**华东交通大学****软件+轨道信号阶段作业一**

**班级：软件+轨控2班**

**姓名：刘洋**

**成绩：**

**一、问答题（每小题10分，共100分）**

1. **System.out.println()和System.out.print()有什么区别？**

print() 输出时不换行，而 println() 输出后换行

1. **Java的注释方式有哪几种，格式为何？**

Java 的注释方式有三种，分别是：

单行注释：以 // 开头，注释内容一直到该行结束。

多行注释：以 /\* 开始，以 \*/ 结束，可以跨越多行。

文档注释：以 /\*\* 开始，以 \*/ 结束，通常用于生成 Javadoc 文档

1. **Java的基本数据类型有哪些？String是基本数据类型吗？**

byte：1 字节，表示整数，范围是 -128 到 127。

short：2 字节，表示整数，范围是 -32,768 到 32,767。

int：4 字节，表示整数，范围是 -2^31 到 2^31-1（约 -21 亿到 21 亿）。

long：8 字节，表示整数，范围是 -2^63 到 2^63-1。

float：4 字节，表示单精度浮点数。

double：8 字节，表示双精度浮点数。

char：2 字节，表示单一字符，范围是 0 到 65,535。

boolean：1 字节，表示布尔值，只有 true 和 false 两个值。

String 不是 基本数据类型，它是一个 引用数据类型。

1. **判断如下程序的运行结果**

*//如下代码的计算结果是：*

int i = 1;

i \*= 0.2;

i++;

System.out.println("i=" + i);

i=1

1. **说明break和continue使用上的相同点和不同点**

相同点：

控制循环流：两者都用于控制循环结构（for、while、do-while）的执行流程。

影响循环执行：都可以影响循环的执行，但它们影响的方式不同。

不同点：

break：用来跳出整个循环或 switch 语句，终止循环的执行，跳出循环后，程序会继续执行循环后面的代码。它会直接跳出循环体，并且不再执行循环内剩余的代码。

1. **switch后面使用的表达式可以是哪些数据类型？**

Byte short char int String（从 Java 7 开始支持） 枚举类型（enum，从 Java 5 开始支持）

包装类：例如 Byte, Short, Character, Integer

1. **不同类型的一维数组元素的默认初始化值各是多少？**

数字类型（byte、short、int、long、float、double）：

默认值：0（对于 byte、short、int 和 long 为 0，float 和 double 为 0.0）

字符类型（char）：

默认值：'\u0000'（即 Unicode 值为 0 的字符，也就是空字符）

布尔类型（boolean）：

默认值：false

引用类型（例如 String、对象等）：

默认值：null

1. **数组中常见的异常有哪些？请举例说明**

ArrayIndexOutOfBoundsException：是最常见的数组异常之一，发生在数组访问时使用了无效的索引。

原因：当你试图访问数组中不存在的索引时，抛出此异常。数组索引从 0 开始，所以对于长度为 n 的数组，合法索引范围是 0 到 n-1。

示例：

int[] arr = new int[3]; // 数组长度为3

System.out.println(arr[5]); // 报错：Index 5 out of bounds for length 3

NullPointerException：当你尝试使用 null 引用来访问数组元素时，会抛出此异常。

原因：如果数组的引用是 null，你不能使用它访问数组中的元素。

示例：

int[] arr = null;

System.out.println(arr[0]); // 报错：NullPointerException

NegativeArraySizeException：当你试图创建一个负长度的数组时，抛出此异常。

原因：数组的长度不能是负数，若尝试用负数创建数组，则会抛出此异常。

示例：

int[] arr = new int[-5]; // 报错：NegativeArraySizeException

ArrayStoreException：当你尝试将一个不兼容类型的对象存入数组时，抛出此异常。

原因：如果你有一个对象数组（例如 Object[]），你可能会尝试向其中存储一个类型不兼容的元素。

示例：

Object[] arr = new String[5];

arr[0] = new Integer(10); // 报错：ArrayStoreException

ClassCastException（与数组有关）：当你尝试将数组的元素转换成不兼容的类型时，抛出此异常。这个异常常见于使用数组时进行类型转换。

原因：即使数组类型正确，若你尝试将其元素强制转换为不兼容的类型，仍然会抛出 ClassCastException。

示例：

Object[] arr = new String[5];

arr[0] = "Hello";

Integer num = (Integer) arr[0]; // 报错：ClassCastException

ArrayIndexOutOfBoundsException ：与多维数组在多维数组中，也可能发生类似的越界访问错误。

原因：同样是索引问题，但在多维数组中，访问某个维度时，某些维度的索引可能被错误地给定。

示例：

int[][] arr = new int[3][3];

System.out.println(arr[5][1]); // 报错：ArrayIndexOutOfBoundsException

1. **异常(Exception)分为哪两大类？**

已检查异常（Checked Exception）：必须显式处理或声明，通常是外部原因引起的异常，程序员可以预见并采取措施。

未检查异常（Unchecked Exception）：运行时异常，不强制要求处理，通常由程序逻辑错误引起。

**10、throw与throws的区别？**

throw 是实际抛出异常的操作，发生在方法体内部。

throws 是方法声明的一部分，表明方法可能会抛出异常，调用者需要处理它。