# Comparable接口

Java.lang

定义:

Public interface Comparable<T>{ //T为要比较的类型

Public int compareTo(T to); //联想到String里的compareTo方法

}

注意:对一个数组排序,数组的元素类型必须实现Comparable接口

compareTo()方法返回3类数据:1(大于),0(等于),-1(小于)

范例:使用比较器

|  |
| --- |
| class test{  public static void main(String args[]) throws Exception{  Book bs[] = new Book[]{  new Book("java开发",3.5),  new Book("java开发",5.5),  new Book("java开发",4.5)  };  Arrays.sort(bs); //compareTo()由sort()自动调用  System.out.print(Arrays.toString(bs));  }  }  class Book implements Comparable<Book>{ T为要比较的类型  private String title;  private double price;  public Book(String title, double price) {  this.title = title;  this.price = price;  }  @Override  public String toString() {  return "Book{" +  "title='" + title + '\'' +  ", price=" + price +  '}';  }  @Override  public int compareTo(Book o) { // Arrays.sort()会自动调用此方法进行比较  if(this.price > o.price){  return 1;  }else if(this.price < o.price){  return -1;  }else{  return 0;  }  }  } |

重写CompareTo方法时,还可以这么写

return new Double(this.price).compareTo(new Double(o.price));

# Comparator挽救的比较器

Java.util

**作用:**Comparable接口在类的定义时就默认实现好了,如果有一个类开发完善了,但是没有实现Comparable接口,就不能排序了,这时就需要Comparator了

**定义:**

@FunctionalInterface

public interface Comparator<T>{

public int compare(T o1,T o2);

public boolean equals(Object obj);

}

**使用Comparable接口时利用的是Arrays类中的sort()方法**

**而使用Comparator接口时用的是Arrays类中的**

public static <T> void sort(T[] a, Comparator<? super T> c)

范例:

|  |
| --- |
| class test{  public static void main(String args[]) throws Exception{  Book bs[] = new Book[]{  new Book("java开发",34.5),  new Book("java开发",31.5),  new Book("java开发",36.5)  };  Arrays.sort(bs,new BookComparator());  System.out.print(Arrays.toString(bs));  }  }  class Book{  private String title;  private double price;  public Book(String title, double price) {  this.title = title;  this.price = price;  }  public String getTitle() {  return title;  }  public void setTitle(String title) {  this.title = title;  }  public double getPrice() {  return price;  }  public void setPrice(double price) {  this.price = price;  }  @Override  public String toString() {  return "Book{" +  "title='" + title + '\'' +  ", price=" + price +  '}';  }  }  //排序工具类  class BookComparator implements Comparator<Book>{  @Override  public int compare(Book o1, Book o2) {  if(o1.getPrice()>o2.getPrice()){  return 1;  }else if(o1.getPrice()<o2.getPrice()){  return -1;  }else{  return 0;  }  }  } |