第5章　详细设计

详细设计的目标和任务

详细设计阶段的根本目标：确定应该怎样具体地实现所要求的系统。

详细设计阶段的任务：设计程序的结构“蓝图”，程序员将根据这个蓝图写出实际的程序代码。

详细设计的结果基本上决定了最终的程序代码的质量。

程序的“读者”有两个，那就是计算机和人。对于长期使用的软件系统而言，读程序的时间比写程序的时间长得多。

衡量程序的质量不仅要看它的逻辑是否正确，性能是否满足要求，更主要的是要看它是否容易阅读和理解。

结构程序设计技术是实现上述目标的关键技术，是详细设计的逻辑基础。

详细设计的目标和任务

1 结构程序设计

2 人机界面设计

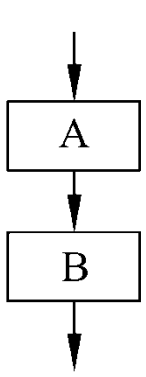
3 过程设计的工具

4 面向数据结构的设计方法

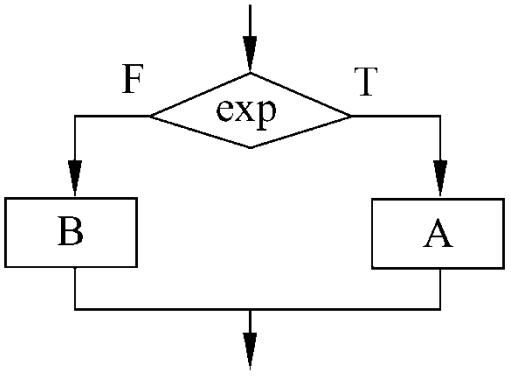
5 程序复杂程度的定量度量

1 结构程序设计

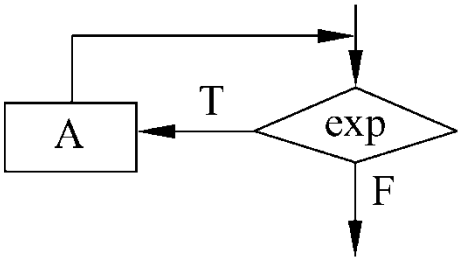
- 1965年结构程序设计的概念最早由E.W.Dijkstra提出：程序的质量与程序中所包含的GO TO 语句的数量成反比
- 1966年Bohm和Jacopini证明了只用“顺序”、“选择”和“循环”控制结构就能实现任何单入口单出口的程序。



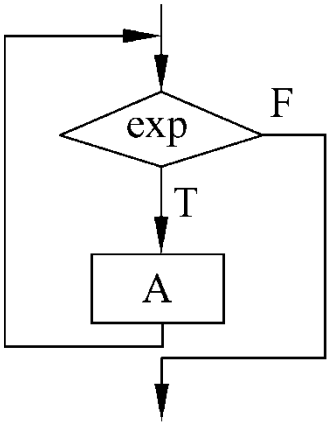
(a)



(b)



或



(c)

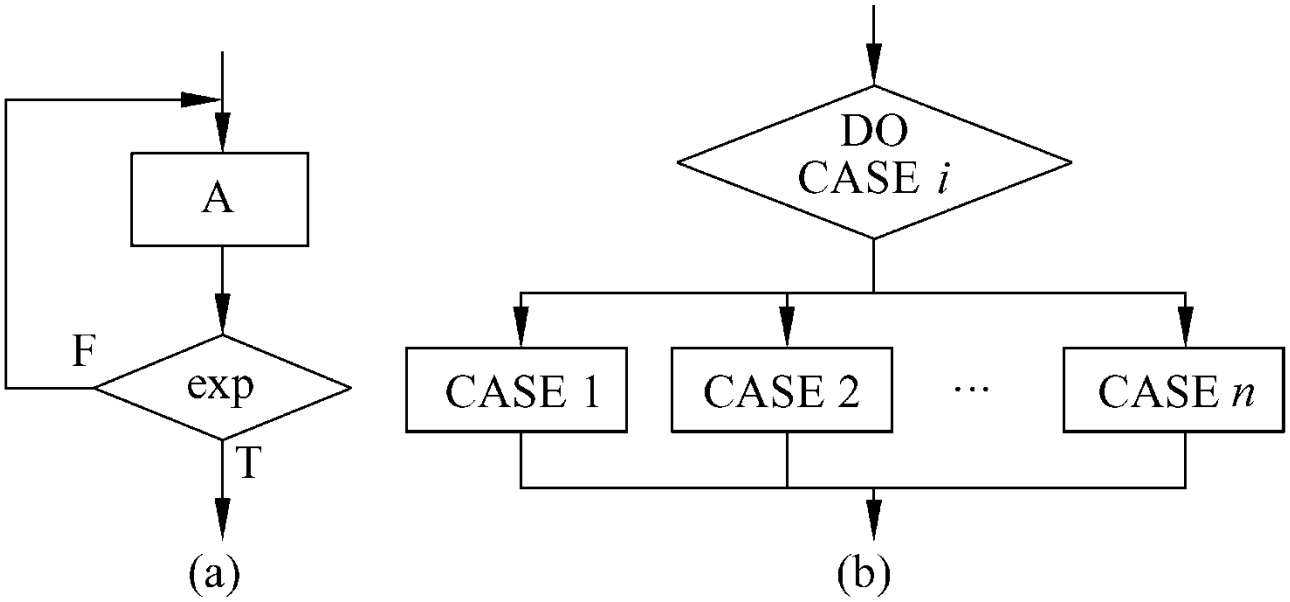
1 结构程序设计

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。

结构程序设计更全面的定义：结构程序设计是尽可能少用GO TO语句的程序设计方法。最好仅在检测出错误时才使用GO TO语句，而且应该总是使用前向GO TO语句。

1 结构程序设计

从理论上说只用上述3种基本控制结构就可以实现任何单入口单出口的程序，但是为了实际使用方便起见，常常还允许使用DO-UNTIL和DO-CASE两种控制结构



1 结构程序设计

如果只允许使用顺序、IF-THEN-ELSE型分支和DO-WHILE型循环这3种基本控制结构，则称为经典的结构程序设计；

如果除了上述3种基本控制结构之外，还允许使用DO-CASE型多分支结构和DO-UNTIL型循环结构，则称为扩展的结构程序设计；

如果再允许使用LEAVE(或BREAK)结构，则称为修正的结构程序设计。

2 人机界面设计

① 系统响应时间。

② 用户帮助设施。

③ 出错信息处理。

④ 命令交互。

2 人机界面设计

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(例如，按回车键或单击鼠标)，到软件给出预期的响应(输出信息或做动作)之间的这段时间。

系统响应时间有两个重要属性，分别是长度和易变性。

1）长度：时间过长，用户就会感到紧张，过短，加快用户操作节奏，可能会犯错误

2）易变性：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 人机界面设计

②用户帮助设施。

常见的帮助设施可分为集成的和附加的两类。

具体设计帮助设施时，必须解决下述的一系列问题。

(1) 在用户与系统交互期间，是否在任何时候都能获得关于系统任何功能的帮助信息?有两种选择：提供部分功能的帮助信息和提供全部功能的帮助信息。

(2) 用户怎样请求帮助?有3种选择：帮助菜单，特殊功能键和HELP命令。

(3) 怎样显示帮助信息?有3种选择：在独立的窗口中，指出参考某个文档(不理想)和在屏幕固定位置显示简短提示。

(4) 用户怎样返回到正常的交互方式中?有两种选择：屏幕上的返回按钮和功能键。

(5) 怎样组织帮助信息?有3种选择：平面结构，信息的层次结构和超文本结构。

2 人机界面设计

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”。一般说来，交互式系统给出的出错信息或警告信息，具有下述属性。

(1) 用用户可以理解的术语描述问题。

(2) 提供有助于从错误中恢复的建设性意见。

(3) 指出错误可能导致哪些负面后果(例如，破坏数据文件)，以便用户检查是否出现了这些问题，并在确实出现问题时及时解决。

(4) 伴随着听觉上或视觉上的提示

(5) 不能带有指责色彩，不能责怪用户。

2 人机界面设计

④ 命令交互。

许多高级用户仍然偏爱面向命令行的交互方式

在提供命令交互方式时，必须考虑下列设计问题。

(1) 是否每个菜单选项都有对应的命令?

(2) 采用何种命令形式?有3种选择：控制序列(例如，Ctrl+P)，功能键和输入命令。

(3) 学习和记忆命令的难度有多大?忘记了命令怎么办?

(4) 用户是否可以定制或缩写命令?

在越来越多的应用软件中，人机界面设计者都提供了“命令宏机制”。

在理想的情况下，所有应用软件都有一致的命令使用方法。

2 人机界面设计

设计过程

用户界面设计是一个迭代的过程，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。

建立起用户界面的原型，就必须对它进行评估，评估可以是非正式的也可以使正式的。

用户界面的评估周期

人机界面设计指南

① **一般交互指南：**涉及信息显示、数据输入和系统整体控制

(1)保持一致性。
(2)提供有意义的反馈。
(3)在执行有较大破坏性的动作之前要求用户确认。
(4)允许取消绝大多数操作。
(5)减少在两次操作之间必须记忆的信息量。
(6)提高对话、移动和思考的效率。
(7)允许犯错误。
(8)按功能对动作分类，并据此设计屏幕布局。
(9)提供对用户工作内容敏感的帮助设施
(10)用简单动词或动词短语作为命令名。

2 人机界面设计

人机界面设计指南

② **信息显示指南：**多种不同方式“显示”信息：用文字、图形和声音；按位置、移动和大小；使用颜色、分辨率和省略。

(1)只显示与当前工作内容有关的信息。

(2)不要用数据淹没用户，应该用便于用户迅速吸取信息的方式来表示数据。

(3)使用一致的标记、标准的缩写和可预知的颜色。

(4)允许用户保持可视化的语境。

(5)产生有意义的出错信息。

(6)使用大小写、缩进和文本分组以帮助理解。

(7)使用窗口分隔不同类型的信息。

(8)使用“模拟”显示方式表示信息，以使信息更容易被用户提取。

(9) 高效率地使用显示屏。

人机界面设计指南

③ **数据输入指南：**用户的大部分时间用在选择命令、输入数据和向系统提供输入。

(1)尽量减少用户的输入动作。

(2)保持信息显示和数据输入之间的一致性。

(3)允许用户自定义输入。

(4)交互应该是灵活的，并且可调整成用户最喜欢的输入方式。

(5)使在当前动作语境中不适用的命令不起作用。

(6)让用户控制交互流。

(7)对所有输入动作都提供帮助。

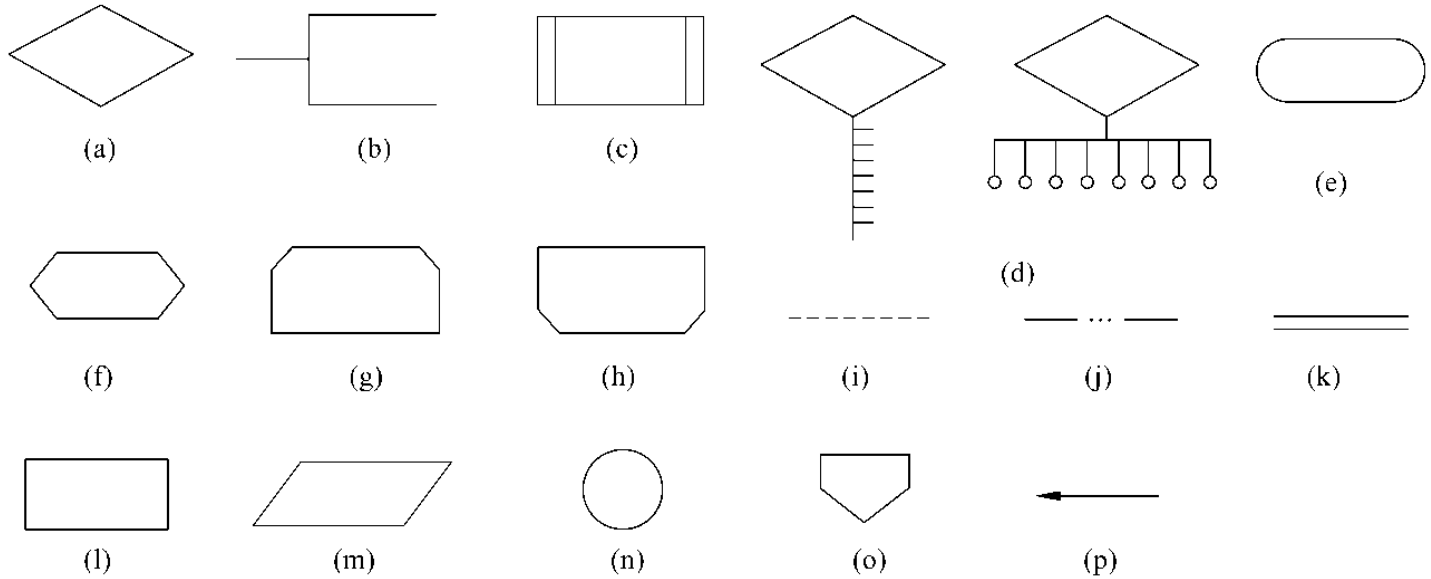
(8)消除冗余的输入。

(9) 高效率地使用显示屏。

3 过程设计的工具

程序流程图

程序流程图又称为程序框图，它是使用最广泛的描述过程设计的方法。程序流程图中使用的符号(a) 选择(分支)； (b) 注释； (c) 预先定义的处理； (d) 多分支； (e) 开始或停止； (f) 准备； (g) 循环上界限； (h) 循环下界限； (i) 虚线； (j) 省略符； (k) 并行方式； (l) 处理； (m) 输入输出； (n) 连接； (o) 换页连接； (p) 控制流



3 过程设计的工具

盒图

出于要有一种不允许违背结构程序设计精神的图形工具的考虑，Nassi和Shneiderman提出了盒图，又称为N-S图。它有下述特点。

(1) 功能域(即一个特定控制结构的作用域)明确，可以从盒图上一眼就看出来。

(2) 不可能任意转移控制。

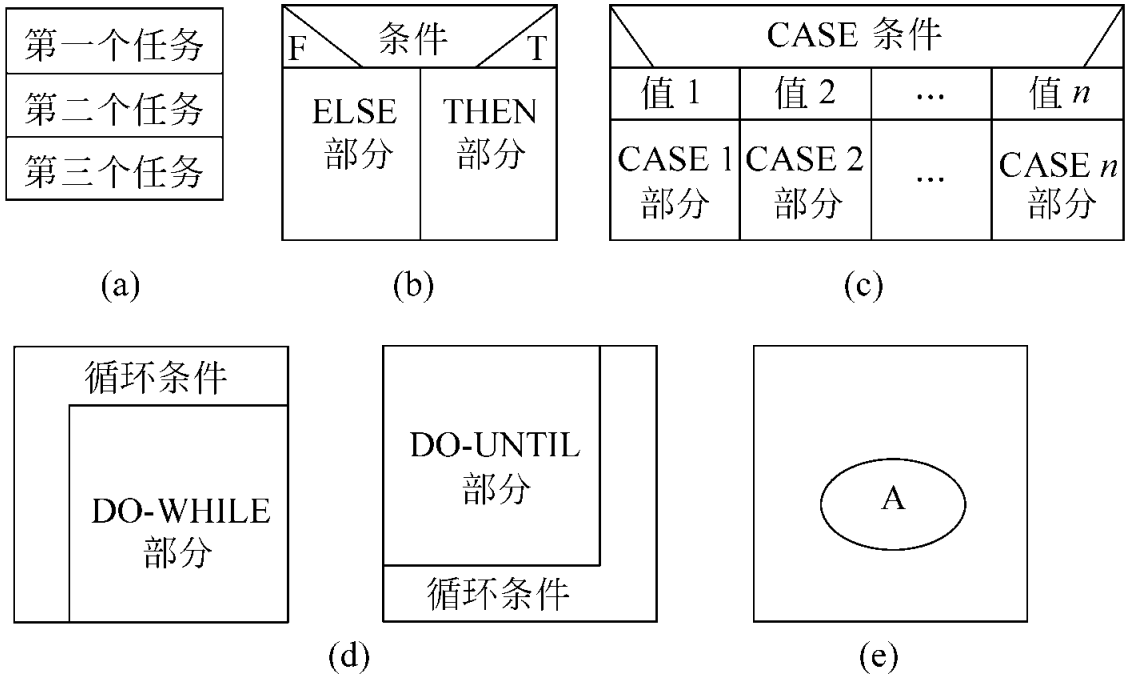
(3) 很容易确定局部和全程数据的作用域。

(4) 很容易表现嵌套关系，也可以表示模块的层次结构。

3 过程设计的工具

盒图

给出了结构化控制结构的盒图表示，也给出了调用子程序的盒图表示方法。其中：基本符号(a) 顺序； (b) IF_THEN_ELSE型分支； (c) CASE型多分支； (d) 循环； (e) 调用子程序A



3 过程设计的工具

其他工具

- **PAD图**
- 判定表
- 判定树