### JAVA第一阶段—DAY05-JAVA 案例

1. 对一维数组进行元素的初始化、元素的修改、元素的获取。

* 参考答案

public static void main(String[] args) {

// 定义一个定长数组

int[] array = new int[3];

// 1、元素赋值

array[0]=1;

array[1]=2;

array[2]=3;

// 2、元素修改

array[0]=11;

// 3、元素获取

int a=array[0];

// 打印数组，Arrays先了解即可

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

}

1. 对二维数组进行元素的初始化、元素的修改、元素的获取。

* 参考答案

public static void main(String[] args) {

// 定义一个指定容量的二维数组

int[][] array = new int[2][3];

// 1、赋值

array[0][0]=0;

array[0][1]=1;

array[0][2]=2;

array[1][0]=3;

array[1][1]=4;

// 2、修改

array[0][0]=110;

// 3、获取

int a=array[0][0];

// 打印数组

System.out.println(Arrays.deepToString(array));

}

1. 使用2种方式完成数组的遍历。

* 参考答案

public static void main(String[] args) {

int[] arr = { 1, 2, 3, 4, 5 };

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

System.out.println(arr[i]);

}

}

public static void main(String[] args) {

int[] arr = { 1, 2, 3, 4, 5 };

for (int e : arr) {

System.out.println(e);

}

}

1. 获取数组元素的最大值。

* 参考答案

public static void main(String[] args) {

int[] arr = { 5, 15, 2000, 10000, 100, 4000 };

//定义变量，保存数组中0索引的元素

int max = arr[0];

//遍历数组，取出每个元素

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

//遍历到的元素和变量max比较

//如果数组元素大于max

if (arr[i] > max) {

//max记录住大值

max = arr[i];

}

}

System.out.println("数组最大值是： " + max);

}

1. 使用冒泡排序和选择排序两种方式完成数组元素排序。

* 参考答案

public static void main(String[] args) {

// 定义一个元素顺序杂乱的数组

int[] arr = { 4, 1, 3, 6, 2, 5};

// 外层循环表示比较轮数

for (int i = 1; i <= arr.length-1; i++) {

// 内层循环对比相邻两个数

for (int j = 0; j < arr.length-i; j++) {

// 定义一个临时变量

int tempMax=0;

// 进行元素位置交换

if(arr[j]>arr[j+1]) {

tempMax=arr[j];

arr[j]=arr[j+1];

arr[j+1]=tempMax;

}

}

}

// 打印

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

}

public static void main(String[] args) {

// 定义一个元素顺序杂乱的数组

int[] arr = { 4, 1, 3, 6, 2, 5};

// 外层循环循环每一个元素，拿着与内存循环的每个元素对比

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

// 内层循环对比相邻两个数

for (int j = i+1; j < arr.length; j++) {

// 定义一个临时变量

int temp=0;

// 进行元素位置交换

if(arr[i] > arr[j]) {

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

// 打印

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

}

1. 完成数组元素的二分查找操作。

* 参考答案

public static void main(String[] args) {

// 定义一个元素顺序有序的数组

int[] arr = { 1, 2, 3, 4, 5, 6};

// 定义一个要查找的元素

int a=61;

// 初始化查找的起始位置、终止位置和中间位置

int begin=0;

int end=arr.length;

int mid=(begin+end)/2;

// 确保不会出现重复查找，越界

while(begin <=end) {

if(a> arr[mid]) {

begin=mid+1;

mid=(begin+end)/2;

}else if(a< arr[mid]) {

end=mid-1;

mid=(begin+end)/2;

}else {

System.out.println("元素位置为："+mid);

break;

}

// 如果起始角标已经超过终止角标还没找到，那就是数组没有这个原生

if(end<=begin) {

System.out.println("没有该元素");

break;

}

}

}

1. 完成数组元素的反转。

* 参考答案

public static void main(String[] args) {

// 定义一个元素顺序杂乱的数组

int[] arr = { 4, 1, 3, 6, 2, 5};

// 对半遍历数组即可

for (int i = 0; i < arr.length/2; i++) {

// 首尾元素进行对换

int temp=arr[i];

arr[i]=arr[arr.length-1-i];

arr[arr.length-1-i]=temp;

}

// 打印

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

}