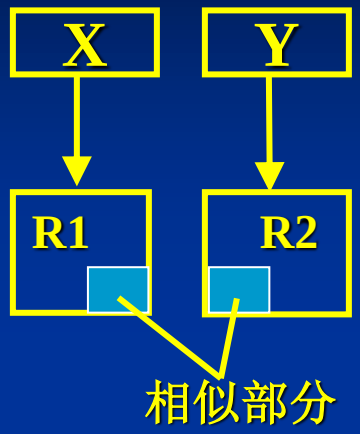
第5章内容概要

- 软件设计过程
- 软件设计原理
- ★ 启发规则
- 描绘软件结构的图形工具
- 面向数据流的设计方法
- 软件体系结构

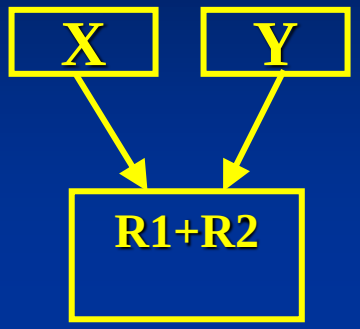
启发规则

1. 改进软件结构提高模块独立性：
   - 设计出软件的初步结构以后，应该审查分析这个结构，通过模块分解或合并，力求降低耦合提高内聚。
   - 模块功能的完善化(一个完整的功能模块,不仅能够完成指定的功能,还应能告诉使用者完成任务的状态)：
     - 执行规定的功能的部分；
     - 出错处理部分；
     - 如需要返回一系列数据给调用者，在完成数据加工或结束时应告诉调用者完成任务的状态（即返回一个该模块是否正确结束的标志）。
   - 消除重复功能，改善软件结构：

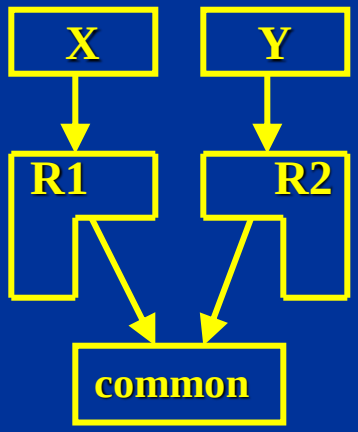
启发规则


X
Y
R1
R2
相似部分

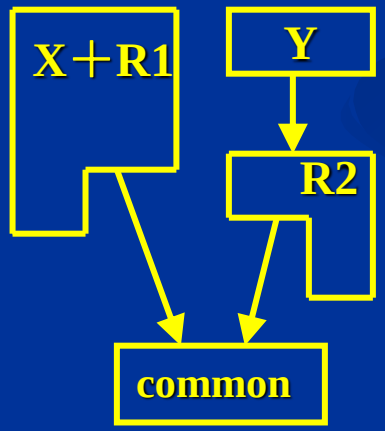
a


X
Y
R1+R2

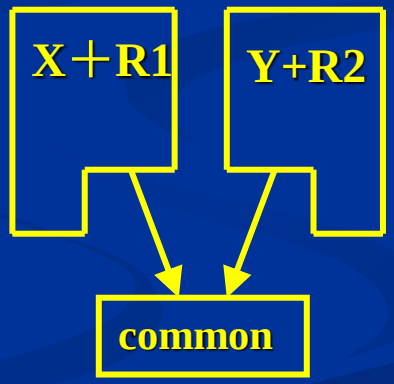
b


X
Y
R1
R2
common

c


X+R1
Y
R2
common

d


X+R1
Y+R2
common

e

启发规则

2. 模块规模应该适中：

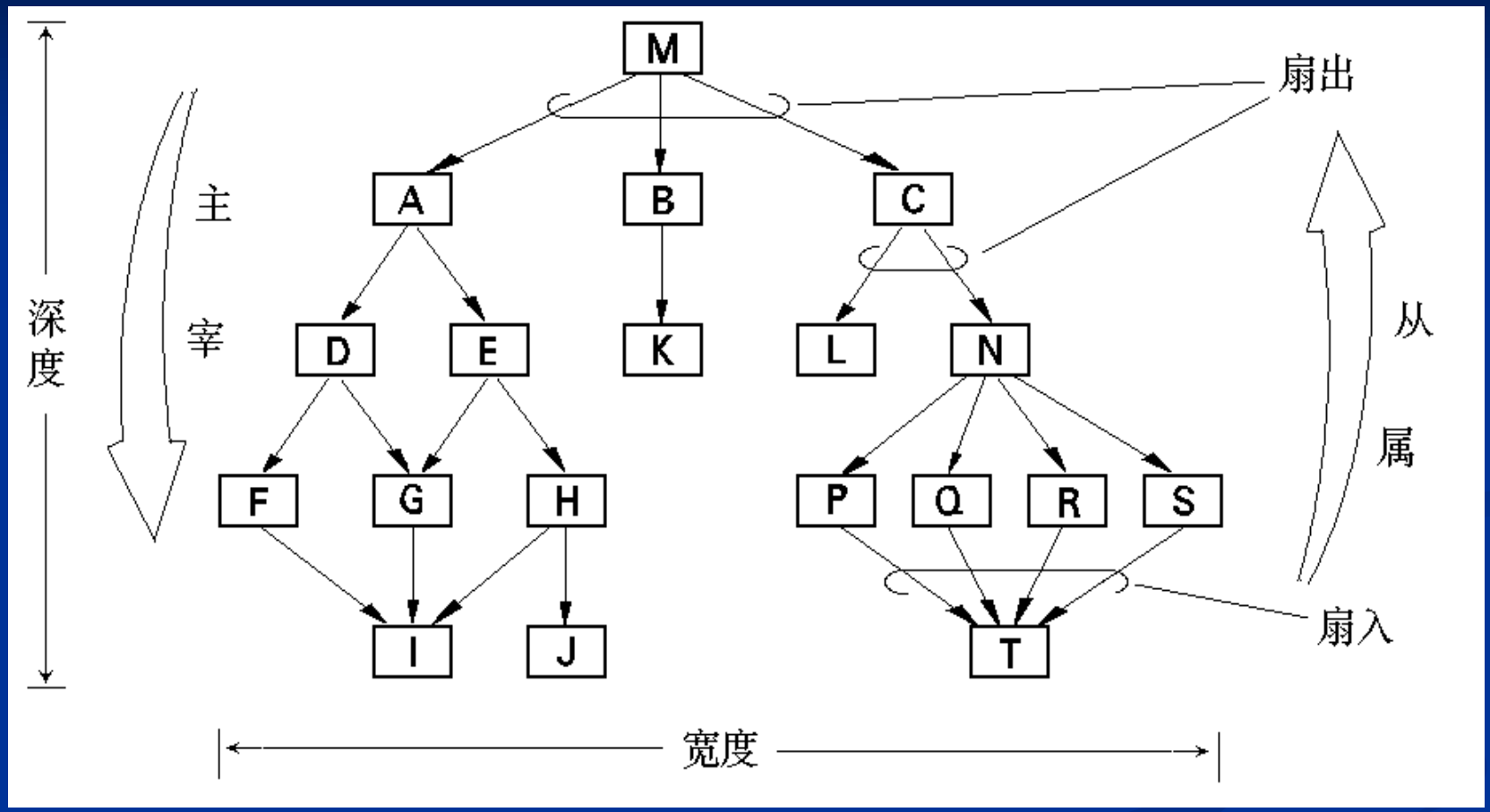
- 过大的模块可理解性差。W.M.Weinberg的研究表明：当模块长度超过30条语句时，其可理解性将迅速下降。F.T.Baker建议：模块长度可选在50句左右，使之能打印在一张打印纸上，免得读程序时要来回翻页
- 过大的模块往往是由于分解不充分，可以对功能进一步分解，生成一些下级模块或同层模块；分解模块不应该降低模块的独立性。
- 过小的模块开销大于有效操作，而且模块数目过多使系统接口复杂。

启发规则

3．深度、宽度、扇出和扇入都应适当：

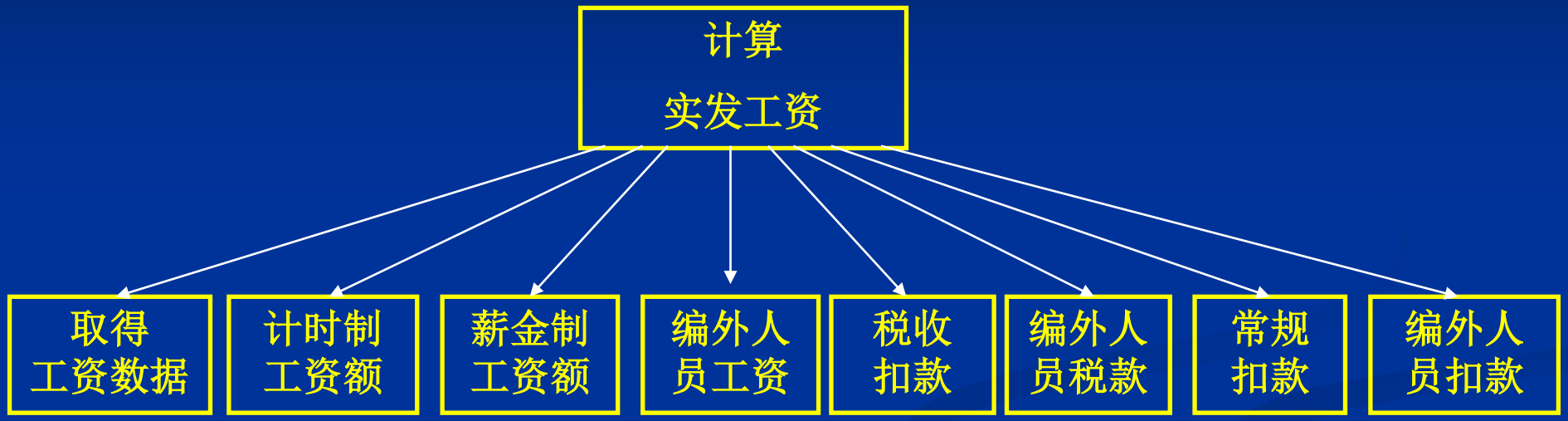
- 深度(depth)：表示软件结构中控制的层数，它往往能粗略地标志一个系统的大小和复杂程度。
- 宽度(width)：是软件结构内同一个层次上的模块总数的最大值。一般说来，宽度越大系统越复杂。
- 扇出(fan-out)：是一个模块直接控制（调用）的模块数目，扇出过大意味着模块过分复杂，需要控制和协调过多的下级模块；扇出过小（例如总是1）也不好。经验表明，一个设计得好的典型系统的平均扇出通常是3或4（扇出的上限通常是5～9）。
- 扇入(fan-in)：表明有多少个上级模块直接调用它。

启发规则

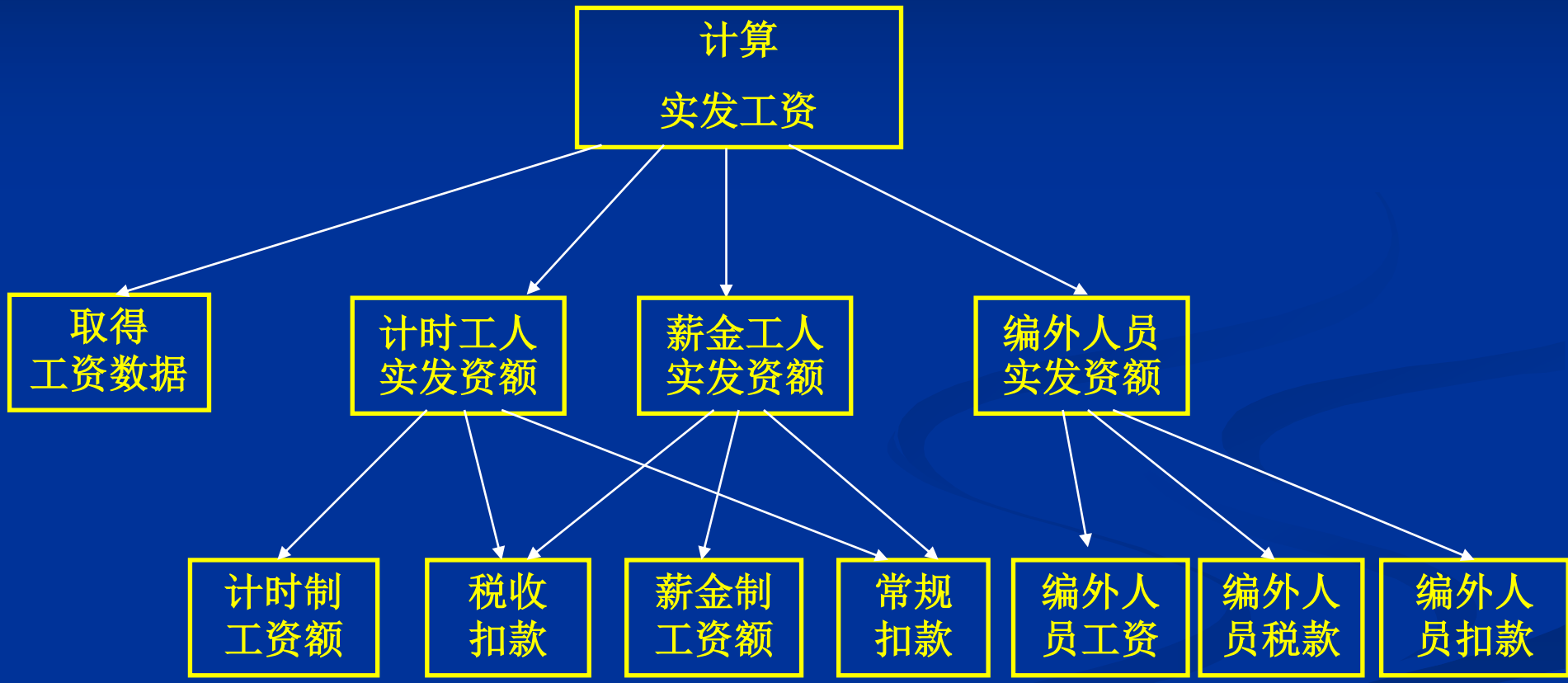


程序的层次结构图示例

启发规则

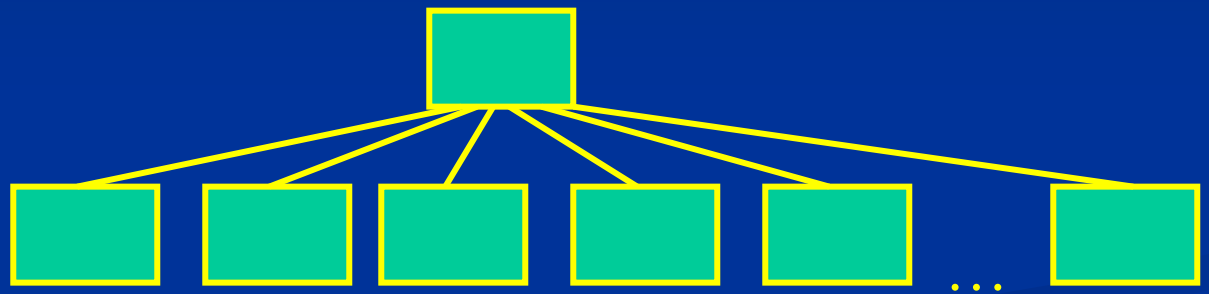

计算
实发工资
取得
工资数据
计时制
工资额
薪金制
工资额
编外人
员工资
税收
扣款
编外人
员税款
常规
扣款
编外人
员扣款

启发规则


计算
实发工资
取得
工资数据
计时工人
实发资额
薪金工人
实发资额
编外人员
实发资额
计时制
工资额
税收
扣款
薪金制
工资额
常规
扣款
编外人
员工资
编外人
员税款
编外人
员扣款

启发规则

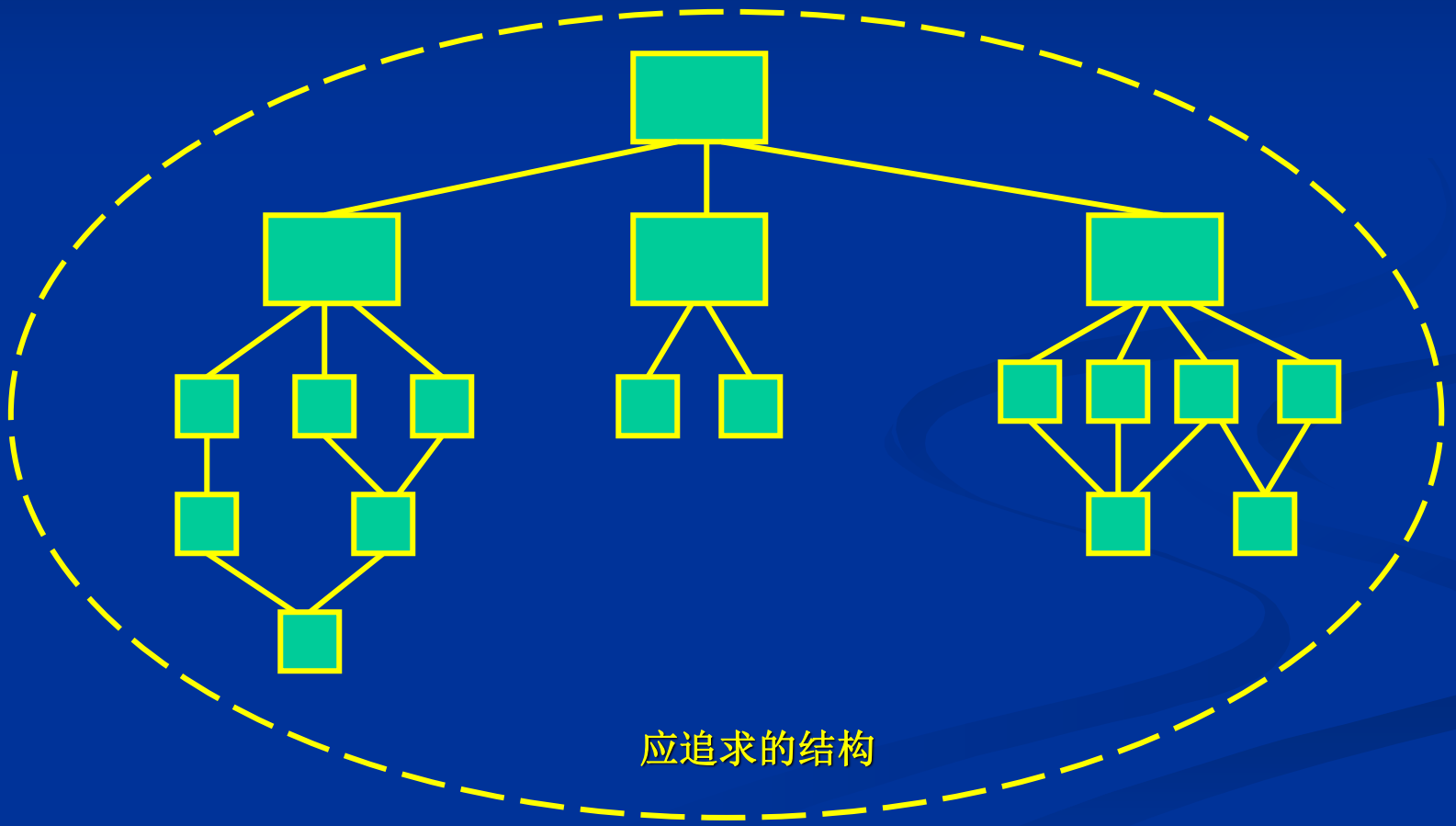
- 应避免“扁平”的结构：



应避免的结构

启发规则

- 应追求“椭圆”的结构：



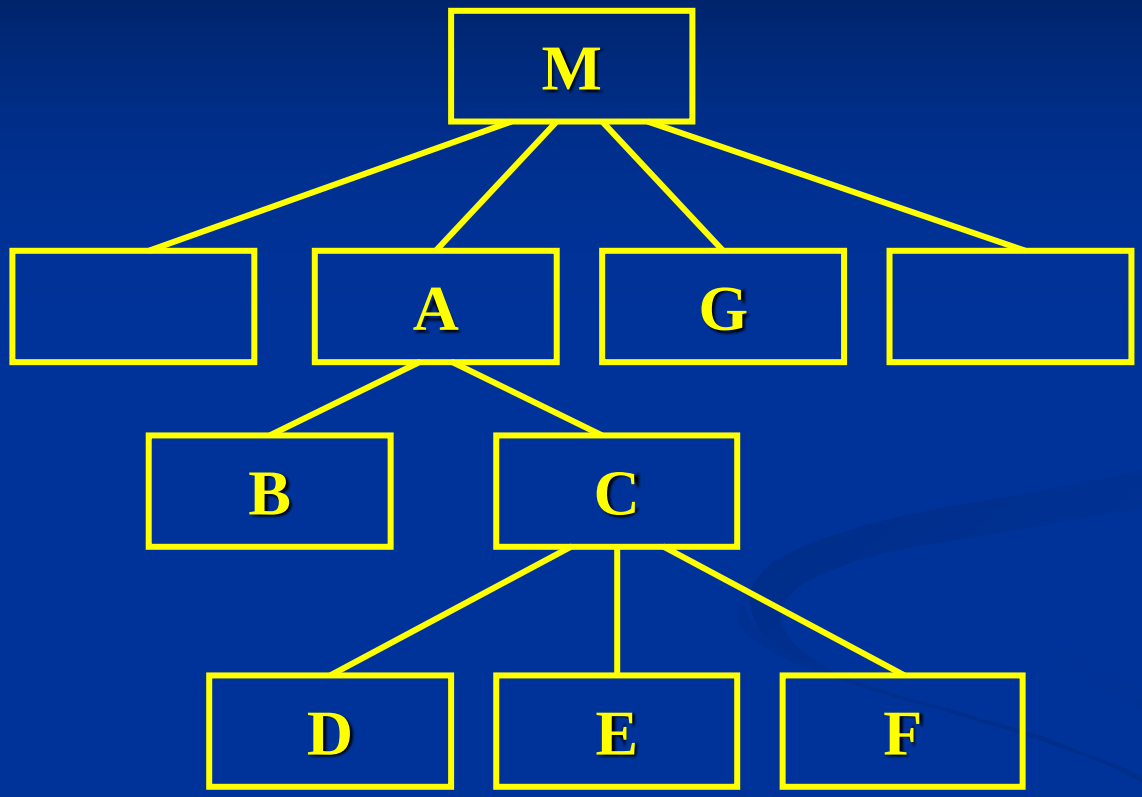
应追求的结构

启发规则

4. 模块的作用域应该在控制域之内：

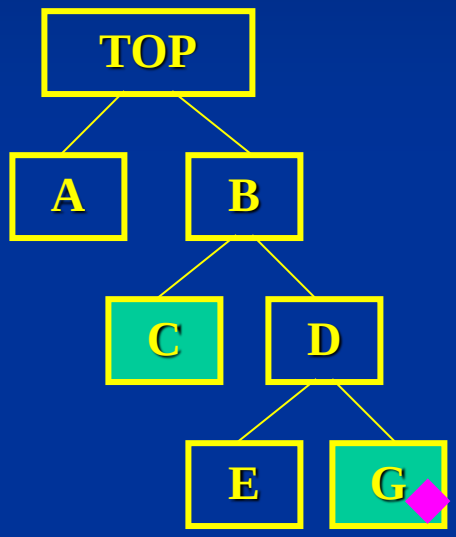
- 模块的作用域：定义为受该模块内一个判定影响的所有模块的集合。
- 模块的控制域：是这个模块本身以及所有直接或间接从属于它的模块的集合。
- 在一个设计得很好的系统中，所有受判定影响的模块应该都从属于做出判定的那个模块，最好局限于做出判定的那个模块本身及它的直属下级模块。

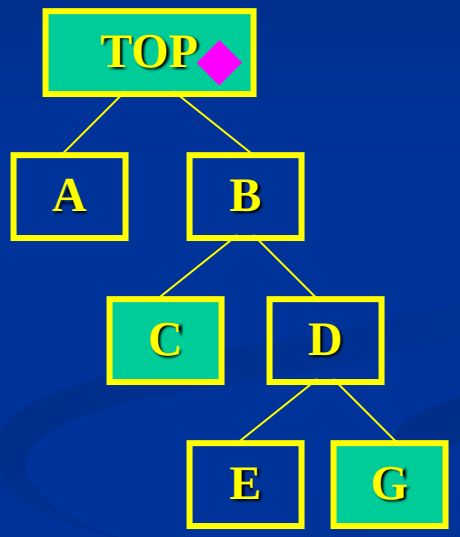
启发规则



模块的作用域和控制域

启发规则

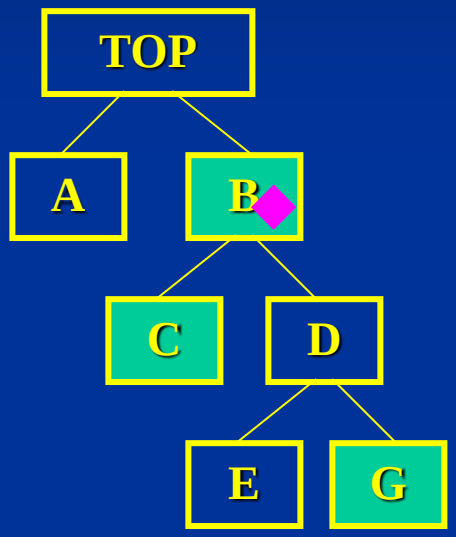


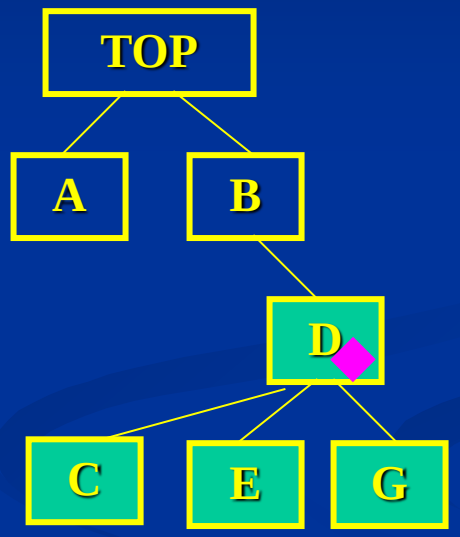


◆ ：表示判定所在位置

■ ：表示判定影响的模块

启发规则





◆ ：表示判定所在位置

：表示判定影响的模块

启发规则

- 将作用范围移动到控制范围的方法：
  - 将判定所在模块合并到父模块中，使判定处于较高层次；
  - 将受判定影响的模块下移到控制范围内；
  - 将判定上移到层次中较高的位置。

启发规则

5. 力争降低模块接口的复杂程度：

- 模块接口复杂是软件发生错误的一个主要原因。应该仔细设计模块接口，使得信息传递简单并且和模块的功能一致。
-
- QUAD_ROOT(TBL,X)
- QUAD_ROOT(A,B,C,ROOT1,ROOT2)

启发规则

6. 设计单入口单出口的模块：
   - 避免出现内容耦合。

启发规则

7．模块功能应该可以预测，避免对模块施加过多限制：

- 如果一个模块可以当做一个黑盒子，也就是说，只要输入的数据相同就产生同样的输出，这个模块的功能就是可以预测的。

启发规则