

Inheritance (Kalıtım)

- İkinci tür yeniden kullanım sınıf yapısının kalıtım yoluyla yeniden kullanılmasıdır. Kalıtım nesneye dayalı programlama dillerinin anahtar elemanlarından biridir.
- Eğer farklı sınıflar benzer durumlara sahipse ve benzer servisler sağlıyorsa kalıtım bu benzer kısımları genelleştirilmiş sınıf bildiriminden tekrar kullanmamıza olanak verir. Bu, soyut temel sınıf yaratılarak ve onun durum ve davranışlarını daha çok özelleştirilmiş sınıflarda tekrar kullanarak başarılır. Eğer bir sınıf temel bir sınıftan türetilmişse, bütün üye değişkenler ve üye yöntemler türetilmiş sınıf tarafından miras alınır (veya kullanılır hale getirilir).
- Temel sınıf **superclass** (süper sınıf) olarak ve türetilmiş sınıf **subclass** (alt sınıf) olarak adlandırılır. Kalıtımla gerçekte **superclass** ile uyumlu olan yeni bir sınıf yaratılır. Yani **subclass** **superclass** türünde bir sınıfmış gibi kullanılabilir. Bu **subclass** örneklerinin **superclass**'a dönüşümünü mümkün kılar. Bu durum polimorfizm olarak tanımlanır ki bu da nesneye dayalı programlama dillerinin bir diğer anahtar elemanıdır.

Inheritance (Kalıtım)

```
class Sekil{
    int pozisyonX;
    int pozisyonY;
    void move(int newX, int newY){
        pozisyonX = newX;
        pozisyonY = newY;
    }
}

class Daire extends Sekil{
    int radius;
    void scale(int scaleFactor){
        radius *= scaleFactor;
    }
}

class Dikdortgen extends Sekil{
    int width;
    int height;
    void scale(int scaleFactor){
        width *= scaleFactor;
        height *= scaleFactor;
    }
}

public class Ana {
    public static void main(String args[]){
        Daire c = new Daire();
        c.pozisyonX = 10;
        c.pozisyonY = 20;
        c.radius = 3;
        c.move(11,11);
        c.scale(5);
    }
}
```

- Türetme işleminde C# : yerine extends kelimesi kullanılır.
- Scale fonksiyonda override kullanılmaz. Direkt yazılır.
- C#'tan farklı olarak Virtual metot virtual kelimesine gerek yoktur.