

第6章 详细设计

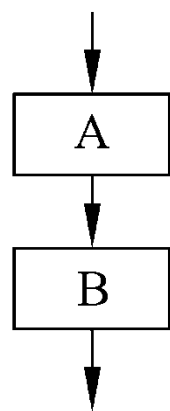
结构程序设计

人机界面设计

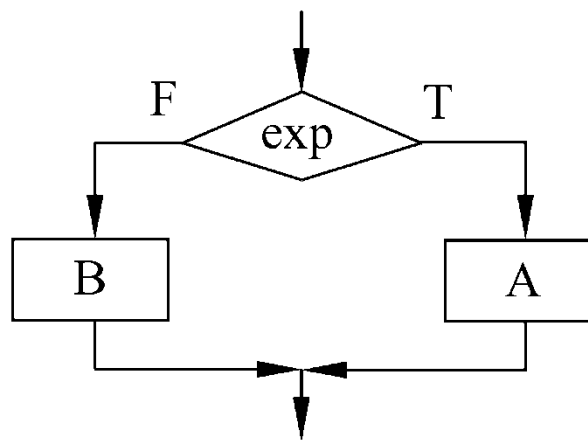
过程设计的工具

6.1 结构程序设计

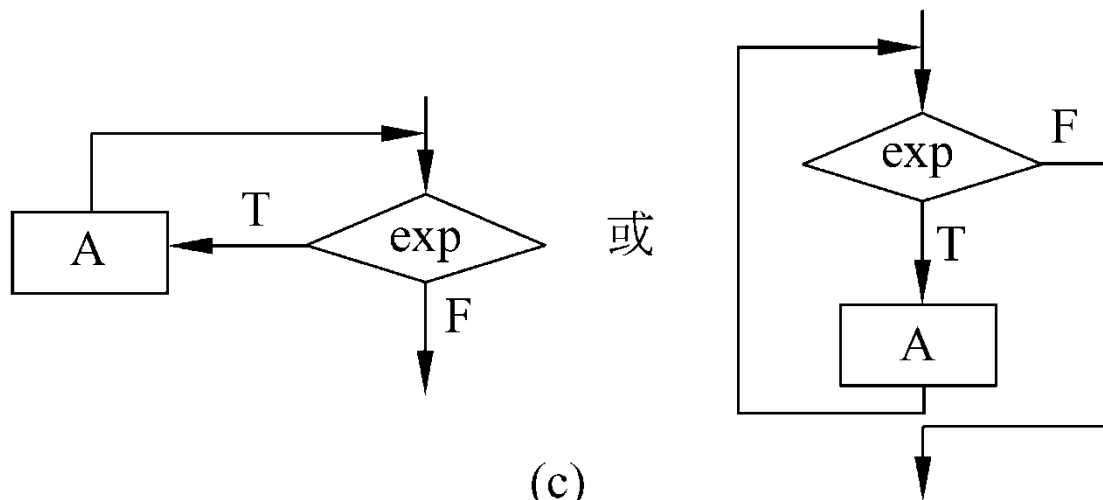
- **经典定义：** 如果一个程序的代码块仅仅通过顺序、选择和循环这3种基本控制结构进行连接，并且每个代码块只有一个入口和一个出口，则称这个程序是结构化的。



(a)

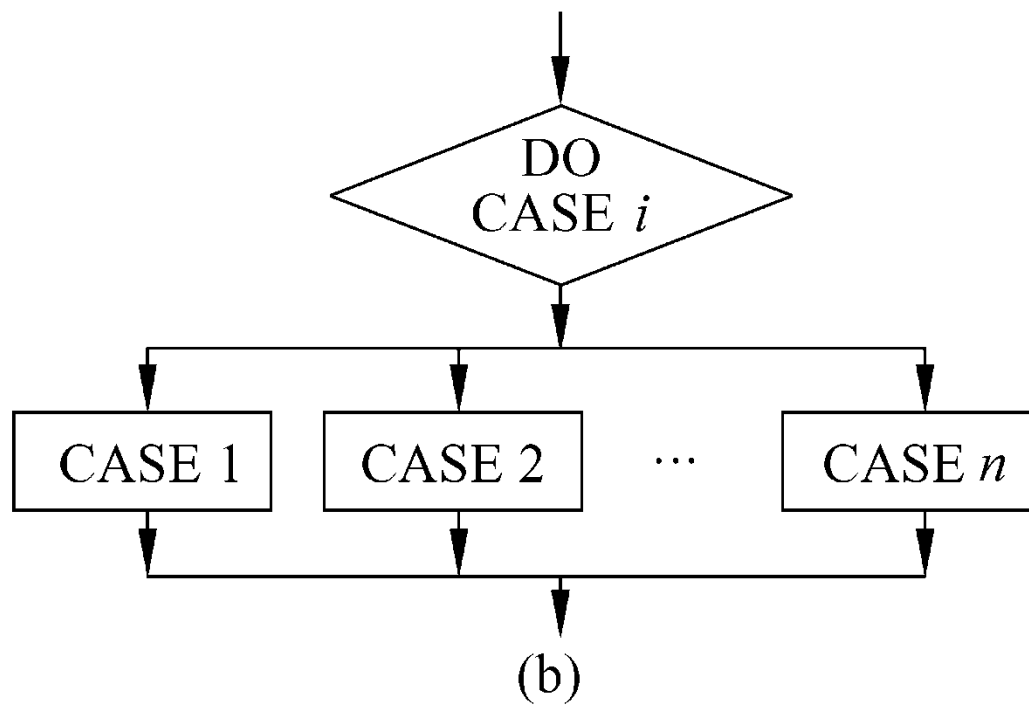
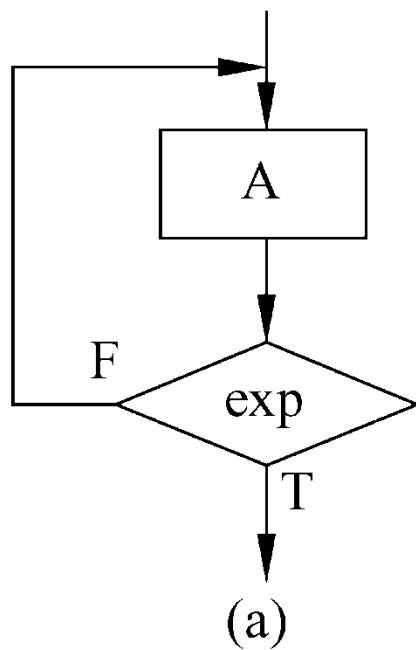


(b)



(c)

3种基本的控制结构



其他常用的控制结构

6.2 人机界面设计

- 人机界面设计是接口设计的重要组成部分。对于交互式系统来说，人机界面设计和数据设计、体系结构设计及过程设计一样重要。
- 人机界面的设计质量，直接影响用户对软件产品的评价，从而影响软件产品的竞争力和寿命，必须对人机界面设计给予足够重视。

6.2.1 设计问题

设计人机界面过程中会遇到的4个问题：

- 系统响应时间
- 用户帮助设施
- 出错信息处理
- 命令交互

1. 系统响应时间

- 系统响应时间指从用户完成某个控制动作，到软件给出预期的响应之间的这段时间。系统响应时间有两个重要属性：长度和易变性。
- **长度：**如果系统响应时间过长，用户就会感到紧张和沮丧；系统响应时间过短会迫使用户加快操作节奏，从而可能会犯错误。
- **易变性：**指系统响应时间相对于平均响应时间的偏差，即使系统响应时间较长，响应时间易变性低也有助于用户建立起稳定的工作节奏。

2. 用户帮助设施

- 大多数现代软件都提供联机帮助设施，用户无须离开用户界面就能解决自己的问题。常见的帮助设施可分为集成的和附加的两类。
- **集成的帮助设施**设计在软件里面，它对用户工作内容是敏感的，用户可以从与刚刚完成的操作有关的主题中选择一个请求帮助。
- **附加的帮助设施**是在系统建成后再添加到软件中的，它实际上是一种查询能力有限的联机用户手册。
- 集成的帮助设施优于附加的帮助设施。

3. 出错信息处理

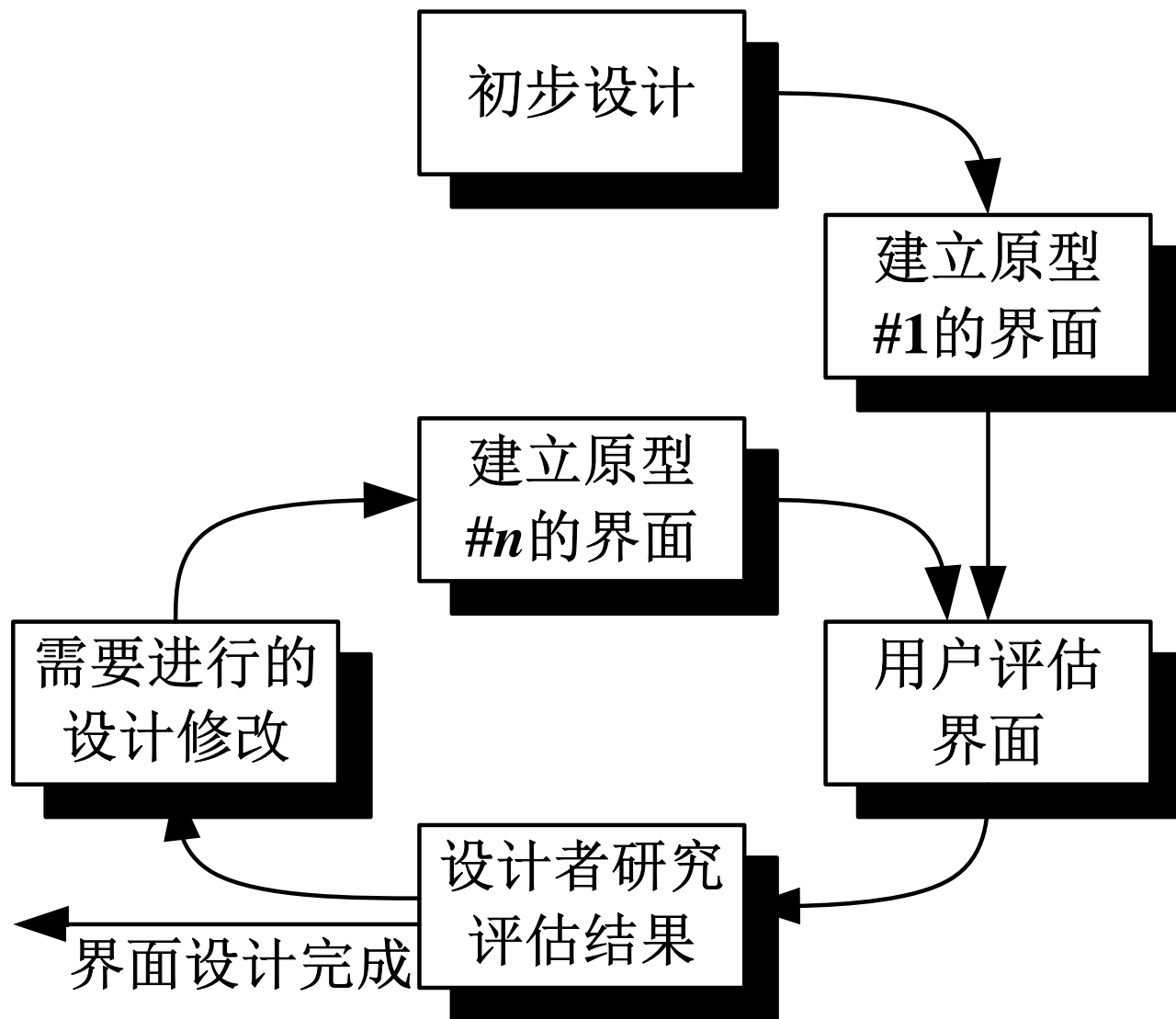
- 出错信息和警告信息，是出现问题时交互式系统给出的“坏消息”。

4. 命令交互

- 多数情况下，用户既可以从菜单中选择软件功能，也可以通过键盘命令序列调用软件功能。
- 在理想的情况下，所有应用软件都有一致的命令使用方法。

6.2.2 设计过程

- 用户界面设计是一个迭代的过程，也就是说，通常先创建设计模型，再用原型实现这个设计模型，并由用户试用和评估，然后根据用户意见进行修改。
- 为了支持上述迭代过程，各种用于界面设计和原型开发的软件工具应运而生。这些工具被称为用户界面工具箱或用户界面开发系统。



界面设计评估周期

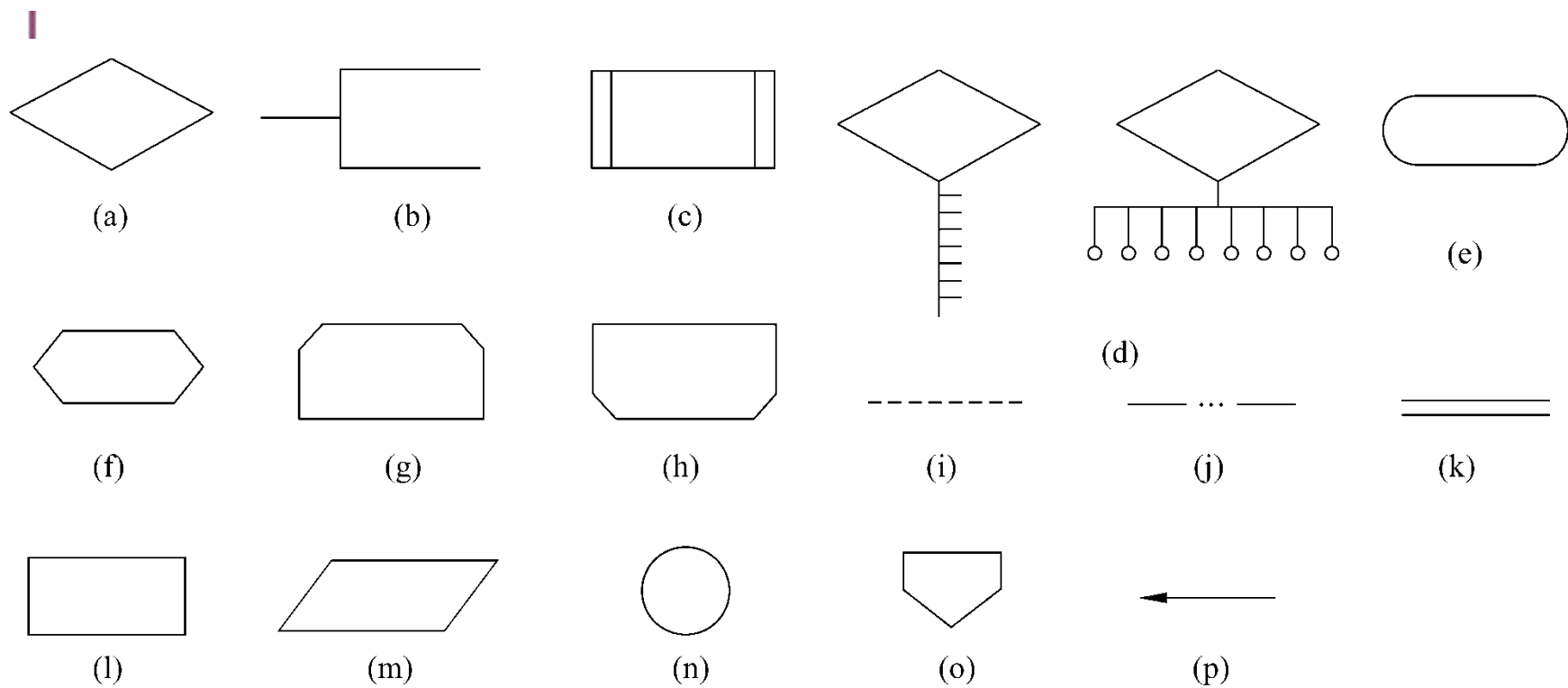
6.2.3 人机界面设计指南

- 一般交互指南
- 信息显示指南
- 数据输入指南

6.3 过程设计的工具

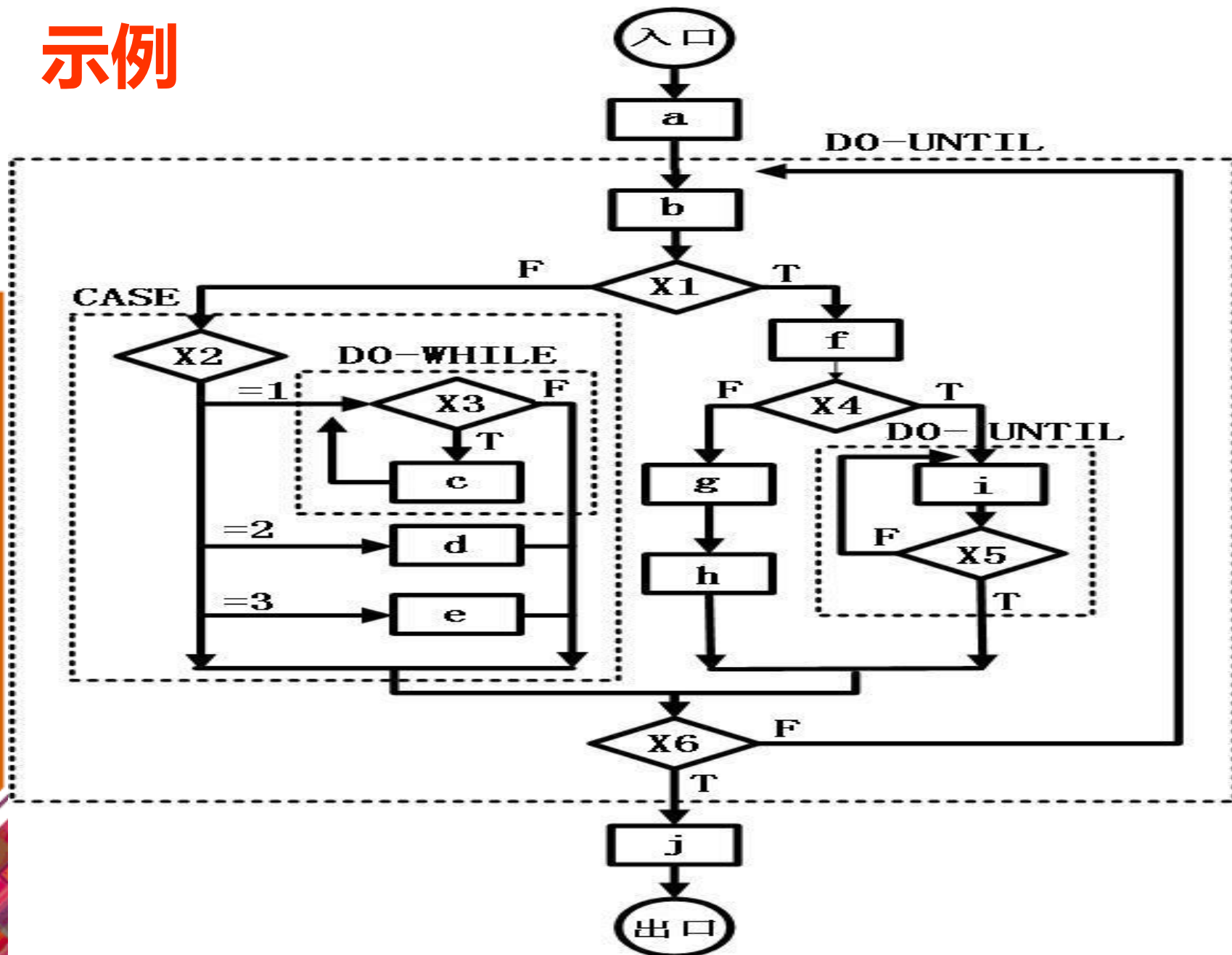
6.3.1 程序流程图

- 程序流程图又称为程序框图，它是历史最悠久、使用最广泛的描述过程设计的方法。
- 它的主要优点是对控制流程的描绘很直观，便于初学者掌握。
- 程序流程图历史悠久，至今仍在广泛使用着。



程序流程图中使用的符号

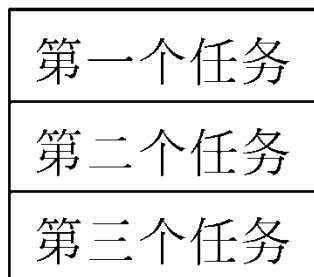
示例



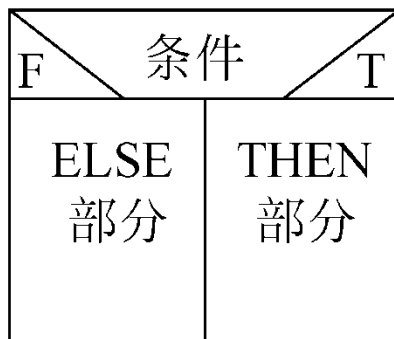
6.3.2 盒图(N-S图)

盒图具有下述特点：

- 功能域明确。
- 不可能任意转移控制。
- 很容易确定局部和全程数据的作用域。
- 很容易表现嵌套关系，也可以表示模块的层次结构。



(a)



(b)



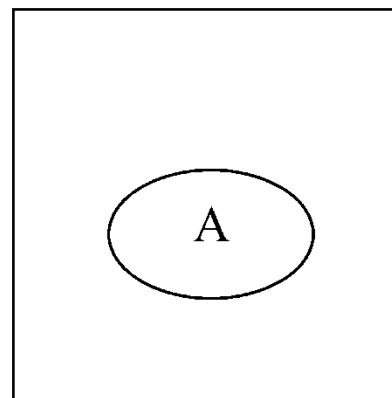
(c)



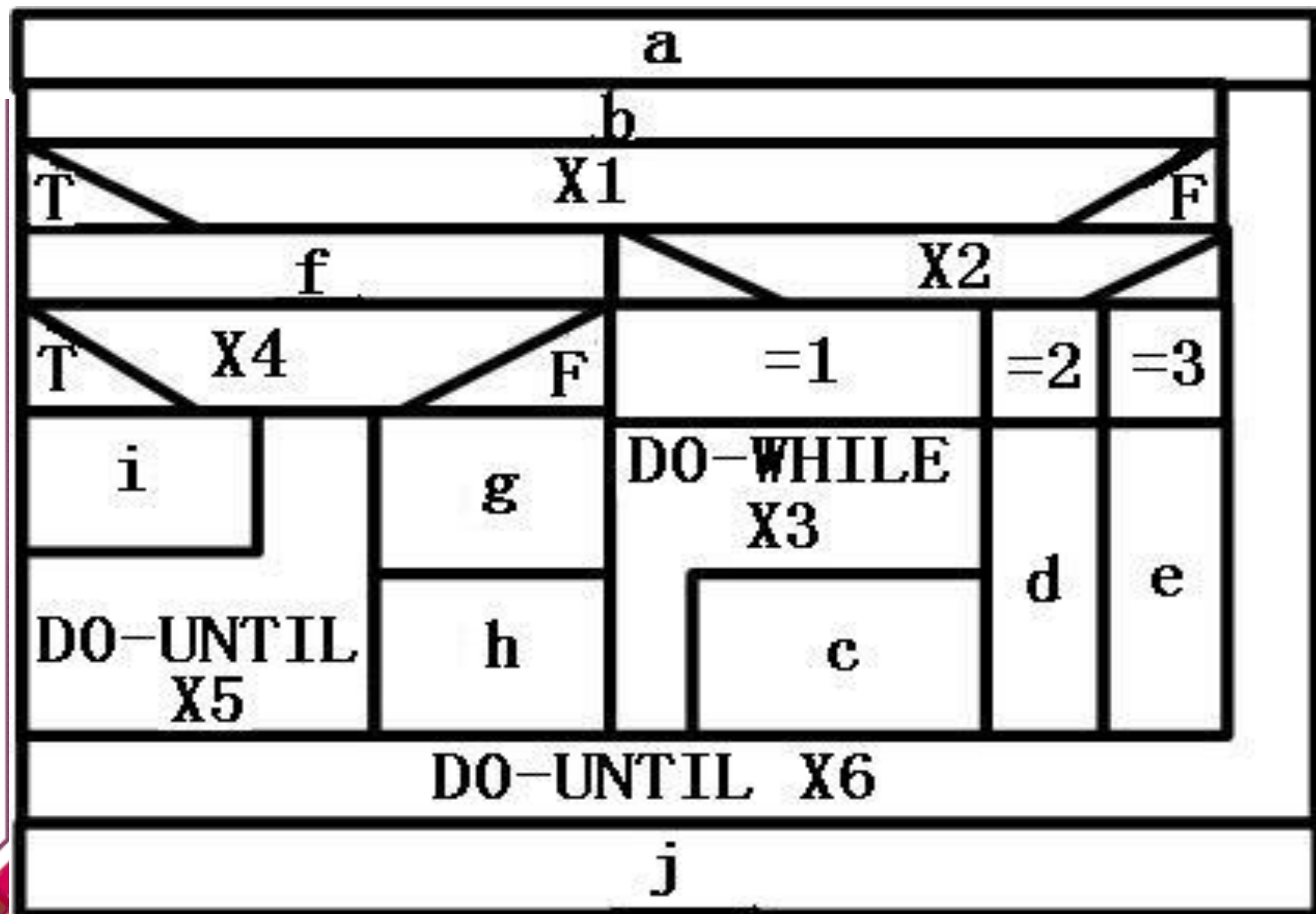
(d)



(e)

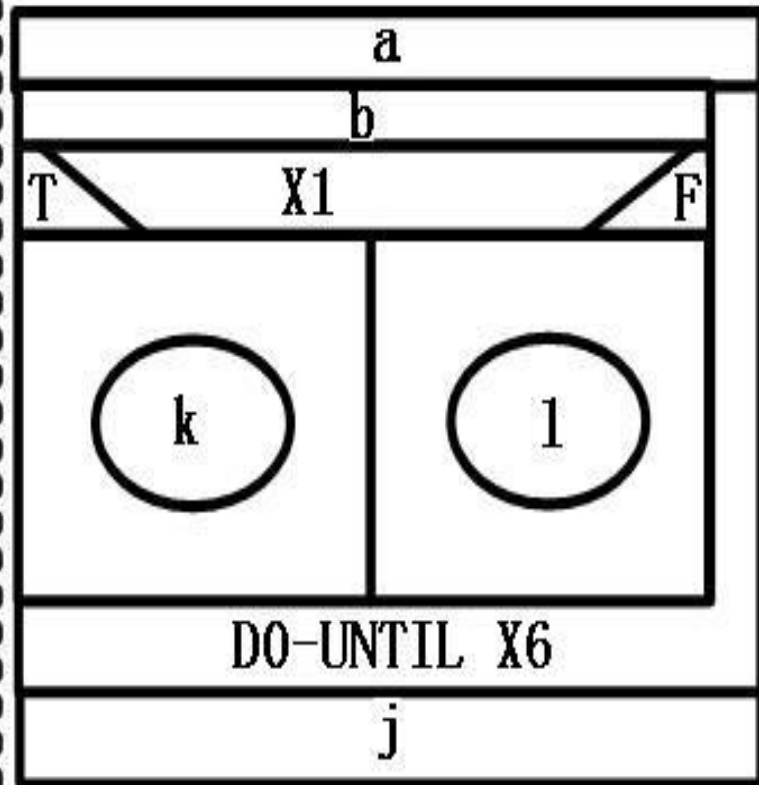


盒图的基本符号

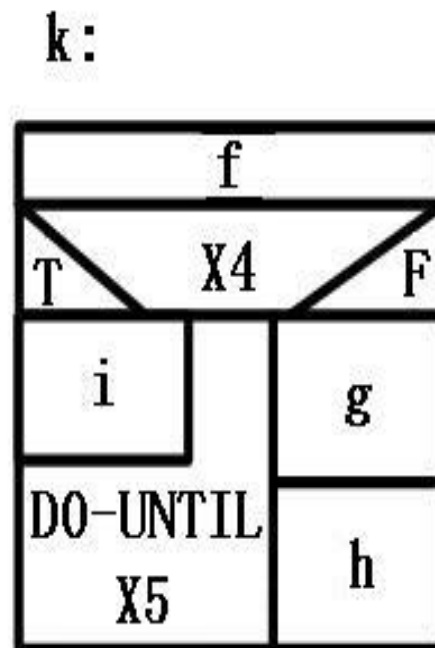


示例

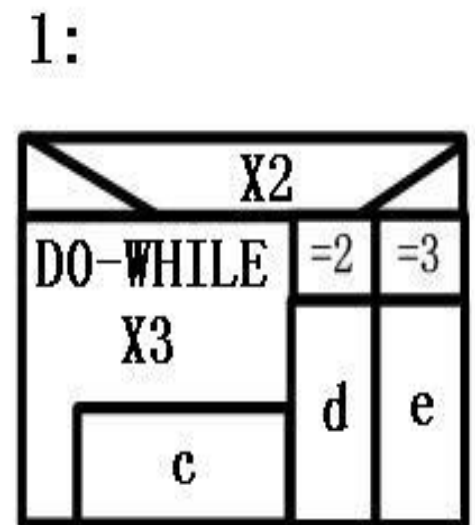
N-S图的嵌套定义形式



(a)



(b)



(c)

■ N-S图的优缺点:

■ 优点:

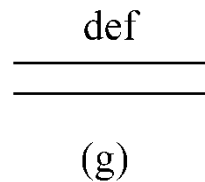
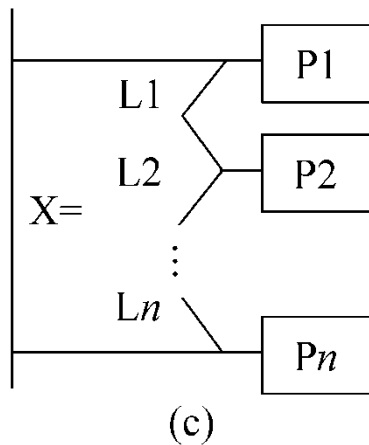
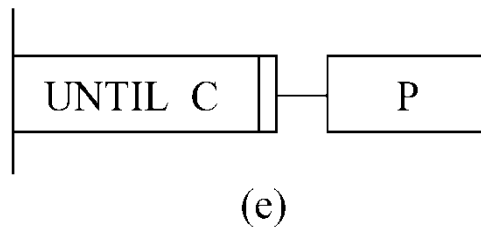
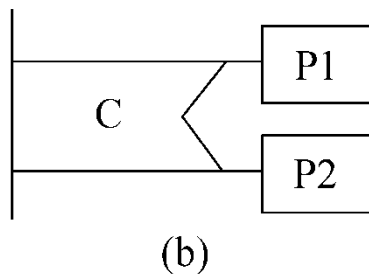
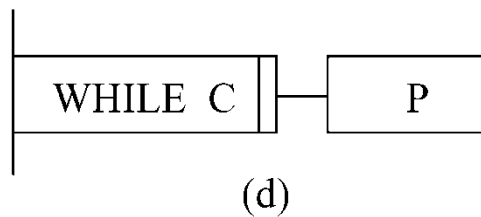
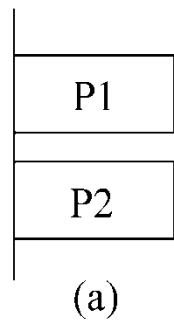
- 功能域（作用域）明确
- 无GOTO
- 容易确定局部或全局数据的作用域
- 容易表示嵌套关系，体现层次结构

■ 缺点:

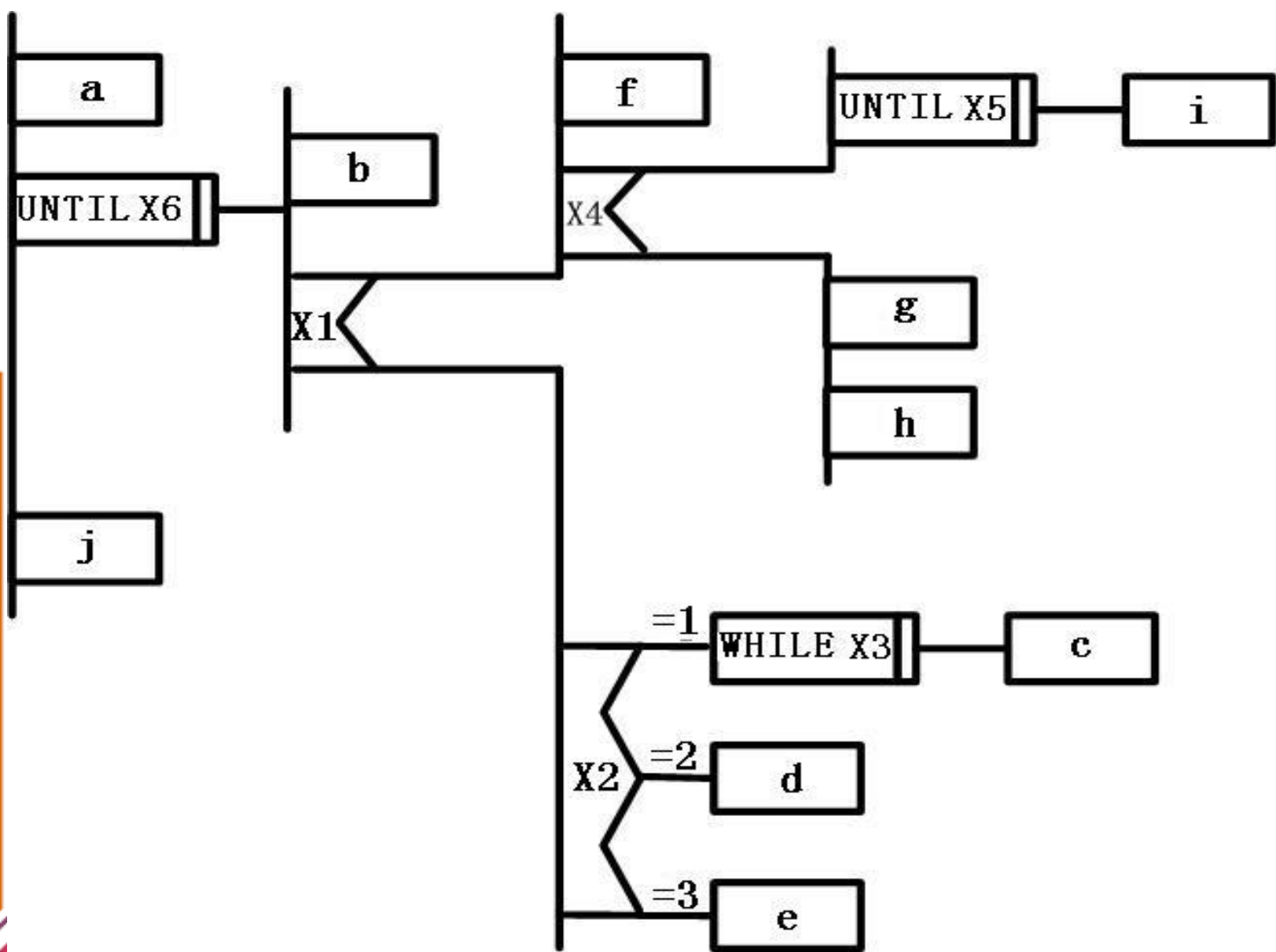
- 修改困难，特别在手工画时
- 分支或嵌套多时不好画

6.3.3 PAD图

- PAD是问题分析图(problem analysis diagram)的英文缩写，自1973年由日本日立公司发明以后，已得到一定程度的推广。
- 它用二维树形结构的图来表示程序的控制流，将这种图翻译成程序代码比较容易。



PAD图的基本符号



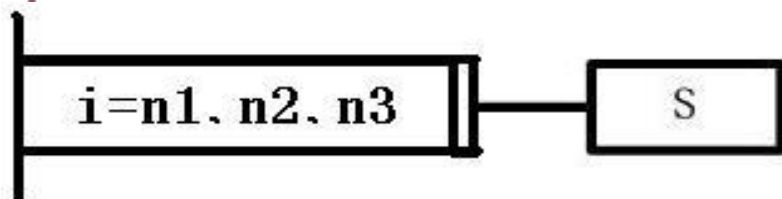
PAD描述的示例

PAD图的主要优点如下：

- 使用表示结构化控制结构的PAD符号设计出来的程序必然是结构化程序。
- PAD图所描绘的程序结构十分清晰。
- PAD图表现程序逻辑易读、易懂、易记。
- 容易将PAD图转换成高级语言源程序，这种转换可用软件工具自动完成。
- 即可表示程序逻辑，也可描绘数据结构。
- PAD图的符号支持自顶向下、逐步求精方法的使用。

对应于增量型循环结构

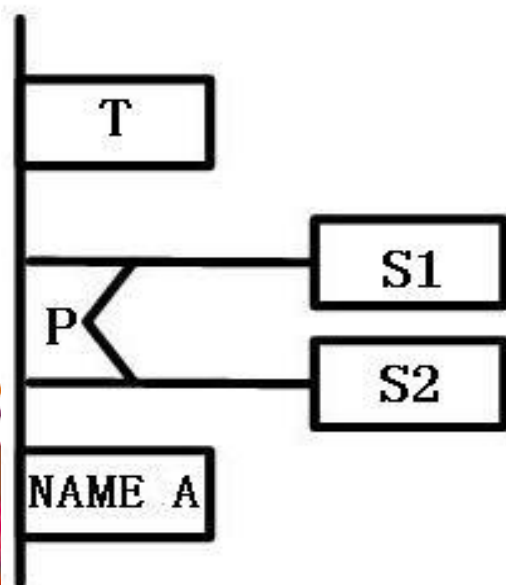
for $i := n1$ to $n2$ step $n3$ do
在**PAD**中有相应的循环控制结构



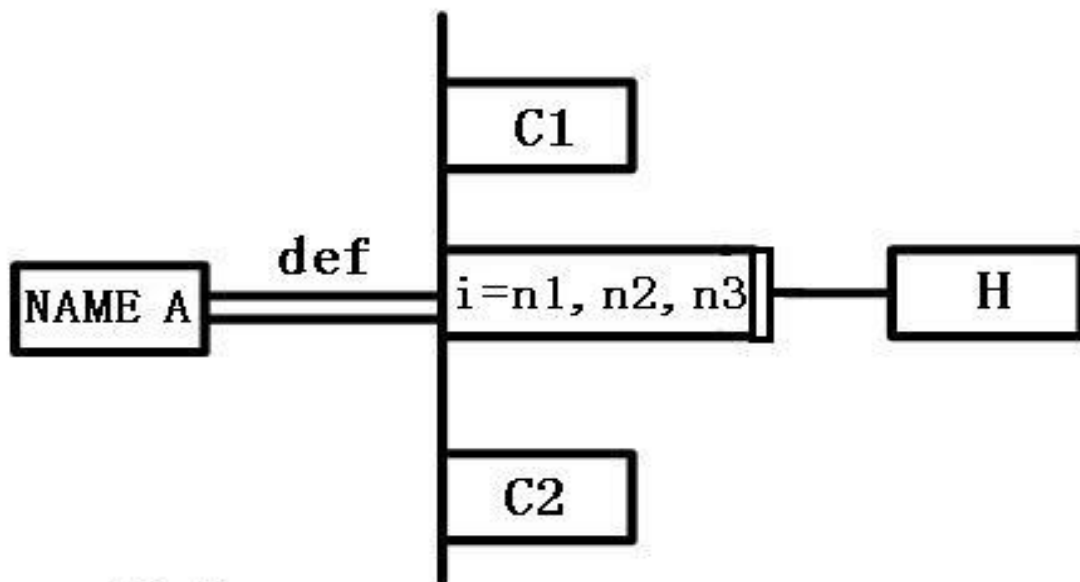
(a) FOR重复型

其中, $i=n1, n2, n3$ 表示
 $i:=n1$ to $n2$ step $n3$

PAD的扩充控制结构

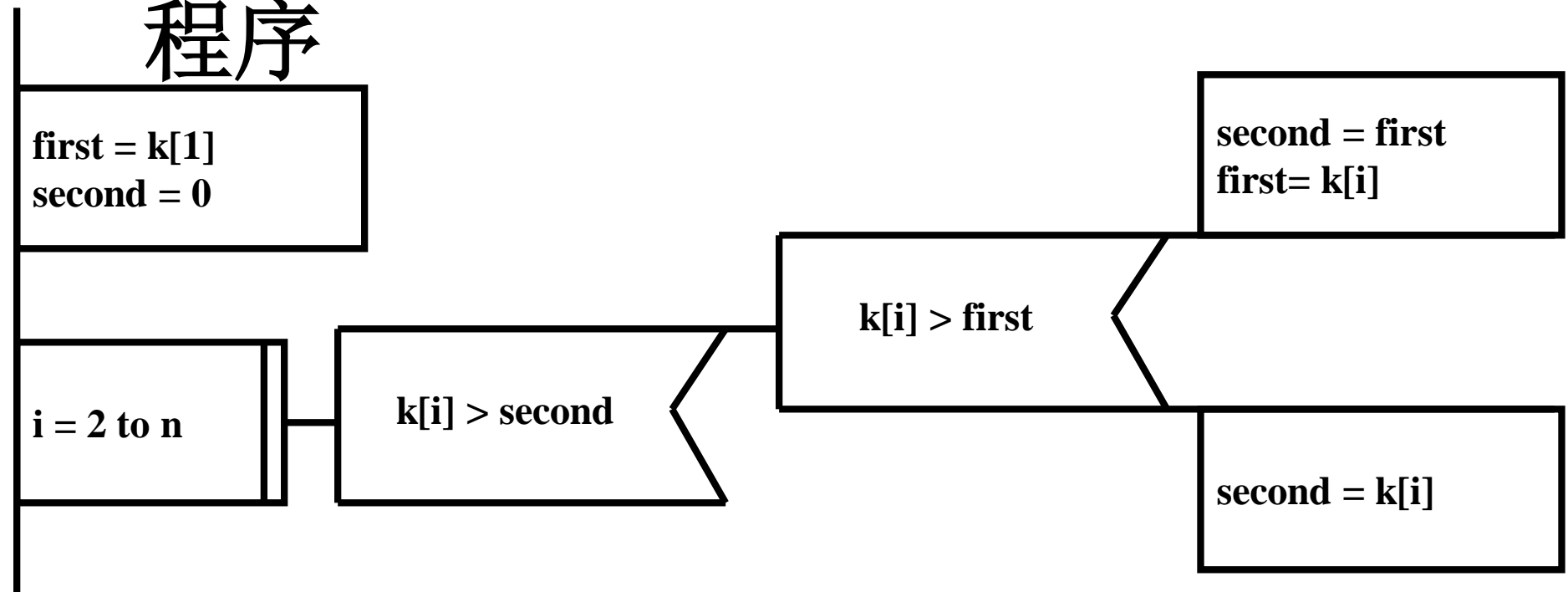


(b) def格式



例如：在数组**K**中找出最大和次大的二个数（假定所有数均大于**0**）

可以通过“走树”（**tree walk**）的方法将**PAD**图转换成高级语言源程序



6.3.4 判定表

- 当算法中包含多重嵌套的条件选择时，用程序流程图、盒图、PAD图或后面即将介绍的过程设计语言(PDL)都不易清楚地描述。
- 判定表却能够清晰地表示复杂的条件组合与应做的动作之间的对应关系。

一张判定表由4部分组成:

- 左上部列出所有条件;
- 左下部是所有可能做的动作;
- 右上部是表示各种条件组合的一个矩阵;
- 右下部是和每种条件组合相对应的动作。

所有条件	条件组合矩阵
所有动作	条件组合 对应的动作

例题：

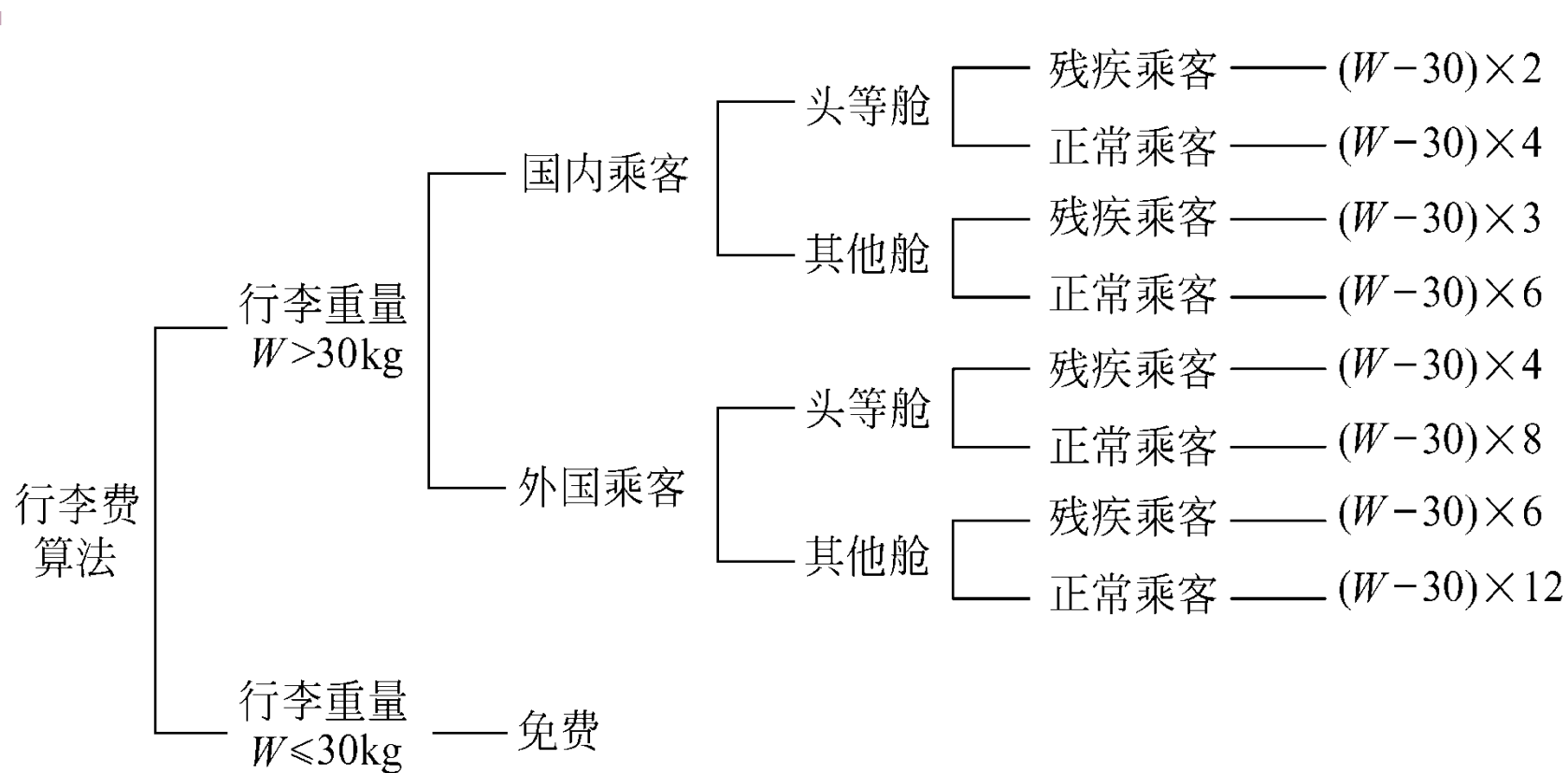
- 假设某航空公司规定，乘客可以免费托运重量不超过30kg的行李。
- 当行李重量超过30kg时，对头等舱的国内乘客超重部分每公斤收费4元，对其他舱的国内乘客超重部分每公斤收费6元。
- 对外国乘客超重部分每公斤收费比国内乘客多一倍，对残疾乘客超重部分每公斤收费比正常乘客少一半。

表6.1 用判定表表示计算行李费的算法

	1	2	3	4	5	6	7	8	9
国内乘客		T	T	T	T	F	F	F	F
头等舱		T	F	T	F	T	F	T	F
残疾乘客		F	F	T	T	F	F	T	T
行李重量 $W \leq 30$	T	F	F	F	F	F	F	F	F
免费	√								
$(W-30) \times 2$				√					
$(W-30) \times 3$					√				
$(W-30) \times 4$		√						√	
$(W-30) \times 6$			√						√
$(W-30) \times 8$						√			
$(W-30) \times 12$							√		

6.3.5 判定树

- 判定树是判定表的变种，也能清晰地表示复杂的条件组合与应做的动作之间的对应关系。
- 多年来判定树一直受到人们的重视，是一种比较常用的系统分析和设计的工具。



用判定树表示计算行李费的算法

6.3.6 过程设计语言

- 过程设计语言(PDL)也称为伪码，它是用正文形式表示数据和处理过程的设计工具。
- PDL具有严格的关键字外部语法，用于定义控制结构和数据结构；另一方面，PDL表示实际操作和条件的内部语法通常又是灵活自由的，可以适应各种工程项目的需要。
- PDL是一种“混杂”语言，它使用一种语言的词汇，同时却使用另一种语言的语法。

伪代码的基本控制结构：

- 简单陈述句结构：避免复合语句。
- 判定结构： **IF_THEN_ELSE**或**CASE_OF**结构。
- 选择结构： **WHILE_DO**或**REPEAT_UNTIL**结构。

例：检查订货单

```
IF 客户订货金额超过5000元 THEN
    IF 客户拖延未还赊欠钱款超过60天 THEN
        在偿还欠款前不予批准
    ELSE (拖延未还赊欠钱款不超过60天)
        发批准书，发货单
    ENDIF
ELSE (客户订货金额未超过5000元)
    IF 客户拖延未还赊欠钱款超过60天 THEN
        发批准书，发货单，并发催款通知书
    ELSE (拖延未还赊欠钱款不超过60天)
        发批准书，发货单
    ENDIF
ENDIF
```

PDL的优点:

- 可以作为注释直接插在源程序中间。有助于保持文档和程序的一致性，提高了文档的质量。
- 可以使用普通的正文编辑程序或文字处理系统，很方便地完成PDL的书写和编辑工作。
- 已经有自动处理程序存在，而且可以自动由PDL生成程序代码。

PDL的缺点:

- 不如图形工具形象直观，描述复杂的条件组合与动作间的对应关系时，不如判定表清晰简单。

小结

- 1、 结构程序设计
- 2、 人机界面设计
- 3、 过程设计的工具