**数组作业**

1. **填空题**
2. 数组会在内存中开辟一块\_\_\_\_\_\_\_\_\_\_\_\_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\_\_\_\_\_\_\_\_\_\_\_\_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. \_\_\_\_\_\_\_\_\_\_\_\_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\_\_\_\_\_\_\_\_\_\_\_\_。
6. 在Java中有二维数组int [ ] [ ] array={{1,2,3},{4,5}} ，可以使用\_\_\_\_\_\_\_\_\_\_\_\_得到二维数组中第二维中第一个数组的长度。
7. **选择题**

|  |  |  |
| --- | --- | --- |
| **1.** | **在Java中，以下程序段能正确为数组赋值的是（ ）。（选择二项）** | |
|  |  |  |
|  | **A** | int a[]={1,2,3,4}; |
|  | **B.** | int b[4]={1,2,3,4}; |
|  | **C.** | int c[];c={1,2,3,4}; |
|  | **D.** | int d[];d=new int[]{1,2,3,4}; |

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

|  |  |  |
| --- | --- | --- |
| **3.** | **已知表达式int [] m={0,1,2,3,4,5,6};下面（ ）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（ ）。（选择二项）** | |
|  |  |  |
|  | **A** | int t[10]=new int[ ]; |
|  | **B.** | char [ ]a=”hello”; |
|  | **C.** | String [ ] s=new String [10]; |
|  | **D.** | double[ ] d [ ]=new double [4][ ]; |

|  |  |  |
| --- | --- | --- |
| **5.** | **在Java中,下面代码的输出结果为（ ）。（选择一项）** | |
|  | **public** **static** **void** main(String[] args) {  **int**[] arrA = { 12, 22, 8, 49, 3 };  **int** k = 0; **int** len = arrA.length;  **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（ ）。（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));  }  } | |
|  |  | 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 |
|  | **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 |

|  |  |  |
| --- | --- | --- |
| **8.** | **分析下面的Java程序，编译运行后的输出结果是（ ）。（选择一项）** | |
|  | **public** **class** Example {  String str = **new** String("good");  **char**[] ch = { 'a', 'b', 'c' };  **public** **static** **void** main(String args[]) {  Example ex = **new** Example( );  ex.change(ex.str, ex.ch);  System.*out*.print(ex.str + "and");  System.*out*.print(ex.ch);  }  **public** **void** change(String str, **char** ch[]) {  str = "test ok";  ch[0] = 'g';  }  } | |
|  |  |  |
|  | **A** | goodandabc |
|  | **B.** | goodandgbc |
|  | **C.** | test okandabc |
|  | **D.** | test okandgbc |

|  |  |  |
| --- | --- | --- |
| **9.** | **以下选项中能够正确创建一个数组的是（ ）。（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][]; |

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. 声明数组并分配空间后，数组的每个元素将会赋予初始值。（ ）
5. 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。（ ）
6. 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。（ ）
7. 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。（ ）
8. 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。（ ）
9. 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（ ）
10. **简答题**
11. 数组的特点。
12. 数组的优缺点
13. 冒泡排序的算法。
14. **编码题**
15. 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”。
16. 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。  
    提示：使用 int num=(int)(n\*Math.random());获取随机数
17. 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。

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： int newArr [] ={1,3,4,5,6,6,5,4,7,6,7,5}

思路： 确定出不为0的个数，这样可以开辟新数组；从旧的数组之中，取出内容，并将其赋给新开辟的数组。

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。
5. 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}}
6. **可选题**
7. 生成一百个随机数，放入数组，然后排序输出。
8. 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。