

第 5 章 总体设计

5.1 基本知识点

5.1.1 软件设计基础

1. 软件设计的目标

传统的结构化方法将软件设计划分为体系结构设计、数据设计、接口设计和过程设计 4 个部分。软件设计的过程和目标,就是根据用信息域表示的软件需求,以及功能和性能要求进行数据设计、体系结构设计、接口设计和过程设计。图 5.1 描绘了软件设计过程中的信息流。

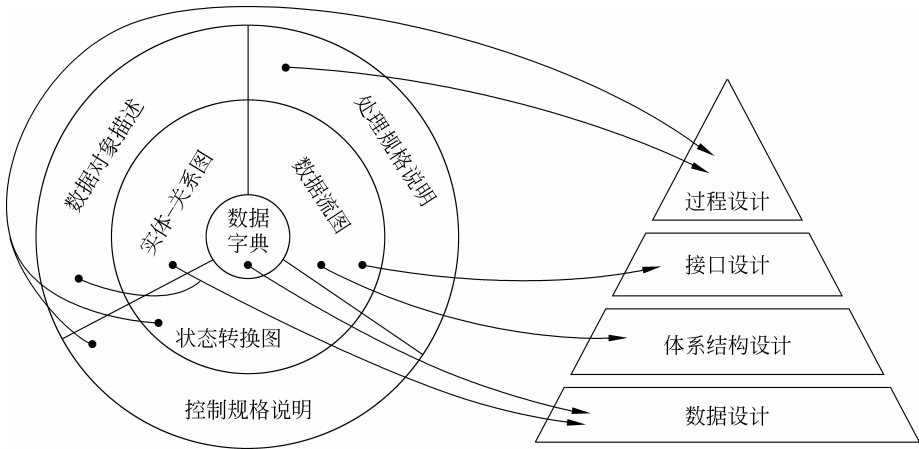


图 5.1 把分析模型转换成软件设计

- (1) 体系结构设计：体系结构设计定义软件的主要结构元素及其之间的关系。
- (2) 数据设计：数据设计将分析阶段创建的信息模型转变成实现软件所需的数据结构。
- (3) 接口设计：接口设计描述用户界面、软件和其他硬件设备、其他软件系统及使用人员的外部接口,以及各种构件之间的内部接口。
- (4) 过程设计：过程设计的主要工作是确定软件各个组成部分内的算法及内部数据结构,并选定某种过程的表达形式描述各种算法。

2. 软件设计的任务

从工程管理的角度来看,传统的软件设计任务通常分为两个阶段完成,如图 5.2 所示。

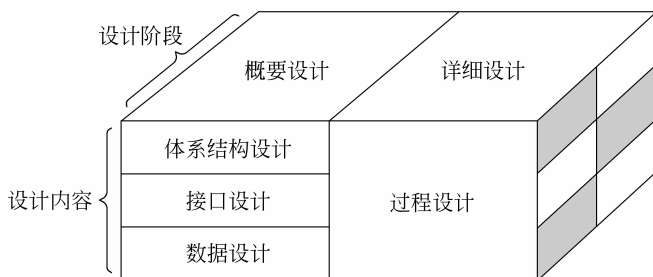


图 5.2 软件设计过程结构图

第一个阶段是概要设计,又称为总体设计或初步设计,将软件需求转化为数据结构或软件的系统结构,包括结构设计和接口设计,并编写概要设计文档。这一阶段主要确定实现目标系统的总体思想和设计框架。

第二阶段是详细设计阶段,即过程设计或构件级设计,其任务是通过结构表示进行细化,确定各个软件构件的详细数据结构和算法,产生描述各软件构件的详细设计文档,详细设计或构件级设计的根本目标是确定应该怎样具体地实现所要求的系统。

3. 总体设计过程

总体设计的目标是综合采用各种技术手段,将系统需求转换为模块结构、数据结构(或对象/类结构)的表达式,并实现系统的性能、安全性、可靠性要求。

5.1.2 软件设计的基本原理

软件设计的主要内容如下。

- (1) 把软件划分成若干独立成分的依据。
- (2) 表示不同的成分内的功能细节和数据结构。
- (3) 统一衡量软件设计技术质量。

1. 模块化设计原理

所谓模块,是指具有相对独立性的,由数据说明、执行语句等程序对象构成的集合。程序中的每个模块都需要单独命名,通过名字可实现对指定模块的访问。在高级语言中,模块具体表现为函数、子程序、过程等。

在软件的体系结构中,模块是可组合、分解和更换的单元,具有 4 个基本属性:接口、功能、逻辑、状态。

功能、状态与接口反映模块的外部特征,逻辑则反映模块的内部特征。

模块是构成程序的基本构件。对于其他模块而言,只需了解被调用模块的外部特性就足够了,不必了解它的内部特性。在软件设计时,通常先确定模块的外部特性,然后再确定它的内部特性。前者是总体设计的任务,后者是详细设计的任务。

模块化是指将整个程序划分为若干个模块,每个模块用于实现一个特定的功能。把这些模块集成起来构成一个整体,可以完成指定的功能满足用户的需求。划分模块对于解决

大型复杂的问题是非常必要的,可以大大降低解决问题的难度。

2. 抽象和逐步求精

人们在认识复杂现象的过程中,使用的最强有力的思维工具就是抽象。所谓抽象就是将事物相似的方面集中和概括起来暂时忽略它们之间的差异。或者说,抽象就是抽出事务本质特性而暂时不考虑它们的细节。

逐步求精是一种先总体、后局部的思维原则,也就是一种逐层分解、分而治之的方法。在面对一个复杂的大问题时,它采用自顶向下、逐步细化的方法,将一个大问题逐层分解成许多小问题,然后每个小问题再分解成若干个更小的问题,经过多次逐层分解,每个最低层问题都足够简单,最后再逐个解决。

3. 信息隐蔽和局部化

信息隐蔽是指在设计和确定模块时,使一个模块内包含的信息(过程和数据),对于不需要这些信息的其他模块来说,是不可访问的。“隐蔽”意味着有效的模块化可以通过定义一组相互独立的模块来实现,这些独立的模块彼此间仅交换那些为了完成系统功能而必须交换的信息,而将自身的实现细节与数据“隐藏”起来。

局部化和信息隐蔽是密切相关的。所谓局部化,就是使一些关系密切的软件元素彼此靠近些。

4. 模块独立性

模块独立性是指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量最少且接口简单。模块独立性的概念是模块化、抽象和信息隐蔽这些软件工程基本原理的直接结果。只有符合和遵守这些原则才能得到高度独立的模块。良好的模块独立性能使开发的软件具有较高的质量,并能有效提高软件的生产率。这是因为有效模块化(即具有独立的模块)的软件比较容易开发,功能分割明确并且接口简单,当多人合作开发时,这点尤其重要。同时,独立的模块也容易测试和维护。所以模块独立是好的设计的关键。

模块独立性可以由两个定性标准度量,反映模块外部特征的标准是耦合,反映模块内部特征的则是内聚。

耦合也称块间联系,是对一个软件结构内不同模块间相互联系紧密程度的度量。模块间联系越紧密,耦合性就越强,模块的独立性就越差。

耦合的类型有:非直接耦合、数据耦合、标记耦合、控制耦合、外部耦合、公共耦合、内容耦合。

内聚也称块内联系,是指一个模块内部各个元素之间关系的紧密程度。若一个模块内各元素(语句间、程序段间)联系得越紧密,则内聚性就越高,模块的独立性就越好。

内聚的类型有:功能内聚、信息内聚、通信内聚、过程内聚、时间内聚、逻辑内聚、巧合内聚。

5. 软件结构设计优化原则

软件总体设计的主要任务是软件结构的设计,为了提高设计质量,必须根据软件设计原理改进软件设计,并提出以下软件结构设计的优化原则。

(1) 模块的独立性原则。即尽量高内聚、低耦合,保持模块相对独立性,并以此为原则优化初始的软件结构。

(2) 保持适中的模块规模和复杂度。一个模块的规模不应过大,模块的总行数应控制

在 10~100 行的范围之内,最好在 30~60 行,以便于阅读和研究。

(3) 软件层次的深度、宽度、扇入、扇出要适当。

(4) 模块的作用范围应在其控制域内。模块的控制域是这个模块本身以及所有直接或间接从属它的模块的集合。模块的作用范围是受该模块内的一个判定影响的所有模块。

(5) 定义单入口、单出口模块。尽量不使模块之间出现内容耦合,不随便使用转向语句。从顶部进入模块并且从底部退出来时,软件是比较容易理解的,因此也是比较容易维护的。

(6) 模块内的数据结构尽可能做到对外部隐蔽和具备灵活的扩充机制,并且模块的功能应该做到可以预测。

5.1.3 表示软件结构的图形

结构化设计方法(SD)方法通常采用层次图(H图)、HIPO图和结构图(Structure Chart, SC)描述软件结构。

1. 层次图和 HIPO 图

通常使用层次图(H图)来描绘软件的层次结构,层次图适合于在自顶向下设计软件的过程中使用。在层次图中一个矩形表示一个模块,矩形框之间的连线表示调用关系,位于上方的矩形框所代表的模块调用位于下方的矩形框所代表的模块。

2. 软件结构图

需求分析用数据流图的形式描述信息在系统加工之间的流动情况。面向数据流的设计方法以数据流图作为设计的依据,定义了一些变换规则,利用这些规则将数据流图变换成软件结构图。软件结构可用树状或网状结构的图形表示,通常采用 Youdon 等提出的结构图工具。

结构图是软件系统的模块层次结构,反映整个系统程序的控制层次体系。软件结构图的主要内容如下。

(1) 模块。用方框表示,并用名字标识该模块,名字体现该模块的功能。

(2) 模块间的调用关系。用单向箭头或直线连接起来表示调用关系。按照惯例,上方的模块调用下放的模块,所以不用箭头也不会产生二义性。

(3) 模块间的信息传递。

5.1.4 结构化设计方法

结构化设计方法通常也叫作面向数据流的设计。结构化设计是基于自顶向下、逐步求精、模块化设计原理等思想的设计方法。结构化设计与结构化分析和结构化编程方法前后呼应,形成了统一、完整的系列方法。

1. 结构化设计的流程

在结构化设计方法中,软件的结构用结构图来描述。结构化设计(SD)方法的步骤如下。

(1) 复查并精化数据流图。对需求分析阶段得出的 DFD 认真复查,必要时可再次进行

修改或细化。不仅要确保 DFD 给出了目标系统正确的逻辑模型,而且应该使 DFD 中每个处理都代表一个规模适中、相对独立的子功能。

(2) 鉴别 DFD 中数据流的类型,典型的数据流类型有变换型数据流和事务型数据流。数据流类型决定了映射的方法。

(3) 导出初始的软件结构图。根据数据流的类型,应用变换型映射方法或事务型映射方法得出初始的结构图。

(4) 逐级分解。对软件结构图进行逐级分解,一般需要进行一级分解和二级分解,如果需要也可以进行更多级的分解。

(5) 精化软件结构。使用设计度量和启发式原则对得到的软件结构进一步精化。

(6) 导出接口描述和全局数据结构。对每一个模块,给出进出该模块的信息,即该模块的接口描述。此外,还需要对所使用全局数据结构给出描述。

2. 典型的数据流类型和系统结构

典型的数据流类型有变换型数据流和事物型数据流。数据流的类型不同,得到的系统结构也不同。

1) 变换型数据流与变换型系统结构

变换型数据处理问题的过程大致分为三步,即取得数据、变换数据和给出数据。信息沿输入通路进入系统,同时由外部形式变换成内部形式。进入系统的信息通过变换中心,经加工处理以后,再沿输出通路变换成外部形式离开软件系统。当数据流图具有这些特征时,这种信息流就叫作变换流。

2) 事务型数据流与事务型系统结构

数据沿输入通路到达一个处理 T 时,这个处理根据输入数据的类型在若干动作序列中选出一个来执行。这类系统具有从多个事务中选择执行某个事务的能力。事务型数据流图至少有一条接收路径,一个事务中心,与若干条动作路径组成,这种数据流称为事务流。处理 T 称为事务中心,它完成下述任务:接收输入数据、分析每个事务以确定它的类型、根据事务类型选取一条活动通路。

3. 变换分析

变换分析是一系列设计步骤的总称,经过这些步骤,将具有变换流特点的数据流图按预先确定的模式映射成软件结构。

变换分析方法由以下 4 步组成。

- (1) 重画数据流图;
- (2) 区分有效输入、有效输出和中心变换部分;
- (3) 进行一级分析,设计上层模块;
- (4) 进行到底二级分解,设计输入、输出和变换中心的下层模块。

4. 事务分析

在数据流具有明显的事务特点时,也就是有一个明显的事务中心时,采用事务分析方法进行软件设计。

事务分析的设计步骤和变换分析的设计步骤大部分相同或类似,主要区别在于由数据流图到软件结构的映射方法不同。

5.1.5 总体设计文档及复审

总体设计完成后要给出概要设计说明书,概要设计说明书的主要内容及结构如下。

1. 引言

1.1 编写目的[阐明编写概要设计说明书的目的,指明读者对象。]

1.2 项目背景[应包括:①项目的委托单位、开发单位和主管部门;②该软件系统与其他系统的关系。]

1.3 定义[列出本文档中所用到的专门术语的定义和缩写词的原意。]

1.4 参考资料[列出有关资料的作者、标题、编号、发表日期、出版单位或资料来源,可包括:①项目经核准的计划任务书、合同或上级机关的批文;②项目开发计划;③需求规格说明书;④测试计划(初稿);⑤用户操作手册(初稿);⑥文档所引用的资料、采用的标准或规范。]

2. 任务概述

2.1 目标

2.2 运行环境

2.3 需求概述

2.4 条件与限制

3. 总体设计

3.1 处理流程

3.2 总体结构和模块外部设计

3.3 功能分配[表明各项功能与程序结构的关系]

4. 接口设计

4.1 外部接口[包括用户界面、软件接口与硬件接口。]

4.2 内部接口[模块之间的接口]

5. 数据结构设计

5.1 逻辑结构设计

5.2 物理结构设计

5.3 数据结构与程序的关系

6. 运行设计

6.1 运行模块的组合

6.2 运行控制

6.3 运行时间

7. 出错处理设计

7.1 出错输出信息

7.2 出错处理对策[如设置后备、性能降级、恢复及再启动等。]

8. 安全保密设计

9. 维护设计

[说明为方便维护工作的设施,如维护模块等。]

设计文档完成以后,就可以对软件设计进行评审。

5.2 习题

5.2.1 填空题

- (1) 软件设计阶段产生的最重要的文档是_____。
- (2) 软件结构是以_____为基础而组成的一种控制层次结构。
- (3) 反映软件结构的基本形态特征是_____。
- (4) 一个模块把数值作为参数送给另一个模块,这种耦合方式称为_____。
- (5) 两个模块通过全程变量相互作用,这种耦合方式称为_____。
- (6) 将与同一张报表有关的所有程序段组成一个模块,该模块的内聚性为_____。
- (7) 一个模块的作用范围指_____的集合。
- (8) 一个模块的控制范围指_____的集合。
- (9) 结构化设计以_____为基础映射成软件结构。
- (10) 为了防止软件概要设计的错误传播到开发的后续阶段,在概要设计文档完成以后,要进行_____。
- (11) 软件设计中划分程序模块通常遵循的原则是使各模块间的耦合尽可能_____。三种可能的模块耦合是:例如,一个模块直接引用另一个模块中的数据为_____;一个模块把开关量作为参数传送给另一个模块为_____;一个模块通过公共数据结构把数据传送给另一个模块为_____。其中,_____的耦合性最强。
- (12) 为高质量地开发软件项目,在软件结构设计时必须遵循_____的原则,_____建立软件系统的模块结构。并且应根据_____评价系统模块划分的质量。此外,在设计模块时,应从三种基本的_____出发,利用它们组合成一个模块的程序块结构。要求每个_____的结构应是单入口和单出口。
- (13) 在完成软件概要设计,并编写出相关文档之后,应当组织对概要设计工作的评审。评审的内容包括:分析该软件的系统结构、子系统结构,确认该软件设计是否覆盖了所有已确定的软件需求,软件每一成分是否可_____到某一项需求。分析软件各部分之间的联系,确认该软件的内部接口与外部接口是否已经明确定义。模块是否满足_____和_____的要求。模块_____是否在其_____之内。
- (14) 结构化设计方法中,要把数据流图转换成软件结构,若某个加工将它的输入流分离成许多发散的数据流,形成许多加工路径,并根据输入的值选择其中一个路径来执行,这种特征的DFD称为_____的数据流图。
- (15) 模块的独立程度可以由两个定性标准度量,这两个标准分别称为_____和_____。

(16) 结构化设计中以数据流图为基础的两具体设计方法是_____设计。

5.2.2 选择题

- (1) 结构化设计方法在软件开发中,用于()。
- A. 测试用例设计 B. 软件概要设计 C. 程序设计 D. 软件详细设计
- (2) 软件结构图中,模块框之间若有直线连接,表示它们之间存在着()关系。
- A. 调用 B. 组成 C. 链接 D. 顺序执行
- (3) 划分模块时,一个模块的()。
- A. 作用范围应在其控制范围之内 B. 控制范围应在其作用范围之内
C. 作用范围与控制范围互不包含 D. 作用范围与控制范围不受任何限制
- (4) 属于软件设计的基本原理是()。
- A. 数据流分析设计 B. 变换流分析设计
C. 事务流分析设计 D. 模块化
- (5) 软件设计阶段一般又可分为()。
- A. 逻辑设计与功能设计 B. 概要设计与详细设计
C. 概念设计与物理设计 D. 模型设计与程序设计
- (6) 好的软件结构应该是()。
- A. 高耦合、高内聚 B. 低耦合、高内聚
C. 高耦合、低内聚 D. 低耦合、低内聚
- (7) 结构图中,不是其主要成分的是()。
- A. 模块 B. 模块间传递的数据
C. 模块内部数据 D. 模块的控制关系
- (8) 软件设计一般分为总体设计和详细设计,它们之间的关系是()。
- A. 全局和局部 B. 抽象和具体 C. 总体和层次 D. 功能和结构
- (9) 在面向数据流的软件设计方法中,一般将信息流分为()。
- A. 变换流和事务流 B. 变换流和控制流
C. 事务流和控制流 D. 数据流和控制流
- (10) 软件设计中一般将用到图形工具,下列哪种(些)可用作设计的图形工具?()
- a. 结构图 b. 实体联系图 c. IPO图 d. 层次图
A. a和b B. c和d C. a,c,d D. 全部
- (11) 软件设计将涉及软件的构造、过程和模块的设计,其中软件过程是指()。
- A. 模块间的关系 B. 模块的操作细节
C. 软件层次结构 D. 软件开发过程
- (12) 模块独立性是软件模块化所提出的要求,衡量模块独立性的度量准则是模块的()。
- A. 抽象和信息隐蔽 B. 局部化和封装化
C. 内聚性和耦合性 D. 激活机制和控制方法
- (13) 模块的独立性是由内聚性和耦合性来度量的,其中内聚性是指()。

- A. 模块间的联系程度 B. 模块的功能强度
C. 信息隐蔽程度 D. 接口的复杂程度

(14) 软件结构是软件模块间关系的表示,下列术语中哪一个不属于对模块间关系的描述? ()

- A. 调用关系 B. 从属关系 C. 嵌套关系 D. 主次关系

(15) 软件设计中划分模块的一个准则是(①)。两个模块之间的耦合方式中,(②)耦合的耦合度最高,(③)耦合的耦合度最低。一个模块内部的内聚种类中(④)内聚的内聚度最高,(⑤)内聚的内聚度最低。

- ① A. 低内聚低耦合 B. 低内聚高耦合 C. 高内聚低耦合 D. 高内聚高耦合
② A. 数据 B. 非直接 C. 控制 D. 内容
③ A. 数据 B. 非直接 C. 控制 D. 内容
④ A. 偶然 B. 逻辑 C. 功能 D. 过程
⑤ A. 偶然 B. 逻辑 C. 功能 D. 过程

(16) 结构化程序设计主要强调的是()。

- A. 程序的规模 B. 程序的效率
C. 程序设计语言的先进性 D. 程序易读性

(17) 结构化设计方法把软件结构划分为(①)和(②)两大类,并提出了对应于以上结构的分析方法。

- ① A. 控制型 B. 变换型 C. 模拟型 D. 处理型
② A. 事务型 B. 分析型 C. 通信型 D. 功能型

(18) 软件结构使用的图形工具,一般采用()。

- A. DFD B. PAD C. SC D. E-R 图

(19) 在软件概要设计中,不使用的图形工具是()。

- A. SC B. IPO C. IDEF0 D. PAD

5.2.3 判断题

(1) () 如果两个模块之间传递的是数据结构,如高级语言中的数组名、记录名等,则这两个模块间存在数据耦合。

(2) () HIPO 法既是需求分析方法,又是软件设计方法。

(3) () 划分模块可以降低软件的复杂度和工作量,所以应该将模块分得越小越好。

(4) () 在网状结构中任何两个模块都是平等的,没有从属关系,所以在软件开发过程中常常被使用。

(5) () 信息隐蔽原则有利于提高模块的内聚性。

(6) () 中心变换型的 DFD 可看成是对输入数据进行转换而得到输出数据的处理,因此可以使用事务分析技术得到初始的模块结构图。

(7) () SD 法是一种面向数据结构的设计方法,强调程序结构与问题结构相对应。

(8) () 模块是可组合、分解和更换的单元,具有 4 个基本属性:接口、功能、逻辑、状态。

- (9) () 当模块的控制范围是其作用范围的子集时,模块之间的耦合度较低。
- (10) () 模块之间联系越紧密,其内聚性越强,耦合性越差。
- (11) () 总体设计的基本目的就是回答:“概括地说,系统应该如何实现?”这个问题。
- (12) () 模块独立要求高耦合低内聚。
- (13) () 数据流图表示了程序系统对数据的算法处理过程,即系统的物理模型。
- (14) () 数据流图一般可分为变换型和事务型两类。
- (15) () 模块化指解决一个复杂问题时,自顶向下逐层把软件系统划分成若干模块的过程。
- (16) () 事务型数据流图是由输入、变换、输出三部分组成。
- (17) () 模块间耦合高低取决于模块间接口的复杂性、调用关系和传递的信息。
- (18) () 总体设计是要把系统的各项功能需求转换成系统的体系结构。
- (19) () 模块间联系越紧密,内聚性就越强,模块的独立性就越高。
- (20) () 一个模块的深度表示软件结构中控制的层数,宽度是指软件结构一层中最大的模块个数。

5.2.4 简答题

- (1) 结构化分析方法的步骤是什么?
- (2) 变换分析设计与事务设计分析设计有什么区别?简述其设计步骤。
- (3) 什么是软件概要设计?该阶段的基本任务是什么?
- (4) 软件设计的基本原理包括哪些内容?
- (5) 衡量模块独立性的两个标准是什么?各表示什么含义?
- (6) 模块的耦合性有哪几种?各表示什么含义?
- (7) 模块的内聚性有哪几种?各表示什么含义?
- (8) 简述软件结构设计的优化准则。
- (9) 什么是模块的作用范围?什么是软件的控制范围?它们之间应建立什么关系?
- (10) 什么是“事务流”?什么是“变换流”?试将相应形式的数据流图转换为软件结构图。
- (11) 按照“降低块间联系,提高块内联系”的设计总则对模块进行修改,具体从哪些方面进行改进?
- (12) 在图 5.3 中,模块 C 中的一个判定对模块 F 产生影响。根据模块的作用范围与控制域的原则,判定图 5.3(a)和图 5.3(b)两图的正确性。
- (13) 图 5.4 中,模块 G 为判定,判断涉及模块 B、F、G,请指出设计中的错误,再根据改进模块图的基本原则,画出一两个改进方案(不改变模块 G 的判断关系),并说明是按照哪条基本原则进行改进的。
- (14) 考察图 5.5 中子图、父图的平衡。