1、Comparator它是一个比较器接口，包括两个函数：compare() 和 equals()。一个类实现了Comparator接口，那么它就是一个“比较器”。其它的类，可以根据该比较器去排序。

public class AgeComparator implements Comparator {

Image.gif

Image(1).gif

  public int compare(Object op1, Object op2) {

Image(2).gif

    Employee eOp1 = (Employee) op1;

    Employee eOp2 = (Employee) op2;

    // 按年龄排序

    return eOp1.getAge()-(eOp2.getAge());

Image(3).gif

  }

Image(4).gif

}

Image(5).gif

Image(6).gif

Image(7).gif

public class NameComparator implements Comparator {

  public int compare(Object op1, Object op2) {

    Employee eOp1 = (Employee) op1;

    Employee eOp2 = (Employee) op2;

    // 按姓名排序

    return eOp1.getName().compareTo(eOp2.getName());

  }

}

public class SalaryComparator implements Comparator {

  public int compare(Object op1, Object op2) {

    Employee eOp1 = (Employee) op1;

    Employee eOp2 = (Employee) op2;

    // 按薪水排序

    return eOp1.getSalary()-(eOp2.getSalary());

  }

}

public class Employee{

  protected String name;

  protected int age;

  protected int salary;

  public Employee(String name,int age,int salary){

    this.name=name;

    this.age=age;

    this.salary=salary;

  }

  public int getAge() {

    return age;

  }

  public void setAge(int age) {

    this.age = age;

  }

  public String getName() {

    return name;

  }

  public void setName(String name) {

    this.name = name;

  }

  public int getSalary() {

    return salary;

  }

  public void setSalary(int salary) {

    this.salary = salary;

  }

}

public class Main{

  public static void main(String[] args){

    List<Employee> employees=new ArrayList<Employee>();

    employees.add(new Employee("Andy",21,2000));

    employees.add(new Employee("Felix",21,3000));

    employees.add(new Employee("Bill",35,20000));

    employees.add(new Employee("Helen",21,10000));

    employees.add(new Employee("Cindy",28,8000));

    employees.add(new Employee("Douglas",25,5000));

    // 按名称排序

      Collections.sort(employees,new NameComparator());

      display(employees);

      // 按年龄排序

      Collections.sort(employees,new AgeComparator());

      display(employees);

      // 按薪水排序

      Collections.sort(employees,new SalaryComparator());

      display(employees);

  }

  public static void display(List<Employee> employees){

    for(Employee e:employees){

      System.out.println("雇员名="+e.getName()+" 年龄="+e.age+" 薪水="+e.getSalary());

    }

    System.out.println();

  }

}

2、Comparable它是一个排序接口，只包含一个函数compareTo()。

一个类实现了Comparable接口，就意味着“该类本身支持排序”，它可以直接通过Arrays.sort() 或 Collections.sort()进行排序。

3、