

■ Object类被子类经常重写的方法

方法	说明
toString()	返回当前对象本身的有关信息，按字符串对象返回
equals()	比较两个对象是否是同一个对象，是则返回true
hashCode()	返回该对象的哈希代码值
getClass()	获取当前对象所属的类信息，返回Class对象

■ Object类的equals()方法

◆ 比较两个对象是否是**同一个**对象，是则返回true

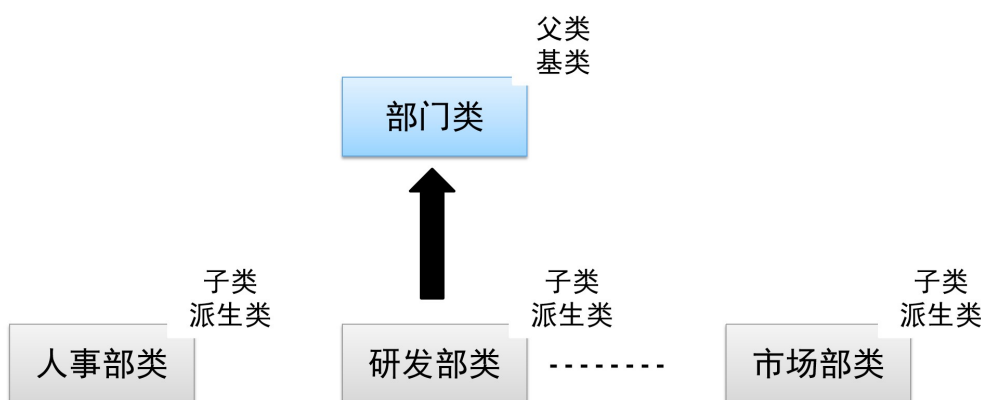
◆ 操作符==

- 简单数据类型，直接比较值。如1==2
- 引用类型，比较两者是否为同一对象

- (1) Object类的equals()方法与==没区别
- (2) 当有特殊需求，如认为属性相同即为同一对象时，需要重写equals()
- (3) Java.lang.String重写了equals()方法，把equals()方法的判断变为了判断其值

继 承

为什么要使用继承



继承是面向对象的三大特征之一，是Java中实现代码重用的重要手段之一。

类的继承

- ✓ 在Java中定义一个类时，让该类通过关键字**extends**继承一个已有的类，这就是类的继承
- ✓ 被继承的类称为**父类**（**超类，基类**），新的类称为**子类**（**派生类**）。
- ✓ 子类继承父类的所有**属性**和**方法**，同时也可以增加自己的属性和方法。

• 继承的好处

- 实现类的重用
- 易维护

```
class Employee {  
    String name;  
    String birth;  
    double salary;  
  
    void getDetail() {  
        System.out.println("name=" + name + " age=" + birth);  
    }  
  
    double getSalary() {  
        System.out.println(salary);  
        return salary;  
    }  
}
```

```
class Worker extends Employee{  
    double dressAllowance(){  
        double allowance = 200  
        return allowance;  
    }  
}
```

• 继承的规则

- ✓ Java中只支持**单继承**，也就是说每个类只能有一个父类，不允许有多重继承
- ✓ 一个父类可以有多个子类
- ✓ 子类**继承**父类**所有的属性**和**方法**，但不继承父类的**构造方法**

```
class Dog extends Pet {  
    //子类特有的属性和方法  
}
```

■ 子类的构造方法 — 子类不能继承父类的构造方法

- ✓ 使用默认的构造方法
- ✓ 在子类中的创建构造方法
- ✓ 在子类中创建构造方法时，必须调用**父类的构造方法**
 - 子类可以在自己的构造方法中使用**super**关键字来调用父类的构造方法

```
super (参数1, 参数2, ...);
```
 - 如果使用super关键字调用父类构造方法，必须写在该子类构造方法的**第一行**
 - 如调用的是父类中无参的构造方法，则可以不写super()
 - 如果子类中调用了父类无参的构造方法，而父类中没有无参构造方法则系统编译出错

类的继承

■ 子类访问父类成员

◆ 访问父类构造方法

```
super();  
super(name);
```

使用**super**关键字，**super**代表父类对象

在子类构造方法中调用且必须是第一句

◆ 访问父类属性

```
super.name;
```

◆ 访问父类方法

```
super.print();
```

■ 方法的覆盖 (override)

- ✓ 重写 (rewrite)
- ✓ 对从父类中**继承来的方法**进行改造
- ✓ 在子类继承父类时发生

• 方法覆盖的规则

- ✓ 在子类中的覆盖方法与父类中被覆盖的方法应具有
 - 相同的方法名
 - 相同的参数列表（参数数量、参数类型、参数顺序都要相同）
 - 相同的返回值类型
 - 子类覆盖方法的访问权限要不小于父类中被覆盖方法的访问权限

小结



提问

■ 构造方法也会被重写吗？

不能被继承，
因此不能重写

■ 方法重写的规则

- ◆ 方法名相同
- ◆ 参数列表相同
- ◆ 返回值类型相同或者是其子类
- ◆ 访问权限不能严于父类

■ 方法重载与方法重写

	位置	方法名	参数表	返回值	访问修饰符
方法重写	子类	相同	相同	相同或是其子类	不能比父类更严格
方法重载	同类	相同	不相同	无关	无关

■ Object类是所有类的父类

```
public class Pet extends Object {  
    .....  
}
```

```
java.lang  
Class Object  
java.lang.Object  
-----  
public class Object  
Class Object is the root of the class hierarchy.  
Since:  
    JDK1.0  
See Also:  
    Class
```