Arkadaşınızın sizin aylık gelirinizi, ATM kartınızın şifresini, vb. bilmesi gerekmez.

SINIFLAR, NESNELER VE ÜYELER

GÖRÜNEBİLİRLİK KURALLARI VE VERİ GİZLEME

- Genel Görünebilirlik Kuralları (Visibility rules):
  - Public: Bu tip üyelere erişimde hiç bir kısıtlama yoktur.
  - Private: Bu tip üyelere başka sınıflardan nesneler erişemez, yalnız kendisi ve aynı türden olan diğer nesneler erişebilir.
- UML Gösterimi:



- Ayrıca (derste sorumlu değilsiniz):
  - protected: #
    - Kalıtım ile ilgili (paketteki diğer sınıflara ve alt sınıflarına açıktır)
  - package: ~
    - Paketteki diğer sınıflara açıktır
    - Java'da varsayılan kural

SINIFLAR, NESNELER VE ÜYELER

GÖRÜNEBİLİRLİK KURALLARI VE VERİ GİZLEME

- Veri gizleme ilkesi her zaman için mükemmel olarak uygulanamaz.
  - Bir sınıftaki değişiklik sadece o sınıfı etkilemekle kalmaz, ilişkide bulunduğu başka sınıfları da etkileyebilir.
  - Veri gizleme ilkesine ne kadar sıkı uyulursa, değişikliğin diğer sınıflara yayılması olasılığı veya değişiklikten etkilenen sınıf sayısı da o kadar azalır.
- Veri gizleme ilkesine uyulmasını sağlamak için:
  - Üye alanlar private olarak tanımlanır, ve:
  - Değer atayıcı ve değer okuyucu metotlar kullanılır.
  - Bu ilkeye uymazsanız gitti en az 5 puan!
- Değer atayıcı ve değer okuyucu metotlar (erişim metotları):
  - Değer atayıcı (Setter) metot: Bir nesnenin bir üye alanına değer atamaya yarar.
  - Değer okuyucu (Getter) metot: Bir nesnenin bir üye alanının değerini öğrenmeye yarayan metot.
  - Adlandırma: getUye, setUye

SINIFLAR, NESNELER VE ÜYELER

GÖRÜNEBİLİRLİK KURALLARI VE VERİ GİZLEME

- Örnek:

- Üyelere erişim kurallarında istenen değişikliklerkolaylıkla yerine getirilebilir.
  - Örneğin plaka içeriğinin okunması serbest olmakla birlikte bu alana değer atanmasının sadece ilgili paket içerisindeki sınıflar tarafından yapılması gerektiğinde, getPlaka metodu public olarak bırakılıp setPlaka metodu paket düzeyi görünebilirliğe alınır

SINIFLAR, NESNELER VE ÜYELER

ÜYELERİN ÖZEL DURUMLARI

- Statik üye alanlar:
  - Aynı türden nesnelerin bile durum bilgisi farklıdır (gördük),ancak:
  - Kimi zaman **aynı tipten tüm nesnelerin ortak bir üye alanı paylaşması** istenilebilir.
  - Bu durumda üye alan **`static`** olarak tanımlanır.
  - **SınıfAdı.üyeAdı** şeklinde sınıf üzerinden kullanılırlar, nesneler üzerinden kullanılmazlar.
  - Örnek: Her binek otomobilin 4 tekerleği vardır.
- Statik üye metotlar:
  - Aynı türden iki nesne, aynı mesaja farklı yanıtlar verir (gördük), ancak:
  - Kimi zaman **aynı tipten tüm nesnelerin aynı mesajın aynı şekilde çalışması** istenilebilir.
  - Bu durumda üye metot **`static`** olarak tanımlanır.
  - Statik metot içerisinde yalnız statik üyeler kullanılabilir.
  - Statik üye alana erişim metotları da statik tanımlanır.
  - **SınıfAdı.üyeAdı()** şeklinde kullanılırlar.

SINIFLAR, NESNELER VE ÜYELER

ÜYELERİN ÖZEL DURUMLARI

- Final üye alanlar:
  - Bir alanın değerinin sürekli olarak aynı kalması istenebilir.
  - Bu durumda üye alan **`final`** olarak tanımlanır.
  - Final üyelere yalnız bir kez değer atanabilir.
  - Örnek: Bir arabanın şasi numarası o araba fabrikadan çıkar çıkmaz verilir ve bir daha değiştirilemez.
- Final üye metotlar:
  - Sınırlı kullanım alanı: Kalıtım ile yeniden tanımlanamazlar (ileride).

DİKKAT EDİLECEK NOKTALAR

- Bir üye, hem final hem de static olabilir.
- Tanımları ve adları gereği final ve static kavramları birbiriyle karıştırılabilir:
  - Final: Bir kez değer atama
  - Static: Ortak kullanım. Sözlük karşılığı durağan, ancak siz ortak diye düşünün.