

FoxBASE⁺

与管理信息系统编程

宗大华 卢雄远 等编著



国防工业出版社

FoxBASE⁺与管理信息系统编程

宗大华 卢雄远 等编著

国防工业出版社

· 北京 ·

图书在版编目(CIP)数据

FoxBASE⁺与管理信息系统编程/宗大华等编著. —北京:
国防工业出版社, 1995. 8

ISBN 7-118-01326-9

I. F... I. 宗... III. 关系型数据库-数据库管理系统 IV.
TP311.13

中国版本图书馆 CIP 数据核字(94)第 07031 号

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京怀柔新华印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 16 3/4 387 千字

1995 年 8 月第 1 版 1995 年 8 月北京第 1 次印刷

印数 1—4000 册 定价:19.50 元

(本书如有印装错误, 我社负责调换)

前 言

本书是专门为已初步掌握 FoxBASE⁺的命令语句、想进一步开发数据库管理软件但又缺乏实际编程经验的人撰写的。计算机本身并不会自动做任何工作,只有当人们使用程序设计语言,根据自己的设计思想编制出各种程序交给计算机去执行时,它才会实现人们的意愿。所以学会编程实在是一件把计算机真正应用起来的最为重要的事情。

虽然各种各样的功能设计都可以借助于顺序、判断、多分支、循环等程序结构描述出来,但真正想要把它们用得得心应手也并非易事。实践表明,初学者多阅读一些别人编写的各式程序,将大有裨益。在此基础上去进一步发挥、提高和创造,就会显得轻松、快捷,少走弯路。这正是本书的编写初衷。书中针对开发应用管理系统中经常见到的建库、录入、查删改、统计打印等问题,列举了大量可供读者参阅的例子。这些程序虽谈不上最佳和最为理想,但却都经过了实际验证,具有较好的可读性和参考价值。相信本书能起到抛砖引玉的作用,给予读者启迪和帮助。

本书由宗大华编写第一章、第六章到第九章;卢雄远编写第二、第三章;唐慧君编写第四章;卢震邈编写第五章,并为第二、三、四章提供了实例;宗涛参与了第六章到第九章的部分程序的调试工作。全书由宗大华统稿。蒋玮自始至终为书稿的整理尽心尽力,付出了辛勤的劳动,在此特表真挚的谢意!

编 者

2015.3.11

内 容 简 介

本书为作者《FoxBASE⁺数据库及其实用技术》(《最流行软件丛书》之一)的姊妹篇(提高篇)。作者积多年从事管理信息系统教学、科研工作的经验,结合软件工程理论,精心编著了本书。

书中不仅有生动的理论阐述,更多的是经验与技巧的敷陈。全书共九章,每一章出发点不同、侧重面各异,但实际上都是对 FoxBASE⁺知识的综合应用;内容包括软件开发的生存周期法,库结构设计原则、建库方法,应用程序中常见的各种数据录入方法,数据的查询,数据的删除与修改,统计和报表的输出,菜单及窗口程序设计,程序调试及系统维护,好程序的编写风格。书中的实例程序涉及教学管理、病案管理、办公室文档管理及房屋资产管理等诸方面。

本书适合于初步掌握 FoxBASE⁺基本知识的读者阅读,对实际从事应用信息系统开发工作的软件人员尤有参考价值。

目 录

第一章	信息系统的开发与 FoxBASE ⁺	1
1.1	计算机应用的分类	1
1.2	信息系统的特点	2
1.3	应用信息系统的开发方法	4
1.4	关于 FoxBASE ⁺	12
1.5	本书的展开方式	14
第二章	数据结构、库设计和库操作	16
2.1	常量、变量和基本数据类型	16
2.2	数据库结构	23
2.3	库结构设计	25
2.4	数据库的建立	30
2.5	库操作	31
第三章	数据录入	41
3.1	概述	41
3.2	数据录入模块的设计	41
3.3	代码技术	54
3.4	表格式录入程序	64
第四章	数据查询	70
4.1	基础知识	70
4.2	查询模块设计	71
第五章	数据删除与修改	85
5.1	删除模块的程序设计	86
5.2	修改模块的程序设计	87
5.3	删除、修改程序实例	88
第六章	数据的统计和报表输出	98
6.1	数据统计的各种命令	98
6.2	报表格式文件.FRM	111
6.3	报表输出程序的编制	120
第七章	菜单及窗口程序设计	135
7.1	各种菜单程序	135
7.2	窗口内容的行式推移	143
7.3	窗口内容的翻页技术	153
7.4	窗口内容的左、右滚动	164
第八章	程序调试及系统维护	173
8.1	TALK、STEP、ECHO、DEBUG 四条命令	173

8.2	断点的设置	188
8.3	程序调试中测试用例的选取	191
8.4	系统初始化与备份	211
第九章	程序设计的风格	223
9.1	好程序的标准	223
9.2	程序的层次结构和模块划分	235
9.3	FoxBASE+的过程文件	255

第一章 信息系统的开发与 FoxBASE⁺

1.1 计算机应用的分类

当前,总是把计算机的软件分为两大类,一是所谓的系统软件,一是所谓的应用软件。系统软件的典型代表是操作系统、各种程序设计语言以及它们的解释程序或编译程序等。这些软件被用来管理、监控和维护计算机的资源,它为使用计算机的用户提供各种便利的条件。在你购买计算机时,系统软件一般都由计算机厂家、公司作为系统的一部分向你提供。应用软件则不然,它们是用户在系统软件的基础上,为了解决各种实际问题而编制出的程序。

自从世界上第一台计算机于 1946 年问世以来,计算机作为一种高科技的象征而得到了突飞猛进的发展,它的应用深入到各个领域,范围越来越大。由于计算机在诞生初期,仅局限于用在快速数值运算上,“计算机”也因此而得名。现在,计算机在非数值运算方面的应用,远远超过了数值运算方面的应用,“计算机”这个名字就有点名不符实了。人们已开始用“电脑”来称谓它,试图用它来作为人脑的“延伸”。据估计,目前应用计算机的领域已在 5000 个以上,它们分布在社会的各个角落,执行着多种多样的任务。概括起来,按其用途大致可以有如下几类。

1. 科学计算,如天气预报,地震预测等。

在数学、物理以及工程技术等学科中,都会遇到大量的计算问题。这些问题的求解,大多采用近似解法,如求方程的根,方程组求解,以及定积分求值,等等。它们的公式复杂,迭代次数多,计算量相当大,以致于只有计算机才是解题的唯一手段。举例说,1500 年前我国的数学家祖冲之,用了若干年的时间计算出 π 值在 3.1415926 到 3.1415927 之间。后来英国数学家香克斯花费了 15 年时间,把 π 值计算到小数点后 707 位,可惜最后 100 多位还是错的。现在,使用计算机来完成这项任务,不仅计算准确,时间也只需几十分钟。

2. 数据处理,如编辑排版,图书检索等。

数据处理的特点是计算公式并不复杂但数据量大。譬如一个航空公司的预约订票文件,就是由表示一年里所有各次航班的座位信息的数据组成的。当然,它里面还应包括关于所有订票旅客的信息。这样一个系统不光要有文件的维护功能(即增加新记录、删除旧记录、修改现有记录等),还要有信息检索功能,各种统计功能,以及对文件采取的安全措施。目前世界上计算机在数据处理上的应用越来越多。

3. 实时控制,如飞机、列车调度,炼钢过程控制等。

借助于计算机快速处理问题的能力,可以用它来及时采集检测数据,对生产过程实现自动控制。譬如火车的货场编组站,把到达的货车按未来的去向加以解体之后,通过“驼峰”(一个高坡),把每一勾车按去向溜放到新的编组线上,以形成一列新的货车。由于勾车

的去向不同,下滑的距离远近不一,因此需要用计算机控制道岔的搬动,保证正确的溜放方向;用计算机对测速、测距雷达进行定时采样,从而实现对缓行器(安在轨道两侧的、类似于闸的东西)的动作控制,以保证溜放勾车的下滑速度和滑行距离等。

4. 计算机辅助系统,如辅助设计,辅助制造,辅助教学。

拿辅助教学来讲,即是充分利用计算机及其各种外部设备,利用声、图、色等来向学生传授知识,并可测试学生对概念的理解程度。学生可在计算机上“做”化学实验,物理实验,可以学习英语。计算机具有无限的耐心,而且能够一天 24 小时的工作,何时向它请教都行,它为人们的学习提供了很大便利。

5. 人工智能,如计算机诊断,机器人和机械手。

计算机下棋就属人工智能范畴。1993 年 10 月 28 日北京晚报在《谢军卫冕战花絮》栏目中曾报导说,参加这次大会的还有一位“电脑大师”,它曾在世界电脑国际象棋赛中获得冠军,这次赛会组织者把这位“电脑大师”带了来,并把它和赛场上的电脑并联上,无论哪方走出一歩棋,它都能把它的对弈着法演示出来。电脑能下棋,实际上就是使计算机具有“推理”和“学习”的功能,以使用它来“模仿”人的智能,这是计算机应用领域中的一枝新秀,前景必然十分广阔。

1.2 信息系统的特点

应用信息系统属于数据处理一类,它能将一个组织机构中的所有信息作为一个整体来加以管理,产生出明显的效益。在基层的组织机构中,有很多的日常事务工作,譬如财务部门的工资管理,办公室部门的公文收发管理,经销部门的产品信息管理、合同管理等,都是常见的文书性的数据处理工作,它们需要数据的录入,查询,修改,删除,打印输出等诸多功能。在中、上层的组织机构里,很多的业务工作是要在底层提供的信息基础上,对收集的信息加以处理、分类、分析,找到解决问题的方法和线索,由此提出可能的行动方案,并验证这些方案的可行性,最后从中选取出一个最佳的具体方案加以实施。这就是通常所说的领导部门作出预测和决策。因此,一个完整的应用信息系统,不仅能完成诸业务职能,提高数据处理的准确度,减轻人们的劳动强度,而且还应该能通过对信息的处理和分析,向领导提供有用而合理的结果,协助作出决策。

现在来看一个企业生产管理系统,它是采用数据库技术和计算机网络技术实现的计算机信息系统,它通过散布在企业各车间、科室的终端,实时地收集和提供信息,从而对企业生产作出安排和控制。

该应用信息系统共有 12 个子系统,如图 1-1 所示。

从图上可以看出,根据企业的工作流程,可以把整个系统划分为三大块:制订生产计划、制订作业计划、执行计划。

“设备与生产数据管理”子系统维持有整个企业产品的设计数据,工艺过程,以及所需材料等基本的技术参数和数据。

“用户订货服务”子系统将接收用户的订货合同单,并做相应的各种处理。

“预测”子系统的主要功能是对企业原始数据的统计、分析、调整,选择适当的预测模型,从而产生出在未来一段时间内的产品需求信息。

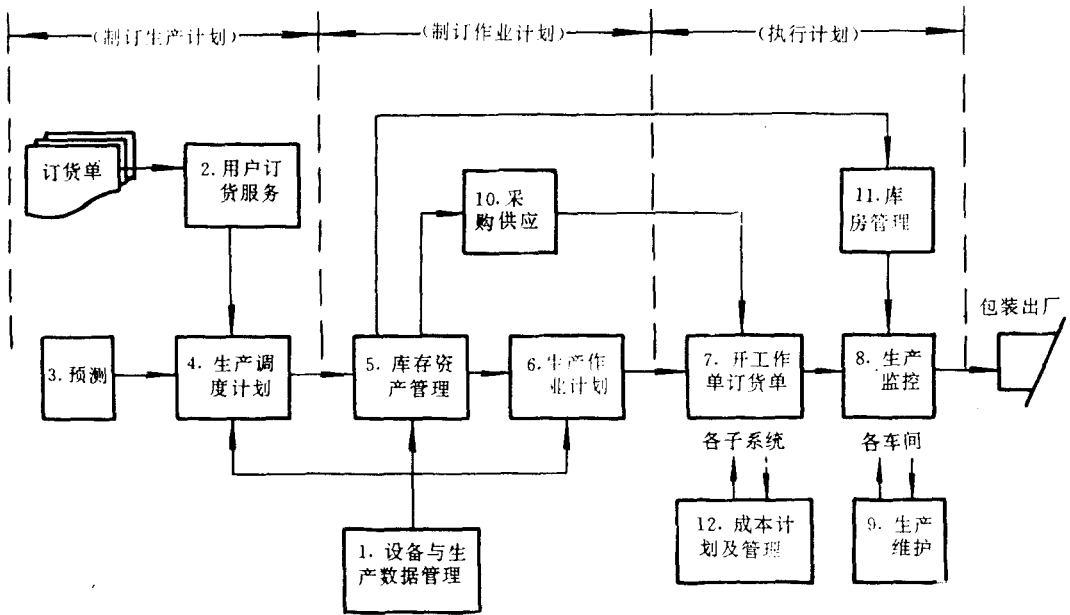


图 1-1 企业生产管理系统功能关系图

以这三个子系统的信息作为输入，“生产调度计划”子系统将制定出全年或较长一段时间内本企业诸产品的生产任务及进度要求，并提供生产中各类材料的需求量，设备的负荷状况，从而标志着制订生产计划阶段的结束，进入作业计划的制订。

“库存资产管理”子系统利用“设备与生产数据管理”子系统的信息以及“生产调度计划”子系统产生的生产计划，对完成这一计划所需要的原、辅材料，配件需求以及资金等作出具体安排，一方面把结果交给“生产作业计划”子系统去编制具体的班、组作业计划，把有关信息传送给“采购供应”子系统，以便提出采购计划；另一方面也把这些需求量提供给“库房管理”子系统，以便保证企业生产有条不紊地进行。

在执行计划阶段，“开工作单订货单”子系统与“成本计划及管理”子系统交往，以便根据作业计划、采购计划等开出生产工作单、下料单、订货单等。“生产监控”子系统将通过各车间的反馈数据以及“库房管理”提供的进货、存放、发料等方面的实物处理和帐目登记信息，及时对计划进行调整，更好地进行人员调配和考勤，以确保生产按计划完成，并包装出厂。

由上面粗略的描述可以看出，应用信息系统的开发，具有如下的几个特点。

第一，应用信息系统是一个复杂的系统，必须先把它分解成若干界线清晰、功能明确、易于解决的子系统，然后逐个加以实现，最后再拼装起来，形成一个“完整的系统”。

第二，应用信息系统诸子系统之间，是通过数据库来建立起联系的，数据库能向各个功能子系统提供必要的数据库，并保证它们之间对数据的有效共享。

第三，应用信息系统与具体的组织机构、管理职能有密切联系。因此在开发时，必须详细地向用户进行调查，以便能适应和满足用户的应用需要。

1.3 应用信息系统的开发方法

随着计算机应用领域的逐步扩大,软件的功能越来越丰富,内部的逻辑关系越来越复杂。因此在软件的开发过程中,虽然投入的人力、物力、财力不断增加,软件的质量却无法得到保证,软件的开发周期也越拖越长,不能按时提交给用户投入使用。这就是所谓的“软件危机”。于是人们开始考虑按照机械工程、建筑工程等中的“工程化”思想来解决软件研制过程中所面临的困惑,从而产生了“软件工程”这一新的学科。软件工程把软件开发过程,从初始到结束划分为若干个阶段,规定每一阶段应实现的目标和任务,然后按这些要求严格地一步一步去做,从而获得投资少、开发周期短、质量高的软件。在这一节里,我们主要介绍软件工程的开发方法——生存周期法。

1.3.1 软件开发的三个阶段

一个建筑工程(如立交桥、宾馆大厦等)从开始到结束,应该经历如图 1-2 所示的设计、施工和验收三个阶段。

在设计阶段,用户以书面的形式对建筑工程提出设计要求,设计人员据此进行设计,绘制出蓝图。在经过用户复审批准后,该工程就可以投入现场施工。施工阶段的依据就是设计图纸,它是不能随意修改变更的。施工完成后,用户将组织人员根据初始的设计要求以及蓝图,对工程进行逐项考核验收。通过后才能交付使用。

完全可以把这种工程化的实践方法运用到软件开发的过程中,形成如图 1-3 所示的定义、开发、维护的软件开发三个阶段。

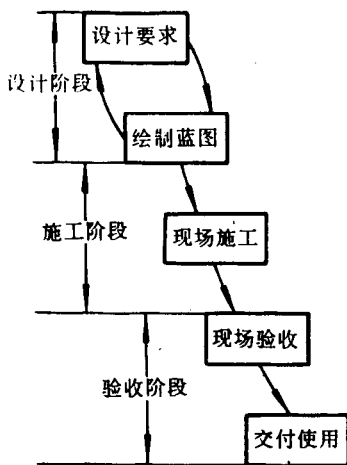


图 1-2 建筑工程的设计—
施工—验收三个阶段

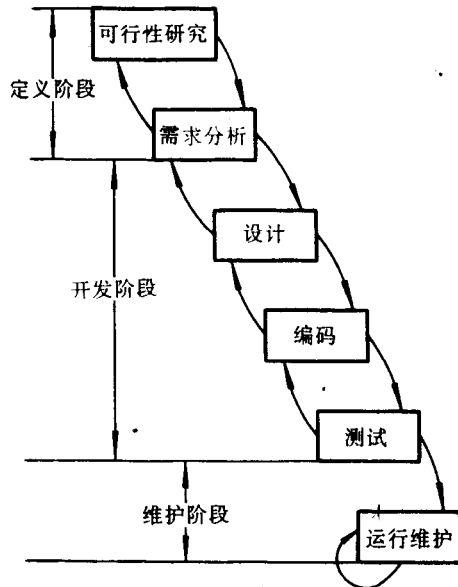


图 1-3 软件开发的三个阶段

由于软件是一种抽象的、逻辑性的产品,研制和维护本身实际都是人的“思考”过程,很难对这种“思考”进行控制。因此软件工程终究与其他工程科学很不相同。譬如说,图1-2中,设计蓝图产生之后,往下的每一步没有回溯的问题。但在软件开发过程中,每一步与它前一步都可能经历不只一次的修改和适应的回溯过程。又譬如软件在交付使用后,还有一个运行维护问题,这不仅是由于系统本身的需要(如发现了以前没发现的隐藏错误、运行环境有了变化等),也是由于用户要求的变更、扩大而随之带来的变动需要。

1.3.2 软件开发的生存周期法

从图1-3可以看出,软件产品与世间的一切事物一样,从发生、发展到成熟,以至淘汰衰亡,有一个历史的发展过程,这一过程在软件工程中被称为软件的生存周期。表1-1列出了生存周期三个阶段中每一步的基本任务和完成标志。

表 1-1 生存周期各步的基本任务

生存周期 各 步	基 本 任 务	结 束 标 志
可行性分析	确定系统实现的可能性	可行性报告
需求分析	确立系统的目标和总体要求	软件需求说明书
设 计	建立系统结构及诸模块的实现方法	设计说明书
编 码	根据设计说明书编写程序	程序文本
测 试	发现和排除诸模块的错误	可以运行的系统
维 护	运行和管理	改进开发的系统

一、可行性分析

可行性分析的基本任务是明确所要开发的系统是否有必要实现以及是否能够实现的问题,并不具体去解决问题。在一个项目投入实施之前,花费一定的人力和财力去论证可行性是很重要的,它会使我们避免盲目上马,陷入困境,甚至使项目中途夭折。

可行性研究的内容主要集中在探讨系统在技术上、经济上和管理上的实现可能性。在当前技术的基础上,该系统可以实现吗?本单位能保证投资吗?新系统实现后的收益会大于投资费用吗?本单位有能力完成该系统的开发或交付使用后本单位能承担得起系统的维护工作吗?这都是一个新系统提出后,必须率先回答和解决的问题。

可行性研究的最终结果应该拿出一份可行性报告。该报告应对下一步的工作作出明确的建议:如果项目不值得去搞,或目前不具备开发的各种条件,那么应该列出理由,建议停止这一项目,以免在人力、物力、财力上造成浪费和损失;如果项目值得投入,并且有解决的办法,那么就应该在报告中列出可供选择的各种方案,阐述理由,推荐其中认为最优的一种,并为项目的实施(进度、资金预算、人力安排等)作一个初步的设想。

二、需求分析

在通过可行性论证、确定项目可行之后,就应该组织人员对该现行管理机制进行详细的调查,内容应该包括现行的组织机构,现行的业务流程、信息流程,数据的输入要求、输

出要求,数据的格式和数据量,等等。需求分析的目的就是要弄清楚新开发的系统必须做些什么,必须具有什么功能。这一步工作是下面各步工作的基础,它进展及完成的好坏,将会直接影响到未来新系统的设计质量。

目前的实际情况是,设计人员具备计算机软件开发的方法和技术,但对使用单位的业务不熟悉和不了解;相反,使用单位(即用户)对本身的业务范围、性质、工作流程等极为清楚,但缺乏计算机知识,不了解计算机都能干些什么事情。因此在进行需求分析时,必须由双方人员密切配合,精诚合作,才能准确、清晰地勾画出目标系统的具体要求,取得共识。

由于需求分析是要弄清楚用户的真正要求,所以常常采用召开各种类型、各种人员参加的座谈会、发调查表、或亲自参与实践的方法来进行。譬如表1-2给出了一种调查报表数据的调查表格格式。由该表可以知道物资公司现有哪些数据表格,它们各自的用途,报表中的数据来源于何处,报表的使用频率(是日报还是月报等)、信息量的大小(即有多少项,或是页数乘行数乘列数),等等,这些信息对于系统开发过程中设计数据库、计算存储量等都是极有价值的。

在开发人员和用户对所要设计的系统达成一致看法之后,就必须把这种共同的认识和理解明确地写成一份文档,这就是所谓的“软件需求说明书”。由于这是双方的共识,因此它将成为以后软件设计、测试等的重要依据。所以它是该软件生存周期中的一份具有举足轻重意义的文档。正由于如此,软件需求说明书一定要简洁、完整、准确,无二义性,条理清楚,易于阅读理解。

表1-2 物资公司报表数据调查表

部门: _____ 编号: _____ 第 _____ 页

序号	表名	用途	数据来源	报表期限					主管单位	保留期限	信息量 (项) 或 (页*行*列)	存在问题
				年报	季报	月报	旬报	日报				

填表人: _____ 填表日期: _____ 年 _____ 月 _____ 日 审批: _____

三、设计

这里所说的设计,并不是动手进行程序编写的意思,而是根据需求分析对目标系统的描述,从整体上解决系统如何实现的问题。在“设计”这一步,应该先按功能将系统划分成若干个子系统,使问题得以由繁化简;应该确定诸子系统功能由哪些更小的模块组成,以及这些模块之间的相互关系;更进一步,还应该设计出系统各数据库的结构,给出各模块实现的算法,模块间的接口,具体的输入输出格式、菜单格式、用户界面等。由此看出,这一步相当于建筑工程中绘制蓝图的作用。任何一个建筑施工队拿到了绘制出的蓝图后,都能建成相同的建筑物。因此在软件工程中,应该由“设计”这一步,产生出编写具体程序

的设计“蓝图”,无论哪一位程序员拿到这一阶段所产生的设计说明书,都应该能编写出功能相同的、满足接口要求的程序。

为了能够反映出各程序模块在系统中的相互关系,它所需要的输入数据和产生的输出,以及它所完成的工作的处理算法,在生存周期的“设计”一步,经常采用一种称为 IPO 图的图形工具。IPO 图是输入-处理-输出图的简称,它是由美国 IBM 公司推出并发展完善起来的,用它能够方便地描绘输入数据、对数据的处理以及输出数据三者之间的关系。

假定有图 1-4 所给的模块调用关系,它反映出在计算某人的工资总额时,要调用“计算正常工资”、“计算加班费”和“计算奖金”三个子模块。

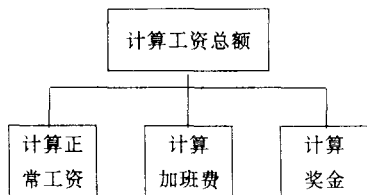


图 1-4 一个模块调用关系图

图 1-5 给出了“计算工资总额”模块相应的 IPO 图,图 1-6 给出了“计算正常工资”模块相应的 IPO 图,图 1-7 给出了“计算加班费”模块相应的 IPO 图,限于篇幅,没有给出“计算奖金”模块的 IPO 图,但由图 1-5 到图 1-7 已完全可以了解如何填写模块的 IPO 图以及该图的重要作用了。

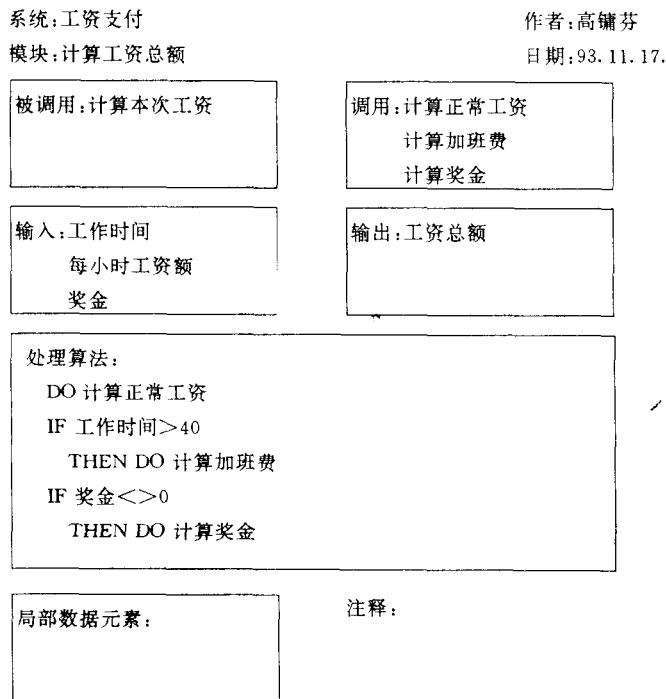


图 1-5 “计算工资总额”模块的 IPO 图

系统:工资支付
模块:计算正常工资

作者:高镛芬
日期:93.11.17.

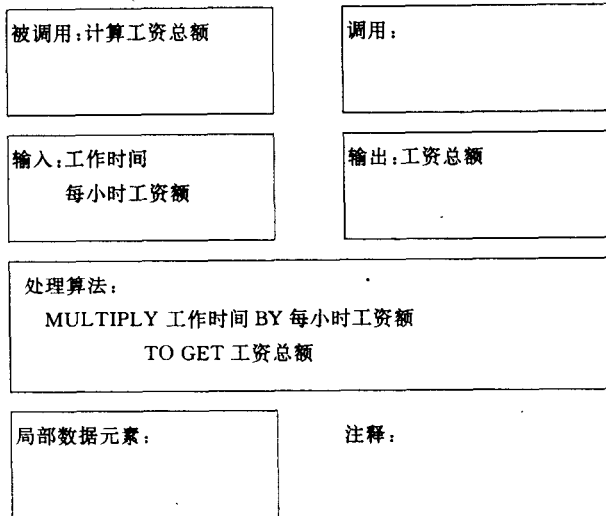


图1-6 “计算正常工资”模块的 IPO 图

系统:工资支付
模块:计算加班费

作者:高镛芬
日期:93.11.17.

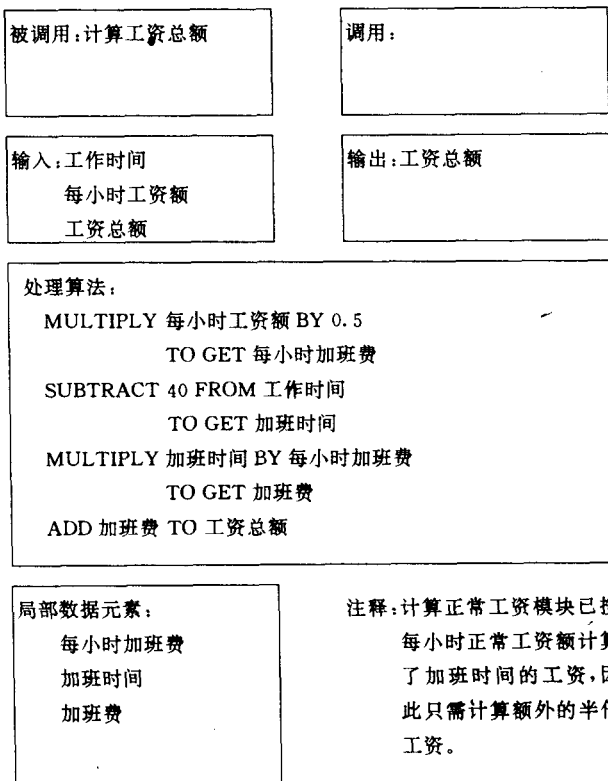


图1-7 “计算加班费”模块的 IPO 图

在图1-5中,“被调用”框给出了这一模块与上级的关系,“调用”框给出了这一模块与下属的关系。由此可见,是计算本次工资模块调用计算工资总额模块,而计算工资总额模块又要调用计算正常工资、计算加班费以及计算奖金三个模块,这正是图1-4所描绘的情形。“输入”框表示在调用该模块时,应该由上级模块提供(或由用户直接键入)工作时间、每小时工资额以及奖金这三个信息。“输出”框表示该模块在获取输入数据后,经过处理算法的加工所产生的结果,就是工资总额。“处理算法”框常用结构化的伪语言(非正式的、可英汉混用的一种描述性语言)来描述该模块的程序主要完成的功能和各部分的逻辑关系。这里:

DO 计算正常工资

表示去调用计算正常工资模块,从而得到某人应发的正常工资。这里:

IF 工作时间 > 40

THEN DO 计算加班费

表示假如输入的工作时间大于40,那么就应该去调用计算加班费模块,以求得此人应得的加班费用。

在图1-6中,“调用”框里没有内容,表示该模块已是最基本、最简单的功能模块了,它只被别的模块调用,而自己无需再调用其他模块就能完成自己承担的任务。在“处理算法”框里,

MULTIPLY 工作时间 BY 每小时工资额

TO GET 工资总额

表示用工作时间乘以每小时工资额,就得到了工资总额。这里要注意,这个工资总额并不是最终的工资总额,它仅是一个中间结果,是计算工资总额模块(见图1-4和图1-5)里调用计算正常工资(即 DO 计算正常工资)后,返回给调用者的输出数据。

在图1-7“处理算法”框里描述的是先求得每小时加班费,然后从工作时间里减去40而得到加班时间,再求出具体的总加班费,最后把总加班费与工资总额(即图1-6中输出的结果)相加,从而返回新的工资总额。可以看出,这时返回给计算工资总额模块的仍是一个中间结果,只有再加上奖金,才能得到最终的工资总额值。

“局部数据元素”框,表示这里列出的变量是该模块自己定义的变量,它与“输入”框列出的变量不同,那里的变量是上级模块传递过来的,局部数据元素只是局部于本模块的变量,退出本模块的作用范围,该变量也就自动消除了。

把如上的 IPO 图交给程序员去进行程序编写,由于所编程序应该完成的功能已由处理算法确定,它与外界的联系(输入与输出)也已经全部约定好,所以就能够使不同的程序员编写出基本相同的程序来。

至于在开发一个系统时,应该如何去建立系统的结构,应该以什么原则去划分模块,模块的尺寸应该以多大为宜,模块之间应该是什么样的关系对程序的编写、测试更为有利等问题,在本书的第九章都有讲述。

四、编码

编码就是根据设计说明书来编写程序。由于在设计阶段已经得到了系统的总体结构,模块的划分,模块的 IPO 图等,因此编码应该说是一项水到渠成的工作。通过编码,将产生整个系统的程序文本,同时也应该产生出详细的用户使用手册。这里主要的问题是对编

码语言的选取。

程序设计语言是人与计算机进行通信的最为基本的工具,某种程序设计语言的特性,不可避免地会影响人们的思维 and 解决问题的方式,会影响到人和计算机通信的质量和便利程度,也会影响到其他人阅读和理解所编写程序的难易程度。因此在选取程序设计语言时,应考虑如下几个方面的问题。

第一,该语言应具有结构化的控制结构和模块化的机制。

第二,该语言应具有很好的可读性。

第三,尽量选择一种已经为程序员所熟悉的语言,因为即使是对一个有经验的程序员来讲,虽然学习一种新的编程语言并不会有有多大困难,但要完全掌握它并能熟练地加以运用,却需要通过一定的实践才行。

第四,如果所开发的系统将由用户负责维护,那么应该尽量选用他们熟悉的语言编程。

五、测试

正如表1-1所示,测试的基本任务是发现和排除程序中的错误,以便能提交一个性能良好、与用户友好、可以正常运行的系统。

怎样强调“测试”在软件开发生存周期中的重要性都不显得过分,测试工作完成的好坏,会直接影响到所开发软件的质量。实际上,在软件生存周期的每一步,都应该安排自身的审核工作,以便能尽早地发现存在的问题和隐患,及时加以排除。之所以这样说,有下面两方面的原因。

第一,从统计资料看,对在“测试”这一步发现的错误进行追根溯源,真正属于编程错误的只占36%,其余64%的错误都出自于需求分析和设计。譬如说,由于需求分析调查时的疏忽,把数据库中某一字段的长度少算了一位。等到测试时,才发现这一字段的某些取值会产生溢出。现象是在对模块测试时表露出来的,但原因却是需求分析结束时未能进一步审核,而把错误“遗传”了下来。

第二,错误发现得越晚,要彻底纠正时所需付出的代价则越高。仍以上述的为例,若在需求分析结束时发现这个错误,则只需在信息长度上加1即可;若在设计时发现,则需要修改数据库的结构,当然还要修改需求分析时产生的“软件需求说明书”;若到测试时发现再修改,前述工作不但要做,还要做程序的修改,它不但会涉及信息的输入程序,还会要改动输入输出的格式等,付出的精力和代价明显成倍增加。

至于如何对程序进行测试,测试有哪些方法,如何设计测试用例等问题,本书第八章将结合 FoxBASE+ 的程序设计做介绍。

六、运行维护

一个软件投入运行之后,还需要进行维护。从软件开发生存周期来讲,运行维护阶段的时间最长,任务也最为繁重。

排除故障是维护的日常性工作,正如后面第八章所讲,一个经过测试验收而投入运行的系统,绝对不能说它是完全正确的,而只能说到目前为止还未能找到它的错误所在。系统投入正式运行,实际上是“测试”工作的延伸。运行过程中肯定还会有这样那样的问题需要解决,这样或那样的错误需要修改、更正。这就是所谓的改正性维护。

第二项维护工作是所谓适应性维护。在系统运行过程中,可能会出现操作系统版本的