1、算法思想：选择排序，从头至尾扫描序列，找出最小的一个元素，和第一个元素交换，接着从剩下的元素中继续这种选择和交换方式，最终得到一个有序序列。

2、算法过程

举个栗子（第一趟的排序过程）

原始序列：49、38、65、97、76、13、27、49

1）在进行选择排序过程中分成有序和无序两个部分，开始都是无序序列

结果：49、38、65、97、76、13、27、49

2）从无序序列中取出最小的元素13，将13同无序序列第一个元素交换，此时产生仅含一个元素的有序序列，无序序列减一

结果：{13、}   {38、65、97、76、49、27、49}

3）从无序序列中取出最小的元素27，将27同无序序列第一个元素交换，此时产生仅两个元素的有序序列，无序序列减一

结果：{13、27、}   {65、97、76、49、38、49}

4）从无序序列中取出最小的元素38，将38同无序序列第一个元素交换，此时产生含三个元素的有序序列，无序序列减一

结果：{13、27、38、}   {97、76、49、65、49}

5）从无序序列中取出最小的元素49，将49同无序序列第一个元素交换，此时产生含四个个元素的有序序列，无序序列减一

结果：{13、27、38、49、}   {76、97、65、49}

6）从无序序列中取出最小的元素49，将49同无序序列第一个元素交换，此时产生含五个元素的有序序列，无序序列减一

结果：{13、27、38、49、49、}   {97、65、76}

7）从无序序列中取出最小的元素65，将65同无序序列第一个元素交换，此时产生含六个元素的有序序列，无序序列减一

结果：{13、27、38、49、49、65}   {97、76}

8）从无序序列中取出最小的元素76，将76同无序序列第一个元素交换，此时产生含七个元素的有序序列，无序序列减一

结果：{13、27、38、49、49、65、76、}   {97}

9）最后一个元素肯定是最大元素，无序排序直接生产一个有序的序列

结果：{13、27、38、49、49、65、76、97}

public class SelectSort {

public static void main(String[] args) {

int [] arr = {49,38,65,97,76,13,27,49};

selectSort(arr,arr.length);

}

public static void selectSort(int [] arr,int n){

for (int i = 0; i < n - 1; i++) {

int index = i;

int j;

// 找出最小值得元素下标

for (j = i + 1; j < n; j++) {

if (arr[j] < arr[index]) {

index = j;

}

}

int tmp = arr[index];

arr[index] = arr[i];

arr[i] = tmp;

System.out.println(Arrays.toString(arr));

}

}

}

————————————————

版权声明：本文为CSDN博主「那人好像一条猿」的原创文章，遵循CC 4.0 by-sa版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/changhangshi/article/details/82740541