

1906002132015

Programlama Dilleri Temelleri

BAİBÜ Bilgisayar Müh.

Ders 7

Dr. Öğr. Üyesi İsmail Hakkı Parlak

Kaynaklar: Watt, David A., Programming Language Design Concepts, Wiley

Bazı şekiller kaynak kitaptan kopyalanmıştır.

OOP'nin 4 Temel Kavramı

- Soyutlama (Abstraction): Detayları kullanıcıdan gizleme. Kullanıcı bir yöntemin ne yaptığıyla ilgilenir; geliştirici nasıl yapılacağı ile.
- Miras (Inheritance): Bir sınıftan o sınıfın özelliklerini miras alan ve yeni özellikler ekleyebilen başka sınıfların türetilmesi.
- Çok biçimlilik (Polymorphism): Alt sınıflar, türetildikleri üst sınıfın kullanıldığı her yerde kullanılabilir (Alt sınıf *IS-A* üst sınıf) ve üst sınıftan aldıkları yöntemleri değiştirebilir / geçersiz kılabilir (overriding).
- Kapsülleme (Encapsulation).

Kapsülleme (Encapsulation)

- Verileri ve veriler üzerindeki eylemleri bir araya getirmek (veya kapsüllemek). Başka bir deyişle, bir sınıf (class) tanımlama yeteneğidir.
- Bir nesnenin iç yapısını veya durumunu (internal state) dış dünyadan gizlemek (information hiding). Sınıfın özel (private) verilerine dış dünyadan doğrudan erişim olmaması. Sınıfın özel verilerinin veya iç durumunun gerektiğinde dış dünyaya yöntemler (methods) aracılığıyla açılması.

OOP

- Nesne yönelimli programlamanın temel kavramları şunlardır:
 - nesneler (objects)
 - sınıflar ve alt sınıflar
 - miras (inheritance)
 - polimorfizm
- Nesnelerin bir veya daha fazla değişken bileşeni vardır ve nesneler bu bileşenler üzerinde çalışan yöntemlerle donatılmıştır.
- Nesnenin değişken bileşenleri tipik olarak özeldir (private) ve bu nedenle yalnızca nesnenin yöntemleriyle (methods) erişilebilir.

OOP

- Nesneler bize gerçek dünyadaki varlıkları modellemek için doğal bir yol sunar.
- Nesneler ayrıca dosyalar, veritabanları, web sayfaları ve kullanıcı arabirimi bileşenleri gibi siber dünya varlıklarını modellemek için de bize doğal bir yol sunar.
- Nesnelerin sınıflandırılması, nesne yönelimli dillerin temel bir özelliğidir. Bir sınıf, benzer değişken bileşenlere ve yöntemlere sahip bir nesne ailesidir. Bir alt sınıf, bir sınıfı ek bileşenler ve/veya ek (veya geçersiz kılan / overriding) yöntemlerle genişletir (extend).
- Her alt sınıfın kendi alt sınıfları olabilir, bu nedenle bir sınıf hiyerarşisi oluşturabiliriz.

OOP

- Kalıtım (inheritance) nesne yönelimli dillerin diğer bir karakteristiğidir. Alt sınıf, bu yöntemlerden herhangi birini açıkça geçersiz kılmadığı sürece, bir alt sınıf, üst sınıfının tüm yöntemlerini miras alır (paylaşır).
- Polimorfizm, bir alt sınıfın nesnesinin üst sınıfının bir nesnesi gibi ele alınmasını sağlayan anahtar bir kavramdır. Bu, koleksiyonun bazı ortak ata sınıfı terimleriyle tanımlanması koşuluyla, farklı sınıflardan nesnelerin heterojen bir koleksiyonunu oluşturmamıza izin verir.

OOP

```
import java.util.ArrayList;

class Sekil {
    public String toString() {
        return "Sekil";
    }
}

class Ucgen extends Sekil {
    public String toString() {
        return super.toString() + " Ucgen";
    }
}

class Kare extends Sekil {
    public String toString() {
        return super.toString() + " Kare";
    }
}
```

```
class Main {
    public static void main() {
        ArrayList<Sekil> sekiller =
            new ArrayList<Sekil>();

        sekiller.add(new Ucgen());
        sekiller.add(new Kare());

        for (Sekil s : sekiller) {
            System.out.println(
                s.toString()
            );
        }
    }
}
```

12.2 Kullanımbilim (Pragmatics)

- Nesne yönelimli bir programın program birimleri sınıflardır. Sınıflar, bağımlılık (dependency: bir sınıfın işlemlerinin başka bir sınıfın işlemlerini çağırması), genişletme (extension: miras) veya kapsama (containment: bir sınıfın objeleri içinde diğer sınıfın objelerinin bulunması) ile birbiriyle ilişkili olabilir.
- Değişken bileşenler (properties) herkese açıksa (public), bunlara diğer sınıflar tarafından doğrudan erişilebilir ve bu da sıkı bağlantıya (tight coupling) yol açar. Değişken bileşenleri özel ise (private), sınıf soyut bir türe benzer (abstract type) ve gevşek bağlantı (loose coupling) sağlar: sınıfın temsilindeki bir değişikliğin diğer sınıflar üzerinde çok az etkisi olur veya hiç etkisi olmaz.