## Java排序

排序算法，基本的高级语言都有一些提供。C语言有**qsort()**函数，C++有**sort()**函数，java语言有**Arrays**类(不是Array)。用这些排序时，都可以写自己的排序规则。

　　Java API对Arrays类的说明是：**此类包含用来操作数组（比如排序和搜索）的各种方法。**

**1.对基本数据类型的数组的排序**

说明：(1)Arrays类中的sort()使用的是“经过调优的**快速排序**法”;

　　 (2)比如int[]，double[]，char[]等基数据类型的数组，Arrays类之只是提供了默认的升序排列，没有提供相应的降序排列方法。

　　 (3)要对基础类型的数组进行降序排序，需要将这些数组转化为对应的封装类数组，如Integer[]，Double[]，Character[]等，对这些类数组进行排序。(其实还不如先进行升序排序，自己在转为将序)。

　　  用**默认**的升序对数组排序

函数原型：

static **void**[**sort**](mk:@MSITStore:F:\Users\liu\Desktop\JAVA%E4%B8%AD%E6%96%87API%E6%96%87%E6%A1%A3.CHM::/api/java/util/Arrays.html#sort(int[]))(int[] a)   对指定的 int 型数组按数字升序进行排序。

static void[**sort**](mk:@MSITStore:F:\Users\liu\Desktop\JAVA%E4%B8%AD%E6%96%87API%E6%96%87%E6%A1%A3.CHM::/api/java/util/Arrays.html#sort(int[], int, int))(int[] a, int fromIndex, int toIndex)  对指定 int 型数组的指定范围按数字升序进行排序。

　　代码实例：

import java.util.Arrays;

public class ArraysSort\_11 {

public static void main(String args[])

{

int[] a={1,4,-1,5,0};

Arrays.sort(a);

//数组a[]的内容变为{-1,0,1,4,5}

for(int i=0;i<a.length;i++)

System.out.print(a[i]+" ");

}

}

**2.对复合数据类型的数据的排序**

 　　函数原型：

(1)public static<T> void sort(T[] a，Comparator c)  根据指定比较器产生的顺序对指定对象数组进行排序。

(2)public static<T> void sort(T[] a，int fromIndex，int toIndex，Comparator c)  根据指定比较器产生的顺序对指定对象数组的指定范围进行排序。

　　说明：这个两个排序算法是“经过调优的**合并排序**”算法。

　　代码实例：

package aa;

import java.util.Arrays;

import java.util.Comparator;

public class Arraysort {

Point[] arr;

Arraysort(){

arr=new Point[4]; //定义对象数组arr，并分配存储的空间

for(int i=0;i<4;i++)

arr[i]=new Point();

}

public static void main(String[] args) {

Arraysort sort=new Arraysort();

sort.arr[0].x=2;sort.arr[0].y=1; //初始化，对象数组中的数据

sort.arr[1].x=2;sort.arr[1].y=2;

sort.arr[2].x=1;sort.arr[2].y=2;

sort.arr[3].x=0;sort.arr[3].y=1;

Arrays.sort(sort.arr, new MyComprator()); //使用指定的排序器，进行排序

for(int i=0;i<4;i++) //输出排序结果

System.out.println("("+sort.arr[i].x+","+sort.arr[i].y+")");

}

}

class Point{

int x;

int y;

}

//比较器，x坐标从小到大排序；x相同时，按照y从小到大排序

class MyComprator implements Comparator {

public int compare(Object arg0, Object arg1) {

Point t1=(Point)arg0;

Point t2=(Point)arg1;

if(t1.x != t2.x)

return t1.x>t2.x? 1:-1;

else

return t1.y>t2.y? 1:-1;

}

}