**白盒测试的基本路径测试法**

**一、白盒测试的主要测试方法**

1、代码检查法2、静态结构分析法3、静态质量度量法4、逻辑覆盖法5、基本路径测试法（应用最广泛）6、域测试7、符号测试8、Z路径覆盖9、程序变异

**二、基本路径测试法**

1、定义：基本路径测试法是在程序控制流图的基础上，通过分析控制构造的环路复杂性，导出基本可执行路径集合，从而设计测试用例的方法。

设计出的测试用例要保证在测试中程序的每个可执行语句至少执行一次。

2、基本路径测试法的基本步骤

1) 程序的控制流图：描述程序控制流的一种图示方法。

2） 程序圈复杂度：McCabe复杂性度量。从程序的环路复杂性可导出程序基本路径集合中的独立路径条数，这是确定程序中每个可执行语句至少执行一次所必须的测试用例数目的上界。
 3） 导出测试用例：根据圈复杂度和程序结构设计用例数据输入和预期结果。

4） 准备测试用例：确保基本路径集中的每一条路径的执行。

3、基本路径测试法的工具方法

1） 图形矩阵：是在基本路径测试中起辅助作用的软件工具，利用它可以实现自动地确定一个基本路径集。

**三、程序的控制流图**

控制流程图是描述程序控制流的一种图示方法。圆圈称为控制流图的一个结点，表示一个或多个无分支的语句或源程序语句流图只有二种图形符号：图中的每一个圆称为流图的结点，代表一条或多条语句。流图中的箭头称为边或连接，代表控制流。任何过程设计都要被翻译成控制流图。



**1、根据程序流程图化成控制流图**

在将程序流程图简化成控制流图时，应注意：

1）在选择或多分支结构中，分支的汇聚处应有一个汇聚结点。

2） 边和结点圈定的区域叫做区域，当对区域计数时，图形外的区域也应记为一个区域。

如下页图所示：



如果判断中的条件表达式是由一个或多个逻辑运算符 (OR, AND, NAND, NOR) 连接的复

合条件表达式，则需要改为一系列只有单条件的嵌套的判断。
例如：
　1 if a or b
　2 x
　3 else
　4 y

对应的逻辑为：



独立路径：至少沿一条新的边移动的路径



**2、实例说明基本路径测试法的步骤：**

**第一步：画出控制流图**

流程图用来描述程序控制结构。可将流程图映射到一个相应的流图(假设流程图的菱形决定框中不包含复合条件)。在流图中，每一个圆，称为流图的结点，代表一个或多个语句。一个处理方框序列和一个菱形决测框可被映射为一个结点，流图中的箭头，称为边或连接，代表控制流，类似于流程图中的箭头。一条边必须终止于 一个结点，即使该结点并不代表任何语句(例如：if-else-then结构)。由边和结点限定的范围称为区域。计算区域时应包括图外部的范围。

例如：用如下基本路径测试法进行测试

Void Sort(int iRecordNum, int iType)

{ 1

int x=0; 2

Int y=0; 3

while ( iRecordNum-->0) 4

{ 5

 If(0==iType) 6

{x=y+2;break;} 7

else 8

if(1==iType) 9

x=y+10; 10

else 11

x=y+20; 12

} 13

} 14

画出其程序流程图和对应的控制流图如下：



**第二步：计算圈复杂度**

圈复杂度是一种为程序逻辑复杂性提供定量测度的软件度量，将该度量用于计算程序的基本的独立路径数目，为确保所有语句至少执行一次的测试数量的上界。独立路径必须包含一条在定义之前不曾用到的边。

有以下三种方法计算圈复杂度：
　A 流图中区域的数量对应于环型的复杂性;
　B给定流图G的圈复杂度V(G)，定义为V(G)=E-N+2，E是流图中边的数量，N是流图

中结点的数量;
　C 给定流图G的圈复杂度V(G)，定义为V(G)=P+1，P是流图G中判定结点的数量。

**第三步：导出测试用例**

根据上面的计算方法，可得出四个独立的路径。(一条独立路径是指，和其他的独立路径相比，至少引入一个新处理语句或一个新判断的程序通路。V(G)值正好等于该程序的独立路径的条数。

路径1：4-14

路径2：4-6-7-14

路径3：4-6-8-10-13-4-14

路径4：4-6-8-11-13-4-14

根据上面的独立路径，去设计输入数据，使程序分别执行到上面四条路径。

**第四步：准备测试用例**

为了确保基本路径集中的每一条路径的执行，根据判断结点给出的条件，选择适当的数据以保证某一条路径可以被测试到，满足上面例子基本路径集的测试用例是：

路径1: 4-14

输入数据：iRecordNum=0，或者取iRecordNum<0的某一个值

预期结果：x=0

路径2: 4-6-7-14

输入数据：iRecordNum=1，iType=0

预期结果：x=2

路径3 : 4-6-8-10-13-4-14

输入数据：iRecordNum=1，iType=1

预期结果：x=10

路径2: 4-6-8-11-13-4-14

输入数据：iRecordNum=1，iType=2

预期结果：x=20