## 重要

**除了构造方法，其余的子类都继承，包括实例方法和get set方法都可以继承** 在进行重新构造构造器的时候，可以通过super关键字快速完成，this.xx=xxx;的语句，比如

class A extends B{  
public A(String name,int age){  
 super(name,age);  
}  
}

需要注意的是，**super语句只能放在构造器内部的第一行**

[!举例] 爹很有钱，儿子以后也会有钱，爹的钱没用完

## 作用

1. 子类继承父类，**代码复用**
2. 主要作用：有了继承关系，才有后续的方法覆盖和多态

## 语法格式

//A为父类，B为子类，B类继承A类  
//除了构造方法之外，无论是实例方法（get，set）还是变量都继承  
class A{  
}  
class B extends A{  
}

## 相关特性

1. B类继承A类，则A类为超类，父类，B为子类，派生类
2. **java中只支持单继承**，不支持多继承，C++支持多继承
3. **缺点是耦合度高**，父类修改，那么子类也会受到牵连
4. java中**除了构造方法之外都能继承，私有的属性（private）无法在子类中直接访问**（父类private修饰的不能在子类中直接访问）
5. 没有指定父类的class，都默认继承的Object,Object是所有类的根类。

## eg

package a006继承;  
  
public class Jicheng {  
 public static void main(String[] args) {  
 Account a1 = new Account("zhangsan123",100);//创建普通账户  
 System.out.println(a1.getAct()+" 的余额是 "+a1.getBalance());  
 CreditAccount c1 = new CreditAccount("lisi321",120,100);  
 System.out.println(c1.getAct()+" 的余额是 "+c1.getBalance()+"信誉度： "+c1.getCredit());  
 }  
}  
  
// 银行卡  
class Account{  
 private String act;  
 private double balance;  
 // get and set   
 public String getAct() {  
 return act;  
 }  
 public void setAct(String act) {  
 this.act = act;  
 }  
 public double getBalance() {  
 return balance;  
 }  
 public void setBalance(double balance) {  
 this.balance = balance;  
 }  
 // 构造方法  
 public Account(){  
 this("",0);  
 }  
 public Account(String act, double balance) {  
 this.act = act;  
 this.balance = balance;  
 }  
   
  
}  
  
class CreditAccount extends Account{  
 private double credit;  
 //get and set  
 public double getCredit() {  
 return credit;  
 }  
  
 public void setCredit(double credit) {  
 this.credit = credit;  
 }  
 //构造方法  
 public CreditAccount(double credit) {  
 this.credit = credit;  
 }  
 //构造方法  
 public CreditAccount(String act, double balance, double credit) {  
 super(act, balance);  
 this.credit = credit;  
 }  
}

super是用来做引用父类的构造方法的。