

# Object 类

## 目 录

- 1. Object类的简介.....2
- 2. Object的equals()方法.....2



## 1. Object 类的简介

在 Java 中，所有的 Java 类都直接或间接地继承了 java.lang.Object 类。Object 类是所有 Java 类的基类。在定义一个类时，没有使用 extends 关键字，那么这个类直接继承 Object 类。所有对象，包括数组，都继承这个类的方法。

Object 类定义了大量的可被其他类继承的方法，表-1 中是 Object 类比较常用的，也是被它的子类经常重写的方法。

表-1 Object 类的常用方法

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
clone()	生成当前对象的一个拷贝，并返回
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

## 2. Object 的 equals()方法

equals() 用来比较两个对象是否是同一个引用对象，例如：

```
ResearchDept rd=new ResearchDept();  
  
ResearchDept rd2=new ResearchDept();  
  
boolean result= rd.equals(rd2);
```

此时 result 的结果为 false。如果单纯的使用 Object 类的 equals() 方法来比较对象是否是同一个引用对象，则和操作符 “==” 没有区别。

在实际的开发中，往往并不是直接使用这些方法，而是要重写后再使用。下面使用研发部门类为例进行说明，代码如示例 1 所示。

示例 1:

```
/**  
 * 研发部门类  
 * @author 北大青鸟  
 */  
  
public class ResearchDept {  
    private String name; //部门名称  
    private String responsibility; //部门职责
```

```

private String manager; //部门经理

public ResearchDept(String name,String manager,String responsibility){
    this.name=name;
    this.manager=manager;
    this.responsibility=responsibility;
}
//省略 getter/setter 方法
//重写 equals()方法, 如果部门名称、经理、职责都相同, 证明是同一个部门
public boolean equals(Object o){
    if(this==o){ //判断是否是同一个对象
        return true;
    }
    if(this.getClass()!=o.getClass()){ //判断是否是同一种类型
        return false;
    }
    ResearchDept obj=(ResearchDept)o; //将对象 o 转换为 ResearchDept 类型
    if(this.name.equals(obj.name)&&this.manager.equals(obj.manager)&&
        this.responsibility.equals(obj.responsibility)){//判断各字段是否相等
        return true;
    }else{
        return false;
    }
}
}

```

编写测试类进行测试, 代码如下示例 2 所示。

## 示例 2:

```

/**
 * 测试类
 * @author 北大青鸟
 */

public class DeptTest {
    public static void main(String[] args) {
        ResearchDept rd=new ResearchDept("研发部","李经理",
            "负责公司新产品的开发");
        ResearchDept rd2=new ResearchDept("研发部","李经理",
            "负责公司新产品的开发");

        /*验证重写 equals()方法*/
        if(rd==rd2){
            System.out.println("二者为同一个对象");
        }else{
            System.out.println("二者不为同一个对象");
        }
        if(rd.equals(rd2)){
            System.out.println("二者为同一个部门");
        }else{

```

```
        System.out.println("二者不是同一个部门");  
    }  
}  
}
```

输出效果如图-1 所示。

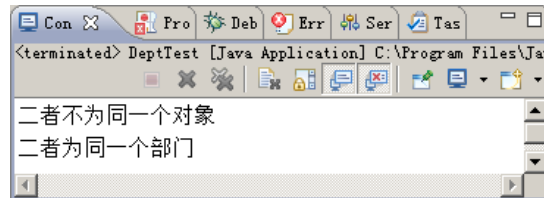


图-1 示例 2 运行效果

从效果图中看到，我们已经成功重写了 `ResearchDept` 类的 `equals()` 方法，只要部门的各个属性值相同，则判断为同一个部门。而我们所熟悉的 `String` 类，也重写了 `equals()` 方法，将原来判断是否是同一个引用对象重写为判断其值是否相等。`Object` 类的其他常用方法的使用可以查阅 API。

