一、填空题

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

2.创建大小为2行4列的二维char型数组的语句为 char[][] ch = 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[] intArray = {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:

保证这个形参的值在方法体运行中不被改变，防止方法参数在调用时被篡改

5. 下列程序存在的错误是:变量i在方法m的形参中已经被定义，因此for循环中不能再定义int i= 0;

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。( true )

2. 一个方法必须要有一个return语句。( false )

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

4. 不能基于函数返回类型来重载函数。（true）

5. 二维数组的行数和列数是相同的。（false）

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(String s :a1){

s = "Java";

}

for(int v :a2){

v++;

}

i++;

}

}

结果为：

Hello

1

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){

}

}

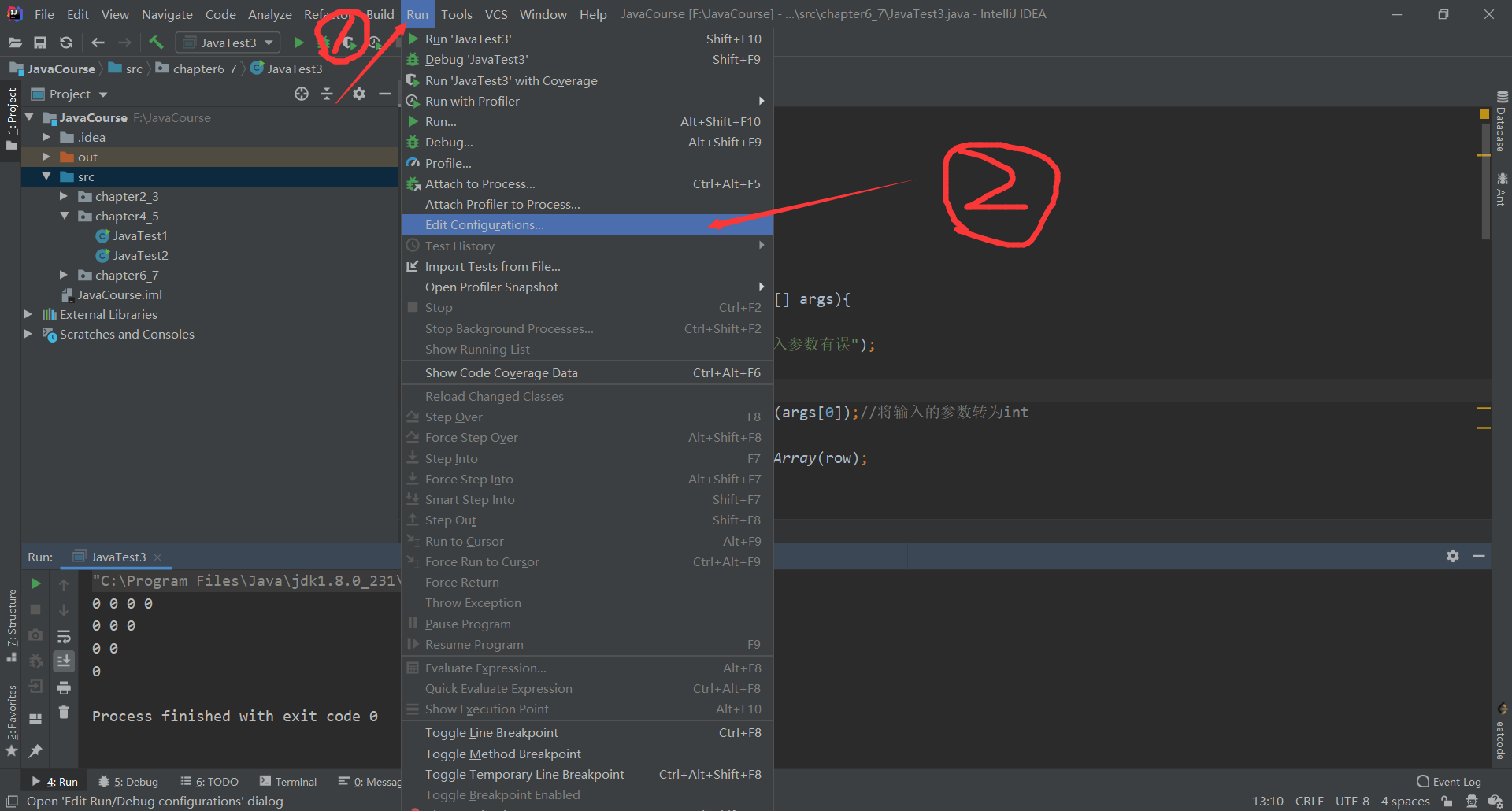
要求：除了提交工程文件外，请给出运行结果屏幕截图并放在Word文档里提交。

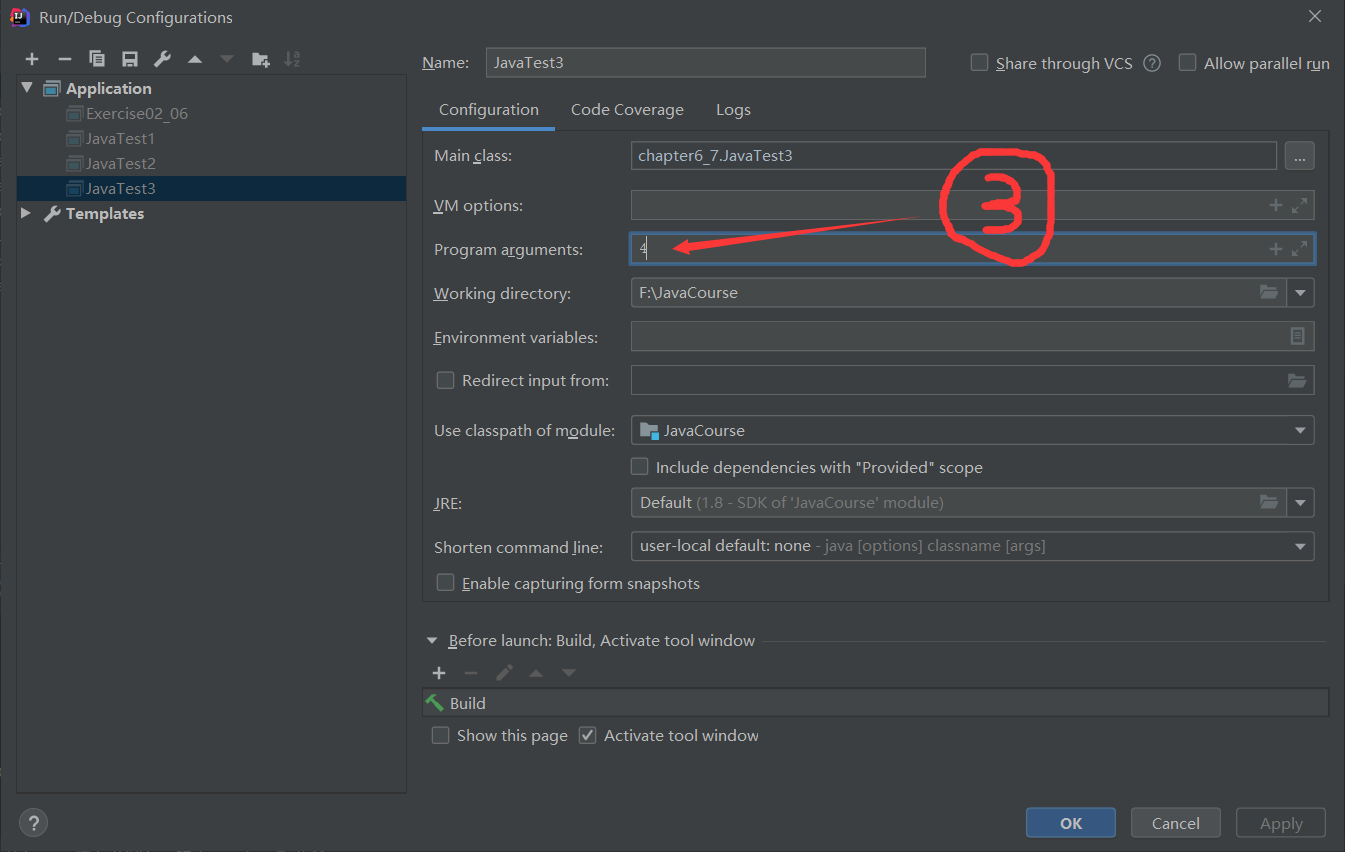
源代码：

*/\*\*  
 \** ***@author*** *WangMingMing  
 \** ***@creat*** *2020-03-11 10:21  
 \*/*public class JavaTest3 {  
 public static void main(String[] args){  
 if(args.length != 1){  
 System.*out*.println("输入参数有误");  
 System.*exit*(1);  
 }  
 int row = Integer.*parseInt*(args[0]);//将输入的参数转为int  
 try{  
 int[][] array = *createArray*(row);  
 *printArray*(array);  
 }catch (Exception ex){  
 System.*err*.println(ex.getMessage());  
 }  
 }  
 */\*\*  
 \* 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) throws Exception {  
 if(row <= 0){//抛出异常  
 throw new Exception("row的值不能为0或者负数");  
 }  
 int[][] resultArray = new int[row][];  
 for(int i = 0; i < row; i++){  
 resultArray[i] = new int[row - i];  
 }  
 return resultArray;  
 }  
  
 */\*\*  
 \* 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){  
 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*.println();  
 }  
 }  
}

运行截图解释：

1. IDEA界面中给main方法附带参数：

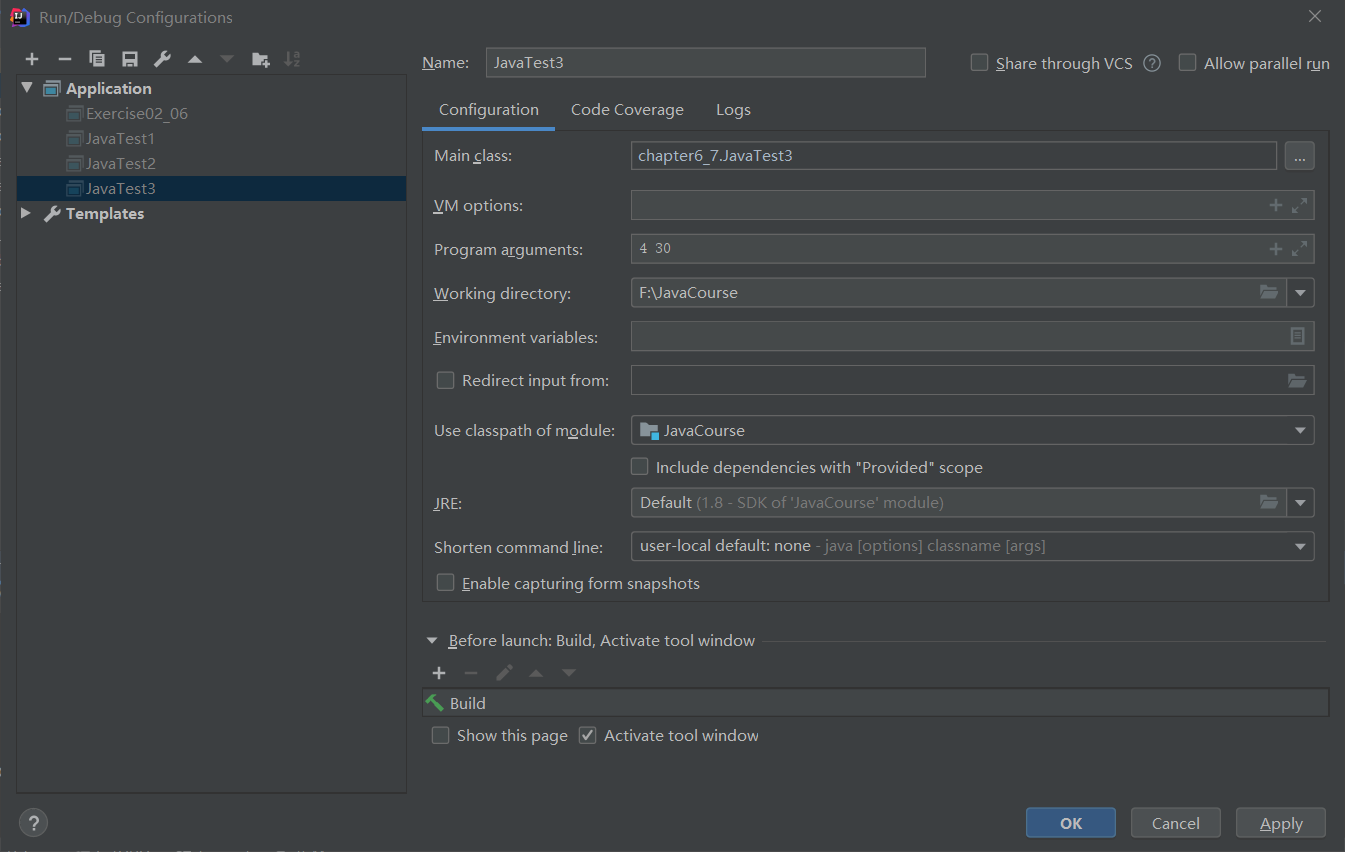


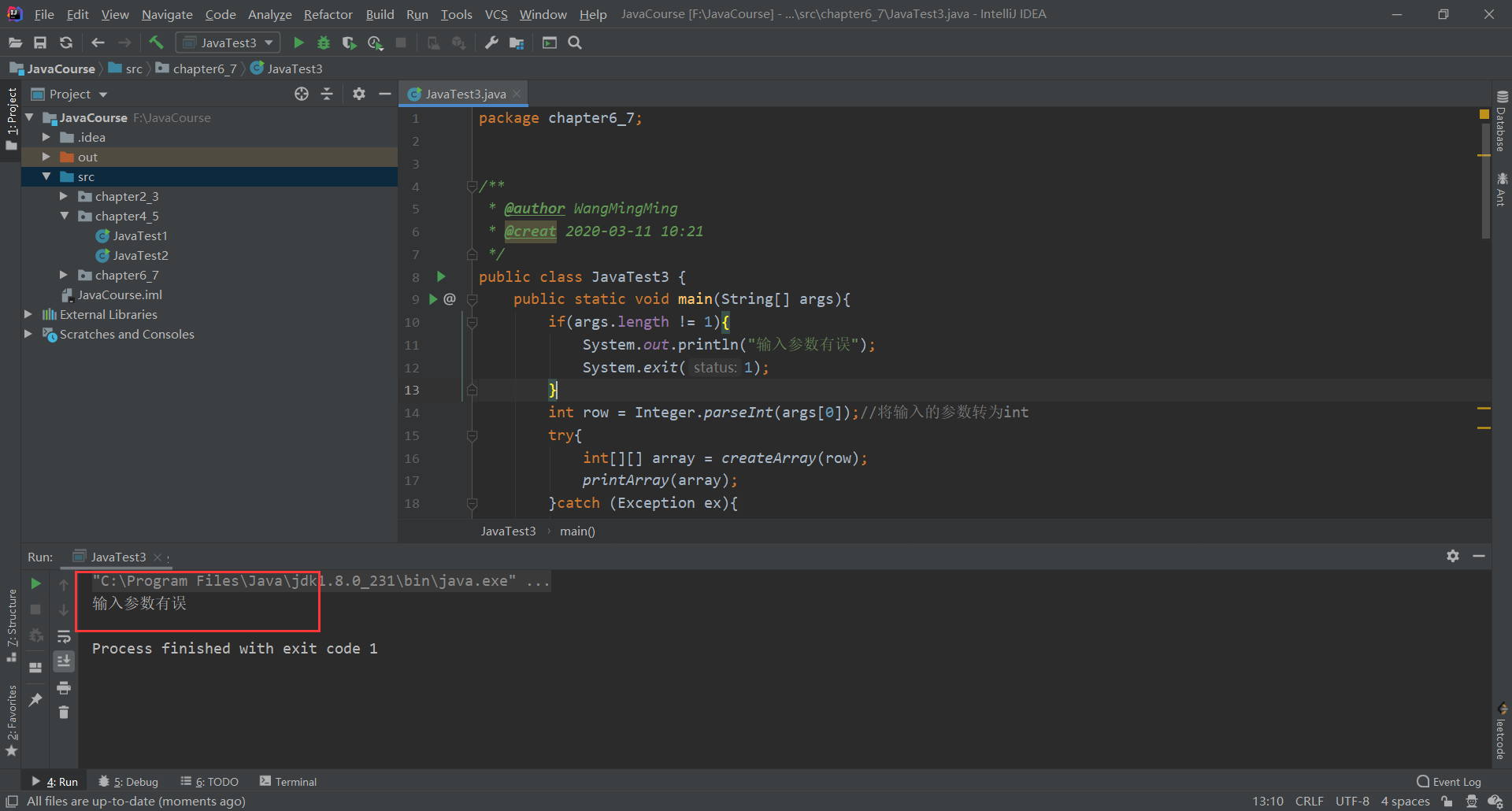


测试情况：参数的个数不是1、参数个数正确但输入的参数<=0、参数个数正确且输入参数正确。

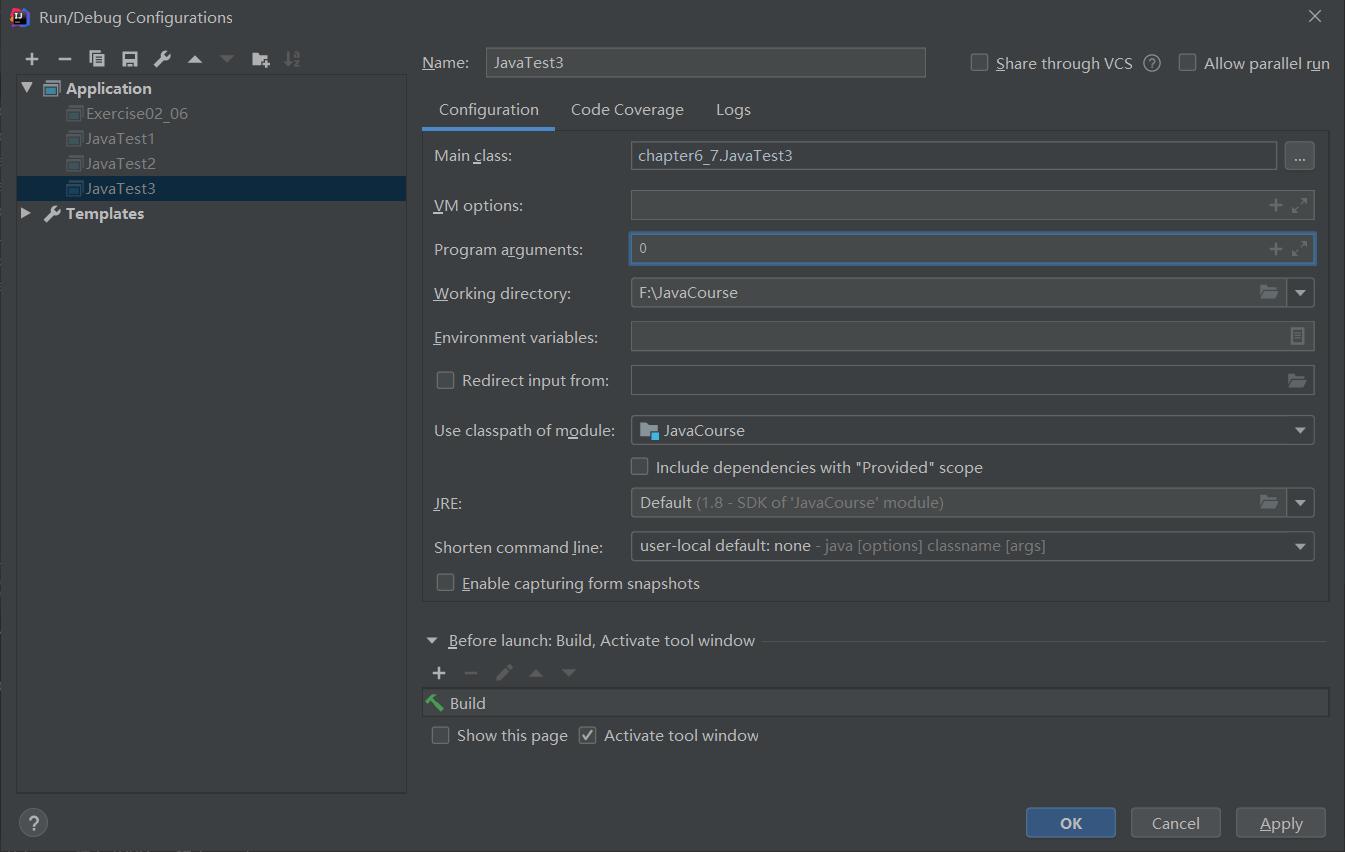
三种情况的运行截图分别如下：

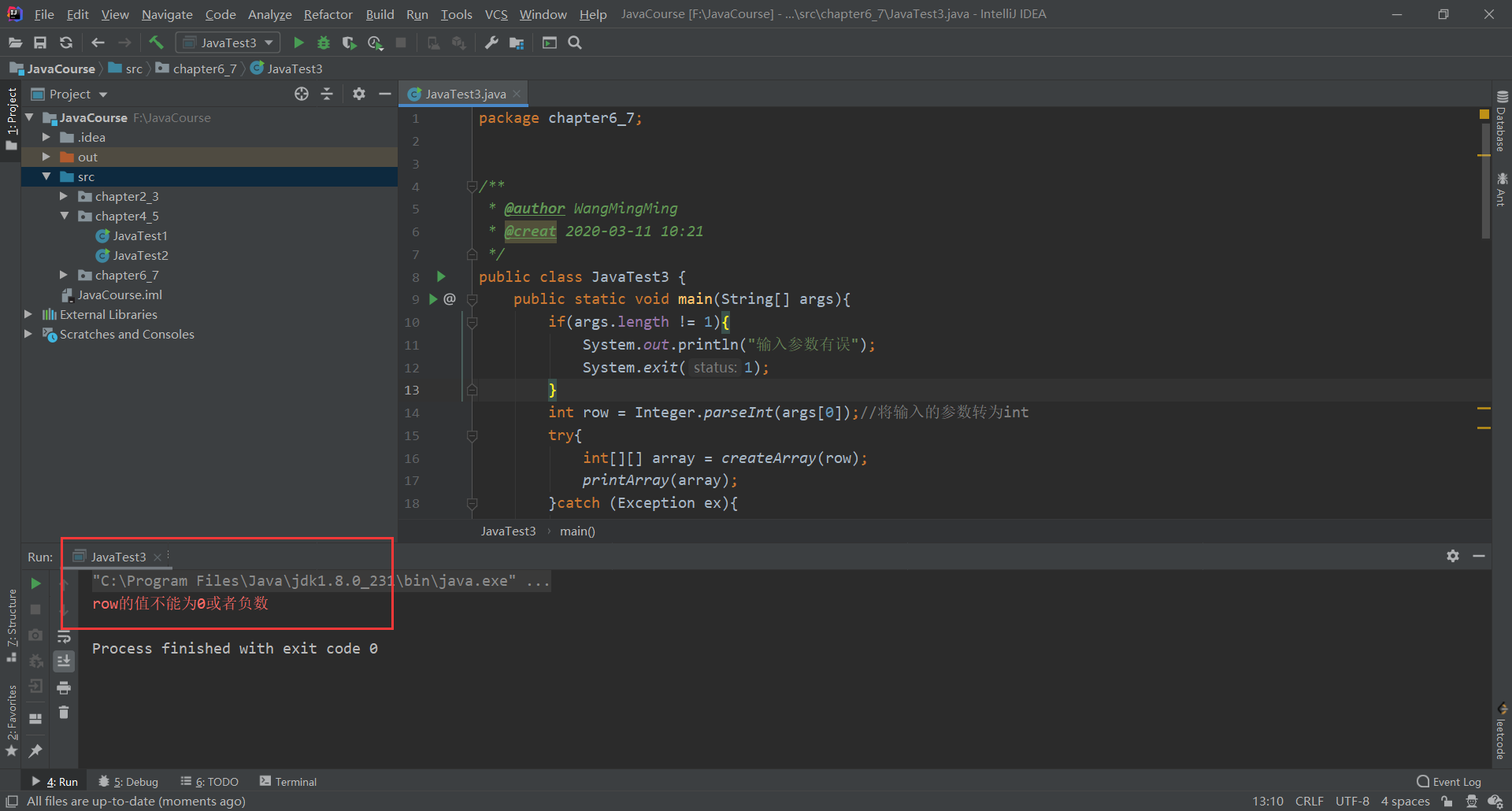
（1）参数的个数不是1





（2）参数个数正确但输入的参数<=0





（3）参数个数正确且输入参数正确

