一、填空题

1.函数重载是指在函数名相同，但\_\_\_\_\_\_\_\_\_\_\_\_\_形参列表\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_不同

2.创建大小为2行4列的二维char型数组的语句为\_\_\_\_\_\_char[][] arr = new char[2][4];\_\_\_\_\_，数组创建后每个元素的值为\_\_\_\_\_‘\u0000’\_\_\_\_\_\_。

3. 创建一个大小为10的整型数组，且数组元素的值分别为1,2,3,4,5,6,7,8,9,10的语句为\_\_\_\_\_\_\_\_int[] arr = {1,2,3,4,5,6,7,8,9,10}; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

4. 用final关键字修饰一个方法形参的含义是\_\_\_\_\_\_\_\_方法内部不允许修改该参数\_\_\_\_\_\_。

5. 下列程序存在的错误是\_\_\_\_\_\_\_\_不能在函数中重新声明形参\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。public static void m(int i){

for(int i = 0 ; i < 10; i++){

System.out.println(i);

}

}

1. 单项选择题

1.下列语句的输出结果是\_\_\_\_\_C\_\_\_\_。

String[][] a = {

{"Beijing","Wuhan"},

{"Shanghai","Guangzhou","Xian"},

{"Chongqing","Chengdu"}

};

System.out.println(a[a.length - 1].length);

System.out.println(a[a.length - 1][a[a.length - 1].length - 1].length());

A．2, 5 B．3, 4 C．2, 7 D．3, 8

2. String[]s={“Monday”,”Tuesday”,”Wednesday”,”Thirsday”,”Friday”,”Sataday”,”Sunday”}；，则下列语句正确的是\_\_\_\_C\_\_\_\_\_。

A．int a = s.length, b = s[1].length;

B．int a = s.length( ), b = s[1].length;

C．int a = s.length, b = s[1].length();

D．int a = s.length( ), b = s[1].length( );

3. 若有下面程序

class C {

public static void main(String[] args) {

int[] array = new int[10];

increase(array);

System.out.print(array[0]);

}

public static void increase(int[] array) {

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

array[i]++;

}

}

}

则输出为\_\_\_\_\_B\_\_\_\_。

A.0 B.1 C.2 D.10

4. 下面的数组申明和初始化语句不合法的是\_\_\_C\_\_\_\_\_\_。

A. int a[ ] = null；

B. int[ ] b = { };

C. int[ ] c = new int{1,2,3,4};

D. int [] d[] = new int[5][ ];

三、判断对错题

1. 局部变量在使用前必须通过初始化或者赋值语句显式地给一个值。( X ) √

2. 一个方法必须要有一个return语句。( X )

3. 如果定义int[] nValues={1,2,3,4}; 那么nValues为引用类型。( √ )

4. 不能基于函数返回类型来重载函数。（√）

5. 二维数组的行数和列数是相同的。（X）

四、阅读下列程序，写出输出结果：

public class Test2 {

public static void main(String[] args){

int[] a = {1};

String[] s = {"Hello"};

int i = a[0];

m(s,a,i);

for(String v:s){

System.out.println(v);

}

for(int v:a){

System.out.println(v);

}

System.out.println(i);

}

public static void m(String[] a1, int[] a2, int i){

for(int j =0; j < a1.length;j++){

a1[j] = "Java";

}

for(int j =0; j < a2.length;j++){

a2[j]++;

}

i++;

}

}

输出结果：

Java

2

1

五、编程题

1：实现下面二个方法，并在Test3里添加入口main函数测试运行。

Tips：注意检查输入参数row的值，当输入负数，0时如何处理也考虑进来，如何处理这种情况不做要求，可以简单地打印出提示信息，或者抛出异常。但最简单的办法就是当出现这些边界条件，直接返回null引用就行了。由这个方法的调用者去处理。另外也不考虑当row的值太大导致内存溢出的情况。

public class Test3 {

/\*\*

\* 创建一个不规则二维数组

\* 第一行row列

\* 第二行row - 1列

\* ...

\* 最后一行1列

\* 数组元素值都为默认值

\* @param row 行数

\* @return 创建好的不规则数组

\*/

public static int[][] createArray(int row){

}

/\*\*

\* 逐行打印出二维数组，数组元素之间以空格分开

\* @param a

\*/

public static void printArray(int[][] a){

}

}

要求：除了提交工程文件外，请给出运行结果屏幕截图并放在Word文档里提交。

|  |
| --- |
| package Test3;  import java.util.Scanner;  public class Test3 {  */\*\*  \* 主函数  \* 输入进行测试  \** ***@param*** *args  \*/* public static void main(String[] args){  Scanner scanner = new Scanner(System.*in*);  System.*out*.println("请输入row: ");  int row = scanner.nextInt();  int[][] array = *createArray*(row);  *printArray*(array);  }  */\*\*  \* 创建一个不规则二维数组  \* 第一行row列  \* 第二行row - 1列  \* ...  \* 最后一行1列  \* 数组元素值都为默认值  \** ***@param*** *row 行数  \** ***@return*** *创建好的不规则数组  \*/* public static int[][] createArray(int row){  if(row<=0) {  System.*out*.println("row小于等于零！");  return null;  }  int[][] arr = new int[row][];  for(int i=0;i<row;i++){  arr[i] = new int[row-i];  }  return arr;  }   */\*\*  \* 逐行打印出二维数组，数组元素之间以空格分开  \** ***@param*** *a  \*/* public static void printArray(int[][] a){  if(a==null){  System.*out*.println("数组为null！");  return;  }  for(int i=0;i<a.length;i++) {  for (int j = 0; j < a[i].length; j++) {  System.*out*.print(a[i][j]+" ");  }  System.*out*.print("\n");  }  } }  结果截图：    结果截图（特殊情况）： |