

对03的回答

1. 基本数据类型

- 整型: byte, short, int, long
- 浮点型: float, double
- 字符型: char
- 布尔型: boolean

2. 整型的大小与范围

- byte: 占用1字节, 范围为 -2^8 到 2^8-1
- short: 占用2字节, 范围为 -2^{16} 到 $2^{16}-1$
- int: 占用4字节, 范围为 -2^{32} 到 $2^{32}-1$
- long: 占用8字节, 范围为 -2^{64} 到 $2^{64}-1$

3. 这是自动类型转换, b的值为52

多说两句, 较小范围的类型向较大范围类型的转换以及字符与字符串的转换是自动类型转换, 由编译器自动进行

而较大范围类型向较小范围类型的转换, 由于会对数据造成精度上的影响而需要程序员进行主动在代码中书写, 称为强制类型转换

另外, 0在ASCII码中对应的10进制数为48, 所以即为 $48+4=52$

4. 输出结果分别是false, true, false

- 第一组创建了两个新的对象, 它们的值相等, 但实际的内存地址是不同的, 所以不等
- 第二组直接调用了缓存池中的数据, 二者指向同一个地址, 相同
- 第三组超过了缓存池默认的数据缓存范围, 即-128到127, 无法调用, 分别生成两个不同的对象, 不同

```
5.  int a = 5 ;
    int b = 7 ;
    int c= (++a) + (b++)
    System.out.println( c );
    System.out.println(a+ " "+b);
```

这个执行出来应该是

```
13
6 8
```

$++a$ 是先自增再调用, $b++$ 是先调用再自增, 所以 $c=6+7=13$, 输出时 a 与 b 都完成了自增, 所以分别是6和8。

6. $a \& (-a) = 0010$, $a = 0010$, 反码为1101, 补码为1110, $\&$ 是按位与的意思, 只有 a 与 $-a$ 的对应位都为1, 才为1, 否则为0

比较0010与1110得到结果0010

此外 a 与 $-a$ 按位与得到的值一定为一个除 a 最低位1所对应位为1, 其余全为0的数, 理由如下

一个任意的正整数，.....1...00，如果最右侧1为最低位的1，右侧省略号以0填充，左侧省略号随机填入，取反码后为...0....11，取补码为...1...00，发现这个1左侧的每一位由于取反码导致都不一样，1右侧则全部为0，根据按位与规则即得到结果。