

BasicAlgorith 帮助文档

本项目涵盖了数据结构中的七个重要的排序算法，选择，插入，冒泡，归并，希尔，快速和堆排序，可实现任何的 List 类型和数组类型，进行排序，除 String 类型外都可以进行排序，为使用开发者和学习者使用这七种经典算法提供方便。

注：若使用本项目进行排序，使用非基本类型，必须实现 `YouCompare<T>` 接口，定义比较的数据。

`public enum Sortord`

两个参数 `ASCE(升序排列),DESC(降序排列);`

`class SortordStaticWay<T>`

方法：

`protected void swap(T[] array, int targer1, int targer2)`

交换数组中位置为 `targer1` 和 `targer2` 两个位置的数据

`protected boolean checkType (Class clazz)`

检查是否是基本数据类型，是 返回 `true`

已知子类：

`BubblSort, HeapSort, InsertSort, QuickSort, SelectSort, ShellSort, TwoWayMergeSort`

`public interface PrintTime`

`void println(Object[] t,int time);`

用于方法回调的参数，`t` 为排序后数组，`time` 为排序躺数，从 1 开始

`interface YouCompare<T>`

方法：

`boolean compare(T compare2);`

比较两个值大小，非基本数据类型排序必须实现该方法

`interface Sort<T>`

方法

`List<T> sort(List<T> list, Sortord asce,PrintTime printTime);`

`list`:待排序集合

`asce` : 排序规律 升序或降序

`printTime`: 回调函数

注：返回类型可能会破坏原有的结构，默认为 `ArrayList`, 还需更新改进

`T[] sort(T[] array, Sortord asce,PrintTime printTime);`

`array`:待排序数组

asce : 排序规律 升序或降序
printTime: 回调函数

```
ArrayList<T> sort(List<T> arrayList, Sortord asce);  
arrayList:待排序集合  
asce : 排序规律 升序或降序
```

```
T[] sort(T[] array, Sortord asce);  
array:待排序数组  
asce : 排序规律 升序或降序
```

已知实现类:

BubblSort, HeapSort, InsertSort, QuickSort, SelectSort, ShellSort, TwoWayMergeSort

class BubblSort<T>(冒泡排序)

```
List<T> sort(List<T> list, Sortord asce,PrintTime printTime);  
list:待排序集合  
asce : 排序规律 升序或降序  
printTime: 回调函数
```

```
T[] sort(T[] array, Sortord asce,PrintTime printTime);  
array:待排序数组  
asce : 排序规律 升序或降序  
printTime: 回调函数
```

```
ArrayList<T> sort(List<T> arrayList, Sortord asce);  
arrayList:待排序集合  
asce : 排序规律 升序或降序
```

```
T[] sort(T[] array, Sortord asce);  
array:待排序数组  
asce : 排序规律 升序或降序
```

class HeapSort<T> (堆排序)

注: 参数 printTime 在堆排序中未起到作用

```
List<T> sort(List<T> list, Sortord asce,PrintTime printTime);  
list:待排序集合  
asce : 排序规律 升序或降序  
printTime: 回调函数
```

```
T[] sort(T[] array, Sortord asce,PrintTime printTime);  
array:待排序数组  
asce : 排序规律 升序或降序
```

printTime: 回调函数

```
ArrayList<T> sort(List<T> arrayList, Sortord asce);
```

arrayList:待排序集合

asce : 排序规律 升序或降序

```
T[] sort(T[] array, Sortord asce);
```

array:待排序数组

asce : 排序规律 升序或降序

class InsertSort<T> (插入排序)

```
List<T> sort(List<T> list, Sortord asce,PrintTime printTime);
```

list:待排序集合

asce : 排序规律 升序或降序

printTime: 回调函数

```
T[] sort(T[] array, Sortord asce,PrintTime printTime);
```

array:待排序数组

asce : 排序规律 升序或降序

printTime: 回调函数

```
ArrayList<T> sort(List<T> arrayList, Sortord asce);
```

arrayList:待排序集合

asce : 排序规律 升序或降序

```
T[] sort(T[] array, Sortord asce);
```

array:待排序数组

asce : 排序规律 升序或降序

class QuickSort <T> (快速排序)

```
List<T> sort(List<T> list, Sortord asce,PrintTime printTime);
```

list:待排序集合

asce : 排序规律 升序或降序

printTime: 回调函数

```
T[] sort(T[] array, Sortord asce,PrintTime printTime);
```

array:待排序数组

asce : 排序规律 升序或降序

printTime: 回调函数

```
ArrayList<T> sort(List<T> arrayList, Sortord asce);
```

arrayList:待排序集合
asce : 排序规律 升序或降序

```
T[] sort(T[] array, Sortord asce);
```

array:待排序数组
asce : 排序规律 升序或降序

class SelectSort<T> (选择排序)

```
List<T> sort(List<T> list, Sortord asce,PrintTime printTime);
```

list:待排序集合
asce : 排序规律 升序或降序
printTime: 回调函数

```
T[] sort(T[] array, Sortord asce,PrintTime printTime);
```

array:待排序数组
asce : 排序规律 升序或降序
printTime: 回调函数

```
ArrayList<T> sort(List<T> arrayList, Sortord asce);
```

arrayList:待排序集合
asce : 排序规律 升序或降序

```
T[] sort(T[] array, Sortord asce);
```

array:待排序数组
asce : 排序规律 升序或降序

class ShellSort<T> (希尔排序)

```
List<T> sort(List<T> list, Sortord asce,PrintTime printTime);
```

list:待排序集合
asce : 排序规律 升序或降序
printTime: 回调函数

```
T[] sort(T[] array, Sortord asce,PrintTime printTime);
```

array:待排序数组
asce : 排序规律 升序或降序
printTime: 回调函数

```
ArrayList<T> sort(List<T> arrayList, Sortord asce);
```

arrayList:待排序集合
asce : 排序规律 升序或降序

```
T[] sort(T[] array, Sortord asce);
```

array:待排序数组
asce : 排序规律 升序或降序

class TwoWayMergeSort<T> (二路归并排序)

```
List<T> sort(List<T> list, Sortord asce,PrintTime printTime);
```

list:待排序集合

asce : 排序规律 升序或降序

printTime: 回调函数

```
T[] sort(T[] array, Sortord asce,PrintTime printTime);
```

array:待排序数组

asce : 排序规律 升序或降序

printTime: 回调函数

```
ArrayList<T> sort(List<T> arrayList, Sortord asce);
```

arrayList:待排序集合

asce : 排序规律 升序或降序

```
T[] sort(T[] array, Sortord asce);
```

array:待排序数组

asce : 排序规律 升序或降序

代码事例:

```
public class Car implements YouCompare<Car> {

    private int height;
    private int weight;
    //省略get和set方法
    public Car(int height, int weight) {
        super();
        this.height = height;
        this.weight = weight;
    }

    public boolean compare( Car compare2) {
        if(this.getHeight()>compare2.getHeight()){
            return true;
        }else if(this.getHeight()<compare2.getHeight()){
            return false;
        }else {
            if(this.getWeight()>=compare2.getWeight()){
                return true;
            }
        }
    }
}
```

```

        }else {
            return false;
        }
    }
}

}

public static void main(String[] args) {
    ArrayList<Car> arrayList = new ArrayList<Car>();
    arrayList.add(new Car(8, 6));
    arrayList.add(new Car(4, 6));
    arrayList.add(new Car(7, 6));
    arrayList.add(new Car(48, 6));
    arrayList.add(new Car(10, 6));
    arrayList.add(new Car(39, 6));
    arrayList.add(new Car(9, 6));
    arrayList.add(new Car(12, 6));
    arrayList.add(new Car(11, 6));
    arrayList.add(new Car(35, 6));
    arrayList.add(new Car(40, 6));
    System.out.println("冒泡排序");
    List<Car> newBubblSort = new BubblSort<Car>().sort(arrayList,
        Sortord.ASCE, new PrintTime() {
            public void println(Object[] t, int time) {
                // TODO Auto-generated method stub
                if (t == null) {
                    System.out.println("null");
                    return;
                }
                System.out.print("遍历第" + time + "躺: ");
                for (int i = 0; i < t.length; i++) {
                    System.out.print((Car) t[i] + " ");
                }
                System.out.println();
            }
        });
}
}

```

作者 游神
 电话 15712375939
 QQ 727204747
 希望互相交流
 正在找工作 ing 随便写点东西*. *