**Java继承**

**继承的概念**

继承是java面向对象编程技术的一块基石，因为它允许创建分等级层次的类。

继承就是子类继承父类的特征和行为，使得子类对象（实例）具有父类的实例域和方法，或子类从父类继承方法，使得子类具有父类相同的行为。

生活中的继承：

兔子和羊属于食草动物类，狮子和豹属于食肉动物类。

食草动物和食肉动物又是属于动物类。

所以继承需要符合的关系是：is-a，父类更通用，子类更具体。

虽然食草动物和食肉动物都是属于动物，但是两者的属性和行为上有差别，所以子类会具有父类的一般特性也会具有自身的特性。

类的继承格式

在 Java 中通过 extends 关键字可以申明一个类是从另外一个类继承而来的，一般形式如下：

类的继承格式

class 父类 {

}

class 子类 extends 父类 {

}

为什么需要继承

接下来我们通过实例来说明这个需求。

开发动物类，其中动物分别为企鹅以及老鼠，要求如下：

企鹅：属性（姓名，id），方法（吃，睡，自我介绍）

老鼠：属性（姓名，id），方法（吃，睡，自我介绍）

企鹅类：

public class Penguin {

private String name;

private int id;

public Penguin(String myName, int myid) {

name = myName;

id = myid;

}

public void eat(){

System.out.println(name+"正在吃");

}

public void sleep(){

System.out.println(name+"正在睡");

}

public void introduction() {

System.out.println("大家好！我是" + id + "号" + name + ".");

}

}

老鼠类：

public class Mouse {

private String name;

private int id;

public Mouse(String myName, int myid) {

name = myName;

id = myid;

}

public void eat(){

System.out.println(name+"正在吃");

}

public void sleep(){

System.out.println(name+"正在睡");

}

public void introduction() {

System.out.println("大家好！我是" + id + "号" + name + ".");

}

}

从这两段代码可以看出来，代码存在重复了，导致后果就是代码量大且臃肿，而且维护性不高(维护性主要是后期需要修改的时候，就需要修改很多的代码，容易出错)，所以要从根本上解决这两段代码的问题，就需要继承，将两段代码中相同的部分提取出来组成 一个父类：

公共父类：

public class Animal {

private String name;

private int id;

public Animal(String myName, int myid) {

name = myName;

id = myid;

}

public void eat(){

System.out.println(name+"正在吃");

}

public void sleep(){

System.out.println(name+"正在睡");

}

public void introduction() {

System.out.println("大家好！我是" + id + "号" + name + ".");

}

}

这个Animal类就可以作为一个父类，然后企鹅类和老鼠类继承这个类之后，就具有父类当中的属性和方法，子类就不会存在重复的代码，维护性也提高，代码也更加简洁，提高代码的复用性（复用性主要是可以多次使用，不用再多次写同样的代码） 继承之后的代码：

企鹅类：

public class Penguin extends Animal {

public Penguin(String myName, int myid) {

super(myName, myid);

}

}

老鼠类：

public class Mouse extends Animal {

public Mouse(String myName, int myid) {

super(myName, myid);

}

}

继承类型

需要注意的是 Java 不支持多继承，但支持多重继承。