1. 实验目的

验证短路求值及带有副作用的表达式对程序执行的影响。

二、实验环境

操作系统： Windows 10

IDE：IntelliJ IDEA 2017.1.3

JDK：jdk1.8.0\_131

三、源程序

<Test.java>

public class Test {  
 public static void main(String[] args){  
 //带有副作用的表达式  
 int a = 2, b = 3;  
 if(a > b || (b++)/3 != 0)//注：由于在JAVA 中 (b++)/3 不能如  
 System.*out*.println(b);// C++ 中一样隐式转换为 boolean 类型

// 故使用 (b++)/3 != 0 进行实验  
 //短路求值  
 a = 3;  
 b = 2;  
 if(a > b || (b++)/3 != 0)  
 System.*out*.println(b);  
 }  
}

四、运行结果及说明

程序执行结果如下：



首先看程序的上半部分：

通过变量声明及初始化得到a = 2, b = 3 。在if(…)的条件判断中 a > b 不成立，因此程序继续判断之后的 (b++) / 3 != 0 部分。在这一部分中 (b++)/3 等于1所以 ‘||’ 号的右半部分成立。并且由 b++ 得到 b的值为 4 。因此程序可以执行 if 语句之后语句块中的内容，即打印 b 的值。由于此时 b 为 4，最终打印 4 。

再看程序的下半部分：

变量 a 和 b 分别赋值为 3 和 2 。在 if(…)的条件判断中，a > b 明显成立。而由于 JAVA 对 ‘||’ 表达式的求值时按照短路求值规则，因此 ‘||’右边的部分不再执行。即 b 的值没有发生改变。由于 ‘||’ 的左半部分成立，因此程序仍然可以执行 if 语句之后语句块中的内容，即打印 b 的值。由于 b 没有发生改变，所以最终打印结果为 2。

可以看出，同一条if(a > b || (b++)/3 != 0) 语句在决定程序行为时，其结果可能由于 a 和 b 的值的不同而带来一些程序设计时可能没有预料到的副作用。因此，我们在程序设计中应尽可能避免出现这样带有副作用的执行结果不确定的语句。