**白盒测试：**  
针对代码逻辑进行的测试，一般和黑盒测试配合使用。主要应用于单元测试阶段。

**特点：**

优点：代码覆盖率高

缺点：覆盖所有代码路径难度大

业务功能可能覆盖不全

测试开销大

**百盒测试设计方法（往往采用：先静态、后动态）：**

**静态**（测试过程中不执行代码）：桌面检查（交叉检测）、代码审查（会议检查，作者讲解）、代码走查（会议，与会人准备测试样例，查看测试走向）、代码扫描工具（代码规范性）

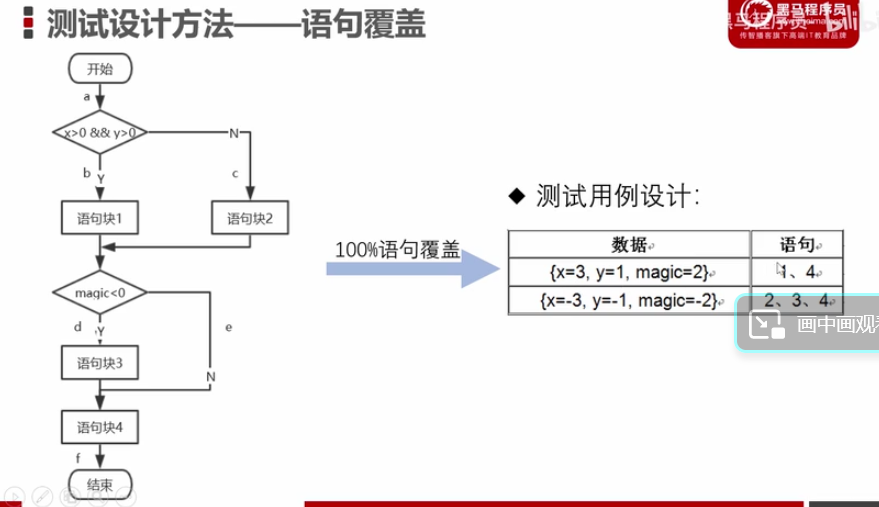
**动态**（测试过程中执行代码）：**逻辑覆盖法、基本路径覆盖法**。

逻辑覆盖法：语句覆盖、判定覆盖、条件覆盖、判定条件覆盖、条件组合覆盖、路径覆盖。

注：设计用例一般使用基本路径测试，重点模块使用多种覆盖率标准。

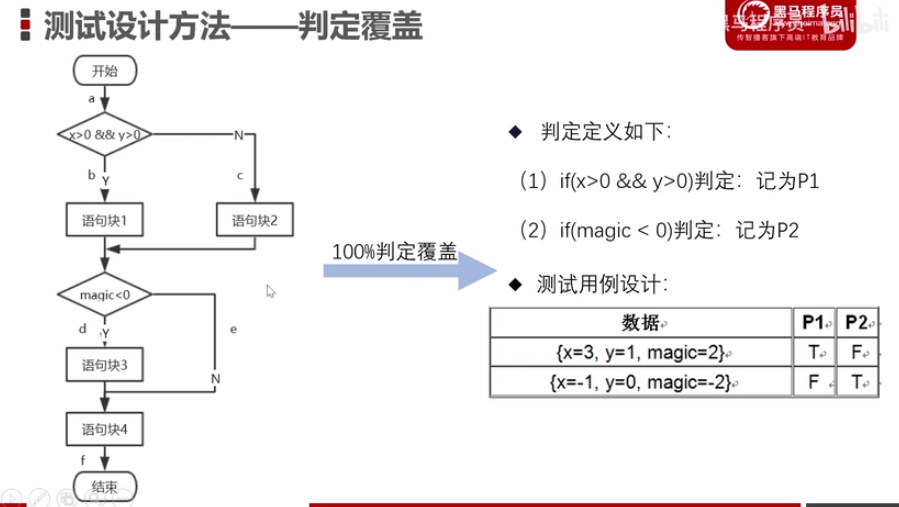
**逻辑覆盖法:**通过对程序逻辑结构的遍历实现程序的覆盖。

**语句覆盖:**设计测试用例,使得程序中每条语句至少被执行一次（在六种逻辑覆盖标准中,语句覆盖标准是最弱的，不能准确判断运算中的逻辑关系错误）



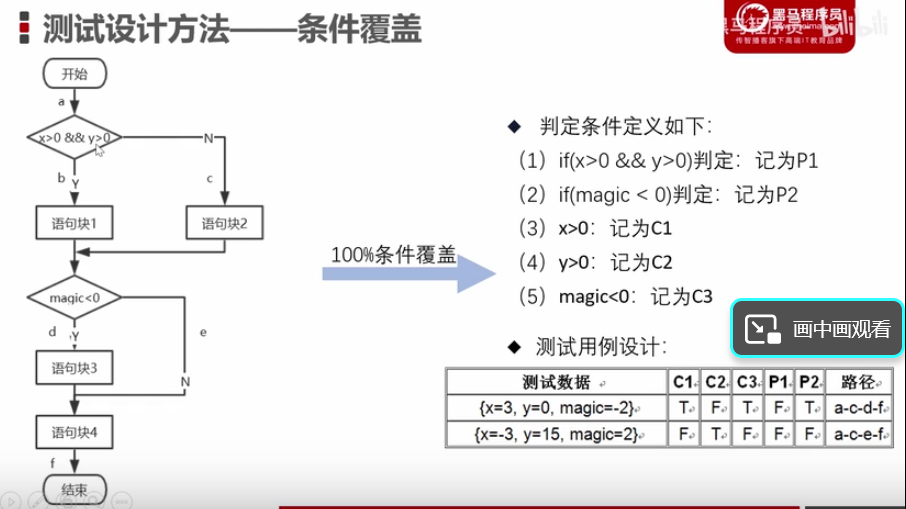
**判定覆盖:**也叫分支覆盖,设计测试用例,使得程序中的每个判断的“真"和“假”都至少被执行一次。即程序中的每个分支至少执行一次（只要满足了判定覆盖标准就一定满足语句覆

盖标准，但也会出现无法判断与或的逻辑错误）



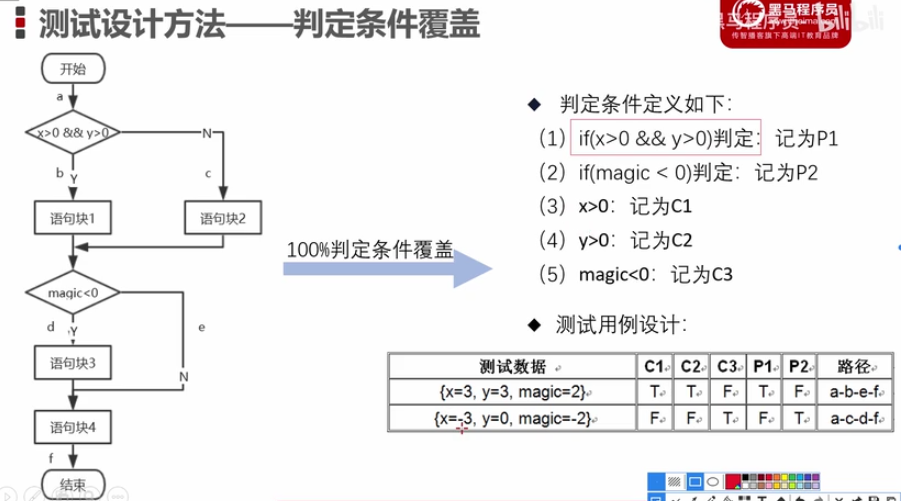
（也可以是TT和FF的真假全覆盖，但一定要覆盖所有判断）

**条件覆盖:**设计测试用例,使得判定中的每个条件至少有一次取真值,有一次取假值。



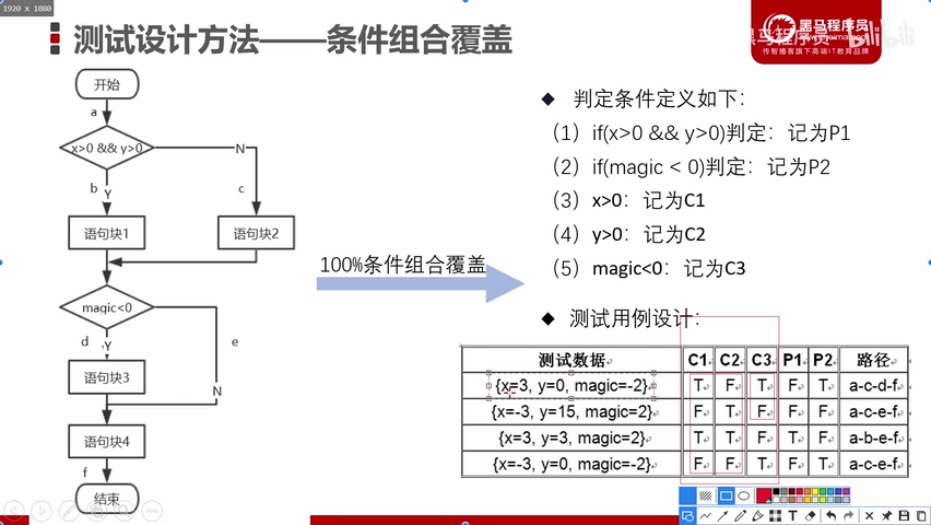
100%条件覆盖不能保证100%判定覆盖，如上P1为真未被覆盖

**判定条件覆盖:**设计测试用例,使得被测试程序中的每个判断本身的判定结果(真假)至少满足一次,同时,每个逻辑条件的可能值(真假)也至少被满足一次。即同时满足100%判定覆盖和100%条件覆盖的标准（满足判定-条件覆盖标准一定能够满足：条件覆盖、判定覆盖和语句覆盖）。



判定条件覆盖会忽略条件中取或(or)的情况：由于上例中C1、C2取的是同真同假，因此&&和||结果无法区别。

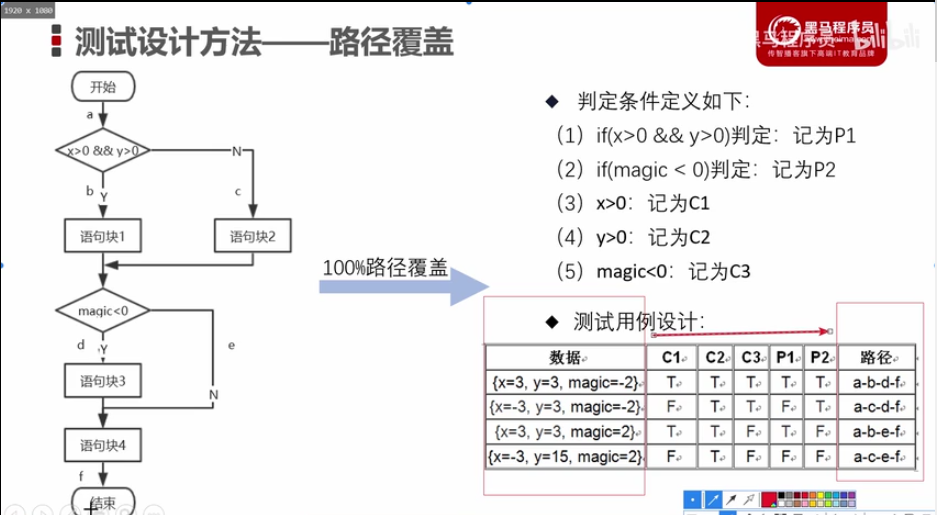
**条件组合覆盖:**设计测试用例,使得被测试程序中的每个判定中条件结果的所有可能组合至少执行一次。由于第一个判断有两个条件，要考虑他们两个的所有组合情况。



条件组合覆盖能满足判定覆盖、条件覆盖、判定-条件覆盖,也就包括语句覆盖

但不能对路径进行全覆盖（如上图中缺少路径a-b-d-f）。

**路径覆盖:**设计测试用例,覆盖程序中所有可能的路径。



路径覆盖可以对程序进行彻底的测试,比前面五种覆盖面都广。

但是满足路径覆盖,并不一定能满足条件覆盖,也就不能满足判定条件覆盖和条件组合覆盖。

**基本路径测试法:**在程序控制流图的基础上,通过分析程序的环路复杂性,导出基本可执行路径集合,从而设计测试用例

具体步骤：

1. 根据代码画出程序控制流图
2. 计算程序的环路复杂度
3. 导出可执行路径
4. 设计测试用例

