

一、填空题

1. 函数重载是指在函数名相同，但_____参数的数量或类型_____不同
2. 创建大小为 2 行 4 列的二维 char 型数组的语句为__char[][]_array = new char[2][4];_____, 数组创建后每个元素的值为__0_____. '\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[]_array = {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__该方法形参为常量，不能修改_____。
5. 下列程序存在的错误是__重复定义变量 i _____。

```
public static void m(int i){  
    for(int i = 0 ; i < 10; i++){  
        System.out.println(i);  
    }  
}
```

二、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___A___。

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___A___。 **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。(1)
2. 一个方法必须要有一个 return 语句。(0)
3. 如果定义 int[] nValues={1,2,3,4}; 那么 nValues 为引用类型。(1)
4. 不能基于函数返回类型来重载函数。(1)
5. 二维数组的行数和列数是相同的。(0)

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++;
    }
}

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。

```

public class Test3 {
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Please input the row:");
        //作为循环标志
        Boolean flag = false;
        while(!flag) {
            int row = scanner.nextInt();
            if (row <= 0) {
                System.out.println("The number is wrong!");
                System.out.println("Please input the row again: ");
            }
            else {
                int[][] array1 = createArray(row);
                printArray(array1);
                flag = true;          //正确输出退出循环
            }
        }
    }

    public static int[][] createArray(int row)
    {
        int[][] array1 = new int [row][];
        int num = row;
        for(int i = 0; i < row ; i++) {
            array1[i] = new int[num];
            num--;
        }
        return array1;
    }

    public static void printArray(int[][] a) {
        for (int i = 0; i < a.length; i++) {
            for (int j = 0; j < a[i].length; j++) {
                System.out.printf("%d ",a[i][j]);
            }
            System.out.println();
        }
    }
}

```