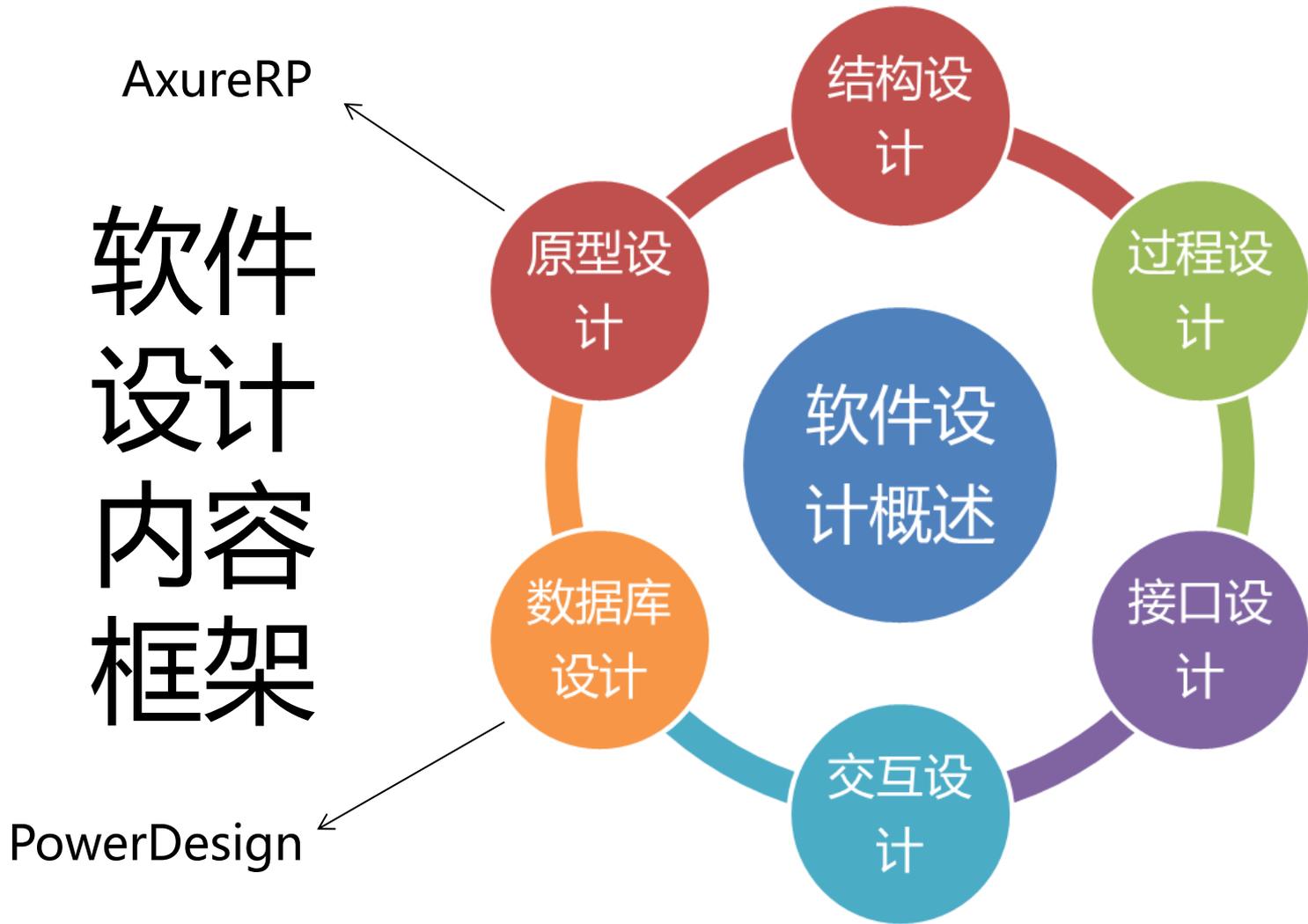


# 软件设计内容框架



# 软件设计概述(一)

杨现民博士  
江苏师范大学

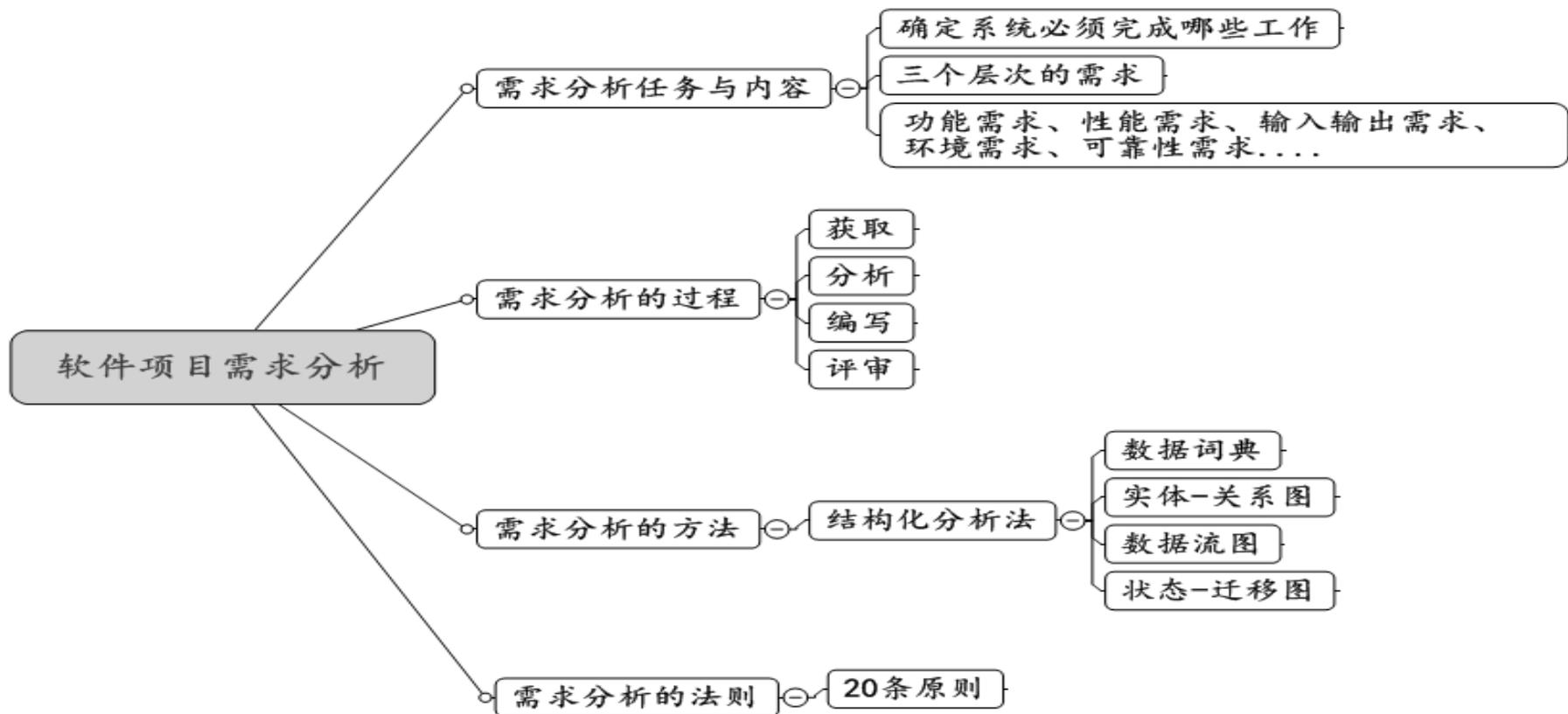
江苏师范大学教育技术学专业本科生课程

2013/10/12



EPD&M

# 课堂回顾



# 学习目标

- 掌握软件设计的概念、目标、任务与一般过程
- 理解软件设计的基本原理
- 掌握软件结构设计方法
- 掌握软件过程设计方法





# 软件设计的概念

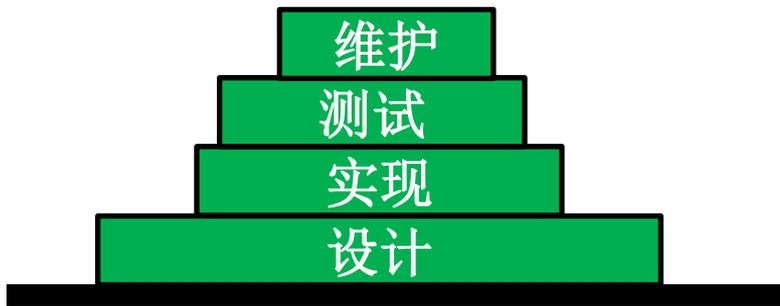
- Software design is a process of **problem solving and planning** for a software solution.
- 应用各种各样的技术和原理，足够详细的定义一个设备、一个程序或系统的**物理实现的过程**。

# 软件设计的任务

- 解决“如何做”的问题!
- 把软件需求变换成为软件表示，根据数据、功能、行为模型表示的软件需求，采用各相应的设计方法进行**概要设计**和**详细设计**。

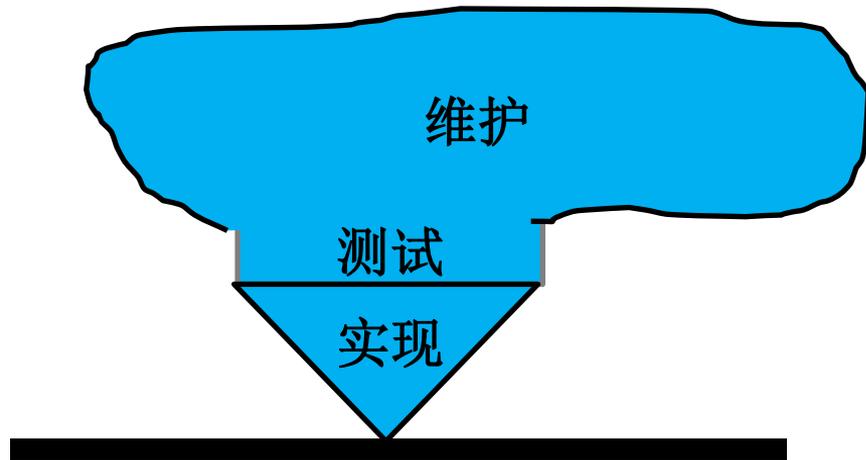


# 软件设计的重要性



有软件设计

结构稳定，易于管理和维护



无软件设计

摇摇欲坠，容易崩溃

# 软件设计的目标

- ✓ 设计必须实现分析模型中描述的所有显式需求，必须满足用户希望的所有隐式需求
- ✓ 设计必须是可读、可理解、易于编程实现、易于测试、易于维护
- ✓ 设计应从实现的角度出发，给出与数据、功能和行为相关的软件全貌

# 软件设计的过程

- 软件设计是一个将需求转变为**软件陈述**（表达）的过程，这种陈述给一个对软件的全局观点。
- 系统通过**逐步求精**使得设计陈述逐渐接近**源代码**。

# 软件设计的过程

- **Step 1** : 概要设计 Preliminary design , 关注于如何将需求转换成数据和软件框架。
- **Step 2** : 详细Detail design , 关注于将框架逐步求精细化为具体的数据结构和软件的算法表达。



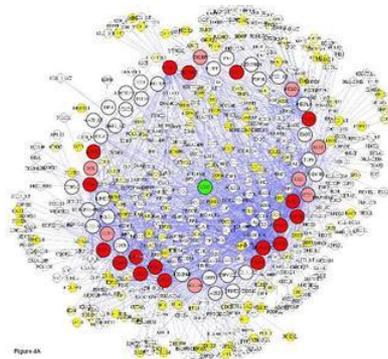
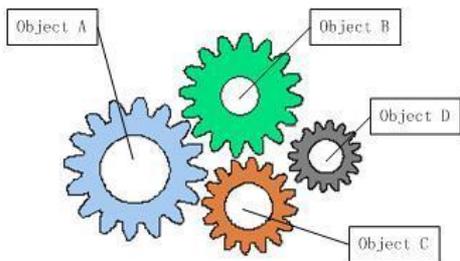
# 软件设计过程中的常见问题

- 设计过程缺乏规范
- 缺少设计文档
- 缺少统一建模语言
- 代码耦合度较高
- 内聚性较低



## 拓展：认识两个重要概念

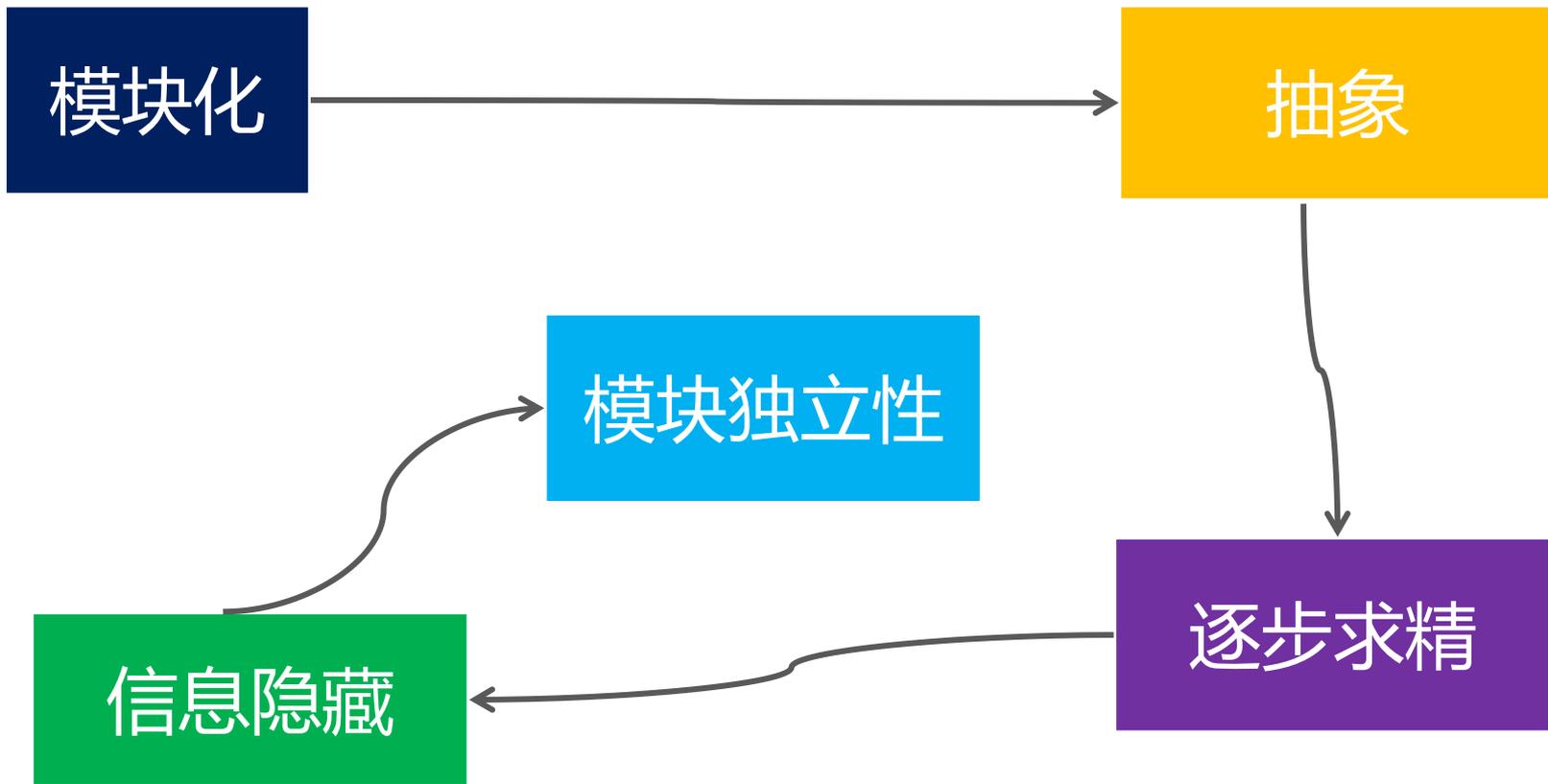
- 耦合度和内聚度是衡量软件模块**独立程度**的标准！
- 耦合度：模块之间的依赖关系的度量。
- 内聚度：模块内部各成分彼此结合的紧密程度。
- 耦合度 VS 内聚度



# 软件设计基本原则

The Basic Principles of Software Design





# 一、模块化

- **模块**又称构件，是具有一定功能、可以单独命名的被访问的数据和程序语句的集合。
- **模块化**是指将程序划分成多个模块，每个模块完成一个子功能。

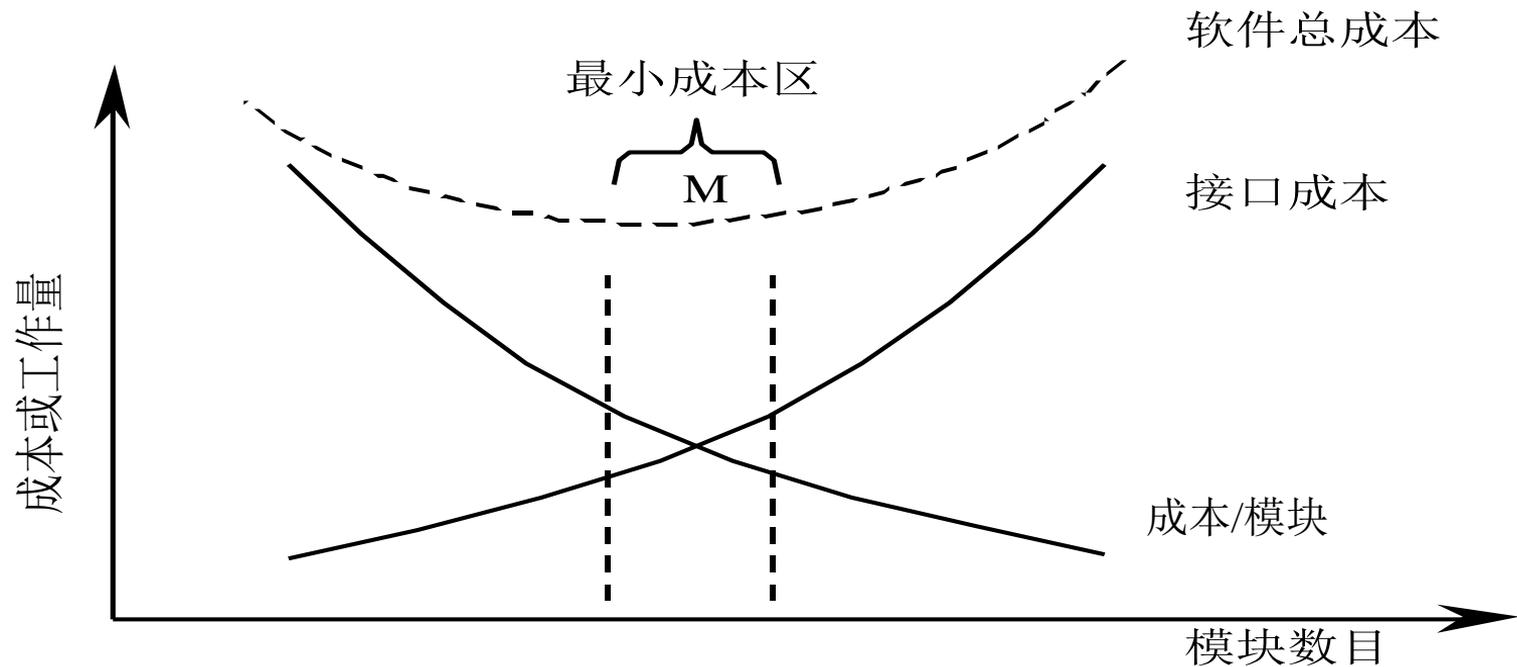


分而治之！



**是时候动动脑筋了!**

**模块化的程度是否越高越好? 模块数量越多越好?**



模块化和软件成本的关系



**抽象是人类认识世界和描述世界过程中常用的思维方法！**

## 二、抽象

- 抽象就是抽出事物的本质特性而暂时不考虑它们的细节。
- 软件工程过程的每一步，都是对软件解法的抽象层次的一次细化。
- 过程抽象、数据抽象、控制抽象。

# 抽象层次从高到低

- 在可行性研究阶段，软件被看作是系统的一个完整部件；
- 在需求分析阶段，软件解决方法是用在问题环境内熟悉的方式来描述，可被抽象为某个加工序列；
- 在概要设计阶段，软件可被抽象为一个模块的层次结构；
- 在详细设计阶段，软件可被抽象为程序流程图；
- 在软件实现阶段，软件可被抽象为某个命名的指令序列；
- 在源程序生成时，达到了抽象的最低层次。



逐步求精是一种自顶向下的设计策略！

### 三、逐步求精

- 逐步求精是将软件**体系结构**按照自顶向下的方式对各个层次的过程细节和数据细节逐步细化，直到能够用**程序设计语言的语句**实现为止，最终确立整个软件的体系结构。
- **抽象和逐步求精是互补的！**



**信息隐藏是一种重要的软件安全设计原则!**

## 四、信息隐藏

- 模块所包含的信息（过程和数据）对于其他模块来说应该是隐藏的。
- 也就是说，模块应当被这样规定和设计，使得包含在模块中的信息（过程或数据）对于其它不需要这些信息的模块来说，是不能访问的，或者说是“不可见”的。

A baby with reddish hair is lying on a green surface, playing with several colorful alphabet blocks. The blocks are yellow and green, with letters like 'f', 'u', 'o', 'e', 't', 'e', and '3' visible. A semi-transparent green banner with white text is overlaid on the image.

模块独立是模块化、抽象、信息隐藏的  
直接结果！

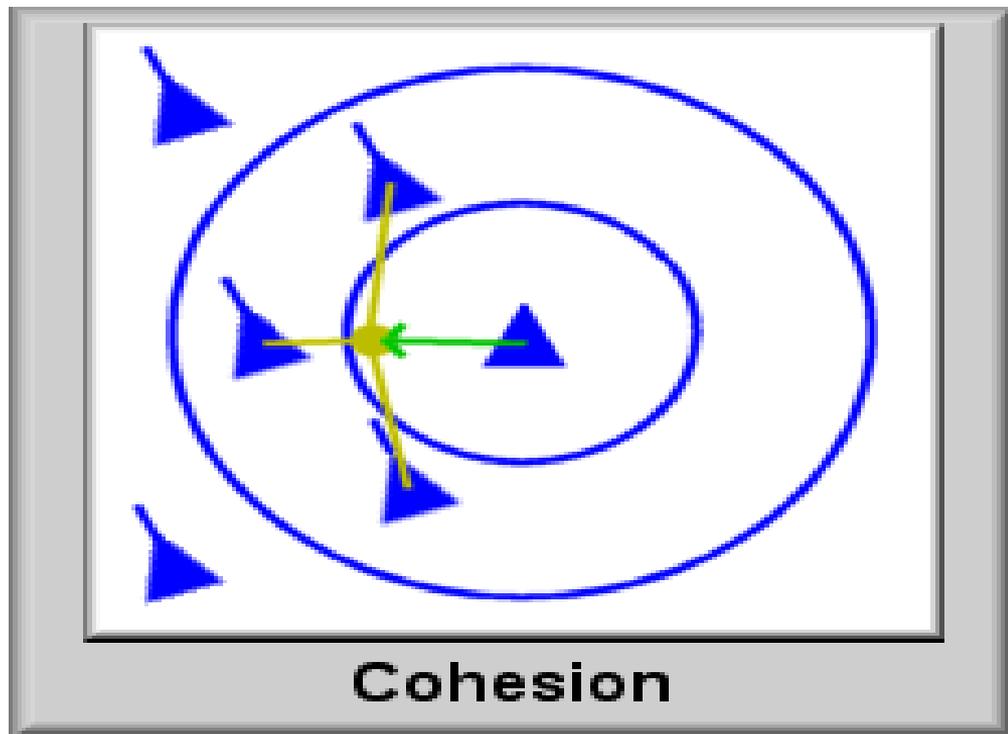
## 五、模块独立性

- 模块独立性是模块化、抽象和信息屏蔽等概念的直接产物。
- 是指软件系统中每个模块只完成系统要求的独立的子功能，并且与其他模块的联系最少且接口简单。

# 模块独立性度量标准

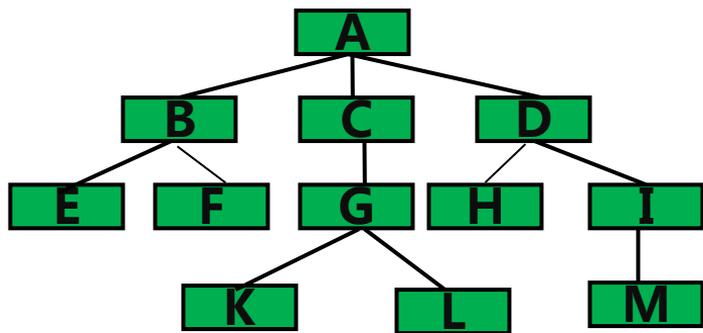
- 模块独立性度量标准：内聚度和耦合度。
- 内聚度：一个模块内部各成分之间相互关联的程度。
- 耦合度：计算机程序中模块之间相互依赖的程度。

提高系统的  
内聚度，而  
降低系统的  
耦合度。



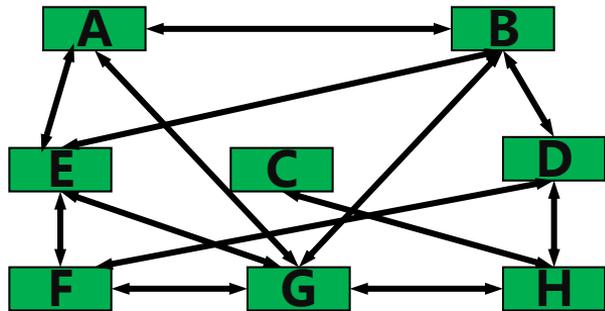


- 软件结构是软件模块之间关系的表示，它决定了整个系统的结构，也决定了系统的质量。
- 软件结构图是总体设计阶段的主要描述工具，它描述了构成系统的基本元素—模块及模块之间的调用关系，模块之间的数据传递关系。



## 树状结构

整个结构只有一个顶层模块，而对于任何一个下属模块来说，它只有一个上级模块，而且同一层模块之间不发生事故。



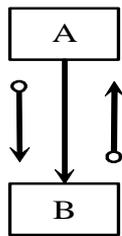
## 网状结构

任何两个模块间都可以有双向的关系，不分层次。任何两个模块都是平等的，没有从属关系。

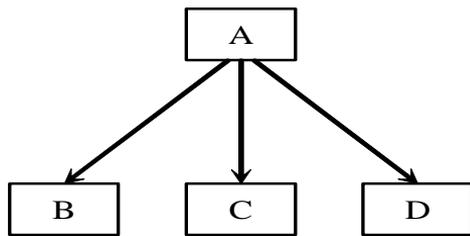
- 对于不加限制的网状结构，由于模块间相互关系的任意性，使得整个结构十分复杂，处理起来势必引起许多麻烦，这与原来划分模块为便于处理的意图相矛盾。
- 所以在软件开发的实践中，人们通常采用树状结构，而不采用网状结构。

# 结构图的基本形式

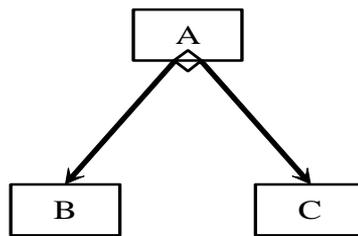
- 方框表示模块
- 带空心的箭头表示数据信息
- 带实心的箭头表示传递控制信息



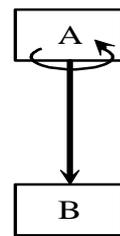
(a) 基本形式



(b) 顺序



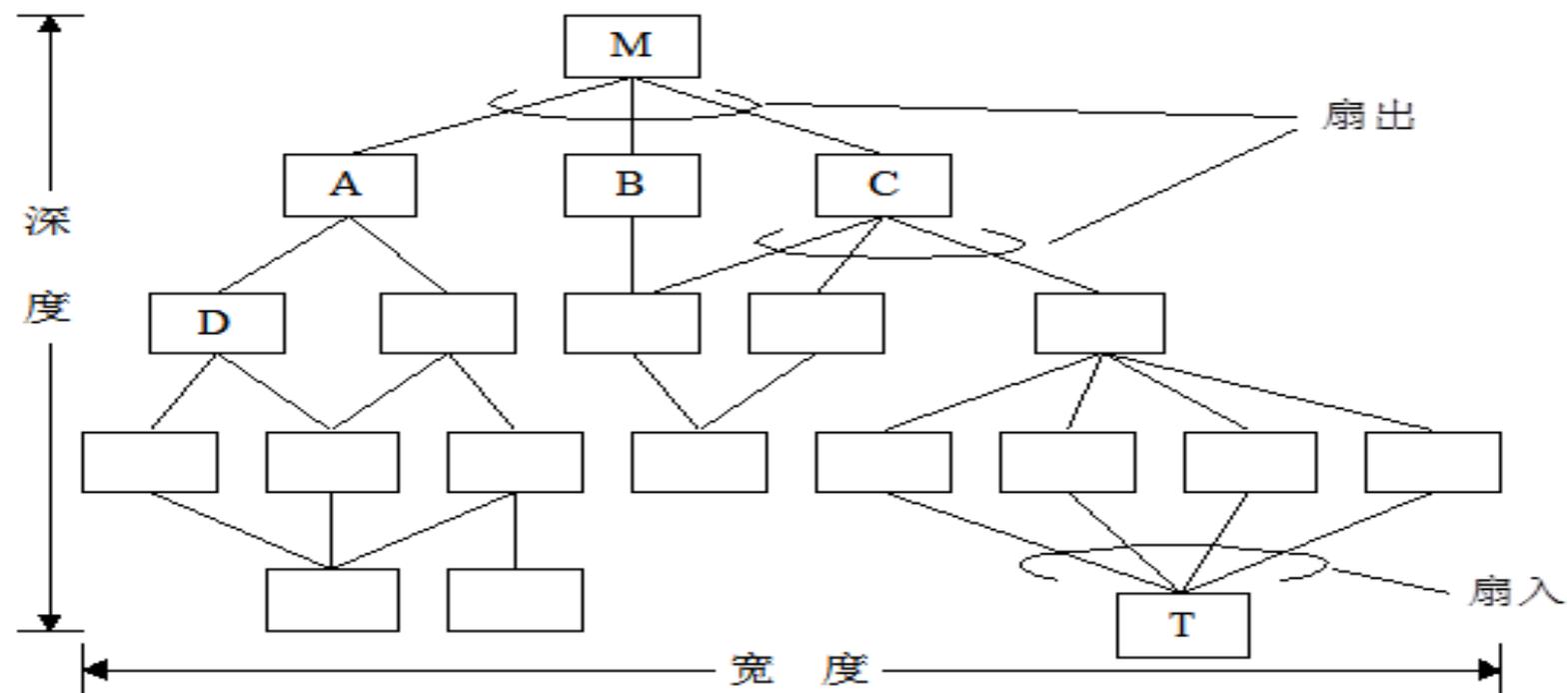
(c) 选择



(d) 重复

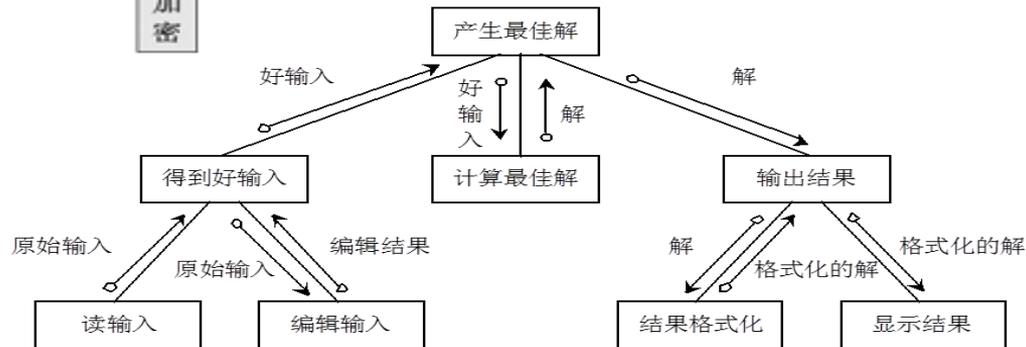
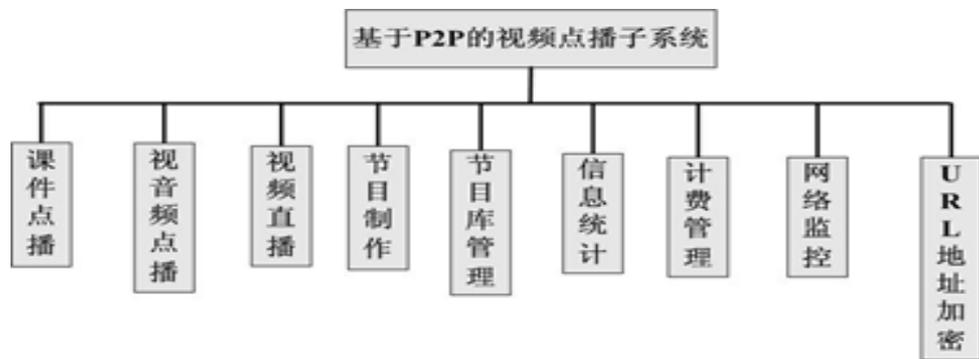
# 核心指标

- **深度**：表示软件结构中从顶层模块到最底层模块的层数。
- **宽度**：指一层中最大的模块个数。
- **扇出数**：指一个模块直接控制下属的模块个数。
- **扇入数**：指一个模块的直接上属模块个数。

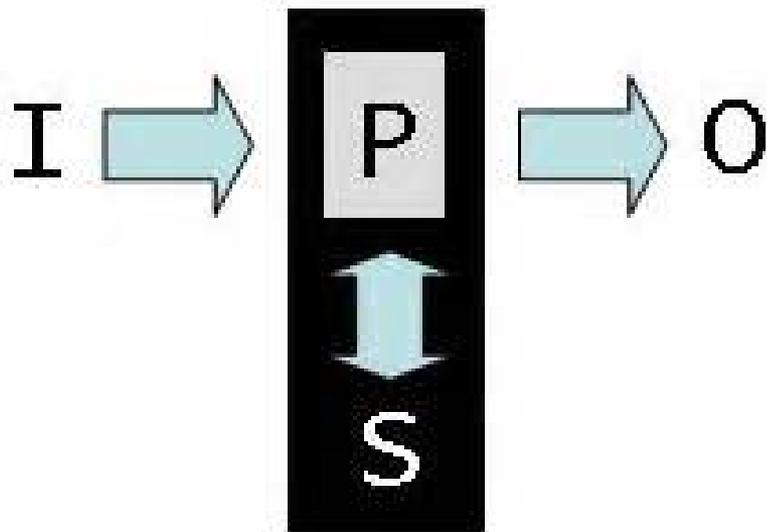


# 几种软件结构图

## ■ 层次图、结构图、HIPO图

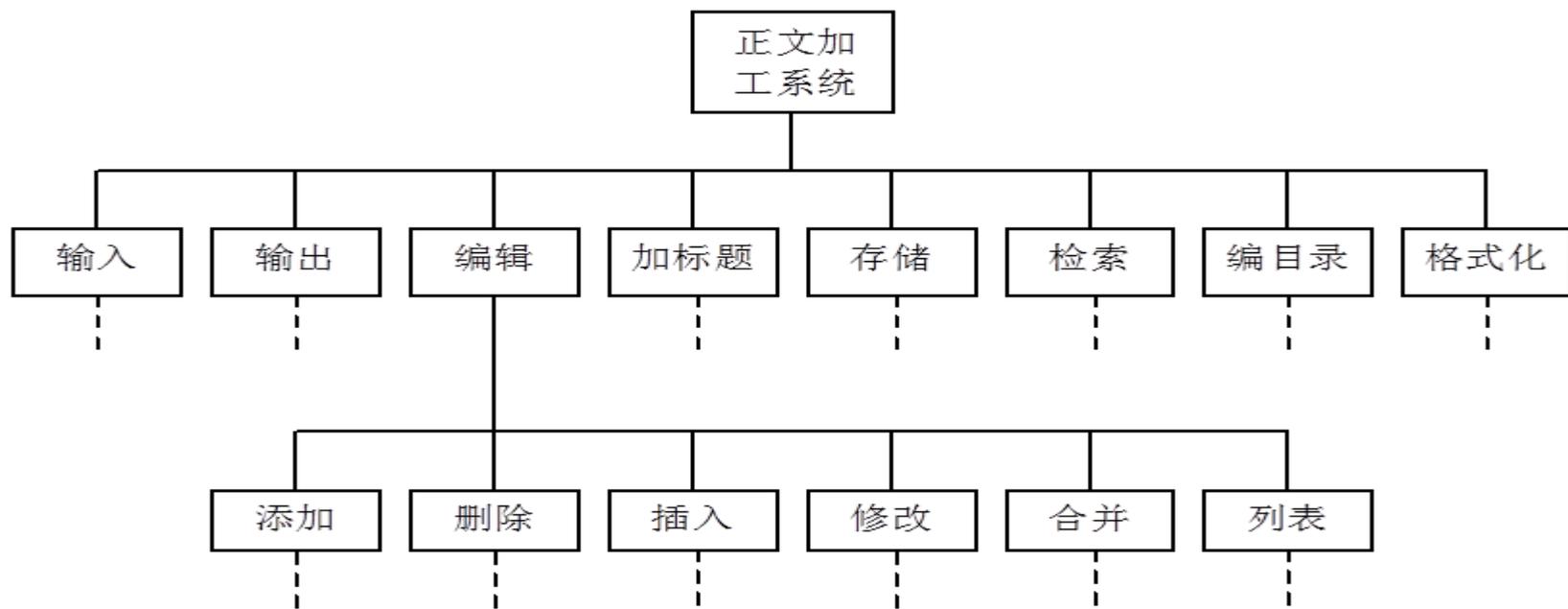


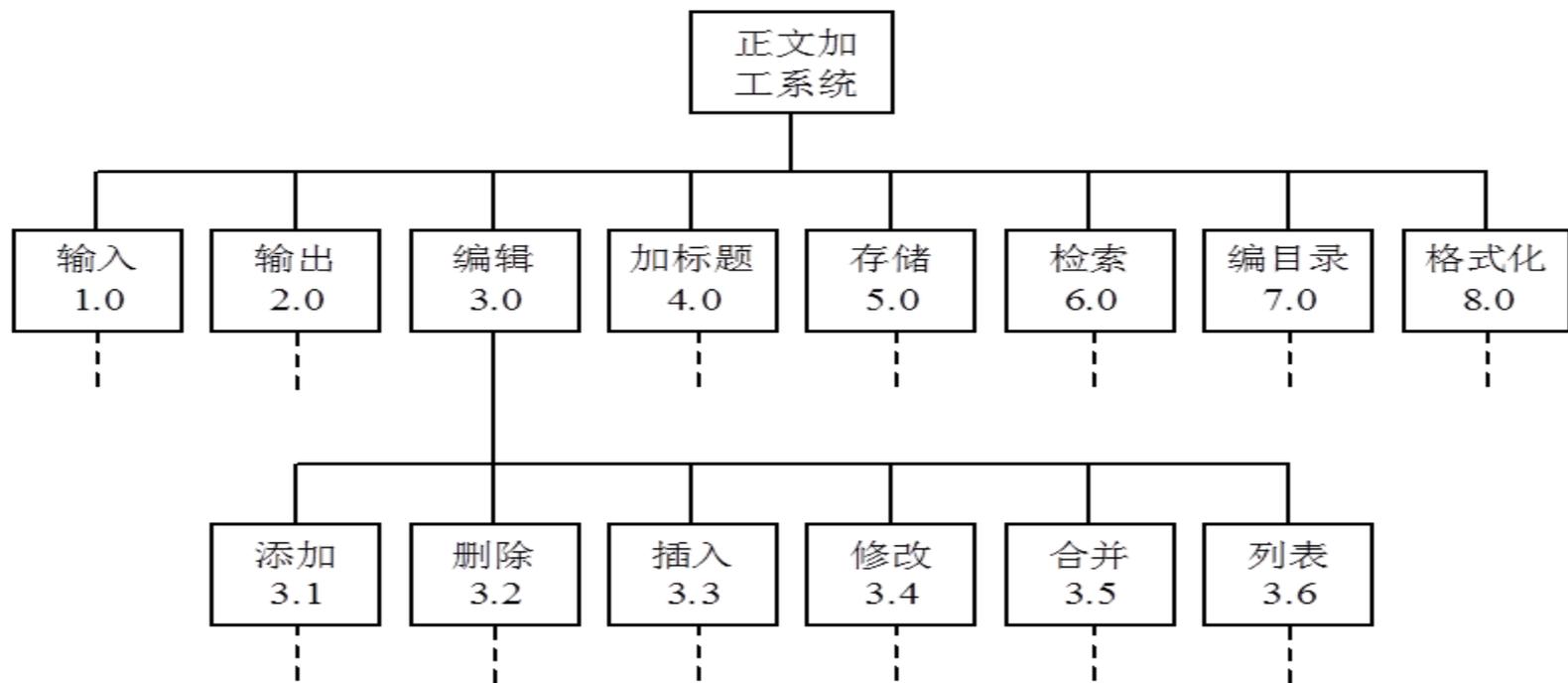
# HIPO Chart

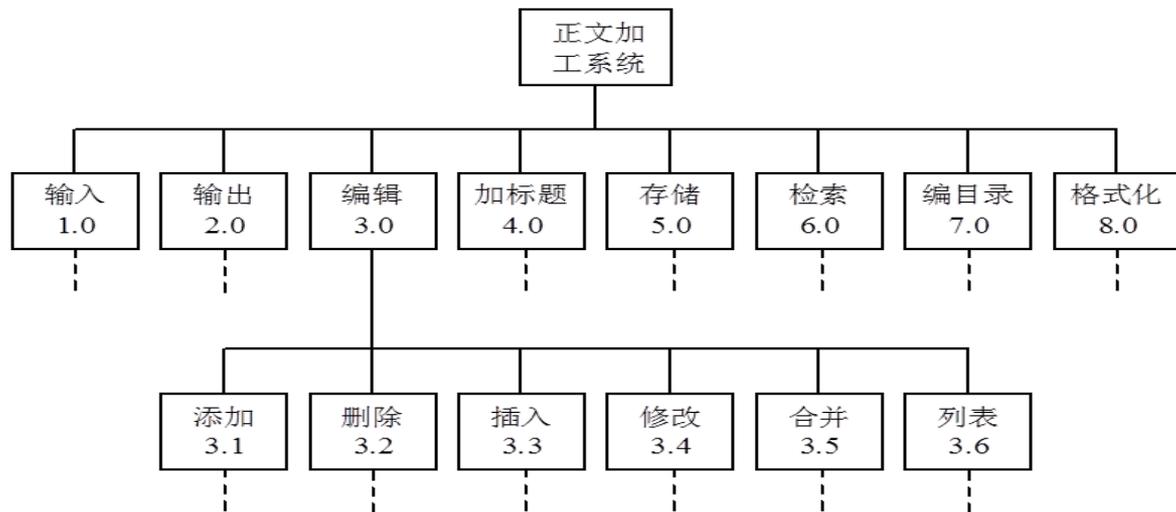


HIPO 图 (Hierarchy Plus

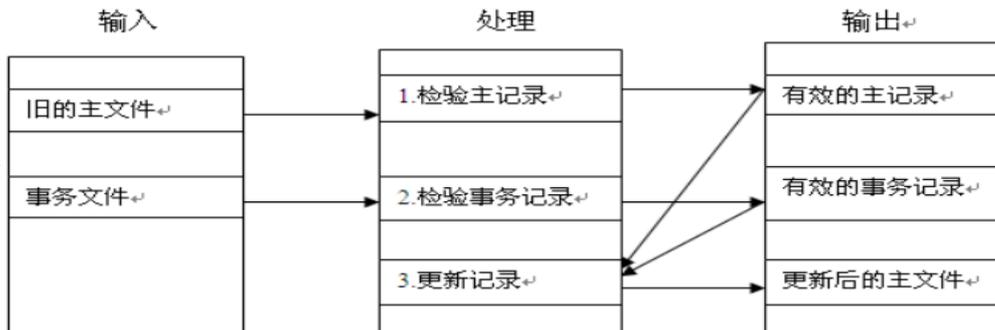
Input/Processing/Output) 是美国 IBM 公司 70 年代发展起来的表示软件系统结构的工具。







**每个模块编号对应一个IPO**





# 软件过程的常用表示工具



# 流程图

- 用统一规定的标准符号描述程序执行具体步骤的图形表示，使用历史悠久、流行最广



起止端点



数据



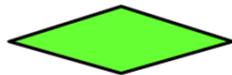
处理



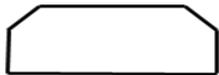
准备或预处理



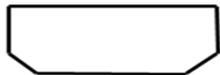
预先定义的处理



条件判断



循环上界限



循环下界限



文档



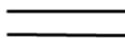
流线



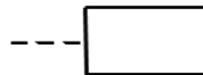
虚线



省略符



并行方式

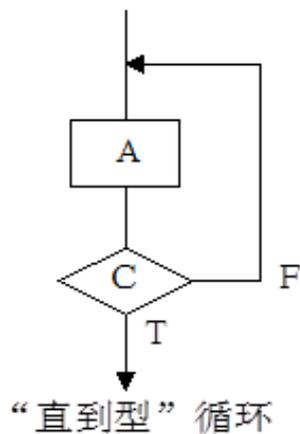
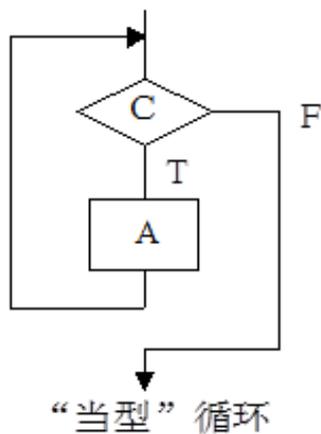
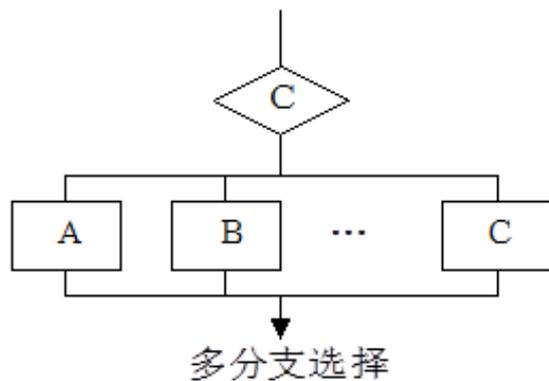
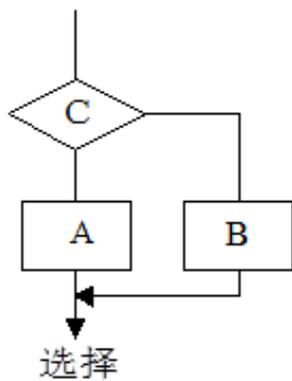
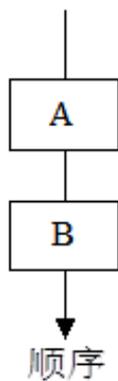


注释



控制流

# 三种基本控制结构



其中A或B为：  
①非转移语句(可以是空)  
②三种基本结构之一  
C为判定条件

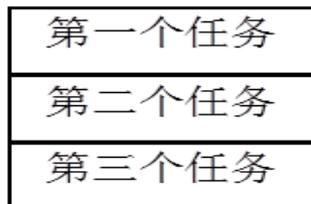
## 流程图优缺点分析

- 优点：流程图虽然比较**直观**，**灵活**，并且比较容易**掌握**
- 缺点：随意性和灵活性却使它不可避免地存在着一些缺点，易使程序员忽略程序的**全局结构**，**随意转移**控制。

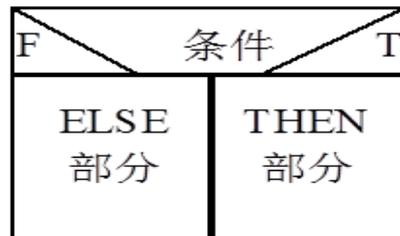
## 盒图（N-S图）

- 1983年，美国的I. Nassi和B. Sheiderman共同提出一种不用Go To语句，不需要流向线的结构化流程图
- 每个处理步骤用一个盒子表示，盒子可以嵌套
- 只能上头进下头出，限制随意的控制转移

# 盒图的基本控制结构



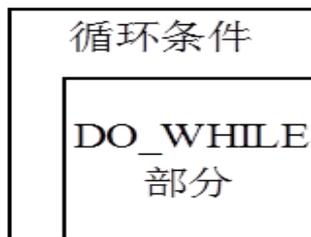
(a) 顺序结构



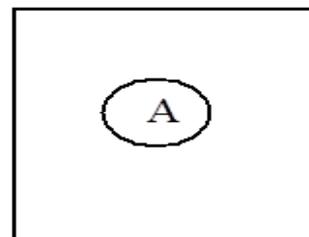
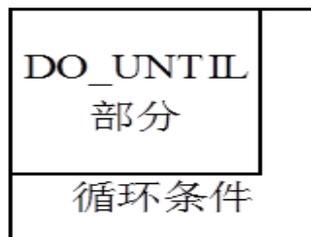
(b) 选择结构



(c) 多分支结构



(d) 循环结构



(e) 调用子程序 A

# 盒图的优缺点分析

## ■ 优点

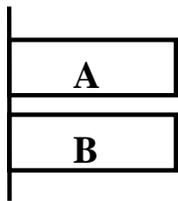
- 功能域（即某一个特定控制结构的作用域）有明确的规定，并且可以很直观地从N-S图上看出来；
- 它的控制转移不能任意规定，必须遵守结构化程序设计的要求；
- 很容易确定局部数据和全局数据的作用域；
- 很容易表现嵌套关系，也可以表示模块的层次结构；

## ■ 缺点：

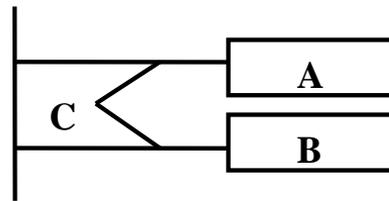
- 画图麻烦，修改麻烦

## 问题分析图（PAD图）

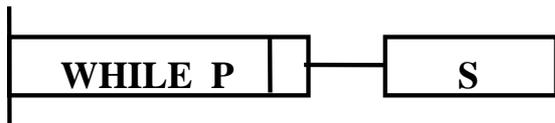
- 由日本日立公司的二村良彦等人于1979年提出，它用二维树形结构的图表示程序的控制流，将这种图转换为程序代码比较容易。
- 既克服了传统流程图不能清晰表达程序结构的缺点，又不像N-S图把全部程序约束在一个方框内的限制。



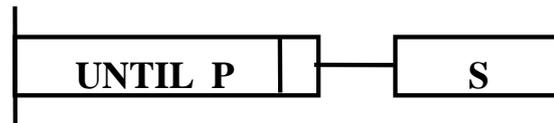
(a) 顺序结构



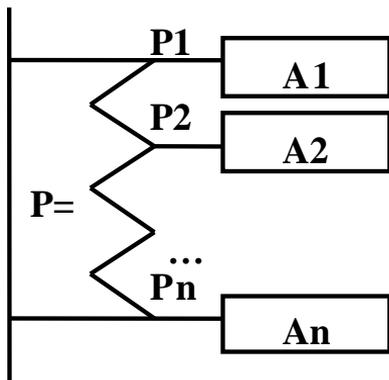
(b) 选择结构



(c) WHILE型循环结构



(d) UNTIL型循环结构



(e) 多分支结构

# PAD图的优缺点分析

## ■ 优点

- 1 . 使用表示结构优化控制结构的PAD符号所设计出来的程序必然是程序化程序；
- 2 . PAD图所描述的程序结构十分清晰；
- 3 . 用PAD图表现程序逻辑，易读、易懂、易记。
- 4 . 很容易将PAD图转换成高级程序语言源程序，有利于提高软件可靠性和软件生产率；
- 5 . 既可用于表示程序逻辑，也可用于描述数据结构；
- 6 . PAD图的符号支持自顶向下、逐步求精方法的使用。

A scenic landscape featuring a vibrant green field in the foreground, a blue sky with scattered white clouds, and a seagull in flight. In the background, there are rocky, light-colored cliffs. The overall scene is bright and clear.

PAD编程自动化系统能否实现?

# 过程设计语言

- PDL(Procedure Design Language, PDL)是用来描述模块内部具体算法的非正式且比较灵活的语言
- 外层语法确定，内层语法不确定
- 又称为伪程序、伪代码

# 过程设计语言

## ■ 顺序结构

- 处理 S1
- 处理 S2
- ...
- 处理 Sn

## ■ 选择结构

- If 条件  
处理 S1
- Else  
处理 S2
- Endif

## ■ 多分支结构

- Case of  
Case(1) 处理 1  
Case(2) 处理2  
...  
Else 处理n
- Endcase

# 过程设计语言

- “当”型循环
  - While 条件
    - 循环体
  - Endwhile

- “直到”型循环
  - Repeat
    - 循环体
  - Until 条件

- “for”型循环
  - For i=1 To n
    - 循环体
  - End For

# 过程设计语言优缺点

## ■ 优点

- 易编写、易理解，也容易转换成源程序；
- 提供的机制较图形全面，为保证详细设计与编码的质量创造了有利条件；
- 可作为注释嵌入在源程序中一起构成程序文档；
- 可自动生成程序代码（PDL/C、PDL/Pascal），提高软件生产率。

## ■ 缺点

- 不够直观、形象，英语准确性较高；
- 描述复杂条件组合与动作间的对应关系时，不够简单。

# 课堂小结与作业

Class Closing and Homework



# 课程作业

- 1. 分别绘制某软件（自选）的层次图、结构图和HIPO图
- 2. 使用流程图、N-S图、PAD图或过程设计语言描述某个软件的过程（比如成绩管理、火车票预订、网上购物等）



# 读书+听课+思考+实践!



杨现民博士

江苏师范大学



手机: 15862183989

E-Mail: [yangxianmin8888@163.com](mailto:yangxianmin8888@163.com)

QQ: 16997036

简历: <http://lcell.bnu.edu.cn/TeamMember/Yang/index.html>

微博: <http://t.sina.com.cn/yangxianmin8888>

通信地址: 江苏省徐州市铜山新区上海路101号 江苏师范大学 教育研究院

邮编: 221116

\* +++++ 态度决定高度 +++++\*