



链滴

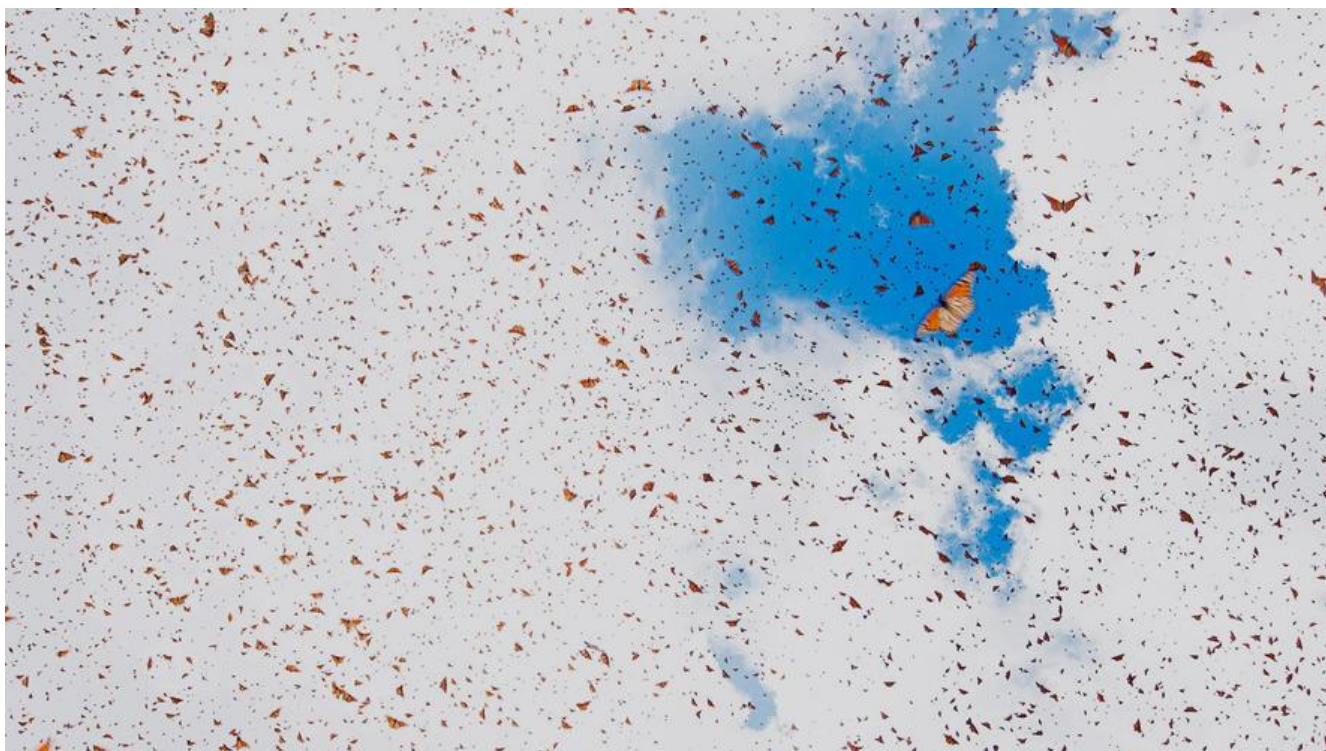
java 基础 | 对象之间的比较及 Comparator 和 Comparable 的区别

作者: [xiaodaojava](#)

原文链接: <https://ld246.com/article/1573605682406>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



javaDEMO

本网站记录了最全的各种JavaDEMO,保证下载,复制就是可用的,包括基础的,集合的, spring的, Mybatis的等等各种,助力你从菜鸟到大牛,记得收藏哦~~

<https://www.javastudy.cloud>

对象之间的比较方式

比较方式只要有以下四种

1. 用 == 来比较
2. 用equals来比较
3. 用Comparator外部比较器来比较
4. 实现Comparable接口来比较

用 == 比较

常用于比较基本类型的值是否相等,包装类型的地址是否相等

使用如下:

```
int a=1 ,b = 1;  
// 输出true  
System.out.println(a==b);
```

用equals比较

equals其实就是java里面的一个普通的方法,当前类有重写的话,就调用重写之后的,如果没有重写的话,

默认调用Object里面的equals方法.

如下面的TestModel,重写了equals方法,只有名字和年龄都想等时才是想等的.

```
public class TestModel{

    private String name;

    private Integer age;

    public String getName() {
        return name;
    }

    public TestModel setName(String name) {
        this.name = name;
        return this;
    }

    public Integer getAge() {
        return age;
    }

    public TestModel setAge(Integer age) {
        this.age = age;
        return this;
    }

    @Override
    public boolean equals(Object obj) {
        TestModel target = (TestModel) obj;
        return this.name.equals(target.name) && age.equals(target.age);
    }
}
```

使用Comparable

用法可参考jdk中Integer类, Integer也是实现了Comparable接口,需要重写compareTo方法,此方法质上也是一个java方法,按需调用即可,如把我们已有的a和传入的目标b比较

a.compareTo(b);

```
void contextLoads() {
    TestModel t1 = new TestModel().setName("java").setAge(25);
    TestModel t2 = new TestModel().setName("study").setAge(12);
    System.out.println(t1.compareTo(t2));
}
```

```
static class TestModel implements Comparable<TestModel>{

    private String name;

    private Integer age;

    public String getName() {
```

```

        return name;
    }

    public TestModel setName(String name) {
        this.name = name;
        return this;
    }

    public Integer getAge() {
        return age;
    }

    public TestModel setAge(Integer age) {
        this.age = age;
        return this;
    }

    @Override
    public int compareTo(TestModel o) {
        if(this.age.equals(o.getAge())){
            return 0;
        }
        if(this.age > o.getAge()){
            return 1;
        }
        return -1;
    }
}

```

使用Comparator进行比较

和Comparable需要当前实体类实现不同, Comparator 需要在外部定义source和target的比较逻辑.

```

/**
 * @Author https://www.javastudy.cloud
 * @CreateTime 2019/11/13
 **/
public class TestMain {

    public static void main(String[] args) {
        Comparator<TestModel> comparator = (source,target)->{
            if(source.getAge().equals(target.getAge())){
                return 0;
            }
            if(source.getAge() > target.getAge()){
                return 1;
            }
            return -1;
        };

        TestModel t1 = new TestModel().setName("java").setAge(25);
        TestModel t2 = new TestModel().setName("study").setAge(12);
        System.out.println(comparator.compare(t1,t2));
    }
}

```

```
}  
  
static class TestModel{  
  
    private String name;  
  
    private Integer age;  
  
    public String getName() {  
        return name;  
    }  
  
    public TestModel setName(String name) {  
        this.name = name;  
        return this;  
    }  
  
    public Integer getAge() {  
        return age;  
    }  
  
    public TestModel setAge(Integer age) {  
        this.age = age;  
        return this;  
    }  
  
}  
}
```

生活中的例子看Comparator和Comparable的区别

如我们一直所诟病的应试教育,教育体系内只按成绩给学生排名,即每个学生都实现了Comparable接口,compareTo方法就是根据学习成绩来的.到了社会之后,排名就多元化了,有根据挣的钱的,有根据社会影响力的,有根据道德思想的等等不同的Comparator来排序.

我们可以看到Comparator是在java.util包下,Comparable是放在java.lang包下,可从命名上做区分,在前学英语时,able结尾的代表有什么什么能力,类似于be able to.这应该是一个实对/对象本身的能力,以Comparable放在了lang包下,是语言层面的.以er/or结尾的代表人/实体,如creator,programmer等,如此看这个比较器不是实体本身的东西,而是和实体并列的,可以为实体提供一定的服务,所以Comparator放在了util下面,加油吧,少年!