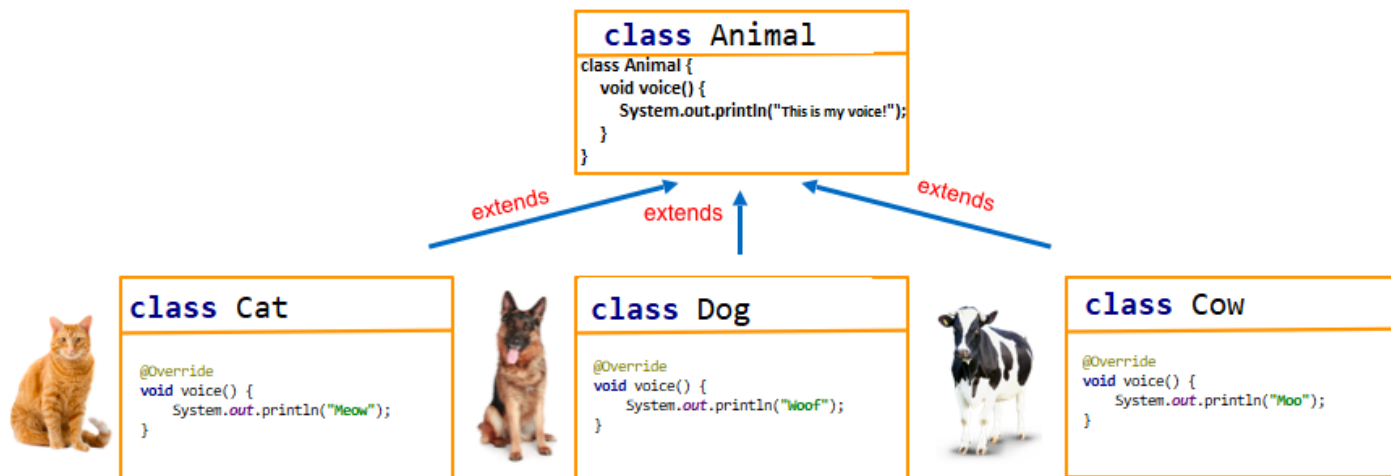


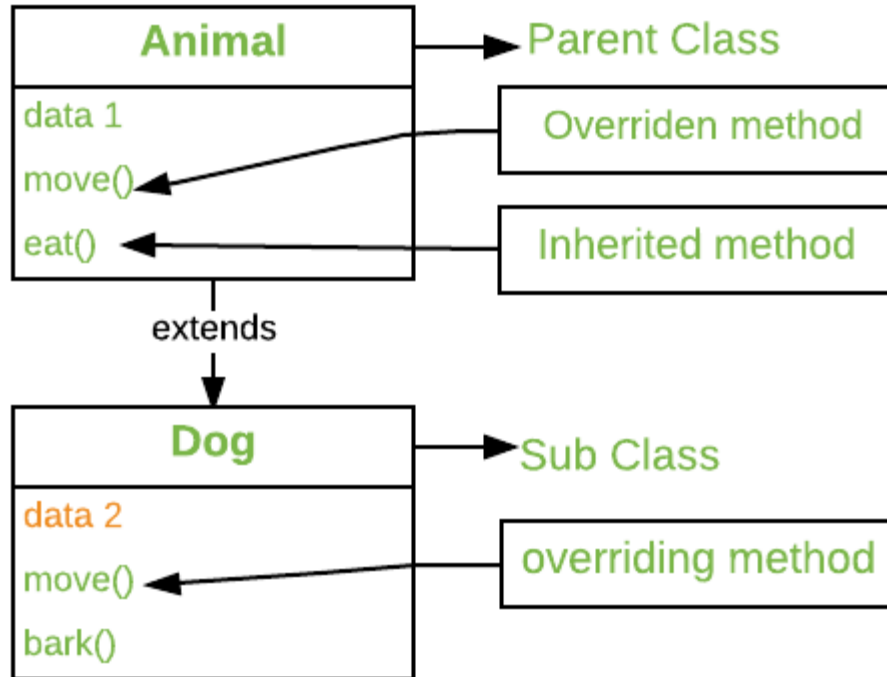
Polymorphism. Overriding. final modifikatori

Overriding – bu polymorphismni amalga oshirish yo'llaridan biri bo'lib , bunda super class methodini voris olingan class ichida header qismini o'zgartirmasdan body qismini qayta yozishga aytiladi.



POLYMORPHISM





Overriding qoidalari

1. **Constructorni override qilib bo'lmaydi**
2. **Override qilingan methodni return type va parametrlari bir xil bo'lish kerak.**
3. **Override qilingan methodni child class methodi ichida murojat qilish**
4. **Overriding va Access-Modifikatorlari**
5. **Static methodni override qilib bo'lmaydi**
6. **Final methodni override qilib bo'lmaydi**



Override qilingan methodni return type va parametrlari bir xil bo'lish kerak.

```
public static void staticMethod(){  
    System.out.println("This is static method");  
}
```

```
@Override  
public void publicMethod() {//private, package yoki protected mumkin emas  
    System.out.println("Overriding public method");  
}
```



Override qilingan methodga child class methodi ichida murojat qilish

Parent class methodini **super** kalit so'zi o'rqali murojat qilish mumkin

```
@Override
protected /*yoki public*/ void packageMethod() { //private mumkin emas
    super.packageMethod();
    System.out.println("Overriding package method");
}
```

Overriding and Access-Modifiers

- voris olingan(child) class ichida override qilingan method access modifikatorini o'zidan yuqori darajali access modifikator bilan almashtirish mumkin, quyi darajalisi bilan mumkin emas.
- private methodni (voris olib bo'lmasligi sababli) override qilib bo'lmaydi.

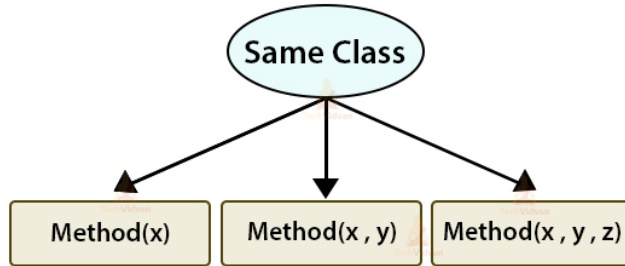



```
public class Parent {  
    private void privateMethod() {  
        System.out.println("This is private method");  
    }  
    void packageMethod() {  
        System.out.println("This is package method");  
    }  
    protected void protectedMethod() {  
        System.out.println("This is protected method");  
    }  
    public void publicMethod() {  
        System.out.println("This is public method");  
    }  
    public static void staticMethod(){  
        System.out.println("This is static method");  
    }  
}
```

```
public class Child extends Parent {
    @Override
    protected /*yoki public*/ void packageMethod() { //private mungkin emas
        super.packageMethod();
        System.out.println("Overriding package method");
    }
    @Override
    public void protectedMethod() { //private yoki package mungkin emas
        System.out.println("Overriding protected method");
    }
    @Override
    public void publicMethod() { //private, package yoki protected mungkin emas
        System.out.println("Overriding public method");
    }
}
```

Overloading and Overriding

Method Overloading in Java



Method Overriding in Java





Java Final keyword



Final Variable



Final Methods



Final Classes

final modifikatori

final modifikatorini **variable** (o'zgaruvchi), **method** va **class** ga nisbatan ishlatish mumkin.

Final Variable → To create constant variables

Final Methods → Prevent Method Overriding

Final Classes → Prevent Inheritance

final/variable

final modifikatori o'zgaruvchi qiymatini o'zgartirishni taqiqlash uchun ishlatiladi

```
static final double PI = 3.141592653589793;  
final int CAPACITY_OF_BOX = 25;
```



final method

final modificatory methodni override qilishni taqiqlash uchun ishlatiladi

```
public final void finalMethod(){  
    System.out.println("This is private method");  
}
```



final class

final modifikatori classdan voris olishni taqiqlash uchun ishlatiladi

```
public final class FinalClass {  
    -----  
}
```

