## 前言

在开发过程中经常会遇到比较排序的问题：

1. 比如说对集合数组的排序等情况，基本类型都提供了默认的比较算法，如string提供了按字母进行排序，而int整数则是根据整数大小进行排序.
2. 但是在引用类型中(具有多个字段)，那么这个排序当然也是取决于我们特定的值。

## IComparable接口

1、说明：

该接口由其值可以排序或排序的类型实现，并提供强类型的比较方法以对泛型集合对象的成员进行排序，例如数字可以大于第二个数字，一个字符串可以在另一个字符串之前以字母顺序出现。他要求实现类型定义的一个方法，

CompareTo(T)该方法指示当前实现在排序顺序中的位置是在同一个类型和第二个对象之前、之后还是与其相同。通常，不会直接从开发人员代码中调用方法。相反他由List.Sort()和Add等方法自动调用。

通常，提供Icomparable实现的类型还IEquatable实现接口。IEquatable接口Equals定义方法，该方法确定实现类型的实例的相等性。

CompareTo(T)方法的实现必须Int32返回具有以下三个值之一的，如下表所示。

| **值** | **含义** |
| --- | --- |
| 小于零 | 此对象在排序顺序中位于CompareTo方法所指定的对象之前。 |
| 零 | 此当前实例在排序顺序中与CompareTo方法参数指定的对象出现在同一位置。 |
| 大于零 | 此当前实例位于排序顺序中由CompareTo方法自变量指定的对象之后。 |

## 二、IComparer接口

IComparable 接口的CompareTo方法一次只能对一个字段进行排序，因此无法对不同的属性进行排序。IComparer接口提供了Compare方法，该方法比较两个对象并返回一个值，该值指示一个对象小于，等于或大于另一个对象。

实现IComparer接口的类必须提供比较两个对象的Compare方法。例如，您可以创建一个StudentComparer类，该类实现IComparer，并具有一个Compare方法，该方法按Name比较Student对象。然后，您可以将StudentComparer对象传递给Array.Sort方法，它可以使用该对象对Student对象的数组进行排序。

以下是综合示例：

电脑屏幕的截图

描述已自动生成