第 5 章　详细设计

本章内容

5.1 过程设计

5.2 用户界面设计

5.3 数据代码设计

5.4 数据输入/ 输出设计

5.5 数据安全设计

5.6 详细设计文档与复审

5.1.1 流程图

01 OPTION 流程图的分类

国家标准GB/T 1526—1989 中规定，流程图分为5种。

数据流程图 01

数据流程图表示求解某一问题的数据通路，同时规定了处理的主要阶段和所用的各种数据媒体

程序流程图 02

程序流程图表示程序中的操作顺序

系统流程图 03

系统流程图表示系统的操作控制和数据流

程序网络图 04

程序网络图表示程序激活路径和程序与相关数据流的相互作用

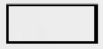
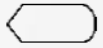
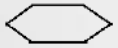
系统资源图 05

系统资源图表示适用于一个问题或一组问题求解的数据单元和处理单元的配置

5.1.1 流程图

02 OPTION 流程图符号

国家标准GB/T 1526—1989 中的主要流程图符号如表所示。

符号	名称	符号	名称
	处理		显示
	判断		循环开始
	准备		循环结束
	人工操作		连接符
	人工输入		端点符
	数据	···	省略符
	存储数据		流线
	文件		内存储器

5.1.1 流程图

03 OPTION 流程图符号使用约定

- 符号的用途是标识它所表示的功能，而不考虑符号内的内容。
- 图中各符号均匀地分配空间，连线应保持合理长度，尽量少用长线。
- 不要改变符号的角度和形状，尽可能统一各种符号的大小。
- 应把理解某符号功能所需的最少量的说明文字置于符号内，若说明文字太多，可使用一个注解符。
- 符号标识符的作用是便于其他文件引用。

5.1.1 流程图

分支符号如图1 所示，多分支符号如图2 所示，每个出口应加标识符，以反映其逻辑通路。

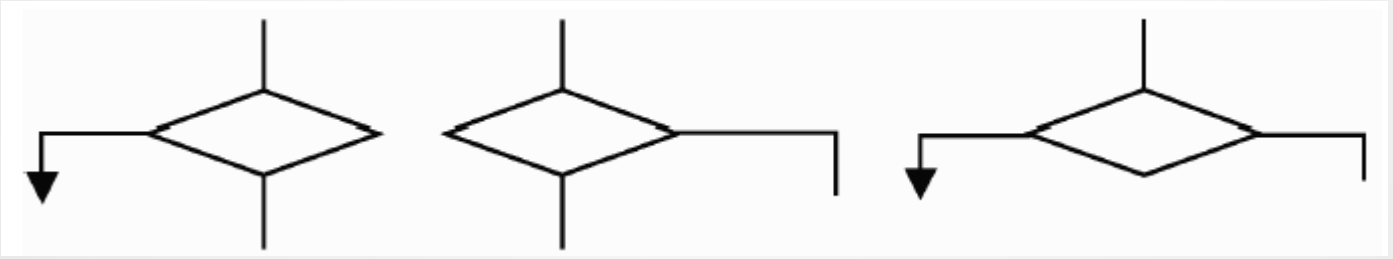


图1

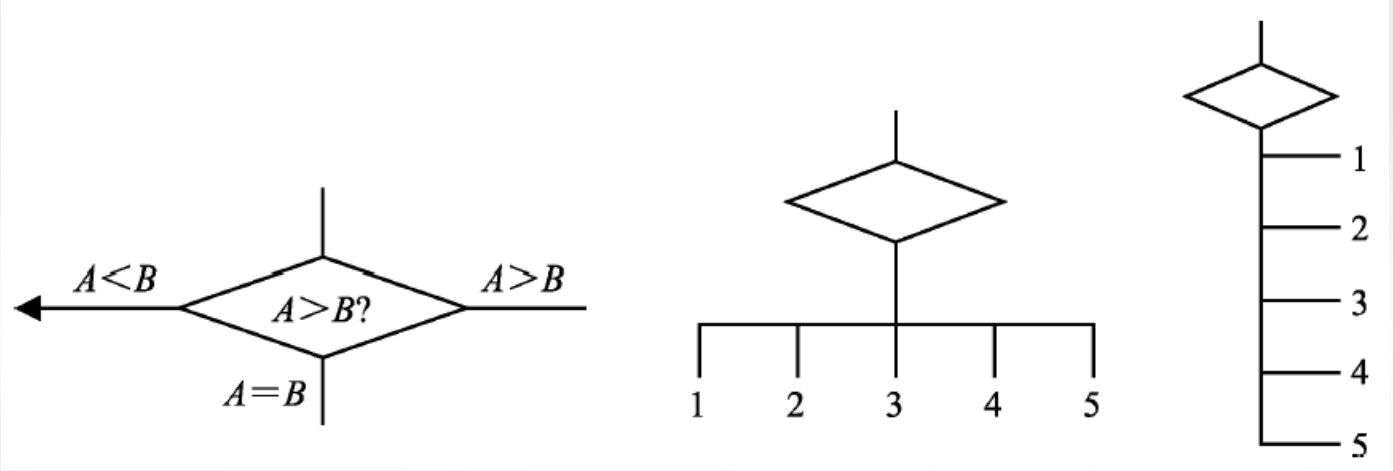


图2

5.1.1 流程图

- 流线可以指示数据流或控制流。可以用箭头指示流程方向（简称流向）。当流向从左到右、自上而下时，箭头可以省略；反之要用箭头指示流向。
- 连接符：连接符往往是成对出现的，在出口连接符与对应的入口连接符中应记入相同的文字、数字、名称等识别符号表示衔接，如图1所示。图1所示是某流程图的一部分，其中有一个分支流向A，而分支A 的详细流程如图2所示。

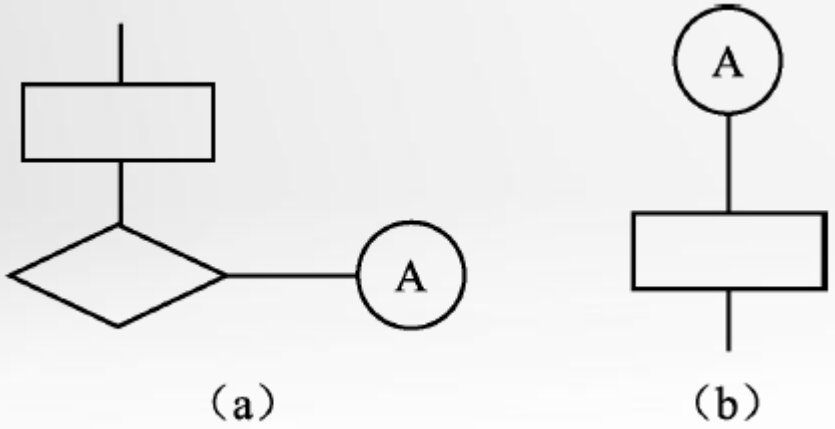


图1

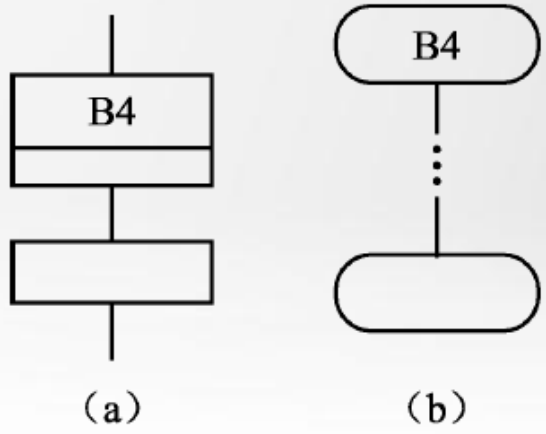
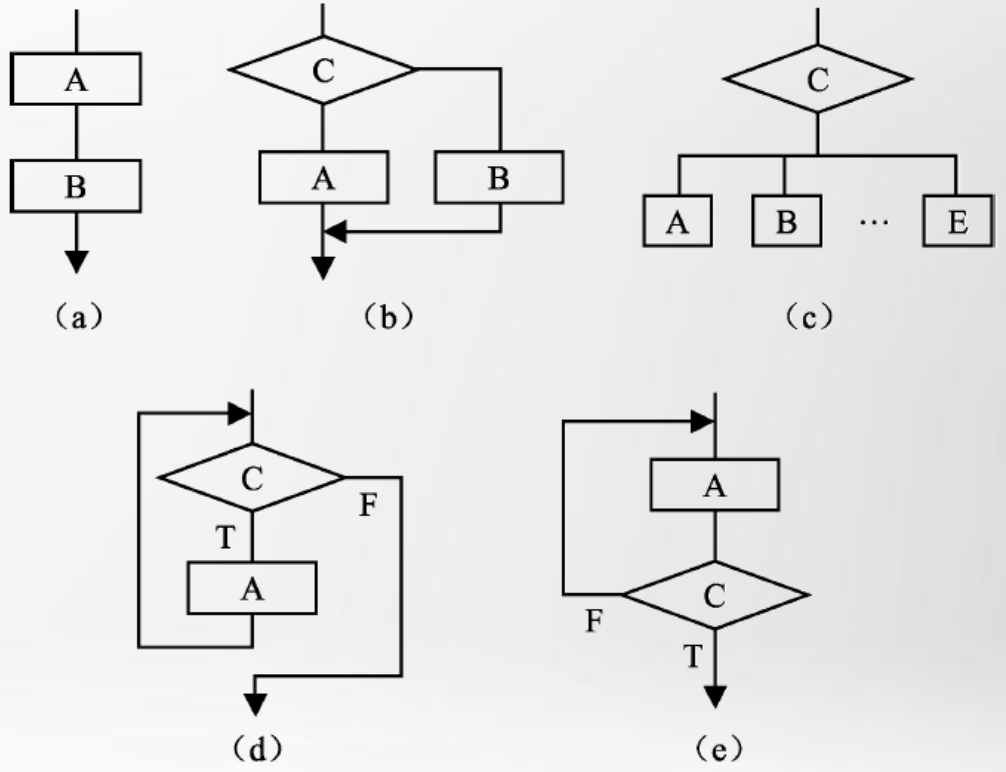


图2

5.1.1 流程图

04 OPTION 流程图的3 种基本结构

流程图的3 种基本结构为顺序结构、条件结构和循环结构，如图1所示，图中的C 是判定条件。顺序结构如图（a）所示。条件结构可分为两种，一种是IF…THEN…ELSE 型条件结构，如图（b）所示；另一种是CASE 型多分支结构，如图（c）所示。

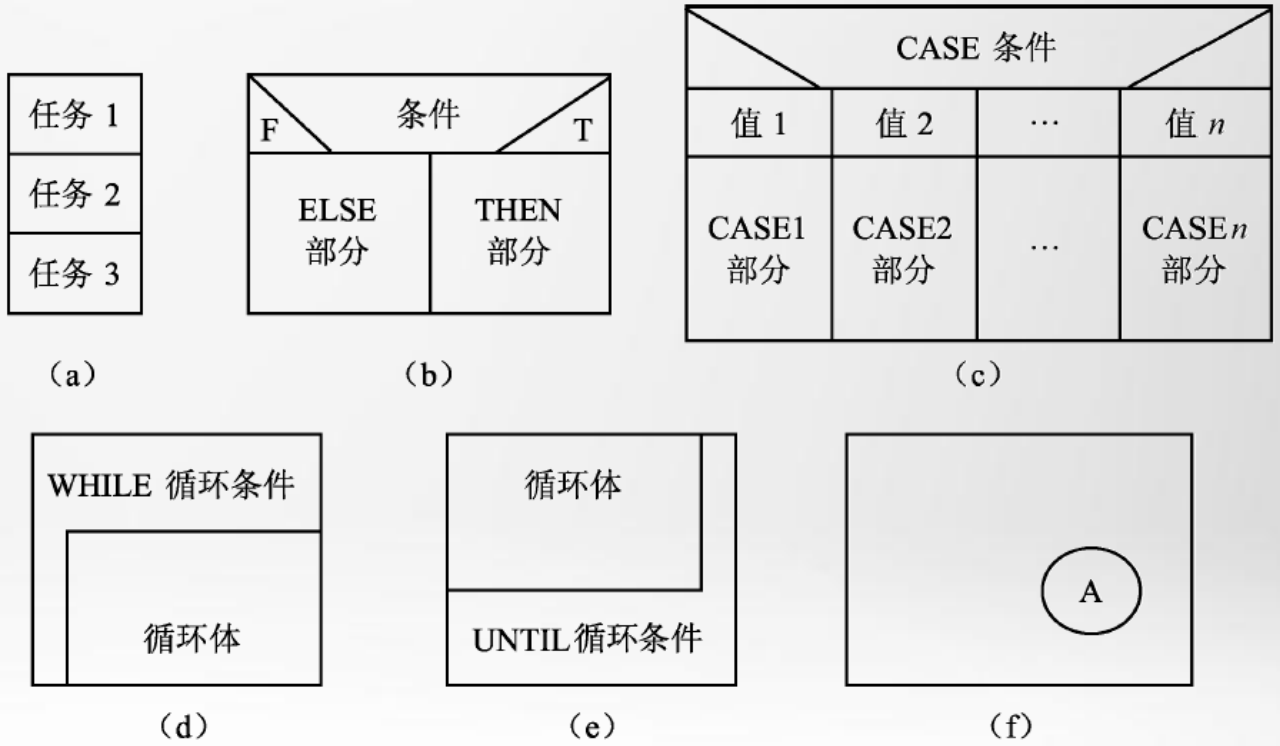


（a） （b） （c） （d） （e）

5.1.2 盒图

01 OPTION 盒图的符号

盒图的符号都画在一个矩形框内，可以根据结构化设计的要求表示软件的层次结构、条件结构、循环结构和嵌套结构。



5.1.2 盒图

02 OPTION 盒图的特点

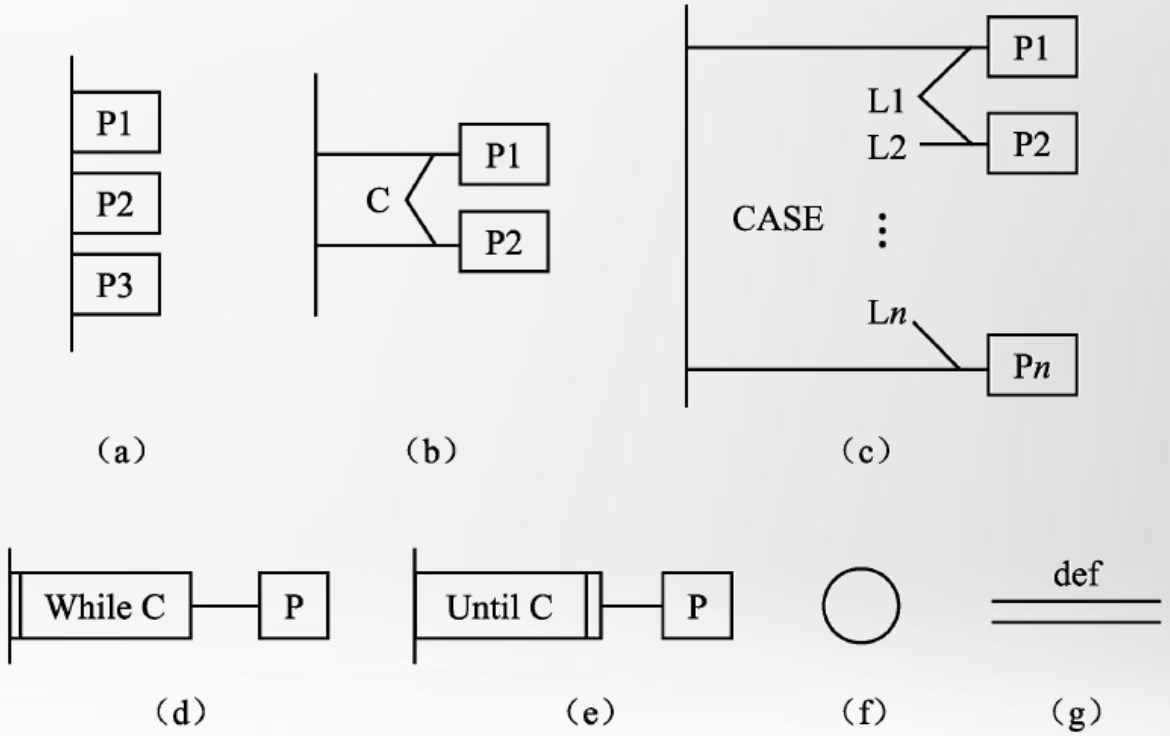
- 能清晰地描述功能域。
- 没有表示任意转移控制的符号，因而只能表示结构化设计的结构。
- 易于确定数据的作用域是全局还是局部。
- 易于描述系统模块的层次结构和嵌套关系。
- 容易转换为高级程序设计语言。

5.1.3 PAD

01 OPTION

PAD 的基本符号

- 顺序结构
- 条件结构
- CASE 型多分支结构
- 先检测循环条件的WHILE 型循环结构
- 后检测循环条件的UNTIL 型循环结构
- 语句标号
- 定义



5.1.3 PAD

02 OPTION PAD 的特点

用PAD 表示的程序从最左边的竖线的上端开始，自上而下、自左向右执行

用PAD 设计的软件结构必然是结构化的程序结构

结构清晰、层次分明

PAD 既可用于表示程序逻辑，也可用于描绘数据结构

PAD 可支持自顶向下、逐步求精的设计方法

5.1.4 判定表

1 判定表的组成

- 左上部列出所有条件。
- 左下部列出所有可能做的工作。
- 右上部每一列表示各种条件的一种可能组合，所有列表示条件组合的全部可能情况。
- 右下部的每一列是和每一种条件组合所对应的应做的工作。

2 判定表中的符号

- 右上部用T 表示条件成立，用 F 表示条件不成立，空白表示条件成立与否不影响。
- 右下部画× 表示在该列上面规定的条件下做该行左边列出的那项工作，空白表示不做该项工作。

5.1.5 判定树

判定树和判定表一样，也能表明复杂的条件组合与对应处理之间的关系。判定树是一种图形表示方式，更易被用户理解。

5.1.6 过程设计语言

01 OPTION PDL 的特点

- 关键字有固定语法，提供结构化的控制结构和数据说明，并在控制结构的头尾都加关键字，体现模块化的特点，如 IF...ENDIF 、 DO WHILE...ENDDO、CASE...ENDCASE 等。
- 用自然语言叙述系统处理功能。
- 有说明各种数据结构的手段。
- 能描述模块定义和调用以及模块接口模式。

5.1.6 过程设计语言

02 OPTION 用PDL 表示程序结构

顺序结构

模块定义和调用

条件结构

数据定义

循环结构

输入/ 输出

本章内容

5.1 过程设计

5.2 用户界面设计

5.3 数据代码设计

5.4 数据输入/ 输出设计

5.5 数据安全设计

5.6 详细设计文档与复审

5.2.1 用户界面设计问题

01 OPTION 系统响应时间

长度

- 响应时间过长，用户会不满意。
- 响应时间过短，会迫使用户加快操作节奏，从而容易引起操作错误。

易变性

- 响应时间相对平均响应时间的偏差称为易变性。
- 响应时间的易变性小，有助于用户建立稳定的工作节奏。
- 响应时间的易变性大，暗示系统工作出现异常。

5.2.1 用户界面设计问题

02 OPTION 用户帮助设施

集成的帮助设施

集成的帮助设施设计在软件里，对用户工作内容敏感，用户可以从与操作有关的主题中选择一个，获取帮助。

附加的帮助设施

附加的帮助设施实际上是一种查询能力有限的联机用户手册。

5.2.1 用户界面设计问题

03 OPTION 出错信息和警告信息的处理

- 信息应以用户可理解的术语描述问题。
- 信息应提供有助于从系统错误中恢复的建设性意见。
- 信息应指出错误可能导致的负面后果（例如破坏数据文件），以便用户检查是否出现了这些问题，并在问题出现时予以改正。
- 信息应伴随听觉上或视觉上的提示，在显示信息的同时发出警告声或用闪烁方式显示，或用明显的颜色表示出错信息。
- 出错信息不能指责用户。

5.2.1 用户界面设计问题

04 OPTION 命令的交互

每个菜单项都应有对应的命令。

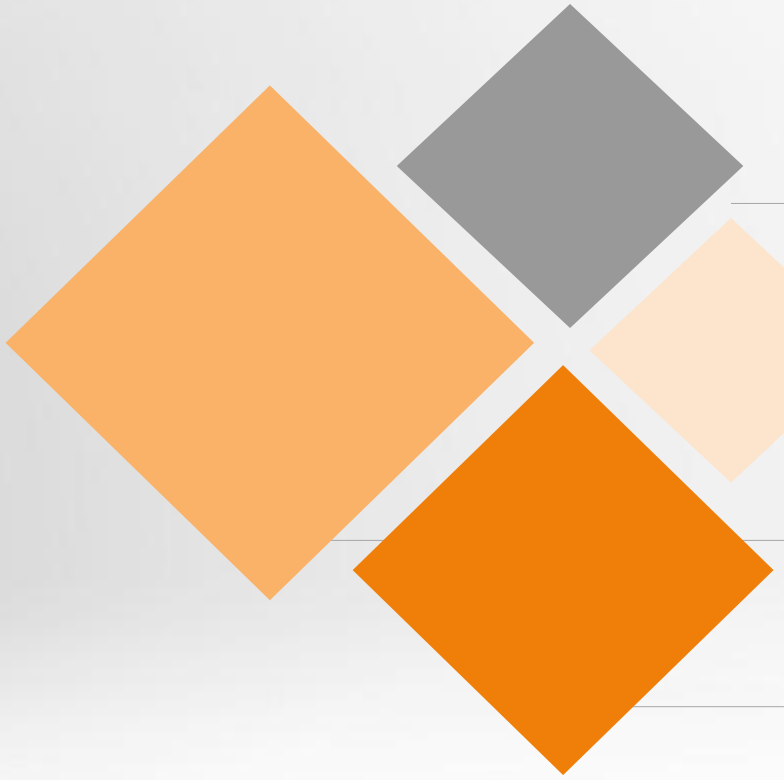
命令形式：控制序列、功能键或输入命令。

考虑学习和记忆命令的难度，命令应当有提示。

宏命令：代表一个常用的命令序列。

所有应用软件都有一致的命令使用方法。

5.2.2 用户界面设计过程



设计和实现用户界面原型。

用户试用该原型，向设计人员提出对界面的评价。

设计人员根据用户的评价修改设计，并实现下一级原型。

不断进行设计、修改，直到用户满意为止。

5.2.3 用户界面设计的基本原则

5.2.4 用户界面设计指南

1 一般交互

- 保持一致性
- 提供有意义的反馈
- 要求确认
- 允许取消操作
- 尽量减少用户的记忆量
- 提高效率
- 允许用户犯错误
- 按功能对动作分类
- 提供帮助设施
- 命令名要简单

2 信息显示

- 用户界面显示的信息应简单、完整、清晰、含义明确
- 可用不同方式显示信息
- 只显示与当前工作内容有关的信息
- 使用一致的标记、标准的缩写和适当的颜色
- 产生必要的出错信息
- 使用大小写、缩进和文本分组等方式帮助理解
- 使用模拟显示方式表示信息
- 高效率地使用显示屏，用窗口分隔不同类型的信息

3 数据输入

- 尽量减少用户的输入动作
- 保持信息显示和数据输入的一致性
- 允许用户自己定义输入
- 允许用户选择输入方式
- 使当前不适用的命令不可用
- 允许用户控制交互流程
- 为所有输入动作提供帮助
- 消除冗余的输入

本章内容

5.1 过程设计

5.2 用户界面设计

5.3 数据代码设计

5.4 数据输入/ 输出设计

5.5 数据安全设计

5.6 详细设计文档与复审

5.3 数据代码设计

代码的定义和作用

数据代码是为了对数据进行识别、分类、排序等操作所使用的数字、文字或符号。数据代码具有识别、分类和排序3 项基本功能。

代码的特性

- 简洁性：减少存储空间、消除二义性。
- 保密性：不了解编码规则的人不知道代码的含义。
- 可扩充性：设计代码时要留有余地，方便在软件生命周期内增加代码。
- 持久性：代码应可以在软件的生命周期内长久使用。代码的变换会影响数据库和程序。

5.3.1 数据代码设计原则

标准化

尽可能采用国际标准、国家标准、部颁标准或习惯标准

唯一性

一个代码只代表一个信息，每个信息只有一个代码

可扩充性

设计代码时要留有余地，方便代码的更新、扩充

简单性

代码结构简单、尽量短，便于记忆和使用

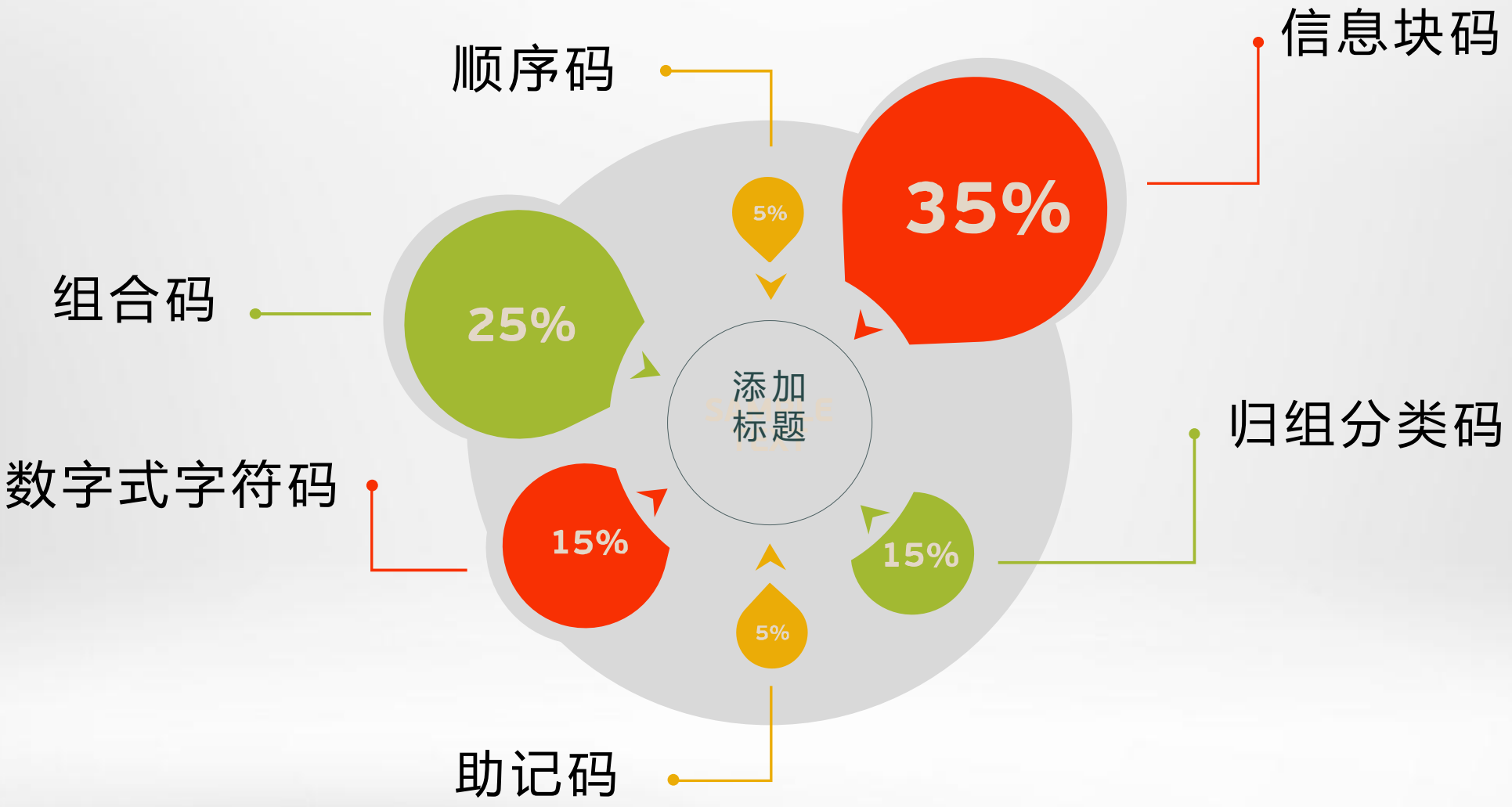
规范化

代码的结构、类型、缩写格式要统一

适应性

代码要尽可能反映信息的特点，唯一地标识某些特征

5.3.2 代码种类



5.3.3 数据代码设计方法

数据代码设计一定要遵循简单性、唯一性、标准化、可扩充性、规范化等原则，设计代码的目的是提高信息的处理效率。

本章内容

5.1 过程设计

5.2 用户界面设计

5.3 数据代码设计

5.4 数据输入/ 输出设计

5.5 数据安全设计

5.6 详细设计文档与复审

5.4.1 数据输入设计

数据输入设计需要对信息的发生、收集、介质化、输入，以及信息内容等方面进行详细的调查、研究后才能进行。

5.4.2 数据输出设计

数据输出设计应充分考虑用户的需求，输出处理要根据输出内容、输出方式、信息分配等具体情况，进行分析、研究后再进行设计。

1 输出内容

输出的内容和格式必须满足用户的要求

2 输出方式

输出的方式可以是屏幕显示、打印或其他

3 信息分配

信息的分配要表明通过什么途径、采用什么方式、什么周期、送给什么人，要防止信息的遗失、泄密、延误等

本章内容

5.1 过程设计

5.2 用户界面设计

5.3 数据代码设计

5.4 数据输入/ 输出设计

5.5 数据安全设计

5.6 详细设计文档与复审

5.5 数据安全设计

01 OPTION 安全事故

数据被破坏或修改

保密的数据被公开

数据和系统不能为用户服务

5.5 数据安全设计

02 OPTION 安全控制方法

5.5 数据安全设计

03 OPTION 数据安全受破坏时的措施

- 要预防由于计算机硬件、软件或操作上的错误，破坏数据的正确性。万一出现错误，应及时发现错误，认真分析错误的影响程度。
- 重要的数据要有备份，避免系统设备被破坏而造成损失。
- 系统发生故障后，在恢复系统的同时要找出事故原因，预防事故再次发生。
- 软件工程的各个阶段都应有专门人员监督检查软件开发、使用中数据的正确性、可信度，保护系统资源不受侵犯，保证系统全部资源和功能的安全性、可靠性。

本章内容

5.1 过程设计

5.2 用户界面设计

5.3 数据代码设计

5.4 数据输入/ 输出设计

5.5 数据安全设计

5.6 详细设计文档与复审

5.6.1 详细设计说明书

01 OPTION 引言

编写目的

背景说明

定义

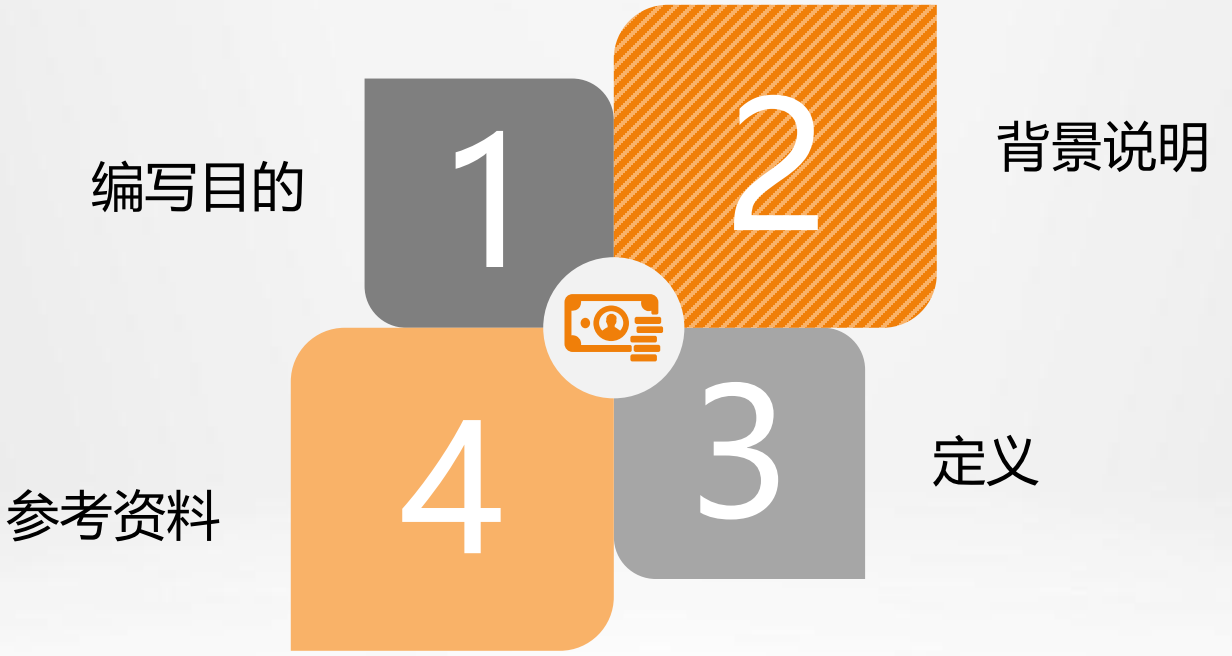
5.6.1 详细设计说明书

02 OPTION 程序系统的结构

- 程序描述。
- 功能。
- 性能。
- 输入项。
- 输出项。
- 算法。
- 流程逻辑。
- 接口。
- 存储分配。
- 注释设计。
- 限制条件。
- 测试计划。
- 尚未解决的问题。

5.6.2 操作手册

01 OPTION 引言



5.6.2 操作手册

02 OPTION

软件概述

1 软件的结构

结合软件系统所具有的功能，包括输入、处理和输出，提供该软件的总体结构图表

2 程序表

列出本系统内每个程序的标识符、编号和简称以及程序的功能

3 文卷表

列出将由本系统引用、建立或更新的每个永久性文卷，说明它们各自的标识符、编号、简称、存储媒体和存储要求

5.6.2 操作手册

03 OPTION 安装与初始化

具体说明为使用本软件而需要进行的安装与初始化过程，包括程序的存储形式、安装与初始化过程中的全部操作命令、系统对这些命令的反应与答复、表明安装工作完成的测试实例等；此外，还应说明安装过程中所需用到的专门软件。

5.6.2 操作手册

04 OPTION 运行说明

运行是指提供一个启动控制信息后，直到计算机系统等待另一个启动控制信息时为止的计算机系统执行的全部过程。

运行表

要列出每个可能的运行，说明每个运行的目的，指出每个运行所执行的程序。

运行步骤

逐个说明每个运行及完成整个系统运行的步骤。以对操作人员最方便、最有用的形式，说明运行的有关信息，如运行控制、操作信息和运行目的。

5.6.2 操作手册

05 OPTION 非常规过程

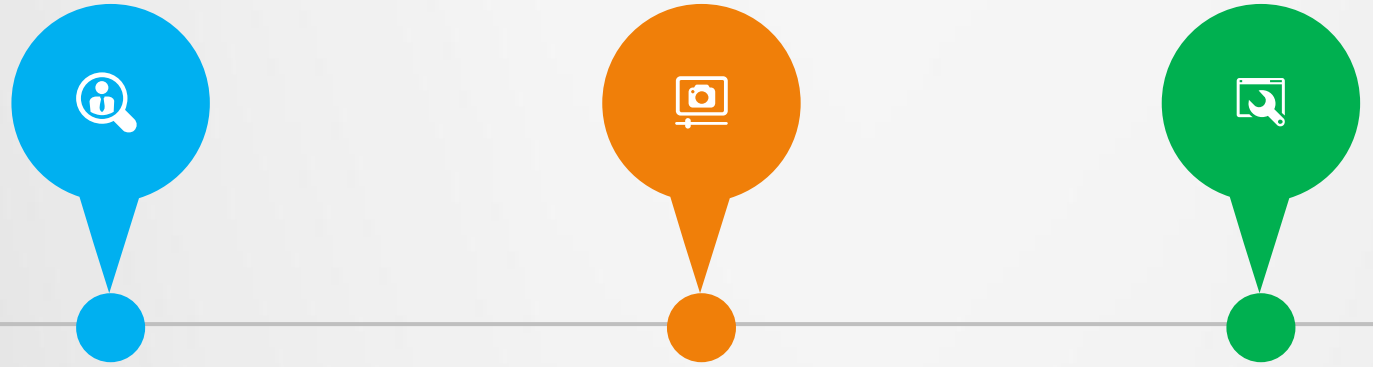
非常规过程提供有关应急操作或非常规操作的必要信息，如出错处理操作、向后备系统的切换操作以及其他必须向程序维护人员交代的事项和步骤。

06 OPTION 远程操作

如果本软件能够通过远程终端控制运行，则说明其操作过程。

5.6.3 详细设计的复审

详细设计的复审软件的详细设计完成后，必须从软件的正确性和可维护性两个方面，对实现功能的逻辑、数据结构和界面等进行审查。



设计人员和设计组的另一成员一起进行静态检查

由检查小组进行较正式的软件结构设计检查

由检查小组进行正式的设计检查，对软件设计质量给出评价