

Nesneye Yönelik Yaklaşım ve Sınıf Kavramı

Nesneye Yönelik Yaklaşım ve Sınıf Kavramı

- Problem seti içerisinde **kavramsal bir olgu** (köpek, bina, hesap...) çözüm setinde **doğrudan bir nesne** olarak karşılık bulur
- Problem modeli ve çözüm modelindeki bu ortak yapılara **nesne** adı verilmektedir
- Problemi oluşturan **olguların** çözüm içerisinde **doğrudan ifade edilebilmesini** sağlar
- Yazılım sisteminde bütün **herşey nesne** olarak ifade edilir

Nesneye Yönelik Yaklaşım ve Sınıf Kavramı

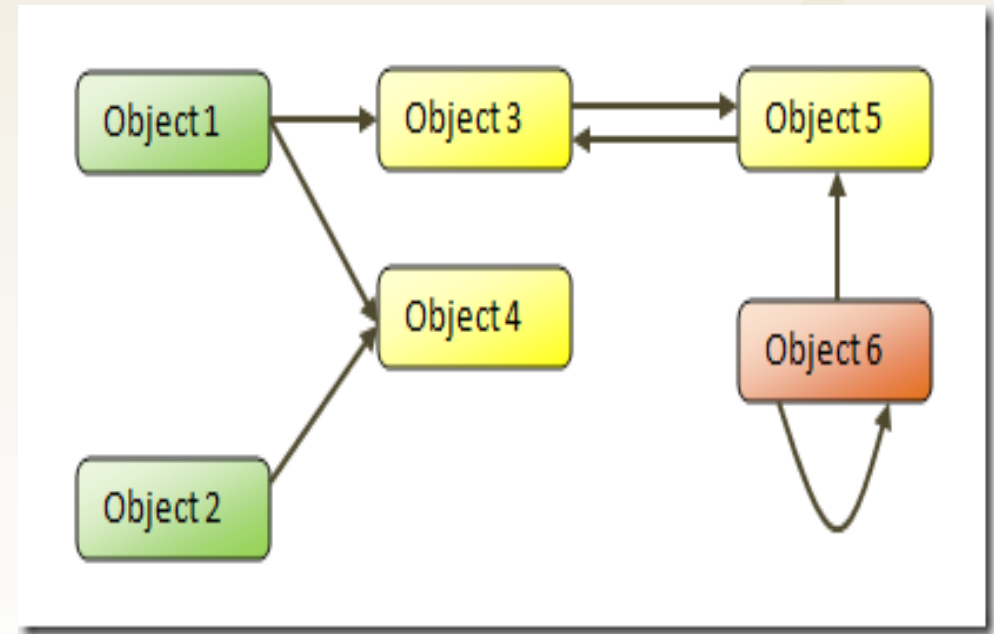
- Nesnelerin özelliklerini ve davranışlarını belirleyen yapılara **sınıf** adı verilir
- Sınıflar **bir nesne grubuna has özellikleri ve davranışları** tanımlar
- Bunlara **attribute** ve **metot** adı verilir
- **Veri (state) ve davranış** bir araya getirir, bunların **bir bütün** olarak ele alınmasını sağlar
- Nesneye yönelik yaklaşımın **temel yapıtaşı** veya **atomik birimi** sınıflardır

Nesneye Yönelik Yaklaşım ve Sınıf Kavramı

- Bir sınıf tanımlandıktan sonra bu sınıftan **istendiği kadar nesne yaratılabilir** ve problemin çözümünde kullanılabilir
- Bu nedenle sınıflara nesne oluşturmak için kullanılan bir **şablon** denebilir

Nesneye Yönelik Yaklaşım ve Sınıf Kavramı

- Ancak yaratılan bu nesneler birbirlerinden **izole ve habersiz olmazlar**
- Nesneler birbirlerini **bilirler**, aralarında **haberleşirler**
- Aslında yazılım sistemi birbirleri arasında haberleşen bir **nesne ağı**ndan oluşmaktadır



UML ve Sınıf Gösterimi



UML, yazılım sistemlerinin analiz, tasarım ve geliştirme süreçlerini modellemek amacı ile ortaya çıkmış bir notasyondur. Nesne yönelimli yazılım sistemlerinde sınıfları, sınıfların özelliklerini, yapılarını, birbirleri ile olan ilişkilerini, nesnelerin evrelerini, nesneler arasındaki mesajlaşmaları ifade etmek için kullanılır.

Java'da Class (Sınıf) Örneği

```
package com.javaegitimleri;
```

Compiler primitif değişkenlere default değer atar, custom tipteki değişkenleri NULL'a init eder

```
public class Araba {  
    private int hiz;
```

instance
attribute/field/property

```
    public Integer hizlan() {  
        return ++hiz;  
    }
```

instance
metot/fonksiyon

```
    public Integer yavasla(Integer azalis) {  
        return hiz-=azalis;  
    }
```

```
}
```

İletişim



www.harezmi.com.tr

www.java-egitimleri.com



info@harezmi.com.tr

info@java-egitimleri.com



[@HarezmiBilisim](https://twitter.com/HarezmiBilisim)

[@JavaEgitimleri](https://twitter.com/JavaEgitimleri)