自动装箱和拆箱有点关系的，代码如下所示：

public class Test03 {

    public static void main(String[] args) {

        Integer f1 = 100, f2 = 100, f3 = 150, f4 = 150;

        System.out.println(f1 == f2);//true

        System.out.println(f3 == f4);//false

    }

}

如果不明就里很容易认为两个输出要么都是true要么都是false。首先需要注意的是f1、f2、f3、f4四个变量都是Integer对象引用，所以下面的==运算比较的不是值而是引用。装箱的本质是什么呢？当我们给一个Integer对象赋一个int值的时候，会调用Integer类的静态方法valueOf，如果看看valueOf的源代码就知道发生了什么。

public static Integer valueOf(int i) {

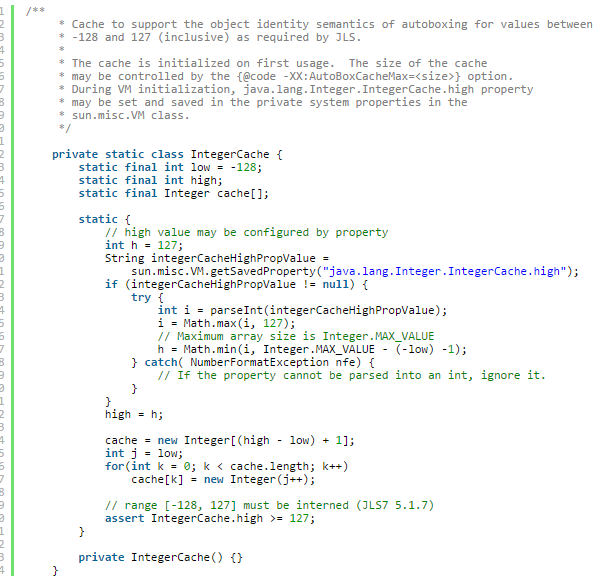
    if (i >= IntegerCache.low && i <= IntegerCache.high)

        return IntegerCache.cache[i + (-IntegerCache.low)];

    return new Integer(i);

}

IntegerCache是Integer的内部类，其代码如下所示：



简单的说，如果整型字面量的值在-128到127之间，那么不会new新的Integer对象，而是直接引用常量池中的Integer对象，所以上面的面试题中f1==f2的结果是true，而f3==f4的结果是false。