



TECHPROED

PROFESSIONAL TECHNOLOGY EDUCATION

WELCOME TO TECHPROED JAVA TUTORIAL

Testi baslatmak icin asagidaki adimlari takip ediniz

Go to www.socrative.com

Click on *Login*

Click on *Student Login*

Room Name: *ALPTEKIN3523*

Kayıtta kullandığınız ismi tam olarak yazınız

Time: 11 Minutes

Abstract Class

abstract class oluşturmak için, “**access modifier**” ile “**class**” keyword arasında “**abstract**” keyword kullanılır.

```
public abstract class Animal { }
```

Note: Abstract class'lardan **object oluşturulamaz**.

```
public abstract class Animal { } ==> Animal animal = new Animal(); X
```

Abstract Method

Note: Body'si olmayan method'lara **abstract method** denir.

Concrete Method

```
public int sumOfTwo(int n1, int n2) { return n1 + n2; }
```

Abstract Method

```
public abstract int sumOfTwo(int n1, int n2);
```

X

Abstract Method'lara Body Yazılmaz

```
public abstract int sumOfTwo(int n1, int n2) { } // Compile Time Error
```

X

Concrete Method'lar Body'siz Yazılmaz

```
public int sumOfTwo(int n1, int n2); // Compile Time Error
```

Note: Bir **abstract method** her zaman **abstract class** icinde oluřturulur

```
public class Animal {  
    public abstract void sound();  
}
```



```
public abstract class Animal {  
    public abstract void sound();  
}
```



Note: Bir **abstract class** hem **abstract method** hem de **concrete method** icerebilir.

```
public abstract class Animal {  
    public abstract void sound();  
}
```



```
public abstract class Animal {  
    public void sound(){ }  
}
```



```
public abstract class Animal {  
    public abstract void eat();  
    public void sound(){ }  
}
```



Note: Bir concrete class **abstract class**'a extends yaparsa, abstract class içindeki tüm abstract method'ları override etmelidir.

```
public abstract class Animal {  
    public abstract void sound();  
}
```

Concrete child classlar abstract
method'ları Override etmelidir.

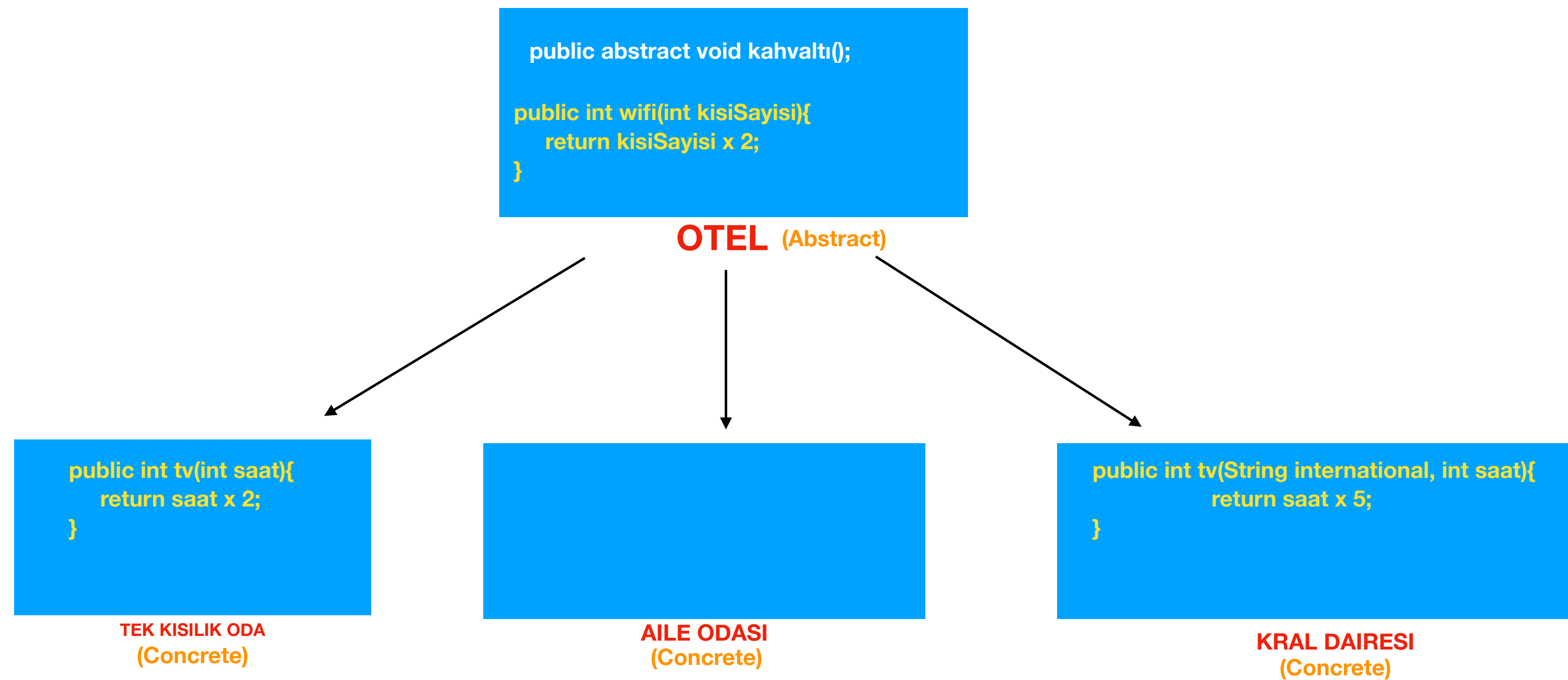
```
public class Dog extends Animal {  
    public void sound(){  
        System.out.println("Barking");  
    }  
}
```

```
public abstract class Animal {  
    public abstract void sound();  
}
```

Abstract child classlar abstract
method'ları Override etmek zorunda değildir

```
public abstract class Dog extends Animal {  
}
```

Abstract Proje



Compile Time Error var mi?

```
public abstract class Animal {  
    public abstract String getName();  
}
```

```
public abstract class BigCat extends Animal {  
    public abstract void roar();  
}
```

```
public class Lion extends BigCat {  
    public String getName() {  
        return "Lion";  
    }  
    public void roar() {  
        System.out.println("The Lion lets out a loud ROAR!");  
    }  
}
```



Not: Herhangi bir parent tarafından override edilen abstract method alttaki concrete child'lar tarafından override edilmese de olur

Compile Time Error var mı?

```
public abstract class Animal {  
    public abstract String getName();  
}
```

```
public abstract class BigCat extends Animal {  
    public String getName() {  
  
        return "BigCat";  
    }  
    public abstract void roar();  
}
```

```
public class Lion extends BigCat {  
    public void roar() {  
        System.out.println("The Lion lets out a loud ROAR!");  
    }  
}
```



Why do we need “abstraction” ?

Bir “Animal” ne tur ses cikarir?

Bir “Animal” ses cikarir ama biz bu ses icin bir tanımlama yapamayız.
Yani; ses var ama tanımlama yapamıyoruz, method var ama body yazamıyoruz.

Her child’in belli fonksiyonlara sahip olmasını istediğimizde abstraction kullanırız?


Mesela belli bir hayvandan bahsederken onun cikardigi sesi tanımlamak zorunda olduğumuz bir application olsun. Bu durumda class’a abstract ses() method’unu koyar ve her child class’da o hayvanin sesini method body’si kullanarak tanımlarız.

Asagidaki ornekte goruldugu gibi Dog, Cat, ve Bird class’ları icin bir ses tanımladık ama Animal class’ındaki ses methodu icin bir tanımlama yapamadık, cunku Animal’ların cikardigi belli bir ses yok.

```
public abstract class Animal {  
    public abstract void ses();  
}  
  
public class Dog extends Animal {  
    public void ses(){  
        System.out.println("Dogs Bark");  
    }  
}  
  
public class Cat extends Animal {  
    public void ses(){  
        System.out.println("Cats Meow");  
    }  
}  
  
public class Bird extends Animal {  
    public void ses(){  
        System.out.println("Birds Sing");  
    }  
}
```

Bir abstract class “final” olabilir mi?

```
public final abstract class Animal {  
    public abstract void ses();  
}
```



```
public class Dog extends Animal{  
    public void ses(){  
        System.out.println("Dogs Bark");  
    }  
}
```

Bir **abstract class** baska class'lar ona extends etsin ve child class olustursun diye oluřturulur. Halbuki final class'lara extends edilemez, bu yuzden Java abstract bir class'in final olmasına muisaade etmez ve Compile Time Error verir.

Bir abstract method “final” olabilir mi?

```
public abstract class Animal {  
    public final abstract void sound();  
}
```



```
public class Dog extends Animal{  
    public void sound(){  
        System.out.println("Dogs Bark");  
    }  
}
```

Bir **abstract method** override edilme zorundadır. Halbuki, final method'lar override edilmezler. Bu yüzden Java abstract method'ların final olmasına müsaade etmez ve Compile Time Error verir.

Bir abstract method “private” olabilir mi?

```
public abstract class Animal {  
    private abstract void sound();  
}
```



```
public class Dog extends Animal{  
    sound() method cannot be accessed form  
    child class because it is private  
}
```

Bir **abstract method** override edilme zorundadır. Halbuki, final method’lar override edilmezler.
Bu yüzden Java abstract method’ların final olmasına müsaade etmez ve Compile Time Error verir.

Abstract Class Kurallari:

- 1. Bir Abstract Class'dan object oluşturulamaz.**
- 2. Abstract class'lar abstract ve concrete method'lar icerebilir.**
- 3. Abstract class private ve final olamazlar.**
- 4. Bir abstract class başka bir another abstract class'a extends ettiginde abstract method'lari override etmek zorunda degildir.**
- 5. Abstract class'a extends eden ilk concrete class abstract class'daki tum abstract method'lari override etmek zorundadir.**

Abstract Method Kurallari:

- 1. Abstract method'lar sadece abstract classlar icinde olusturulabilir.**
- 2. Abstract methodlar private ve final olamazlar.**
- 3. Abstract method'larin method body'si olamaz.**