

二. 十个最常见的经典 Java 基础面试、笔试题

发布时间： 2011-01-04 11:27:26 作者：张孝祥 来源：程序员网校 浏览次数：262880 我要评

论(2)

摘要： 一个".java"源文件中是否可以包括多个类（不是内部类）？有什么限制？Java 有没有 goto？说说&和&&的区别，等经典 Java 基础面试题。

1、一个".java"源文件中是否可以包括多个类（不是内部类）？有什么限制？

可以有多个类，但只能有一个 public 的类，并且 public 的类名必须与文件名相一致。

2、Java 有没有 goto？

java 中的保留字，现在没有在 java 中使用。

3、说说&和&&的区别。

&和&&都可以用作逻辑与的运算符，表示逻辑与（and），当运算符两边的表达式的结果都为 true 时，整个运算结果才为 true，否则，只要有一方为 false，则结果为 false。&&还具有短路的功能，即如果第一个表达式为 false，则不再计算第二个表达式，例如，对于 if(str != null && !str.equals(""))表达式，当 str 为 null 时，后面的表达式不会执行，所以不会出现 NullPointerException 如果将&&改为&，则会抛出 NullPointerException 异常。If(x==33 & ++y>0) y 会增长，If(x==33 && ++y>0)不会增长，&还可以用作位运算符，当&操作符两边的表达式不是 boolean 类型时，&表示按位与操作，我们通常使用 0x0f 来与一个整数进行&运算，来获取该整数的最低 4 个 bit 位，例如，0x31 & 0x0f 的结果为 0x01。

备注：这道题先说两者的共同点，再说出&&和&的特殊之处，并列举一些经典的例子来表明自己理解透彻深入、实际经验丰富。

4、在 JAVA 中如何跳出当前的多重嵌套循环？

在 Java 中，要想跳出多重循环，可以在外面的循环语句前定义一个标号，然后在里层循环体的代码中使用带有标号的 break 语句，即可跳出外层循环。例如，

```
ok: for (int i = 0; i < 10; i++) {  
    for (int j = 0; j < 10; j++) {  
        System.out.println("i=" + i + ",j=" + j);  
    }  
}
```

```

        if (j == 5)
            break ok;
    }
}

```

另外，我个人通常并不使用标号这种方式，而是让外层的循环条件表达式的结果可以受到里层循环体代码的控制，例如，要在二维数组中查找到某个数字。

```

int arr[][] = { { 1, 2, 3 }, { 4, 5, 6, 7 }, { 9 } };
boolean found = false;
for (int i = 0; i < arr.length && !found; i++) {
    for (int j = 0; j < arr[i].length; j++) {
        System.out.println("i=" + i + ",j=" + j);
        if (arr[i][j] == 5)
        {
            found = true;
            break;
        }
    }
}

```

5、switch 语句能否作用在 byte 上，能否作用在 long 上，能否作用在 String 上？

在 switch (expr1) 中，expr1 只能是一个整数表达式或者枚举常量（更大字体），整数表达式可以是 int 基本类型或 Integer 包装类型，由于，byte,short,char 都可以隐含转换为 int，所以，这些类型以及这些类型的包装类型也是可以的。显然，long 和 String 类型都不符合 switch 的语法规定，并且不能被隐式转换成 int 类型，所以，它们不能作用于 switch 语句中。

6、short s1 = 1; s1 = s1 + 1;有什么错？ short s1 = 1; s1 += 1;有什么错？

对于 short s1 = 1; s1 = s1 + 1; 由于 s1+1 运算时会自动提升表达式的类型，所以结果是 int 型，再赋值给 short 类型 s1 时，编译器将报告需要强制转换类型的错误。

对于 `short s1 = 1; s1 += 1;` 由于 `+=` 是 `java` 语言规定的运算符，`java` 编译器会对它进行特殊处理，因此可以正确编译。

7、char 型变量中能不能存贮一个中文汉字?为什么?

`char` 型变量是用来存储 `Unicode` 编码的字符的，`unicode` 编码字符集中包含了汉字，所以，`char` 型变量中当然可以存储汉字啦。不过，如果某个特殊的汉字没有被包含在 `unicode` 编码字符集中，那么，这个 `char` 型变量中就不能存储这个特殊汉字。补充说明：`unicode` 编码占用两个字节，所以，`char` 类型的变量也是占用两个字节。

备注：后面一部分回答虽然不是在正面回答题目，但是，为了展现自己的学识和表现自己对问题理解的透彻深入，可以回答一些相关的知识，做到知无不言，言无不尽。

8、用最有效率的方法算出 2 乘以 8 等於几?

`2 << 3`，因为将一个数左移 `n` 位，就相当于乘以了 2 的 `n` 次方，那么，一个数乘以 8 只要将其左移 3 位即可，而位运算 `cpu` 直接支持的，效率最高，所以，2 乘以 8 等於几的最效率的方法是 `2 << 3`。

9、请设计一个一百亿的计算器

首先要明白这道题目的考查点是什么，一是大家首先要对计算机原理的底层细节要清楚、要知道加减法的位运算原理和知道计算机中的算术运算会发生越界的情况，二是要具备一定的面向对象的设计思想。

首先，计算机中用固定数量的几个字节来存储的数值，所以计算机中能够表示的数值是有一定的范围的，为了便于讲解和理解，我们先以 `byte` 类型的整数为例，它用 1 个字节进行存储，表示的最大数值范围为 -128 到 +127。-1 在内存中对应的二进制数据为 `11111111`，如果两个 -1 相加，不考虑 `Java` 运算时的类型提升，运算后会产生进位，二进制结果为 `1,11111110`，由于进位后超过了 `byte` 类型的存储空间，所以进位部分被舍弃，即最终的结果为 `11111110`，也就是 -2，这正好利用溢位的方式实现了负数的运算。-128 在内存中对应的二进制数据为 `10000000`，如果两个 -128 相加，不考虑 `Java` 运算时的类型提升，运算后会产生进位，二进制结果为 `1,00000000`，由于进位后超过了 `byte` 类型的存储空间，所以进位部分被舍弃，即最终的结果为 `00000000`，也就是 0，这样的结果显然不是我们期望的，这说明**计算机中的算术运算是会发生越界情况的，两个数值的运算结果不能超过计算机中的该类型的数值范围**。由于 `Java` 中涉及表达式运算时的类型自动提升，我们无法用 `byte` 类型来做演示这种问题和现象的实验，大家可以用下面一个使用整数做实验的例子程序体验一下：

```
int a = Integer.MAX_VALUE;
```

```
int b = Integer.MAX_VALUE;

int sum = a + b;

System.out.println("a=" + a + ",b=" + b + ",sum=" + sum);
```

先不考虑 `long` 类型，由于 `int` 的正数范围为 2 的 31 次方，表示的最大数值约等于 2^{31} ，也就是 20 亿的大小，所以，要实现一个一百亿的计算器，我们得自己设计一个类可以用于表示很大的整数，并且提供了与另外一个整数进行加减乘除的功能，大概功能如下：

1. 这个类内部有两个成员变量，一个表示符号，另一个用字节数组表示数值的二进制数
2. 有一个构造方法，把一个包含有多位数值的字符串转换到内部的符号和字节数组中
3. 提供加减乘除的功能

```
public class BigInteger {
    int sign;
    byte[] val;

    public BigInteger(String val) {
        sign = ;
        val = ;
    }

    public BigInteger add(BigInteger other) {
    }

    public BigInteger subtract(BigInteger other) {
    }

    public BigInteger multiply(BigInteger other) {
    }

    public BigInteger divide(BigInteger other) {
```

```
}  
}
```

备注：要想写出这个类的完整代码，是非常复杂的，如果有兴趣的话，可以参看 jdk 中自带的 `java.math.BigInteger` 类的源码。面试的人也知道谁都不可能在短时间内写出这个类的完整代码的，他要的是你是否有这方面的概念和意识，他最重要的还是考查你的能力，所以，你不要因为自己无法写出完整的结果就放弃答这道题，你要做的就是你比别人写得多，证明你比别人强，你有这方面的思想意识就可以了，毕竟别人可能连题目的意思都看不懂，什么都没写，你要敢于答这道题，即使只答了一部分，那也与那些什么都不懂的人区别出来，拉开了距离，算是矮子中的高个，机会当然就属于你了。另外，答案中的框架代码也很重要，体现了一些面向对象设计的功底，特别是其中的方法命名很专业，用的英文单词很精准，这也是能力、经验、专业性、英语水平等多个方面的体现，会给人留下很好的印象，在编程能力和其他方面条件差不多的情况下，英语好除了可以使你获得更多机会外，薪水可以高出一千元。

10、使用 **final** 关键字修饰一个变量时，是引用不能变，还是引用的对象不能变？

使用 **final** 关键字修饰一个变量时，是指引用变量不能变，引用变量所指向的对象中的内容还是可以改变的。例如，对于如下语句：

```
final StringBuffer a=new StringBuffer("immutable");
```

执行如下语句将报告编译期错误：

```
a=new StringBuffer("");
```

但是，执行如下语句则可以通过编译：

```
a.append(" broken!");
```

有人在定义方法的参数时，可能想采用如下形式来阻止方法内部修改传进来的参数对象：

```
public void method(final StringBuffer param) {}
```

实际上，这是办不到的，在该方法内部仍然可以增加如下代码来修改参数对象：

```
param.append("a");
```

