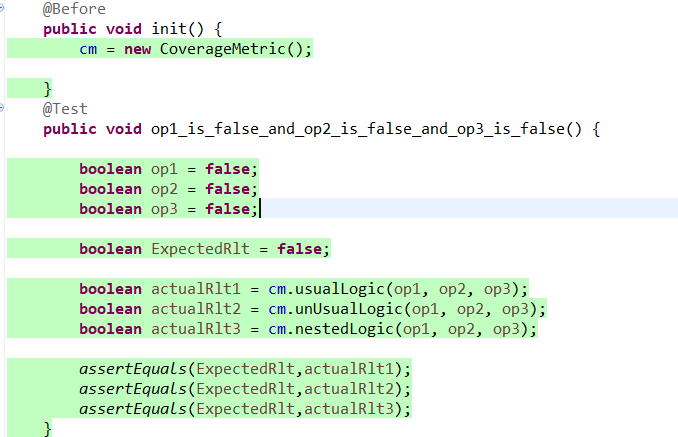


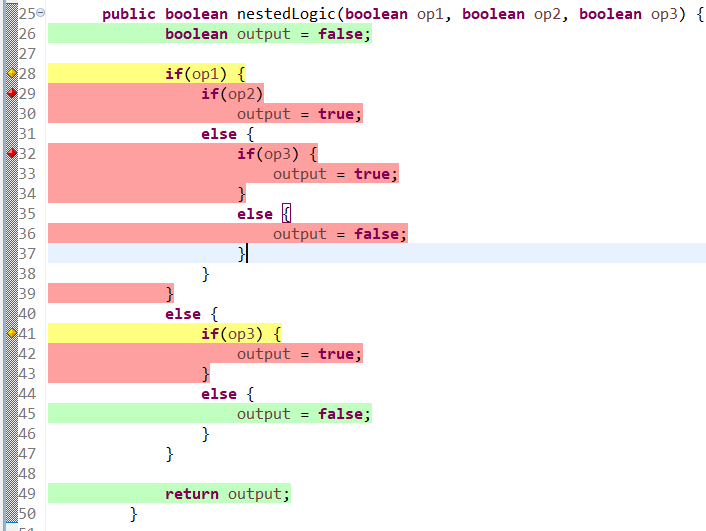
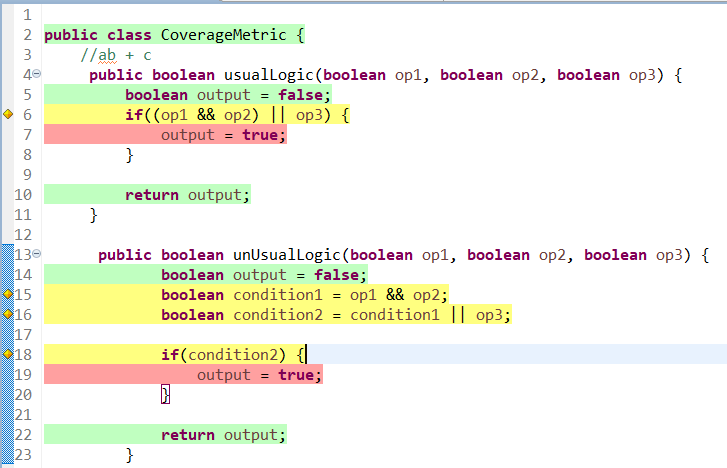
Instruction coverage 指令覆盖

Line coverage: 代码行覆盖用于衡量被执行（覆盖）的程序语句或者代码行；

Branch coverage: 分支覆盖用于衡量被执行的每个判定中的每个分支；

1. 这道题的测试案例只使用了 a=false, b=false, c=false测试代码如下图





usualLogic 和 unUsualLogic

在Instruction coverage上有差别，因为指令覆盖用于衡量被执行指令，这两个函数指令字节不同

在Line coverage: 上有没有差别，因为代码行覆盖用于衡量被执行（覆盖）的程序语句或者代码行，除了output = **true**;这一行是可能不被执行，其他行都一定会被执行，而且output = **true**;这一行代表最终的结果，如果usualLogic执行，unUsualLogic也一定会执行，反之亦然

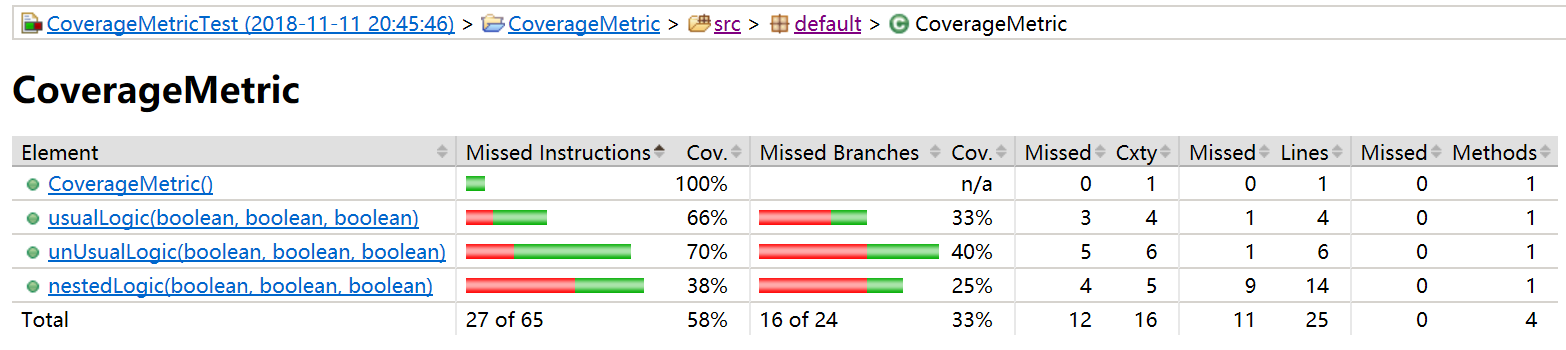
在Branch coverage上有差别，因为分支覆盖用于衡量被执行每个判定中的每个分支，这两个函数所拥有的分支数量以及分支类型就不相同。

usualLogic运行时候**if**((op1 && op2) || op3)有6个分支

unUsualLogic运行时候**boolean** condition1 = op1 && op2;拥有4个分支**boolean** condition2 = condition1 || op3;也拥有4个分支 **if**(condition2)拥有2个分支

所以分支覆盖的结果自然不同

覆盖结果如下图



1. UsualLogic 和 nestedLogic

在Instruction coverage上有差别，因为这两个函数的指令不同

在Line coverage上也有差别，对usualLogic函数来说，**if**((op1 && op2) || op3)执行时，如果op1=false,那么该条件下的行都不会被执行， 对于nestedLogic来说不仅要判断op1 还要判断op3

在Branch coverage上有差别，这两个函数的分支不同

1. 求UsualLogic 满足 MC/DC 的测试案例

解Ba 得到a=true, b=true, c=false; a=false, b=true, c=false;

解Bb 得到a=true, b=true, c=false; a=true, b=false, c=false;

解Bc 得到三例测试用例：

a=true, b=false, c=true; a=true, b=false, c=false;

a=false, b=true, c=true; a=false, b=true, c=false;

a=false, b=false, c=true; a=false, b=false, c=false;

最终得到满足usualLogic MC/DC的测试案例

a=true, b=true ,c=false; a=false, b=true, c=false;

a=true, b=false, c=false; a=false, b=true, c=true;

满足nestedLogic branch coverage的一组测试案例为：

a=true, b=true, c=false;

a=true, b=false, c=false;

a=true, b=false, c=true;

a=false, b=false, c=true;

a=false, b=false, c=false

由此可见满足nestedLogic的branch coverage 并不满足usualLogic的MC/CD (没a=false, b=true, c=false;这一测试案例)

满足usualLogic的MC/DC不满足nestedLogic的branch coverage：（下图为使用usualLogicMC/DC测试案例的覆盖结果）

