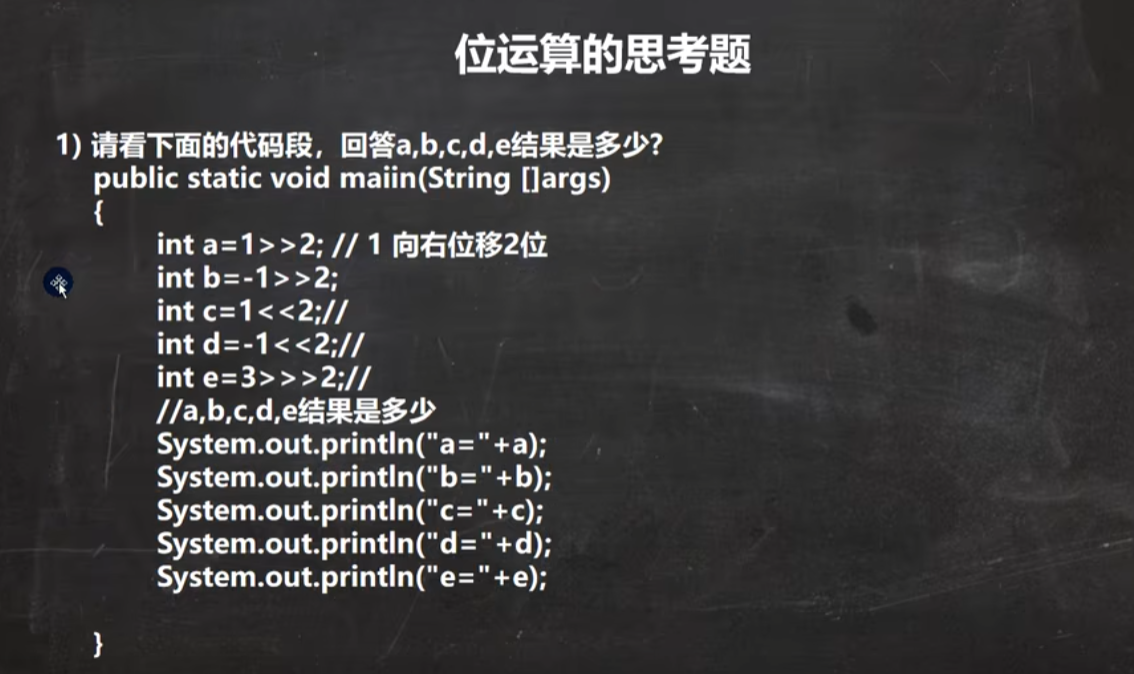
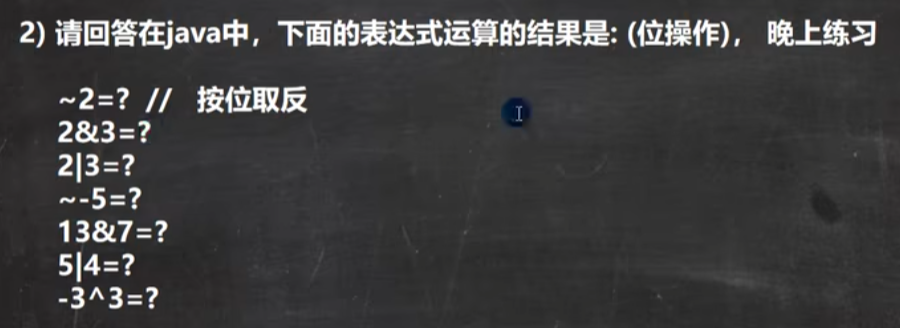
Java7.14 位运算题目思考 原码 反码 补码

（1）位运算的思考题目：







（2）二进制在运算中的说明：

1. 二进制是逢2进位的进位制 0和1是基本算符

2. 现代的电子计算机技术全部采用的是二进制 因为它只使用0，1两个数字符号 非常简单方便 易于用电子方式实现

计算机内部处理的信息都是采用二进制数来表示的 二进制(Binary)数用0和1两个数字及其组合来表示任何数 进位规则是逢2进1 数字1在不同的位上代表不同的值 按从右至左的次序 这个值以二倍递增

（3）原码 反码 补码----重点

网上对于源码 反码 补码的解释过于复杂 这里精简几句话下来：（记住背下来）

对于有符号的而言：

1. 二进制的最高位是符号位：0表示正数 1表示负数（怎么记？🡪 0 旋转还是0；

1旋转一下变成了—（-） 就是负号负数）

2. 正数的原码 反码 补码都一样（三码合一）

3. 负数的反码=它的原码符号位不变 其他位取反（0->1 1->0）

4. 负数的补码=它的反码+1 负数的反码=它的补码-1

5. 0的反码和补码都是0

6. java没有无符号数，换言之，java中的数都是有符号的

7. 在计算机运算的时候 都是以补码的方式来运算的

8. 当我们看运算结果的时候，要看它的原码（重点）

（它在展示结果的时候 它是以原码的方式来展示的）

比如求2&3（更多举例可以看7.14的位运算word）

public class BitOperatorExercise1 {  
 public static void main(String[]args){  
 //详细解析位运算符  
 //解析： 因为在计算机运行的时候 都是以补码的方式来运行的 所以2&3 先求出2和3的补码  
 //正数的三码合一（原码补码反码相同 可以看7.14word里面有写）  
 //第一步 求出2和3的补码 （int类型四个字节） 正数的符号位是0 负数的符号位是1  
 //2的补码 00000000 00000000 00000000 00000010  
 //3的补码 00000000 00000000 00000000 00000011  
 //第二步 求2&3的补码 按位与（两者都为一 结果为1 否则为0）  
 //2的补码 00000000 00000000 00000000 00000010  
 //3的补码 00000000 00000000 00000000 00000011  
 //2&3的补码 00000000 00000000 00000000 00000010  
 //第三步 2&3的补码已得到 但是在计算机中是以原码为展示结果的（原码反码补码知识见7.14word）  
 //2&3的原码 00000000 00000000 00000000 00000010  
 //所以输出结果为2  
 System.*out*.println(2&3);//输出为2  
 }  
}