

## 代码覆盖测试 单元测试

产品名称	代码覆盖测试 单元测试
公司名称	广州思谋信息科技有限公司
价格	.00/个
规格参数	
公司地址	广州市白云区人和镇华成西街118号
联系电话	020-29178595 13342852518

## 产品详情

在做单元测试时，代码覆盖率常常被拿来作为衡量测试好坏的指标，甚至，用代码覆盖率来考核测试任务完成情况，比如，代码覆盖率必须达到80%或90%。于是乎，测试人员费尽心思设计案例覆盖代码。用代码覆盖率来衡量，有利也有有弊。代码覆盖是由系统化软件测试所衍生的方式。第一份出版的相关参考资料是Miller及Maloney1963年在ACM通讯上发表的论文。覆盖测试是衡量测试质量的一个重要指标。在对一个软件产品进行了单元测试、组装测试、集成测试以及接口测试等繁多的测试之后，我

们能不能就此对软件的质量产生一定的信心呢？这就需要我们对测试的质量进行考察。如果测试仅覆盖了代码的一小部分，那么不管我们写了多少测试用例，我们也不能相信软件质量是有保证的。相反，如果测试覆盖到了软件的绝大部分代码，我们就能对软件的质量有一个合理的信心。度量方式

### 函数覆盖

函数覆盖 (Function Coverage)，有执行到程式中的每一个函数 (或副程式) 吗。语句

### 覆盖

语句覆盖 (Statement Coverage)，又称行覆盖 (Line Coverage)，段覆盖 (Segment Coverage)，基本块覆盖 (Basic Block Coverage)，这是最常用也是最常见的一种覆盖方式，就是度量被测代码中每个可执行语句是否被执行到了。这里说的是“可执行语句”，因此就不会包括像C++的头文件声明，代码注释，空行，等等。非常好理解，只统计能够执行的代码被执行了多少行。需要注意的是，单独一行的花括号 {} 也常常被统计进去。语句覆盖

常常被人指责为“最弱的覆盖”，它只管覆盖代码中的执行语句，却不考虑各种分支的组合等等。假如你的上司只要求你达到语句覆盖，那么你可以省下很多功夫，但是，换来的确实测试效果的不明显，很难更多地发现代码中的问题。判

### 断覆盖

判断覆盖(Decision Coverage)，又称分支覆盖(Branch Coverage)，所有边界覆盖(All-Edges Coverage)，基本路径覆盖(Basic Path Coverage)，判定路径覆盖(Decision-Decision-Path)。它度量程序中每一个判定的分支是否都被测试到了。这句话是需要进一步理解的，应该非常容易和下面说到的条件覆盖混淆。因此我们直接介绍第三种覆盖方式，然后和判定覆盖一起来对比，就明白两者是怎么回事了。条件覆盖

### 条件覆盖

条件覆盖(Condition Coverage)，它度量判定中的每个子表达式结果true和false是否被测试

### 到了。路径覆盖

路径覆盖(Path Coverage)，又称断言覆盖(Predicate Coverage)

e)。它度量了是否函数的每一个分支都被执行了。这句话也非常好理解，就是所有可能的分支都执行一遍，有多个分支嵌套时，需要对多个分支进行排列组合，可想而知，测试路径随着分支的数量指数级别增加。总结编辑通过上面的学习，我们再回头想想，覆盖率数据到底有多大意义。总结如下几个观点：覆盖率数据只能代表你测试过哪些代码，不能代表你是否测试好这些代码。(比如上面第一个除零Bug)不要过于相信覆盖率数据。不要只拿语句覆盖率(行覆盖率)来考核你的测试人员。路径覆盖率 > 判定覆盖 > 语句覆盖测试人员不能盲目追求代码覆盖率，而应该想办法设计更多更好的案例，哪怕多设计出来的案例对覆盖率一点影响也没有。

**服务热线：400-669-0203**  
020-29178595 QQ2557064750  
2649046091