

第 5 章 软件详细设计

【本章引言】

软件设计是软件工程的核心技术。与其他领域的工程设计一样，也需要有好的分析策略和好的方法等。从项目管理的角度来看，软件设计分为总体设计(Preliminary Design)和详细设计(Detail Design)。前面一章介绍了总体设计，总体设计完成的是软件系统的总体设计，规定了各个模块的功能及模块之间的联系，而接下来介绍的软件详细设计也称为模块设计、物理设计，是指体系结构选择阶段之后所进行的技术性的活动。在此期间对每个模块（或类）进行详细设计。详细设计中除了体系结构设计和过程设计之外，许多现代应用系统里还包括人机交互界面设计。换句话说，详细设计主要集中在体系结构表达式的细化，选择详细的数据结构和算法。

【本章重点】

- 软件详细设计的任务及步骤
- 软件详细设计的图形工具
- 软件详细设计的方法

【学习目标】

- 了解软件详细设计的任务及步骤
- 了解 PDL 语言的使用
- 掌握程序流程图、N-S 图、PAD 图、判定树及判定表
- 掌握 Jackson 设计方法和 Warnier 设计方法及其设计过程

5.1 详细设计概述

5.1.1 详细设计的任务

软件设计是一个把软件需求转换为软件表示的过程。这种表示先勾勒出软件的框架模块，然后，为软件结构图中的每个模块确定所必需的算法和详细的数据结构，最终给出程序设计蓝图，也就是所谓的详细设计阶段。从技术的角度划分，详细设计可以分为数据设计、系统结构设计以及过程代码设计。

具体的说，详细设计阶段就是要确定如何具体实现所需求的系统，得到一个接近于源代码的软件表示。这一阶段需要完成的任务，具体地讲有以下几个方面：

(1) 确定为实现系统每个模块功能需求所使用的算法和模块间的控制方式（性能设计）。算法的性能设计评测主要指标包括周转时间、响应时间、吞吐量和精度四个方面。

- (2) 结合算法设计，确定详细的数据结构。
- (3) 分析软件各部分之间的联系，确认接口，复审详细说明书。
- (4) 对详细设计进行评审，为编码阶段做准备。

5.1.2 详细设计的步骤

详细设计从体系结构设计阶段开始，直到得到一张编码阶段之前的详细完整的设计图。第一步，设计领域类和结构类中相关的类和模式。第二步，细化模型以确保拥有一个完整的设计，并为每个类指定必需的属性类型和操作。第三步，使用详细设计描述工具（如程序流程图、N-S图和PAD图等）为每个类的每一个需求指定一个最优的方法。第四步，拟定单元测试计划，确定单元测试的范围和优先级。第五步，评审类、方法的属性（名称、返回值和参数类型）以及多个模型之间的类和方法的关系，记录数据错误的类型及严重性。如图5-1所示，即为软件详细设计过程。

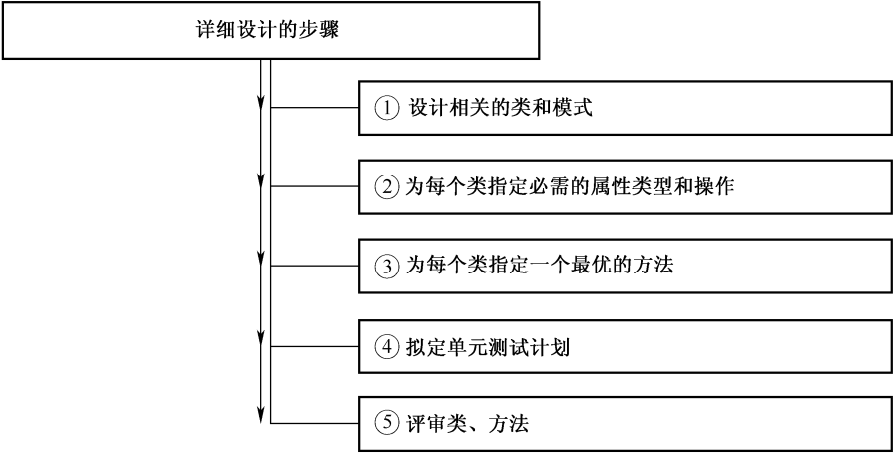


图 5-1 详细设计步骤

5.2 详细设计的图形描述工具

描述程序处理过程的工具称为详细设计的工具，无论哪种工具，对它们的基本要求都是能够提供对设计的准确描述，能够指明控制流程、处理功能、数据组织以及实现细节，从而到了编码阶段就能达到把对设计的描述直接翻译成程序代码。详细设计中的算法描述理论上讲应该采用自然语言来表达，但是“一图胜千言”（一张清晰的图形形象，胜过一千句话的描述）。对于不熟悉软件的人来说，理解算法规格说明时，还要考虑到语法语义以及上下文的过程细节，实在太过复杂。这样，详细设计中就引入了图形描述工具，可以通过图形模式很容易地给出算法过程细节。图形描述工具又称为详细设计工具，目前流行的工具可以分为三类：

- 图形工具：如程序流程图、盒图、PAD图、序列图、数据流图。
- 表格工具：如判定表。
- 语言工具：如程序设计语言PDL。

5.2.1 程序流程图

程序流程图又称为程序框图，是详细设计中最古老、使用最广泛的图形描述工具。是流经一个系统的信息流、观点流或部件流的图形代表。

流程图是由一些图框和流程线组成的，其中图框表示各种操作的类型，图框中的文字和符号表示操作的内容，流程线表示操作的先后次序。其图形绘制简单，最常用的流程图结构是矩形和菱。矩形称为处理，代表一个处理步骤；菱形称为决策，代表一个逻辑判断条件。例如，在企业中，流程图主要用来说明某一过程。这种过程既可以是生产线上的工艺流程，也可以是完成一项任务必需的管理过程。一张流程图能够成为解释某个零件的制造工序，甚至组织决策制定程序的方式之一。这些过程的各个阶段均用图形块表示，不同图形块之间以箭头相连，代表它们在系统内的流动方向。下一步何去何从，要取决于上一步的结果，典型做法是用“是”或“否”或者“真”或“假”的逻辑分支加以判断。

对于流程图所使用的符号，为了统一，国际标准化组织提出并被中国国家技术监督局批准，规定了一些程序流程图标准符号，如图 5-2 所示。

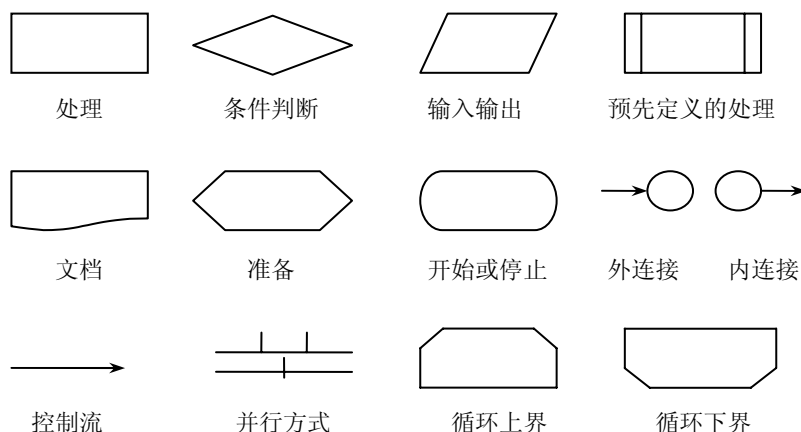


图 5-2 标准程序流程图的规定符号

1966 年 Bohm 和 Jacopini 提出了只用 3 种基本的控制结构就可以实现单入口、单出口的结构程序。这 3 种基本控制结构是顺序型结构、选择型结构以及循环型结构。1972 年 IBM 公司的 Mills 进一步从理论上论证了问题的任何算法都可以综合表示为上述结构化构造。由此，使用流程图描述结构化程序，规定任何复杂的程序流程图都可由以下 3 种基本控制结构组成。

1. 顺序型结构

顺序结构由带箭头的控制线依次连接几个处理方框构成。流程如图 5-3 所示。

2. 选择型结构

选择型结构是流程图最为常用的结构，其结构构造有两种，一种是条件选择结构又称为 If-Then-Else 结构，使用菱形表现逻辑

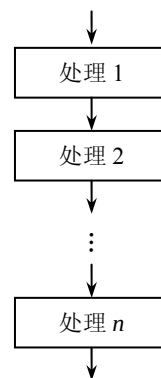


图 5-3 顺序结构流程图

判定条件，条件结果决定选择两个处理方框中的一个。具体的说，条件为真，处理 Then 部分控制线的方框；条件为假，处理 Else 部分控制线的方框。流程如图 5-4 所示。 一种是多向选择结构，又称为 Select-Case 结构，是条件选择结构的形式，列 多种判定 ，一个参数被连 的判定，然后 据控制 量的取值，选择 行其一。流程如图 5-5 所示。

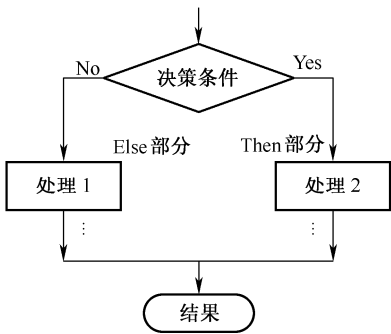


图 5-4 条件选择结构流程图

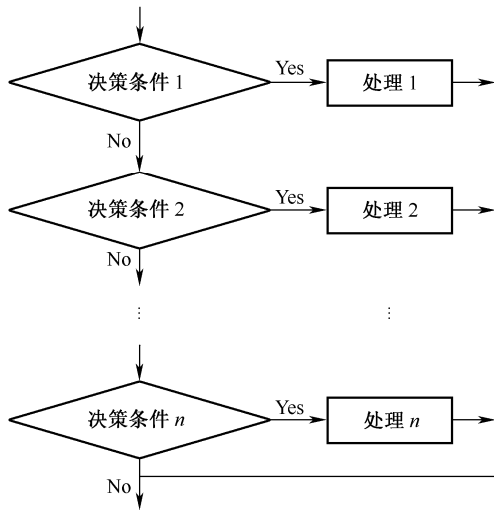


图 5-5 多向选择结构流程图

3. 循环型结构

重复 行某种功能的程序时就用到循环结构，流程图中循环型结构分为两种，一种是先判定型循环，又称为 Do-While 型循环结构，测试一个循环控制条件为真时，就重复 行 定的处理。流程如图 5-6 所示。 外一种是后判定型循环，又称为 Repeat-Until 型循环结构，先重复 行 定的处理，然后测试条件，直到循环控制条件为真时，停止 行。流程如图 5-7 所示。

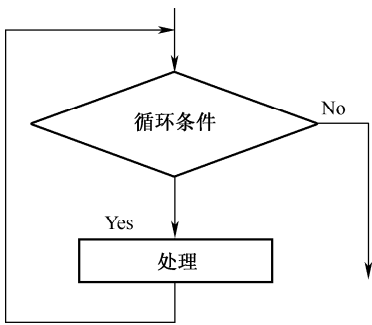


图 5-6 先判定型循环结构流程图

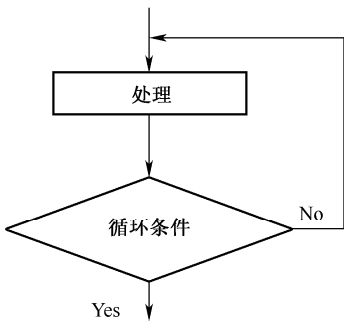


图 5-7 后判定型循环结构流程

以上 3 种结构的流程图通常不 单 使用，都是相 结合和 使用，如图 5-8 所示给出了一个结构化程序流程图。图中 加了一些以 线构成的框， 于理解控制结构的结合 关系。为了 于 地理解结构化流程图的使用，本书 后又给出了一个关于合同评审的实例，详细流程如图 5-9 所示。

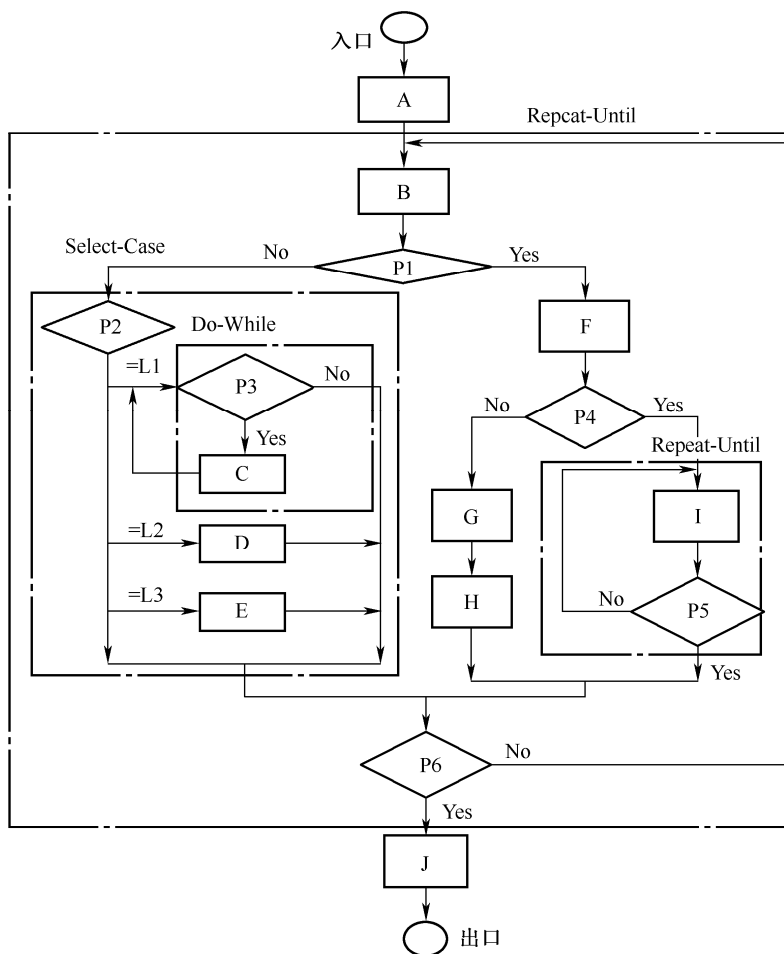


图 5-8 综合结构流程图

5.2.2 N-S 图

Nassi 和 Schneiderman 提出了一种符合结构化程序设计的图形描述工具，称为 N-S 图，又称盒图。其基本元是盒，也就是形象化的矩形，盒图有箭头，此不转控制，解决了流程图中控制流箭头的性的点。盒图具有以下：图中矩形框明确定义功能域；控制流不能够任转控制；很容易确定局部和局作用域；易于表现关系和模块次结构。

盒图也分为 3 种基本控制结构，两个盒（矩形）或多个盒的部部连接来表示顺序结构；条件盒下接一个 Then 盒和一个 Else 盒表示条件选择结构；多向选择结构条件选择结构盒图类，就是 Case 条件盒下接 n 个条件取值盒；循环结构式采用了划界的方式，把 Do-While 循环和 Repeat-Until 循环部分分框于盒的上部和下部表示重复构造的过程。结构复杂时，一个盒图不能够完整地表述细节，可以在图中相应用框出用的从属模块，从属模块部分可以在外的盒图中进一步开表述。具体的盒图的结构化构造的图形表达式如图 5-10 所示。

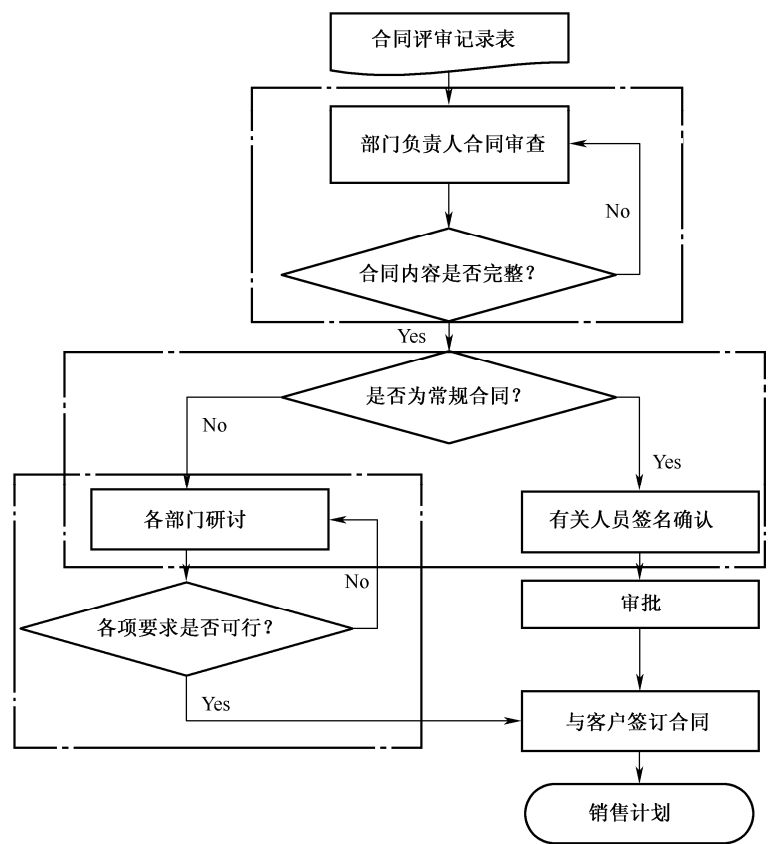


图 5-9 合同评审程序流程图实例

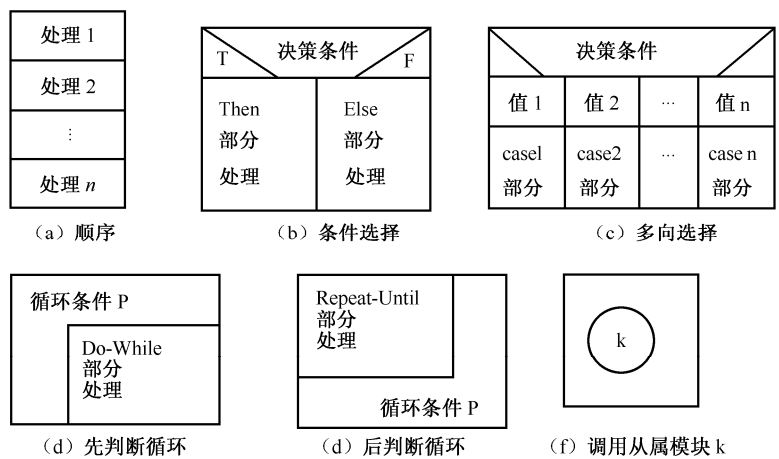


图 5-10 盒图的基本控制结构

为了说明 N-S 图的使用以及和其 图形工具的 ，这 然使用相同的例 （图 5-8 综 合 结构流程图）来描述盒图的表示，如图 5-11 所示。

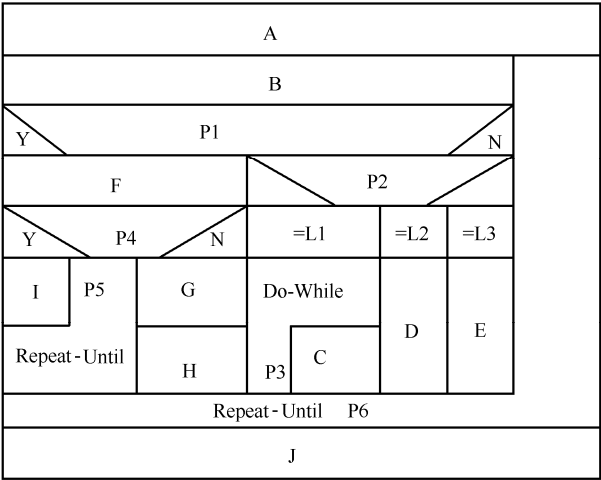


图 5-11 综合结构 N-S 图

5.2.3 PAD 图

PAD 图是问题分析图（Problem Analysis Diagram）的文本形式，它使用二维型结构的图形来描述程序的逻辑，流程图容易理解，结构清晰。PAD 图中从最左线上的节点开始行，自上而下，从及顺序行，所有节点来表示程序的控制流。此外，程序中包含的次数就是 PAD 图的线数，一目了然。

PAD 图是面向高级程序设计语言的，为 BASIC 和 C 语言等常用高级编程语言提供了相应的基本图形图式，每个图式都有相应的控制语句标准模式相对应，有了 PAD 图的基本图式，根据系统的要求就可以画出系统的 PAD 图，然后根据语言的标准模式进行编程就可以很顺利地得到相应的源程序。

PAD 图也分为 3 种控制结构，其结构图式如图 5-12 所示。

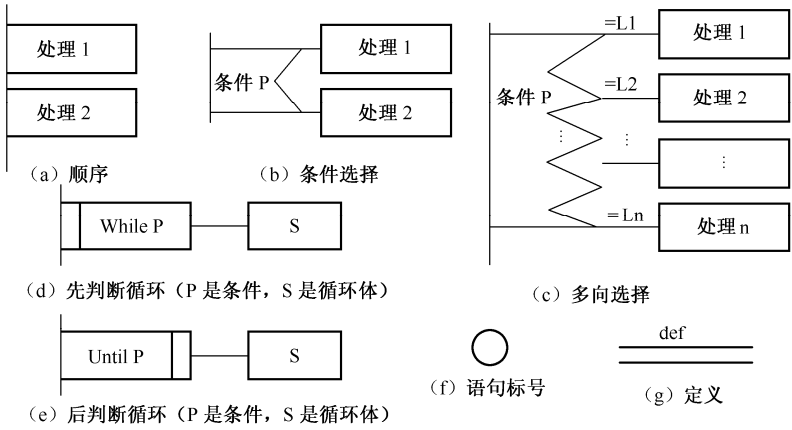


图 5-12 PAD 图的结构的基本图式

PAD 图的行顺序从最左上的节点开始，每到判断选择和循环时，就自上而下进入下一行，从下一行的线上节点开始行，直至该行的下，返回上一行的转

入处。如此 复，直到主 线的最下 节点为止。

作为 PAD 应用的实例，这 然使用相同的例 （图 5-8 综合 结构流程图）来描述 PAD 图的表示，如图 5-13 所示。

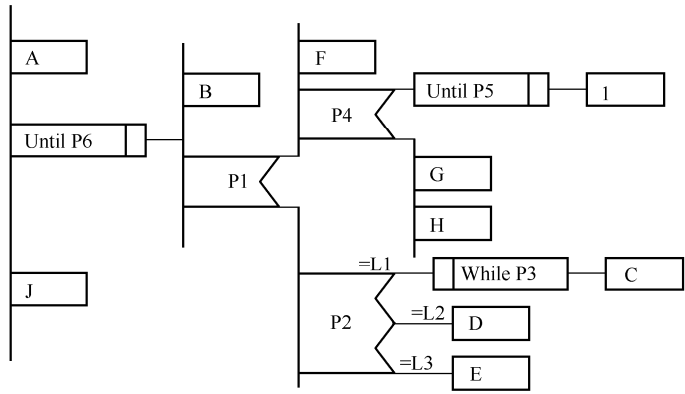


图 5-13 PAD 图式

目前，使用 PAD 图编程有两种方式，一种是 PAD 图由人工 入； 一种是在有 PAD 图系统支 的计算 系统上，直接输入 PAD 图， 即就自动生成语言的源程序。

5.2.4 过程设计语言 PDL

PDL 是过程设计语言 (Program Design Language) 的 文 ，又称为 码 (Pseudocode)，是一种具有 式格式数据表示和处理过程的设计工具， 级语言，用于描述功能模块的算法设计和加工细节。它采用某种语言 (如 语、 语等) 的 ，同时采用 一种结构化编程语言的 部语法，并把说明性的文字直接 入到 PDL 语句 。

码的语法规 一 分为“外语法”和“内语法”，PDL 就是这样的 码，它具有严格的关 字外部语法，用于定义控制结构和数据结构；同时表示实际操作和条件的内部语法又是自由的，可以使用自然语言 ， 应各种工程项目的需要。

PDL 作为一种用于描述程序设计逻辑的语言，可以是语言的简化 本， 如 PDL/C，但是无论 样，都应该具备以下 ：

- 关 字必 具备 定语法，以 提供了结构化控制结构、数据说明和模块化的 点。为了使结构清晰和可 性 ， 通常在所有可能 使用的控制结构的头和 都有关 字等。
- 内语法使用自然语言描述处理 性，语法自由。
- 数据说明的 段。应该既包括简单的数据结构 (如标量和数组)，又包括复杂的数据结构 (如 表或 次的数据结构)。
- 模块 程序的定义和 用的技术，提供各种接口描述。

以下引用一个很多 所使用的 PDL 的实例， 引 。

例如： 错单 的程序。

Procedure spell check is
Begin


```

split document into single words
look up words in dictionary
display words which are not in dictionary
treat a new dictionary
End spell check
-----
Procedure      错单
Begin
  把整个文件分 成单
    字典 到这些单
    示字典中 不到的单
    一个 字典
End
-----

```

由此例的 Procedure、Begin-End 不难看出，PDL 的基本语法是有具体规定的，至应该包括：数据说明、块构造技术、程序定义、条件构造、循环构造、接口描述和 I/O 构造。接下来，就来具体 PDL 的格式和语义。

1. 数据说明

在 PDL 程序中它的功能是指明数据的类型及作用域。

其格式为：Type< 量名>Is< 定 1>< 定 2>

说明：量名是一个模块内部使用的量或模块间用的局量；

< 定 1>标明数据类型，如 SCALAR 标量、ARRAY 数组、LIST 列表、STRING 字符、STRUCTURE 结构；< 定 2>标明该量的作用域。

例如，Type number Is STRING LENGTH(12)。

2. 块程序结构

PDL 的过程元是由块结构构成的，而块作为一个单个的实体来行。

其格式为：Begin<程序块名>

<PDL 语句>

End

3. 输入/输出描述

PDL 中输入/输出说明语句分，化也很多。

其格式为：Read/Write To <设备> <I/O 表>

或者 Ask < 问>Answer <响应选项>

说明：<设备>表示理的 I/O 设备，如、；

<I/O 表>表示被的量。

例如，Ask "select the target direction" Answer "left","right"。

I/O 输入输出描述通常可以，如输出、面向口、下单界面等。

4. 子程序结构

把 PDL 中的过程称为程序。

其格式为：Procedure< 程序名><属性>

Interface<参数表>

< PDL 语句>

End

说明：属性描述了它的引用 以及其 相关属性。Interface 是用来确定模块的自 量参数表，包括所有输入输出信息的标 符。

5. 基本控制结构

用 PDL 表示的程序结构一 有以下几种：

(1) 顺序结构：语句 顺序先后 行。

(2) 条件选择结构：PDL 的条件构造，采用 If-Then-Else 的形式。

其格式：If <决策条件>

Then< PDL 语句或处理 S1>

Else

< PDL 语句或处理 S2>

Endif

决策条件为真值时， 行 Then 后 的语句；决策条件为假值时， 行 Else 后 的语句。If-Then-Else 是可以 使用的。

下面以某系统主控模块详细设计为例，说明如何用 PDL 来描述。

Procedure 模块名()

清 ；

示 xx 系统用 界面；

Put(" 输入用 口 ： ")；

Get(password)；

If password<>系统口 Then

提示 信息；

出 行

Endif

示本系统主 单；

While(true)

接 用 选择 ABC；

If ABC=" 出"

Break；

Endif

用相应下 模块完成用 选择功能；

Endwhile

清 ；

Return

End

(3) 多向选择 Case 结构：条件很多的 下，不能 依 真值和假值来判断语句 行方向，或者 If 的一组语句， 选择应用 Case 结构。

其格式：Case Of <case 量名>；

```

When <case 条件 1>Select <PDL 语句>;
When <case 条件 2>Select <PDL 语句>;
    :
When <case 条件 n>Select <PDL 语句>;
Default:  <PDL 语句>
Endcase

```

此种结构 先 对的是一组条件来测试一个 的参数 量，如果 一个条件，就可
以 行 此条件对应的 PDL 语句。

(4) 重复结构:

PDL 的重复结构包括后测试循环 (PostTest Loops)、先测试循环 (Pretest Loops) 和下标
循环 (Index Loops) 3 种循环。

1) 后测试循环 (Posttest Loops)

其格式: Repeat Until <循环条件>

<PDL 语句>

Endrep

或者 Do Loop

<PDL 语句>

Exit When<循环条件>

End Loop

2) 先测试循环 (Pretest Loops)

其格式: Do While <循环条件>

<PDL 语句>

Enddo

3) 下标循环也称为 引 量型循环 (Index Loops)

其格式: Do For <下标=下标列表、表达式或者序列>

<PDL 语句>

Endfor

代码 例如下:

```

Begin
Enter a vector
Set Maximum to the value of the first element
    in the vector
Do For each second one to the last
If value of element is greater than the Maximum value
    Then Set Maximum to the value of the element
Endif
Endfor
Print the Maximum value
End

```

```

Input arrayA
Max=A(1)
Do For i=2 to n
IF Max<A(i)
Set Max=A(i)
Endif
Endfor
Print Max

```

从以上例 可以 到 PDL 的 体结构 一 程序完 相同。外语法同相应程序语言一 ，内语法使用自然语言，易编 ，易理解，也很容易转换成源程序。

PDL 作为一种设计工具有如下一些优点：

可以作为 释直接 在源程序中间。这样做能 使 人 在 程序代码的同时也相应地 PDL 释， 此有 于保 文档和程序的一 性，提 了文档的 量。

可以使用 通的 文编辑程序或文字处理系统，很方 地完成 PDL 的书 和编辑工作。

经有自动处理程序 在，而 可以自动由 PDL 生成程序代码。

然 PDL 不是详细设计的 一工具，但是在 现设计中的 ，容易生成文件以及动方 方面具有不可 拟性。而 PDL 的 点 是不如图形工具形象直观，描述复杂的条件组合 动作间的对应关系时，不如判定表清晰简单。

5.2.5 判定表和判定树

1. 判定表

判定表（Decision Table）也是描述加工的一种图形工具， 表格形。 描述数据流中的加工时，如果判断的条件 多，各条件又相 组合，相应采取的 （策 ） 多，在这种 下用程序流程图、盒图、PAD 图或者 PDL 都不易清 描述，然而采用判定表来描述这类处理逻辑 合 ，它表达清晰、简 。

判定表的编制， 先要明确加工处理的功能 目标，然后要 响策 的各项 即条件，列出这些 可能出现的 ，并制定出判定的规 。

判定表 分四 部分，如图 5-14 所示：

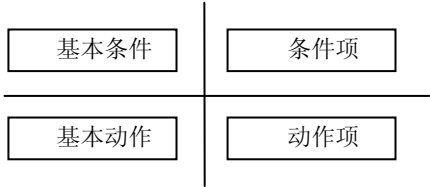


图 5-14 判定表组成图

判定表由四部分组成。第一部分即 表示的部分，判定表的 上部称为基本条件项，列出各种可能的条件。第二部分即 表示的部分，判定表的 上部称为条件项，它列出了各种可能的条件组合。第三部分即 表示的部分，判定表的 下部称为基本动作项，它列出了所有的操作。第四部分即 表示的部分，判定表的 下部称为动作项，它列出在对应条件组合下所选的操作。

以下面 生的 评定为例，说明判定表的组织方法和应用。 的目的在于 生的 优，此处理功能是要合理确定 评定等级。决定 的条件为：成 优 70%或 50%以上，成 为中或中以下 15%或 20%以下， 结 为优 或一 者。 方 为一等 、二等 、三等 、 四种。 为 条件有些是相容的，相 组合的项 多。描述此 生 策的判定表如表 5-1 所示。

表 5-1 学生奖励政策的判定表

条件	程 各 成	优 70%	Y	Y	Y	Y	N	N	N	N
		优 50%					Y	Y	Y	Y
		中以下 5%	Y	Y	N	N	Y	Y	N	N
		中以下 20%			Y	Y			Y	Y
方	结 评分	优	Y	N	Y	N	Y	N	Y	N
		一	N	Y	N	Y	N	Y	N	Y
方	一 等 二 等 三 等		*							
				*	*		*			
						*		*	*	
										*

由表 5-1 可 见，判定表 能 将 复 杂 的 决 策 问 题 简 单 、 明 确 、 一 目 了 然 地 描 述 出 来 ， 它 是 描 述 条 件 多 的 决 策 问 题 的 有 效 工 具 。 判 定 表 能 够 把 一 定 条 件 下 系 统 应 做 的 动 作 准 确 无 误 的 表 示 出 来 ， 但 是 不 能 描 述 循 环 的 处 理 性 ， 循 环 处 理 还 需 结 构 化 语 言 。

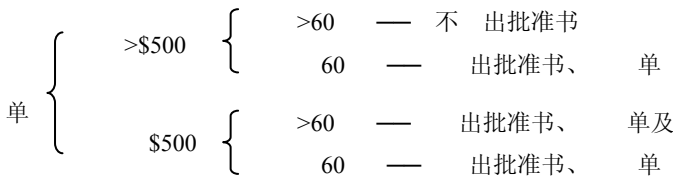
2. 判定树

判定 树 又 称 决 策 树 ， 是 一 种 描 述 加 工 的 图 形 工 具 ， 合 适 描 述 问 题 处 理 中 具 有 多 个 判 断 ， 而 且 每 个 决 策 条 件 有 关 。 使 用 判 定 树 进 行 描 述 时 ， 应 该 从 问 题 的 文 字 描 述 中 分 清 哪 些 是 判 定 条 件 ， 哪 些 是 判 定 的 决 策 ， 根 据 描 述 树 中 的 连 接 线 找 出 判 定 条 件 的 从 属 关 系 、 并 列 关 系 、 选 择 关 系 ， 根 据 它 们 构 造 判 定 树 。 判 定 树 本 质 上 同 判 定 表 是 一 样 的 ， 使 用 不 易 接 受 判 定 表 这 种 描 述 方 式 时 ， 可 以 用 判 定 树 的 形 式 ， 判 定 树 是 一 种 图 形 表 示 ， 易 于 被 理 解 。 下 面 给 出 判 定 表 如 表 5-2 所 示 。

表 5-2 货单操作判定表

规则		1	2	3	4
条件	单	>\$500	>\$500	\$500	\$500
		>60	60	>60	60
操作	不 出 批 准 书	√			
	出 批 准 书		√	√	√
	出 单		√	√	√
	出			√	

其对应的判定 树 如下：



这一判定 树 文 字 描 述 ， 使 人 一 目 了 然 ， 清 晰 地 表 达 了 在 何 种 条 件 下 采 取 何 种 策 略 ， 不 易 产 生 逻 辑 上 的 混 淆 。 而 判 定 表 是 描 述 基 本 处 理 逻 辑 功 能 的 有 效 工 具 。

判定表或判定 都是以图形形式描述数据流的加工逻辑，它结构简单，易 易 。 其 到组合条件的判定， 用判定表或判定 可以使问题的描述清晰，而 于直接 到程序代码。在表达一个加工逻辑时，判定 、判定表都是 的描述工具， 据需要可以 使用。

5.3 Jackson 设计方法

5.3.1 Jackson 方法概述及其图例

1975 年，M.A.Jackson 提出了一种至 广泛使用的软件开 方法。这一方法从目标系统的输入、输出数据结构入 ， 出程序框架结构， 其 细节，就可得到完整的程序结构图。这一方法对输入、输出数据结构明确的中 型系统 有 ，如 业应用中的文件表格处理。该方法也可 其 方法结合，用于模块的详细设计。

Jackson 方法有时也称为面向数据结构的软件设计方法，简称 JSP 方法。Jackson 方法定义了一组以数据结构为指 的过程，它 据输入、输出的数据结构， 一定的规 成软件的过程描述，即程序结构。

Jackson 方法把问题分解为可由 3 种基本结构形式表示的各部分的 次结构。3 种基本的结构形式就是顺序、选择和重复。3 种数据结构可以进行组合，形成复杂的结构体系。

Jackson 系统开 方法（JSD，Jackson System Development）是一种典型的面向数据结构的分析设计方法。Jackson 系统开 方法的系统模型就是相 通信的一组进程的 合。进程间的通信方式有以下 3 种：

- 进程同步 生。
- 通过数据通 /接 动 生。
- 问公用 信息。

Jackson 图对于种类 多的程序中使用的数据结构，各数据元 之间的逻辑关系只有顺序、选择、重复 3 种，所以逻辑数据结构也只有 3 种，如图 5-15 所示。

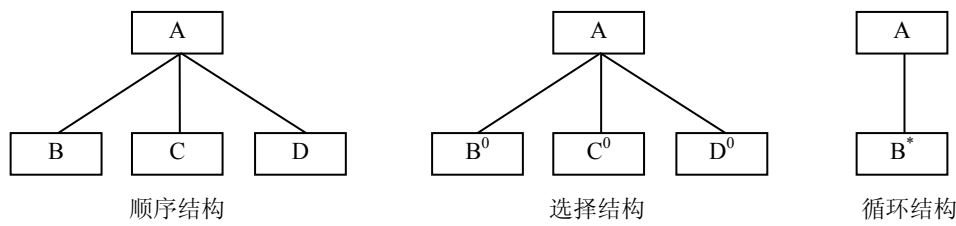


图 5-15 Jackson 图的 3 种结构

- 顺序结构。
- 选择结构。
- 循环结构。

Jackson 图的特点是：用这种图形工具表示选择或重复结构时，选择条件或循环结 条件不能直接在图上表示出来， 响了图的表达能 ，也不易直接把图翻译成程序，此外，框间连线为 线，不易在行式 上输出。以下是设计的 进的 Jackson 图，如图 5-16 所示。

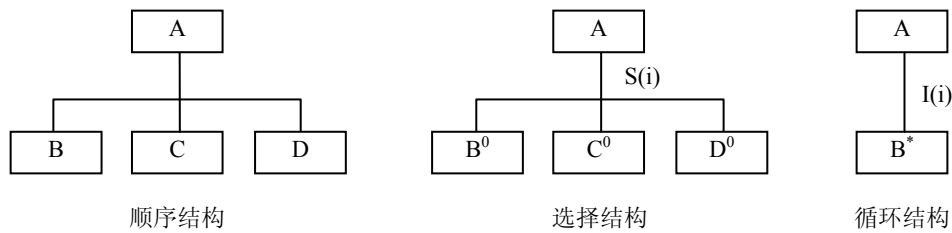


图 5-16 基本的 Jackson 图

例描述，使用 Jackson 图表示数据结构。用 Jackson 图表示如表 5-3 所示的二维表格。

表 5-3 学生点名册

表头	名	性	级	号
表体	⋮	⋮	⋮	⋮

这个 Jackson 图 先 明了该 生名 表格由表头和表体两部分组成。其中表头又顺序包括表名和字段名。而表体可由任 行（0 行或多行）组成，每行包括 生的 名、性 、 级和 号。 级是本 生的， 号项是本 生 号； 级是 生的， 号项是 生 号。

例如，要用 Jackson 图表示产生上面的 生名 文件的程序的程序结构：把 生名 生成为一个计算 文件， 该程序结构可以用如图 5-17 所示的 Jackson 图来表示。

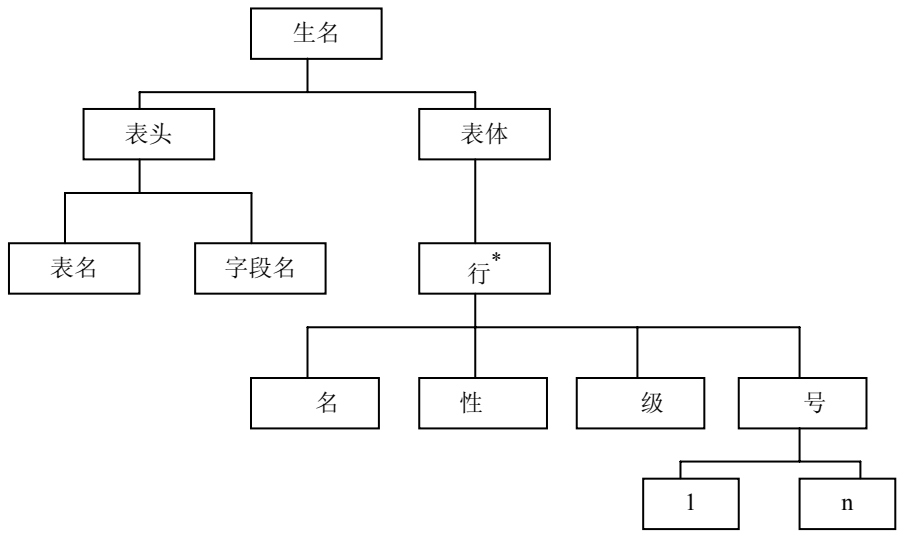


图 5-17 用 Jackson 图表示 生名

Jackson 代码表示方法有 3 种，对应的 Jackson 结构图如图 5-18 所示。

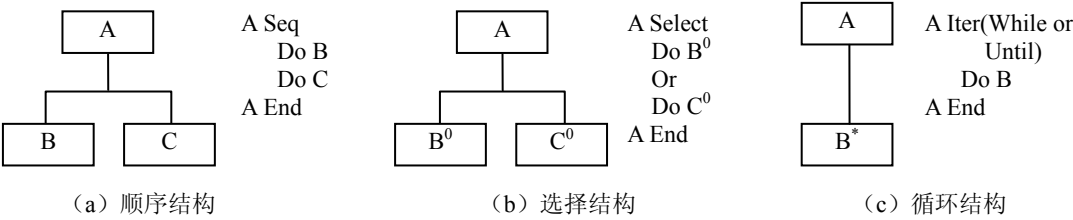


图 5-18 三种结构 Jackson 图表示

1. 顺序结构

顺序结构的代码如下，其中 Seq 和 End 是关键字：

```
A Seq
B
C
D
A End
```

2 选择结构

Select、Or，和 End 是关键字，Cond1、Cond2 和 Cond3 分别是行 B、C 或 D 的条件。

选择结构对应的代码如下：

```
A Select Cond1
B
A Or Cond2
C
A Or Cond3
D
A End
```

3 循环结构

Iter、Until、While 和 End 是关键字（循环结构有 Until 和 While 两种形式），Cond 是条件，重复结构对应的代码图如下：

```
A Iter Until（或 While） Cond
B
A End
```

下面以成绩单为例描述 Jackson 图的设计方法。

输入数据结构的 Jackson 图如图 5-19 所示。

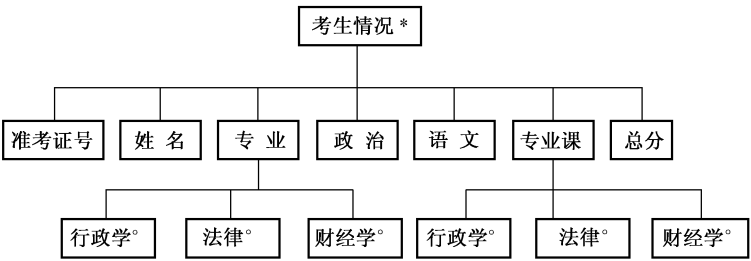


图 5-19 输入数据结构的 Jackson 图表示

成绩单样式和输出数据结构的 Jackson 图如图 5-20 所示。

字符 1:	考试成绩单						
字符 2:	准考证号	名	业		语言	法	分
字符 3:	准考证号	名	业		语言	行	分
字符 4:	准考证号	名	业		语言	经	分
字符 5:							

图 5-20 (a) 考试成绩单样式

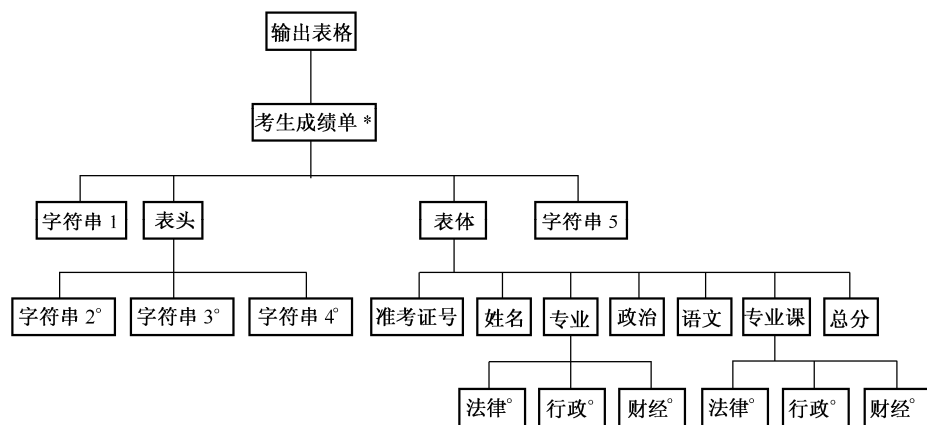


图 5-20 (b) 输出数据结构的 Jackson 图表示

输入、输出数据结构的 Jackson 图对应关系如图 5-21 所示。

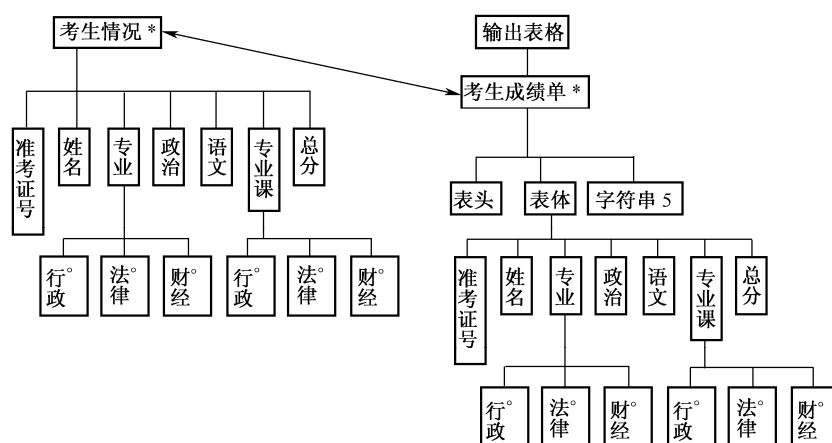


图 5-21 输入、输出数据结构的 Jackson 图对应关系表示

后 输入、输出数据结构关系 出程序结构 Jackson 图。

5.3.2 Jackson 程序设计过程

Jackson 方法一 通过以下五个步骤来完成设计：

- (1) 分析并确定输入数据和输出数据的逻辑结构，并用 Jackson 结构图来表示这些数据结构。
- (2) 出输入数据结构和输出数据结构中有对应关系的数据单元。
- (3) 以下的规 由输入、输出的数据结构 出程序结构，如图 5-22 所示。

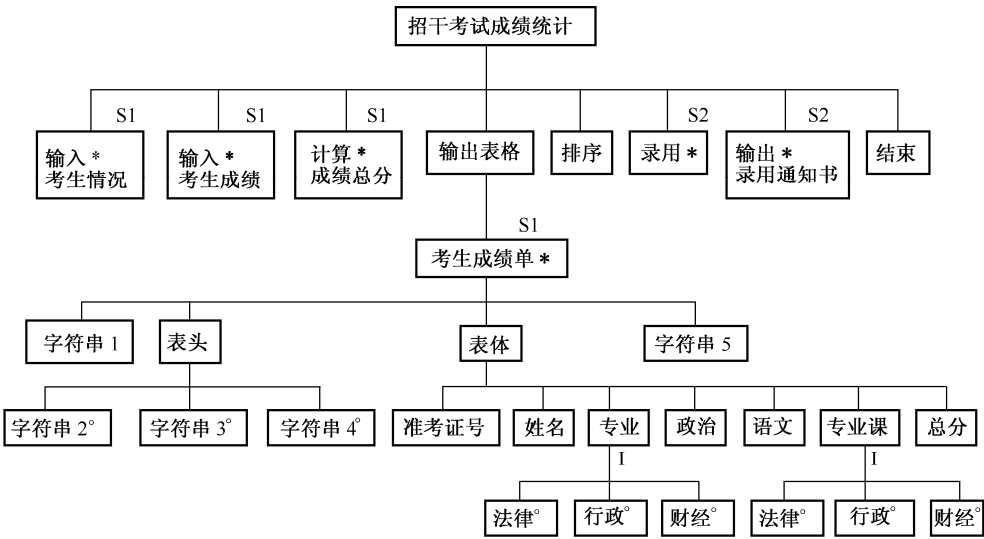


图 5-22 输入、输出数据结构关系 出程序结构 Jackson 图

- 1) 为每一对在输入数据结构和输出数据结构中有对应关系的单元 一个处理框。
- 2) 为输入和输出数据结构中 的数据单元 一个处理框。
- 3) 所有处理框在程序结构图上的 ，应 由它处理的数据单元在数据结构 Jackson 图上的 。
- 4) 必要时，可以对 出的程序结构图进行进一步的细化。
- (4) 列出基本操作 条件，并把它它们分 到程序结构图的 。
- (5) 用 码 出程序。

5.4 Warnier 设计方法

5.4.1 Warnier 方法概述及其图例

Warnier 方法又称为逻辑构造程序的方法，简称 LCP (Logical Construction of Program) 法。Warnier 方法也是一种软件开 方法。1974 年，J.D.Warnier 提出的软件开 方法 Jackson 方法类 。 有三点：一是它们使用的图形工具不同，分 使用 Warnier 图和 Jackson 图；一个 是使用的 码不同；最主要的 是在构造程序框架时，Warnier 方法 考虑输入数

据结构，而 Jackson 方法不 考虑输入数据结构，而 还考虑输出数据结构。

Warnier 程序设计方法是一种面向数据结构的设计方法。它也是从数据结构出 设计程序，但是这种方法的逻辑 严格。Warnier 图在 Warnier 方法中使用一种 用的表达工具。用 Warnier 图可以表明信息的逻辑组织，也就是说，它可以指出一类信息或一个信息量是重复出现的，也可以表示 定信息在某一类信息中是有条件地出现的。Warnier 图是由 的 括号、代码以及 量说明和符号组成的 次 。 括号： 分数据结构的 次，在一个 括号内的所有名字都属于同一类信息。 或符号 $\oplus$ ：表明一类信息或一个数据元 在一定条件下 出现，而 在这个符号上、下方的两个名字所代表的信息只能出现一个。 括号：中间的数字指明了这个名字代表的信息类（或元 ）在这个数据结构中重复出现的次数。

Warnier 图也具有表达数据结构和表达程序结构的 重功能。Warnier 图是表示数据 次结构的一种图形工具，它用 型结构来描绘数据结构。它还能指出某一类数据或某一数据元 重复出现的次数，并能指明某一 定数据在某一类数据中是否是有条件的出现。在进行软件设计时，从 Warnier 图入 ，能够很容易转换成软件的设计描述。

以 的自动编辑系统为例。通常 的 面采用以下格式。

头 部分	论部分	部分
头条	论	体
国内		业
地	者来信	广

上面给出的 概观就是一个数据的 次结构。如图 5-23 所示给出了用 Warnier 图表示的 的数据 次结构。

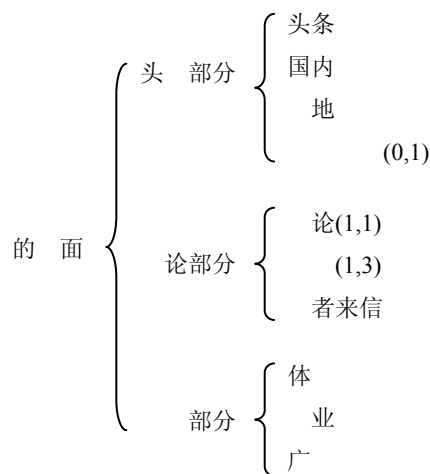


图 5-23 数据 次结构

在这个 Warnier 图中，用 括号“{”表示 次关系，在同一括号下，自上到下是顺序列的数据项。在有些数据项名字后面 加了 括号，给出该数据项重复的次数。

例如， 论(1,1)表示 论 1 ； (1,3)表示 1 到 3 ； (0,1)表示

可有可无，有就 1。外，Warnier 图可以通过细化组合数据项进一步分解信息域。
输入数据结构 Warnier 图如图 5-24 所示。

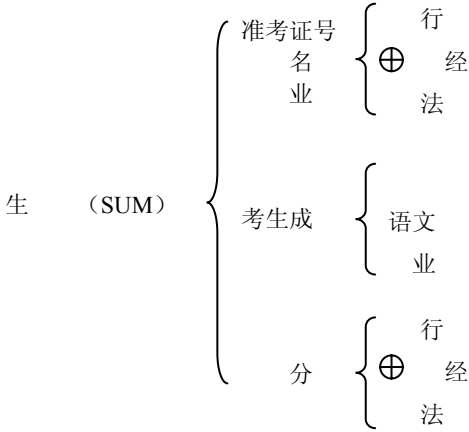


图 5-24 输入数据结构

输出数据结构 Warnier 图如图 5-25 所示。

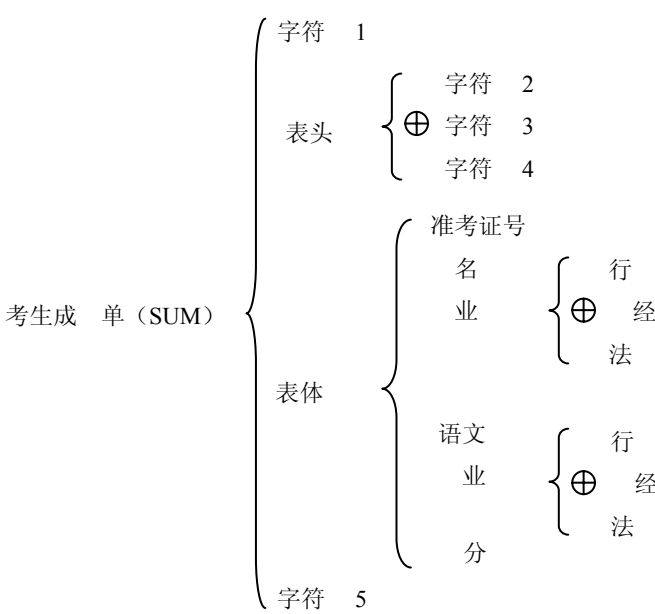


图 5-25 输入数据结构

输入、输出数据结构关系 出程序结构 Warnier 图如图 5-26 所示。

5.4.2 Warnier 程序设计过程

(1) 通过分析和确定输入数据和输出数据的逻辑结构，确认数据流程并 出 Warnier 图的数据结构。

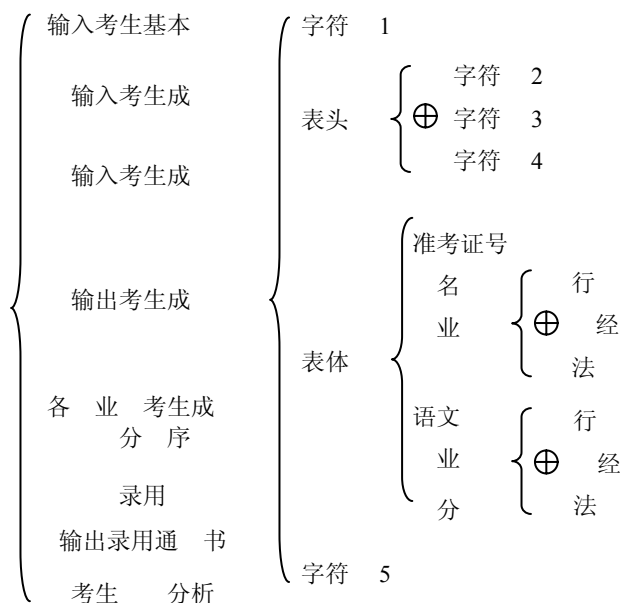


图 5-26 关系 出程序结构

- (2) 据输入的数据结构得出程序的逻辑结构，用 Warnier 图 出程序的处理过程。
- (3) 据以上两步的结构， 出程序的流程图。
- (4) 出程序流程图并自上而下给每个处理框编序号，分类 出 指 。
- (5) 对 (4) 的 指 进行分类 序，并描述出程序处理过程的 代码。

5.5 结

详细设计常用的描述方式包括流程图、盒图、问题分析图和程序设计语言 (PDL) 等，本节 一个支 问题分析图的工具系统。详细设计的工具：

- 图形工具。用图形工具可以把过程的细节用图形描述出来。
- 表格工具。用一张表来描述过程的细节，列出各种可能的操作和相应的条件。
- 语言工具。用某种 级语言 (称之为 码) 来描述过程的细节。

程序流程图又称之为程序框图，它是软件开 者最熟悉的一种算法表达工具。它 于任何一种程序设计语言， 直观和清晰地描述过程的控制流程，易于 。此，至是软件开 者最 采用的一种工具。流程图中只能使用下述的 5 种基本控制结构。

- 顺序型由几个连 的处理步骤依次 列构成，如图 5-27 所示。
- 选择型是指由某个逻辑判断式的取值决定选择两个处理中的一个，如图 5-28 所示。
- While 型循环是先判定型循环，在循环控制条件成 时，重复 行 定的处理，如图 5-29 所示。
- Until 型循环是后判定型循环，重复 行某些 定的处理，直到控制条件成 为止，如图 5-30 所示。

多 型选择列 多种处理 ， 据控制 量的取值，选择 行其一，如图 5-31 所示。

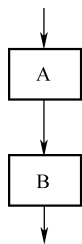


图 5-27 顺序型

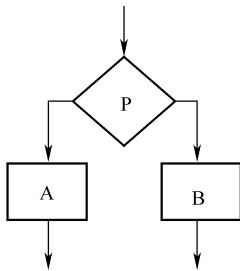


图 5-28 选择型

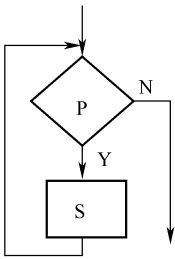


图 5-29 While 型循环

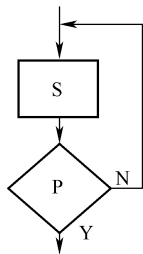


图 5-30 Until 型循环

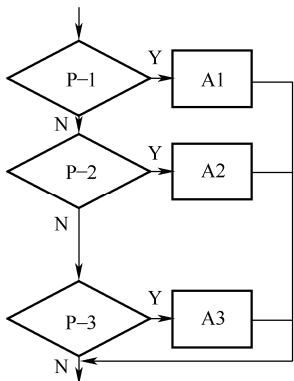


图 5-31 多型循环

盒图是一种符合结构化程序设计的图形描述工具，又称之为 N-S 图。在 N-S 图中，为了表示 5 种基本控制结构，规定了 5 种图形构件。顺序型如图 5-32 所示，在顺序型中，先行 A，后行 B。

如图 5-33 所示，在选择型结构中，如果条件 P 成立，可行 T 下面的 A 的内容，如果条件 P 不成立，可行 F 下的内容。

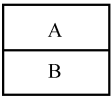


图 5-32 顺序型结构

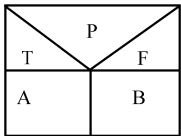


图 5-33 选择型结构

如图 5-34 所示，在 While 重复型循环结构中，先判断 P 的值，可行 S。其中 P 是循环条件，S 是循环体。

如图 5-35 所示，在 Until 重复型循环结构中，先行 S，后判断 P 的值。

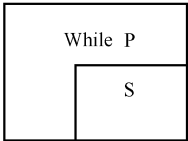


图 5-34 While 重复型循环结构

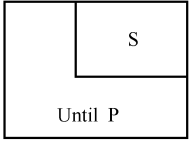


图 5-35 Until 重复型循环结构

多分支选择型如图 5-36 所示。

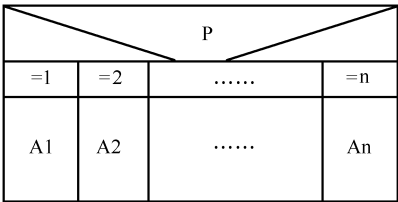


图 5-36 多分支选择型

PAD 图是用结构化程序设计表现程序逻辑结构的图形工具。问题分析图（PAD）用顺序、选择、循环等 3 种标准图式描述一个模块内部的一行过程，而用“规”（Tree Walk）地 PAD 翻译成编程语言。制以下两种工具可以支 PAD 方法：图形和文字编辑，从 PAD 自动产生程序的程序生成。本公司 20 80 年代在 M 系列上开了一个试性的系统 SDL/PAD（Software Design Language/PAD），该系统接用 SDL 方法确定的模块说明书，然后设计人以方式进行详细设计（采用 PAD 或 SDL 描述法），而可以直接产生用编程语言书的程序。

PAD 也设了 5 种基本控制结构的图示，并使用。

顺序型如图 5-37 所示 顺序先 行 A， 行 B。选择型如图 5-38 所示，给出了判断条件为 P 的选择型结构。 P 为真值时 行上面的 A 框，P 为假值时 行下面的 B 框。如果这种选择型结构只有 A 框， 有 B 框，表示该选择结构中只有 Then 后面有可 行语句 A，有 Else 部分。

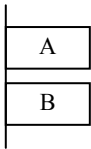


图 5-37 顺序型结构

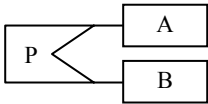


图 5-38 选择型结构

While 重复型和 Until 重复型。如图 5-39 所示，P 是循环判断条件，S 是循环体。循环判断条件框的 为 线，表示该矩形域是循环条件，以 于一 的矩形功能域。

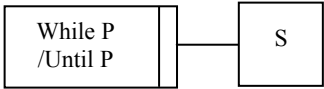


图 5-39 While 重复型和 Until 重复型结构

多分支选择型是 Case 型结构。 判定条件 P 等于 1 时 行 A1 框的内容，P 等于 2 时 行 A2 框的内容，P 等于 N 时 行 An 框的内容。

PDL（过程设计语言）是所有 文形式的过程设计工具的统称，到目前为止 出现多种 PDL 语言。PDL 具有“ ”的编程语言的一点。过程设计语言（Process Design Language，简称 PDL）是一个 统的名字，目前有 多种不同的过程设计语言。过程设计语言用于描述

模块中算法和加工的具体细节，以 在开 人 之间 精确地进行 流。

过程设计语言由外 语法和内 语法构成。外 语法描述结构，采用 一 编程语言类的确定的关 字（如 If-Then-Else、Do-While 等），内 语法描述操作，可以采用任 的自然语句（ 语或 语）。

判定表（Decision Table）也是描述加工的一种图形工具， 表格形。判定表由四部分组成。第一部分即 表示的部分，判定表的 上部称为基本条件项，列出各种可能的条件。第二部分即 表示的部分，判定表的 上部称为条件项，它列出了各种可能的条件组合。第三部分即 表示的部分，判定表的 下部称为基本动作项，它列出了所有的操作。第四部分即 表示的部分，判定表的 下部称为动作项，它列出在对应条件组合下所选的操作。判定 又称决策 ，是一种描述加工的图形工具， 合描述问题处理中具有多个判断，而 每个决策 条件有关。使用判定 进行描述时，应该从问题的文字描述中分清哪些是判定条件，哪些是判定的决策， 据描述 中的连接 出判定条件的从属关系、并列关系、选择关系， 据它们构造判定 。

Jackson 系统开 方法（JSD, Jackson System Development）是一种典型的面向数据结构的分析设计方法。Jackson 系统开 方法的系统模型就是相 通信的一组进程的 合。Jackson 图对于种类 多的程序中使用的数据结构，各数据元 之间的逻辑关系只有顺序、选择、重复 3 种，所以逻辑数据结构也只有 3 种。Warnier 方法又称为逻辑构造程序的方法，简称 LCP 法（Logical Construction of Program）， Jackson 方法类 。 有三点：一是它们使用的图形工具不同，分 使用 Warnier 图和 Jackson 图； 一个 是使用的 码不同；最主要的 是在构造程序框架时，Warnier 方法 考虑输入数据结构，而 Jackson 方法不 考虑输入数据结构，而 还考虑输出数据结构。

5.6 题

- 1 概述软件详细设计的任务及步骤。
- 2 简述软件详细设计的图形工具有哪些。
- 3 详细设计的任务是
- 4 是结构化程序设计 用 N-S 图表示 3 种基本结构。
- 5 简述 Jackson 程序设计方法的主要内容。
- 6 简述 Warnier 程序设计方法的主要内容。
- 7 某 对部分 工重 分 工作的 策是：年 在 20 以下者， 中文化程度 产 ， 中文化程度 工。20 至 40 之间，中 文化程度， 性 工， 性 工， 文化程度都 技术 。年 在 40 以上者，中 文化程度 ， 文化程度 技术 。用判定 出此分 法。

8 就下述 码程序 出 N-S 图。

Start

Input (A,B,C)

If A > 6

Then X = 5


```

Else X = 2
End If
If B < 15
    Then Y = 10
    Else Y = 4
End If
If C < 25
    Then Z = 15
    Else Z = 6
End If
Print(X,Y,Z)
Stop

```

9 Jackson 结构图解和图解逻辑 Warnier 图有 不同

10 用 Jackson 图描绘下述的一系列 的构成：一系列 最多有两个 头。只有一个 头时 于列 最前面， 还有第二个 头时， 第二个 头 于列 最后面。 头 既可能是内 也可能是 。 分为 、 和软 等。 在所有 的前面部分，软 在所有 的后面部分。此外，在 和软 之间 还有一节 。

- 11 软件测试流程中的各个方法。
- 12 概括测试用例的设计过程。
- 13 据决策表方法设计具体测试用例。