**数组作业shu**

1. **填空题**
2. **数组会在内存中开辟一块\_\_连续\_\_\_\_\_\_的空间，每个空间相当于之前的一个变量，称为数组的元素。数组的长度一经确定，就无法再改变。**
3. **要获取一个数组的长度，可以通过\_.length\_\_\_\_\_属性来获取，但获取的只是为数组分配的空间的数量，而不是数组中实际已经存放的元素的个数。**
4. **\_声明\_\_仅仅是给出了数组名字和元素的数据类型，要想真正的使用数组还必须使用new关键字为它分配内存空间。**
5. **创建数组后，系统会给每一个数组元素一个默认的值，如String类型元素的默认值是\_\_\_\_null\_\_\_\_\_\_\_\_。**
6. **在Java中有二维数组int [ ] [ ] array={{1,2,3},{4,5}} ，可以使用\_array[1]\_.length\_\_\_\_\_\_\_\_得到二维数组中第二维中第一个数组的长度。**
7. **数组元素下标(或索引)的范围是\_\_0—数组名.length-1\_\_\_\_\_\_\_\_\_。**
8. **选择题**

|  |  |  |
| --- | --- | --- |
| **1.** | **在Java中，以下程序能正确为数组赋值的是（ACD）。（选择二项）** | |
|  |  |  |
|  | **A** | **int a[]={1,2,3,4};** |
|  | **B.** | **int b[4]={1,2,3,4};** |
|  | **C.** | **int c[];**  **c=new int[] {1,2,3,4}; //大部分情况和A一样 少部分c可以A不可以** |
|  | **D.** | **int d[];**  **d=new int[]{1,2,3,4};** |

|  |  |  |
| --- | --- | --- |
| **2.** | **数组元素的索引可以是（ D ）。（选择一项）** | |
|  |  |  |
|  | **A** | **整型常量**  **scores[0]** |
|  | **B.** | **整型变量**  **scores[i]** |
|  | **C.** | **整型表达式**  **scores[i+1]** |
|  | **D.** | **以上都可以** |

|  |  |  |
| --- | --- | --- |
| **3.** | **已知表达式int [] m={0,1,2,3,4,5,6};下面（ B ）表达式的值与数组最大下标数相等。（选择一项）** | |
|  |  |  |
|  | **A** | **m.length()** |
|  | **B.** | **m.length-1** |
|  | **C.** | **m.length()+1** |
|  | **D.** | **m.length+1** |

|  |  |  |
| --- | --- | --- |
| **4.** | **在Java中，以下定义数组的语句正确的是（ CD ）。（选择二项）** | |
|  |  |  |
|  | **A** | **int t[10]=new int[ ];** |
|  | **B.** | **char [ ]a=new char[5];**  **char []a={‘a’,’b’};** |
|  | **C.** | **String [ ] s=new String [10];//静态动态？** |
|  | **D.** | **double[ ] d [ ]=new double [4][ ];**  **double[][] d;**  **double d[][];**  **Int[][] ints = new int[3][];**  **Int[][] ints = new int[][3];**  **N维数组的每一个元素的类型是n-1维数组类型，也就是n维数组去掉一个[]** |
| **5.** | **在Java中,下面代码的输出结果为（ A ）。（选择一项）** | |
|  | **public static void main(String[] args) {**  **int[] arrA = { 12, 22, 8, 49, 3 };**  **int k = 0;**  **int len = arrA.length;//5**  **for (int i = 0; i < len; i++) {**  **for (int j = i + 1; j < len; j++) {**  **if (arrA[i] > arrA[j]) {**  **k = arrA[i];**  **arrA[i] = arrA[j];**  **arrA[j] = k;**  **}**  **}**  **}**  **for (int i = 0; i < arrA.length; i++) {**  **System.*out*.print(arrA[i]);**  **if (i < arrA.length - 1) {**  **System.*out*.print("，");**  **}**  **}**  **}** | |
|  |  |  |
|  | **A** | **3，8，12，22，49** |
|  | **B.** | **12，22，8，49，3** |
|  | **C.** | **49，22，12，8，3** |
|  | **D.** | **编译错误** |

|  |  |  |
| --- | --- | --- |
| **6.** | **分析下面的Java源程序，编译后的运行结果是（ B ）。（选择一项）** | |
|  | **import java.util.\*;**  **public class Test {**  **public static void main(String[ ] args) {**  **int [ ] numbers=new int[ ]{1,2,3};**  **System.*out*.println(Arrays.*binarySearch*(numbers, 2));**  **数组的二分查找的效率很高，但是得有个前提：数组有序**  **Java中有很多类似Arrays的工具类，比如Collections，封装了很多经典算法的实现，好用的方法。**  **}**  **}** | |
|  |  | **输出** |
|  | **A** | **输出：0** |
|  | **B.** | **输出：1** |
|  | **C.** | **输出：2** |
|  | **D.** | **输出：3** |

|  |  |  |
| --- | --- | --- |
| **7.** | **以下选项中关于Java中方法的可变参数的说法正确的是（ ）。（选择二项）** | |
|  |  |  |
|  | **A** | **可变参数是JDK1.5增加的内容，用来表示方法的形参 int … a** |
|  | **B.** | **一个方法可以没有可变参数，可以有1个或者多个可变参数** |
|  | **C.** | **可变参数可以被当作数组参数来处理** |
|  | **D.** | **可变参数对应的实参可以1个，2个，多个，但不能0个，还可以是一个数组** |

**1 掌握可变长参数的语法 int... 和本质 就是数组**

**2 掌握可编程参数的注意点 2.1 注意不能2个 2.2 如果有多个参数，可变长参数必须是最后一个**

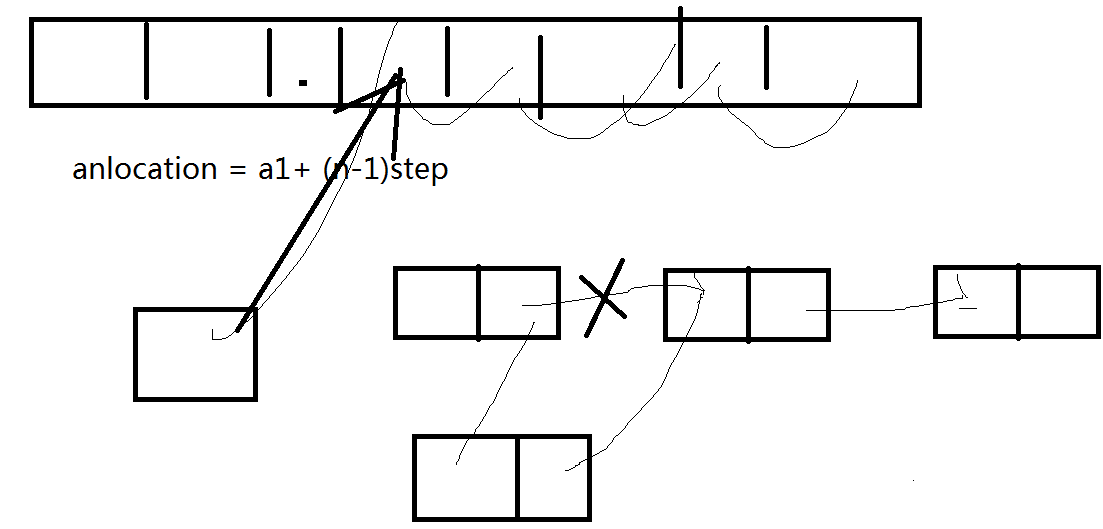
|  |  |  |
| --- | --- | --- |
| **8.** | **以下选项中能够正确创建一个数组的是（A D ）。（选择二项）** | |
|  |  |  |
|  | **A.** | **float []f[] = new float[6][6];** |
|  | **B.** | **float f[][] = new float[][];** |
|  | **C.** | **float [6][]f = new float[6][6];** |
|  | **D.** | **float [][]f = new float[6][];**  **f[0]=new float[2];**  **f[1]=new float[3];**  **….** |

|  |  |  |
| --- | --- | --- |
| **9.** | **下面的数组定义哪些是正确的？（ AC ）。（选择二项）** | |
|  |  |  |
|  | **A** | **int a[][] = new int[3,3];** |
|  | **B.** | **int a[3][3] = new int[][];** |
|  | **C.** | **int a[][] = new int[3][3];** |
|  | **D.** | **int []a[] = new int[3][3];** |

1. **判断题**
2. **数组可以声明为任何数据类型，包括任何基本数据类型和引用数据类型。（ √ ）**
3. **数组的长度是确定的，数组一旦被创建，它的大小就是不可以改变的。但是其元素类型可以是不同类型，允许出现混合类型。（×）**
4. **数组中的元素的类型必须是相同的，并且数组中的元素是有序的。（** **错的 ）**
5. **声明数组并分配空间组的每个元素将会赋予初始值。（ √ ）**
6. **创建数组后，系统会给每个数组元素一个默认值，如double型元素的默认值是0.0。（√ ）**
7. **数组的主要优点是按照索引查找某个元素效率高，同时按照元素值查询某个元素效率也很高，但是添加和删除元素需要大量移动元素，效率低下。（ √ ）**

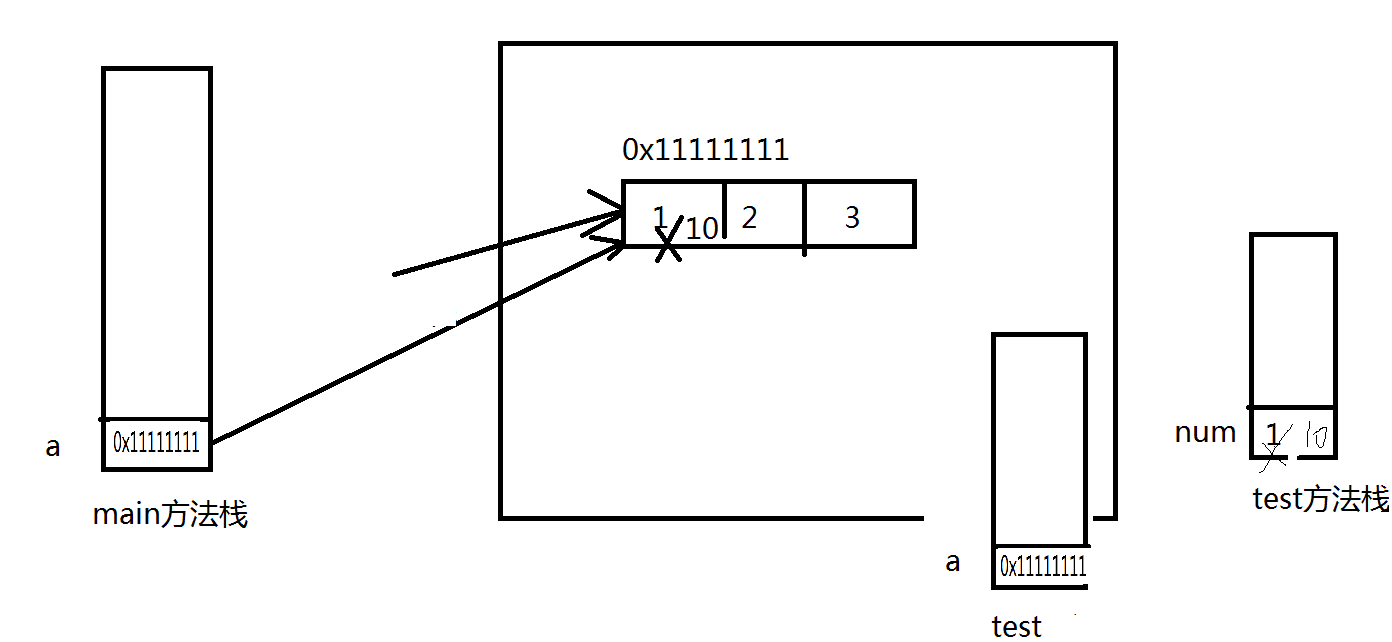
**数组擅长查找**

**链表擅长增删**



1. **数组的某个元素被传递给一个方法并被该方法修改，当被调用方法执行完毕时，这个元素中含有修改过的数值。（ √ ）**

|  |
| --- |
| ***public static void main(String[] args) {***  ***int[] a = {1,2,3};***  ***test(a[0]);***  ***System.out.println(a[0]);***  ***}***  ***public static void test(int num){***  ***num = 10;***  ***}*** |



1. **Java允许创建不规则数组，即Java多维数组中各行的列数可以不同。（ √ ）**

**int a[][]={{1,2,3},**

**{4,5},**

**{7,8, 9}**

**};**

1. **对于数组int[][] t={{1,2,3},{4,5,6}}来说，t.length等于3，t[0].length等于2（ × ）**
2. **数组是引用类型，数组也是对象。（ √）**
3. **简答题**
4. **数组的特点。**

**1）数组本身是引用数据类型，而数组中的元素可以是任何数据类型，包括基本数据类型和引用数据类型。**

**2）创建数组对象会在内存中开辟一整块连续的空间，而数组名中引用的是这块连续空间的首地址。**

**3）数组的长度一旦确定，就不能修改。**

**4）可以直接通过下标(或索引)的方式调用指定位置的元素，速度很快。**

1. **数组的优缺点**

**优点：可以直接通过下标或者索引的方式调用指定位置的元素，速度很块。**

**缺点：添加和删除元素需要大量移动元素，效率低下**

1. **冒泡排序的算法。**

**比较相邻的元素。如果第一个比第二个大（升序），就交换他们两个。**

**对每一对相邻元素作同样的工作，从开始第一对到结尾的最后一对。这步 做完后，最后的元素会是最大的数。**

**针对所有的元素重复以上的步骤，除了最后一个。**

**持续每次对越来越少的元素重复上面的步骤，直到没有任何一对数字需要 比较为止。**

1. **数组的三种初始化方式是什么?**

**Int[][]arr=new int[3][2]**

**Int[][]arr=new int[3][]**

**Int[][]arr=new int{{1,2,3},{4,5}}**

1. **编码题**
2. **数组查找操作：定义一个长度为10 的一维字符串数组，在每一个元素存放一个单词；然后运行时从命令行输入一个单词，程序判断数组是否包含有这个单词，包含这个单词就打印出“Yes”，不包含就打印出“No”。**

**String []str =new String[10];**

**str[0]="apple";**

**str[1]="appl";**

**str[2]="app";**

**str[3]="ap";**

**str[4]="bananer";**

**str[5]="banane";**

**str[6]="banan";**

**str[7]="bana";**

**str[8]="ban";**

**str[9]="can";**

**Scanner input =new Scanner(System.in);**

**System.out.println("请输入一个单词:");**

**String str1=input.next();**

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

**if(str1.equals(str[i])) {**

**System.out.println("Yes");**

**break;**

**}else {**

**System.out.println("No");**

**break;**

**}**

**}**

**2.获取数组最大值和最小值操作：利用Java的Math类的random()方法，编写函数得到 0到n之间的随机数，n是参数。并找出产生50个这样的随机数中最大的、最小的数， 并统计其中>=60的有多少个。  
 int [] num=new int[50];**

**Scanner input =new Scanner(System.in);**

**System.out.println("请输入一个参数n:");**

**int n =input.nextInt();**

**int count =0;**

**for (int i = 0; i < 50; i++) {**

**num[i]=(int)(Math.random()\*n);**

**}**

**int max =num[0];**

**int min =num[0];**

**for (int i = 1; i < num.length; i++) {**

**if (num[i]>max) {**

**max=num[i];**

**}**

**if(num[i]<min){**

**min=num[i];**

**}**

**if(num[i]>=60) {**

**count++;**

**}**

**}**

**System.out.println(count);**

1. **数组逆序操作：定义长度为10的数组，将数组元素对调，并输出对调前后的结果。**

**思路：把0索引和arr.length-1的元素交换，把1索引和arr.length-2的元素交换…..**

**只要交换到arr.length/2的时候即可。**

**package com.uek.java;**

**public class zuoye {**

**public static void main(String[] args) {**

**int []arr=new int[] {1,2,3,4,5,6,7,8,9,10};**

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

**int temp =0;**

**temp=arr[i];**

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

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

**}**

**for (int i : arr) {**

**System.out.print(i+"\t");**

**}**

**}**

**}**

1. **合并数组操作：现有如下一个数组：   int oldArr[]={1,3,4,5,0,0,6,6,0,5,4,7,6,7,0,5}   要求将以上数组中值为0的项去掉，将不为0的值存入一个新的数组，生成的新数组为：**

**package com.uek.java;**

**public class zuoye {**

**public static void main(String[] args) {**

**int oldArr[]={1,3,4,5,0,0,6,6,0,5,4,7,6,7,0,5};**

**int count =0;**

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

**if(oldArr[i]==0) {**

**count++;**

**}**

**}**

**int []newArr=new int[oldArr.length-count];**

**int j = 0;**

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

**if(oldArr[i]!=0) {**

**newArr[j++] =oldArr[i];**

**}**

**}**

**for (int temp : newArr) {**

**System.out.println(temp);**

**}**

**}**

**}**

1. **二分法查找操作：使用二分法查找有序数组中元素。找到返回索引，不存在输出-1。**

**分析：二分法查找的前提是数组有序。**

**假如有一组数为3，12，24，36，55，68，75，88要查给定的值24.可设三个变量front，mid，end分别指向数据的上界，中间和下界，mid=（front+end）/2.**

1. **开始令front=0（指向3），end=7（指向88），则mid=3（指向36）。因为mid>x，故应在前半段中查找。**
2. **令新的end=mid-1=2，而front=0不变，则新的mid=1。此时x>mid，故确定应在后半段中查找。**
3. **令新的front=mid+1=2，而end=2不变，则新mid=2，此时a[mid]=x，查找成功。**
4. **如要查找的数不是数列中的数，例如x=25，当第三次判断时，x>a[mid]，按以上规律，令front=mid+1，即front=3，出现front>end的情况，表示查找不成功。**

**int []Arr=new int[] {3,12,24,36,55,68,75,88};**

**int start =0;**

**int end =Arr.length-1;**

**int select =24;**

**boolean flag =true;**

**int middle =0;**

**while(start<=end) {**

**int middleIndex=(start+end)/2;**

**int middleValue=Arr[middleIndex];**

**if(middleValue==select){**

**System.out.println("找到了！索引是："+middleIndex);**

**middle=middleIndex;**

**flag=false;**

**break;**

**}else if(middleValue>select) {**

**end=middleIndex-1;**

**}else {**

**start=middleIndex+1;**

**}**

**}**

**if(flag) {**

**System.out.println(-1);**

**}else {**

**System.out.println(middle);**

**}**

**}**

**}**

1. **二维数组遍历求和操作：用二重循环求出二维数组b所有元素的和：  
    int[][] b={{11},{21,22},{31,32,33}}**

**int[][]arr=new int[][] {{11},{21,22},{31,32,33}};**

**int sum=0;**

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

**for (int j = 0; j < arr[i].length; j++) {**

**sum+=arr[i][j] ;**

**}**

**}**

**System.out.println(sum);**

1. **可选题**
2. **生成一百个随机数，放入数组，然后排序输出。**
3. **题目：输入某年某月某日，判断这一天是这一年的第几天？**

**分析：以3月5日为例，先把前两个月的加起来，然后再加上5天即本年第几天，特殊情况，闰年且输入月份大于3需考虑多加一天。可定义数组存储1-12月各月天数。**

1. **使用二分法查找有序数组中元素。找到返回索引，不存在输出-1。使用递归实现**
2. **数组A：1，7，9，11，13，15，17，19；数组b：2，4，6，8，10**

**两个数组合并为数组c，按升序排列。 要求：使用Arrays类的方法快速实现。**