

JISUANJI
RUANJIAN SHEJI
YU KAIFA CELUE

计算机软 件设计 与开发策略

索红军◎著



北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

014034265

TP311.52
386

渭南师范学院出版专项经费资助项目

渭南师范学院项目 (12YKS030)

计算机软件设计与开发策略

索红军 著



TP311.52
386

 **北京理工大学出版社**
BEIJING INSTITUTE OF TECHNOLOGY PRESS



北航

C1722518

内 容 提 要

本书是作者参加工作以来在西安卫星测控中心开发计算机软件及渭南师范学院任教期间积累的经验结晶和知识探索。全书结合软件产业和当前教育,主要介绍了程序与软件的关系,说明了早期人们和现在人们对程序和软件的认识差异,提出了作者对于软件危机的一些认识及应用软件工程的方法来缓解软件危机的一些见解,总结了作者在开发软件项目时的一些注意事项和经验小技巧,特别是对于初学软件设计开发的人员,给出了软件与程序设计方面容易忽视的问题及在软件设计策略方面的一些知识。

全书内容丰富,讲解清晰、易懂,详略得当,体现了现代计算机软件设计开发的精髓,帮助读者领悟真实的软件项目设计开发过程中的重点和注意事项,适合学习计算机软件设计与开发及相关专业的人士阅读。

版权专有 侵权必究

图书在版编目(CIP)数据

计算机软件设计与开发策略/索红军著. —北京:北京理工大学出版社, 2014. 1

ISBN 978 - 7 - 5640 - 8665 - 7

I. ①计… II. ①索… III. ①软件设计 IV. ①TP311.5

中国版本图书馆 CIP 数据核字 (2013) 第 301336 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

82562903 (教材售后服务热线)

68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 天津紫阳印刷有限公司

开 本 / 710 毫米 × 1000 毫米 1/16

印 张 / 12

字 数 / 216 千字

版 次 / 2014 年 1 月第 1 版 2014 年 1 月第 1 次印刷

定 价 / 45.00 元

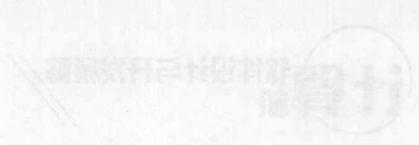
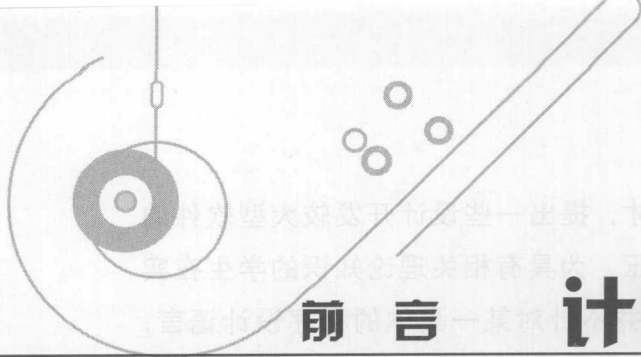
责任编辑 / 王俊洁

文案编辑 / 侯瑞娜

责任校对 / 周瑞红

责任印制 / 马振武

图书出现印装质量问题,请拨打售后服务热线,本社负责调换



前言

软件设计与开发策略 计算机

信息技术的发展进入了蓬勃发展的时期，作为信息技术的支柱——软件，在目前社会无处不在，计算机、通信、工业控制、仪器仪表、智能家电等设备都离不开软件。然而由于软件缺陷或错误造成的系统死机或崩溃，经常会造成难以弥补的损失，这种现象随处可见。而且和硬件相比，软件更是集中了大量的人工智力劳动成果，软件危机从20世纪70年代就开始出现并且一直持续，世界各国对软件人才的需求越来越大。设计与开发软件无形中成为一个范围非常广泛的工作，培养软件人才也成为社会的必然选择。我国大学本科阶段开设的计算机软件方面课程主要集中在高级语言、数据结构、算法分析、软件工程等课程，这些课程一般偏重理论，学生学完之后很难真正获得软件设计与开发方面的实际经验，需要经过长期的工作过程，才能逐步理解、体会到软件设计与开发的精髓。

反观当今中国，在软件的研发上也问题成堆。诸如：为什么当代很多大学生学了程序设计课程却不会写程序？对于同一问题，一百个中国人就能写出一百个变量命名不同、程序风格各异的软件，而印度、以色列等软件强国的学生却能写出几乎完全相同、如出一人之手的软件。软件工业化生产的流程是什么？软件设计中必须注意哪些问题？软件设计过程中最重要、最应该关注的是什么？这一切都需要认真思考。

鉴于此，作者针对大学本科计算机软件及相关专业学生以及步入

软件研发领域不久的初级软件人才，提出一些设计开发较大型软件的方法、技巧和策略，以求抛砖引玉，为具有相关理论知识的学生在实践方面提供一点帮助。尽管本书并不针对某一具体的程序设计语言，对于未学过程序设计语言的人来说读了它也的确不能进行设计程序，但对于某种程序设计语言已成为必修课的当代中国大学生，对于想尽快遨游于软件设计领域的初级软件设计人员来说却一定会有很大的启迪。因为全书以证实思维、逻辑思维、计算思维三大科学思维为指导，自始至终贯穿着软件工程的思想，较好地回答了困扰大学生们诸如学习程序设计而不会设计程序等许多问题。它对于培养大学生们的证实思维、逻辑思维、计算思维三大科学思维很有启迪和帮助。

本书的编写得到了李尧龙教授、张同琦教授、谢膺白教授、张郭军教授等的大力支持，阴国富老师、贺清兰老师也为本书的完成提供了很大帮助，北京理工大学出版社在本书的出版过程中提供了很大方便，在此一并表示感谢。本书的出版受“渭南师范学院出版专项经费资助项目”资助。

索红军

2013年8月于渭南

目 录

第 1 章 概论	001
1.1 什么是软件	002
1.2 软件的发展	003
1.3 软件的版本	004
1.4 软件的设计开发	005
第 2 章 软件与程序	006
2.1 什么是程序	006
2.2 软件与程序关系	009
第 3 章 软件与软件工程	010
3.1 软件生命周期	010
3.1.1 软件生命周期概述	010
3.1.2 软件生命周期模型	012
3.2 软件危机	014
3.2.1 软件危机的主要表现	015
3.2.2 软件危机的原因	016
3.2.3 软件危机的解决途径	018
3.2.4 相关因素	019
3.2.5 软件危机的具体表现	020
3.3 软件工程	021
3.3.1 什么是软件工程	022
3.3.2 软件工程的原理和目标	023
3.3.3 软件工程的原则	023
3.3.4 软件工程项目管理的任务	024
3.4 面向对象软件设计开发	025
3.4.1 面向对象的基本概念	026
3.4.2 面向对象的基本特征	026

3.4.3 面向对象的软件设计开发	030
3.5 软件测试	034
3.5.1 单元测试	035
3.5.2 集成测试	036
3.5.3 确认测试	044
3.5.4 系统测试	045
3.5.5 验收测试	046
3.6 软件测试的现状与前景	056
3.7 软件测试的相关问题	059
第4章 软件设计经验与技巧	064
4.1 软件工程过程中经验与技巧	064
4.2 程序设计经验与技巧	075
4.3 个人与团队是软件开发永远的核心	080
4.4 用最少代码降低安全风险	084
4.5 程序员必须注意的十大安全技巧	087
第5章 软件设计策略	098
5.1 软件架构	098
5.2 软件架构设计理解	112
5.3 软件项目计划与管理	115
5.3.1 软件项目的计划	117
5.3.2 软件项目的控制	118
5.3.3 软件项目管理的组织模式	119
5.3.4 软件项目管理的内容	120
5.3.5 编写软件项目计划书	120
5.3.6 软件配置管理	121
5.3.7 人员组织与管理	121
5.3.8 软件过程能力评估	124
5.3.9 为什么要有项目管理	125
5.3.10 为什么要有专职的项目经理	126
5.3.11 项目组的范围界限在哪里	126

5.3.12	怎样才能算是一个成功的项目	127
5.3.13	软件项目管理的成功原则	127
5.3.14	软件项目管理在管理思维中的空白	131
5.4	软件项目管理总体流程设计	134
5.5	项目计划书的编写	137
5.5.1	引言	137
5.5.2	项目概述	140
5.5.3	项目团队组织	143
5.5.4	实施计划	144
5.5.5	支持条件	148
5.5.6	预算	149
5.5.7	关键问题	150
5.5.8	专题计划要点	150
5.6	软件项目管理中十个误区	150
第6章 软件质量保证		154
6.1	软件质量保证的基本目标	154
6.2	软件质量	155
6.2.1	软件质量的由来	155
6.2.2	软件质量的现在	156
6.2.3	软件质量的未来	156
6.3	软件质量的理论探索	157
6.3.1	过程的认识	157
6.3.2	生产线的隐喻	158
6.3.3	SQA 和其他工作的组合	158
6.3.4	QA 和 QC	159
6.3.5	QA 和 SEPG	160
6.3.6	QA 和组织级的监督管理	161
6.4	SQA 的工作内容和工作方法	161
6.4.1	计划	161
6.4.2	审计/证实	162
6.4.3	问题跟踪	162
6.5	SQA 的素质	162

6.6	SQA 活动	162
6.7	正式技术评审	163
6.7.1	目标	164
6.7.2	评审会议	164
6.7.3	评审总结报告	164
6.7.4	评审指导原则	164
6.8	质量保证与检验	165
第7章 软件保护		166
7.1	软件的许可分类	166
7.2	保护软件知识产权	168
7.3	软件获得法律保护的条件	169
7.4	软件著作权的归属	169
7.5	软件的技术保护	171
第8章 后记		173
8.1	我国软件产业的发展现状	173
8.2	我国发展软件产业存在的问题	174
8.3	中国发展软件产业的对策建议	175
附录 计算机软件保护条例		178

第1章

概 论

当前,计算机已被广泛地应用到生产、生活中的各种领域。借助计算机这一有效工具,人类不仅可以进行各种繁重的科学计算,还可以进行信息处理、工业控制、辅助设计以及智能模拟等工作。

计算机的种种强大功能都是建立在硬件和软件的基础上,两者相辅相成,缺一不可。硬件是存储、处理数据的基础,提供了一个平台环境;而软件是依附于硬件,并实现与用户交互,间接控制硬件来发挥强大的功能。简单地说,硬件犹如计算机的躯体,而软件似于计算机的灵魂。

近年来,软件技术的发展十分迅速,支持不同领域的各种软件层出不穷,它们逐渐改变着人们的生产、生活方式,尤其是各种工具软件,使计算机发挥出更大的效能。本章将以当前计算机软件技术的发展为依托,对计算机软件的一些基础知识进行详细的介绍。

软件是一系列按照特定顺序组织的计算机数据和指令的集合。计算机中的软件,不仅指程序,也包括各种关联的文档、数据以及其他有关处理问题的方法等。根据计算机软件的用途,可以将其分为两大类,即系统软件和应用软件。系统软件的作用是控制并协调计算机硬件的工作,提供一个统一的接口给应用软件;而应用软件则只针对某一特定任务或特殊目的而开发,完成特定任务或特殊目的。

系统软件的作用是协调各部分硬件的工作,并为各种应用软件提供支持,使计算机用户和其他软件将计算机当作一个整体,不需要了解计算机底层的硬件工作方式,即可使用这些硬件实现各种功能。

系统软件主要包括操作系统和一些基本的工具软件,如各种编程语言的编译软件、硬件检测与维护软件以及其他一些针对操作系统的辅助软件等。

应用软件是为完成某一特定任务或特殊目的而开发的软件,可以是一个

特定的程序，也可以是一组功能紧密协作的软件集合体，或由众多独立软件组成的庞大软件系统。应用软件是基于系统软件工作的，因此不面向最基础的硬件，只根据系统软件提供的各种资源进行运作。

应用软件包括专用软件和通用软件两大类。专用软件是指专门为某一个指定的任务设计或开发的软件，例如，专门为某个单位开发设计的财务管理软件等，这类软件只能为一个用户或小部分用户所使用，购买或委托软件公司开发设计这类软件一般费用比较高。通用软件是指可完成一系列相关任务的软件，例如，处理文本、制作网页的各种软件等，这类软件用户量大，购买这类软件一般费用较低。

1.1 什么是软件

具体来说，完整的计算机系统由两部分组成，即计算机硬件系统和软件系统。前边对软件做了一些概要性的说明，那么究竟什么是软件呢？软件（中国大陆及中国香港用语，中国台湾地区称作软体，英文：Software）是一系列按照特定顺序组织的计算机数据和指令以及文档资料的集合。早些时候，软件也是程序的代名词。软件并不只是包括可以在计算机（这里的计算机是指广义的计算机）上运行的电脑程序，与这些电脑程序相关的文档一般也被认为是软件的一部分。简单地说，软件就是程序、数据和文档资料的集合体。

为了理解软件，可以假想一下领导要做报告，做报告的场所，做报告用的音响设备、桌子、凳子等都被称为硬件，而领导的讲话稿，领导讲话的风格、语气、方式等都被称为软件。

可以理解，计算机软件（Computer Software）是计算机系统中除硬件以外的所有事物，一般包括计算机程序、应用数据、程序说明以及其他文档资料等。

软件的正确含义应该是：

- （1）运行时，能够提供所要求功能和性能的指令或计算机程序集合。
- （2）程序能够满意地处理信息的数据结构。
- （3）描述程序功能需求以及程序如何操作和使用所要求的文档。

软件具有与硬件不同的特点：

- （1）表现形式不同。

硬件有形，有色，有味，看得见，摸得着，闻得到。软件无形，无色，

无味,看不见,摸不着,闻不到。软件大多存在于人们的大脑里、纸面上或存储设备中的数据信息中,它的正确与否,是好是坏,只有到程序在机器上运行时才能知道。这就给设计、生产、维护和管理带来许多困难。

(2) 生产方式不同。

软件是开发,是人的智力的高度发挥,不是传统意义上的硬件制造。尽管软件开发与硬件制造之间有许多共同点,但这两种活动是根本不同的。

(3) 要求不同。

硬件产品允许有误差,而软件产品却不允许有误差。

(4) 维护不同。

硬件是会用旧用坏的,在理论上,软件是不会用旧用坏的,但在实际上,软件也会变旧。因为在软件的整个生存期中,一直处于改变(维护)状态。

在现代社会中,软件应用于多个方面。典型的软件比如有电子邮件、嵌入式系统、人机界面、办公套件、操作系统、编译器、数据库、游戏等。同时,各个行业几乎都有计算机软件的应用,比如工业、农业、银行、航空、政府部门等。这些应用促进了经济和社会的发展,使得人们的工作更加高效,同时提高了生活水平。

1.2 软件的发展

从计算机产生开始,计算机软件就开始存在,其发展大致经历了下列四个阶段:

(1) 早期阶段。

在计算机发展的早期阶段,人们认为计算机的主要用途是快速计算,软件编程简单,不存在什么系统化的方法,开发没有任何管理,程序的质量完全依赖于程序员个人的技巧。

(2) 第二阶段。

计算机软件发展的第二阶段跨越了从20世纪60年代中期到70年代末期的十余年,多用户系统引入了人机交互的新概念,实时系统能够从多个源收集、分析和转换数据,从而使得进程的控制和输出的产生以毫秒而不是分钟来进行,在线存储的发展产生了第一代数据库管理系统。

在这个时期,出现了软件产品和“软件作坊”的概念,设计人员开发程序不再像早期阶段那样只为自己的研究工作需要,而是为了用户能更好地使

用计算机。人们开始采用“软件工程”的方法来解决“软件危机”问题。

(3) 第三阶段。

计算机软件发展的第三阶段始于 20 世纪 70 年代中期, 分布式系统极大地提高了计算机系统的复杂性, 网络的发展对软件开发提出了更高的要求, 特别是微处理器的出现和广泛应用, 孕育了一系列的智能产品。软件开发技术的度量问题受到重视, 最著名的有软件工作量估计 COCOMO 模型 (Constructive Cost Model, 构造性成本模型, 是一种精确的、易于使用的基于模型的成本估算方法)、软件过程改进模型 CMM (CMM 是指“能力成熟度模型”, 其英文全称为 Capability Maturity Model for Software, 英文缩写为 SW-CMM, 简称 CMM) 等。

(4) 第四阶段。

计算机软件发展的第四阶段是强大的桌面系统和计算机网络迅速发展的时期, 计算机体系结构由中央主机控制方式变为客户机/服务器方式, 专家系统和人工智能软件终于走出实验室, 进入了实际应用, 虚拟现实和多媒体系统改变了与最终用户的通信方式, 出现了并行计算和网络计算的研究, 面向对象技术在许多领域迅速取代了传统软件开发方法。

在软件的发展过程中, 软件从个性化的程序变为工程化的产品, 人们对软件的看法发生了根本性的变化, 从“软件 = 程序”发展为“软件 = 程序 + 数据 + 文档”。软件的需求成为软件发展的动力, 软件的开发从自给自足模式发展为在市场中流通以满足广大用户的需要。软件工作的考虑范围也发生了很大变化, 人们不再只顾及程序的编写, 而是涉及软件的整个生命周期。

在使用软件之前, 首先需要获取软件的安装程序或使用程序。在获取软件的程序文档后, 还需要掌握软件安装和卸载的技巧等。

1.3 软件的版本

随着用户需求的发展变化以及硬件环境的变化、软件暴露出的缺陷等, 软件在功能上、性能上逐渐不能适应用户的需求或不能适应硬件环境的变化, 这时就必须对软件进行修改, 但又不足以设计开发新的软件, 于是就在原先软件的基础之上做简单的修改, 形成原先软件的一个新版本, 这就引出了软件版本。软件版本是体现软件开发进度的一种标志, 也是帮助用户了解软件发布情况的重要工具。

软件是一种虚拟化的商品，但和现实中的各种商品一样，生产（在计算机领域中被称作开发）的时间有先有后。

由于计算机程序不断发展，各种软件程序的代码越来越复杂。因此，任何软件都难以避免出现各种漏洞或错误（在软件开发领域被称作 Bug，即虫子）。因此，软件发行以后，开发者通常会开始为用户提供各种更新的补丁程序。

当软件的更新积累到某种程度，或增加了重要的功能后，开发者往往会重新将软件封装，再次发行。对于同一个软件而言，版本就是标识这些不同时间发布的软件产品的一种重要标志。通常，每一个版本的软件，都会包括一个唯一的版本号。

当软件在功能上、性能上或缺陷方面的修改比较大时，其修改成本将大大增加，若修改软件的成本大到不值得修改原有软件而应该开发设计新的软件时，原有软件就可以舍去，我们称之为软件生命期结束，于是就不再为该软件修改产生新版本，而应该设计开发新的软件了。

1.4 软件的设计开发

一个完整的计算机系统，离不开软件系统。缺少软件的计算机就好比是没有任何电视信号的电视机一样，没有任何意义。那么软件怎么来的，是谁做出软件的呢？

人们知道，自从计算机诞生以来，计算机软件就随之而有。但早期的计算机软件是应用计算机的人员自己编写，也就是自给自足，软件的设计也没有任何统一的方法，全是个人的创作结晶。随着计算机硬件技术的大力发展，计算机的应用范围迅速扩大，出现专门为别人服务、进行软件设计开发的专职人员，同时软件危机也慢慢显现，也有人研究软件设计与开发的方法等。到现在，为软件设计与开发出现了软件工程学科，各种软件设计的方法及管理技术也应运而生，软件设计呈现出蓬勃发展的前景。

软件设计思路和方法的一般过程包括设计软件的功能和实现的算法、方法，软件的总体结构设计和模块设计，编程和调试，程序联调和测试以及编写、提交程序等。

第2章

软件与程序

虽然前面说明了什么是软件以及软件的发展等，但是，对于一些初学计算机知识，特别是学习计算机高级程序设计语言的人员来说，对软件的理解很大程度上集中在程序上，认为软件就是程序，开发软件就是编写程序。这实际是对软件的一个很大误解，程序只是软件的一部分，编写程序是开发软件的一个环节，而且一般是所占比重比较小的一个环节。那么，具体什么是程序，软件又和程序有什么关系呢？在此进行简单的说明。

2.1 什么是程序

什么是程序？简单地说，程序就是计算机指令的集合，是指挥计算机进行一定操作、完成一定功能的一组指令的集合。

人们使用计算机，就是要利用计算机处理解决各种不同的问题。但是，计算机不会自己思考，它是人类手中的木偶，因此人们要明确告诉它做什么工作以及做哪几步才能完成这个工作。试想一下，计算机程序执行的整个过程是怎样的呢？计算机完成一件我们分配给它的任务，就像“军训”这件工作，它按照人们的命令去做，人们说“立正”，它不能“稍息”，就这样在人们的支配下完成预定工作。这里，人们所下达的每个命令称为指令，它对应着计算机执行的一个基本动作。人们告诉计算机按照某种顺序完成一系列指令，这一系列指令的集合称为程序。程序一般以文件的形式存放于计算机存储器中，称之为程序文件。编写程序，就是根据不同计算机指令的不同格式编写相应的程序文件。

为了理解指令和程序，让我们来想象一个游戏：

游戏中有两个人，其中一个被布蒙上双眼，另一个人是你。场地中混乱

地摆上许多啤酒瓶。游戏任务是由你发号施令，指挥被蒙眼者从场地一端穿越到另一端，其间不允许碰倒任意一个啤酒瓶。

附加一个游戏前提：你不哑，他不聋。

现在，你就会明白什么叫“指令”。指令就是一套符号，这套符号的含义，你懂，他也懂。

你会根据现场情况，向他发出类似这样的指令：“step1：向前 0.5 步；step2：向左 1 步；step3：向后 2 步；step4：向左 0.5 步；step5：向前 4 步；step6：停”。

“向前”，这就是指令。“0.5 步”，这就是指令的参数。在不需要具体区分时，我们也往往将“指令和指令的参数”通称为“指令”。

不同的工具，往往有不同的指令集。

比如锤子，它的指令应该是“锤”（或是“砸”）。剪刀则是“剪”。这样的工具功能单一，所以指令自然也简单了些。

换成汽车呢？如果你是初学者，正好，你的师傅坐在副座上，你就有幸听到相对复杂的指令了：左转……太大了！右转一点点，踩离合！油门！又大了！刹车！

对于计算机而言，不同的“处理器”类型——不同厂家生产的处理器，甚至同一厂家生产的不同版本的处理器，往往都会有不同的指令集合。为了商业利益，有些厂家之间就会进行“连横”，相互之间保持尽量大的兼容，当然也各自保留了一手特立指令。典型的如 Intel 和 AMD 两家 CPU 厂商。还有一些厂家则特定独行，采用完全不同于竞争对手的指令集合，比如曾经的 IBM 公司或苹果电脑，就采用各自独立的 CPU。

人们通常使用的电脑，都使用 Intel 或 AMD 生产的中央处理器（CPU），这一类指令集被称作“80x86 CPU 汇编指令”。

指令集的定义，可以分为“复杂指令”和“精简指令”。比如锤子，我们说它的指令是“锤”。其实是“锤”这个动作，它可以分解为“抡起”“瞄准”“砸下”。而剪刀，也至少可以再分为：“张开剪”“合上剪”两个指令。“锤”“剪”是复杂指令，而其余的则是精简指令。

指令过程，必然存在一个“发号者”和一个“接受者”。接受者接收到某一指令之后，必须将该指令继续细分为多个步骤（子指令）执行，这称为“复杂指令集”。采用这种指令集的处理器的计算机，称为 CISC（Complex Instruction Set Computer）。相反，每一个指令尽量仅定义一个最精简的动作，

接受者不需要再细化单一指令的动作，这样的指令集称为“精简指令集”。采用“精简指令集”的计算机，称为 RISC (Reduced Instruction Set Computer)。

通常我们使用的 PC 计算机，属于“CISC”。

程序 = 指令的逻辑组合。这是给出的第一个有关“程序是什么”的表达式。计算机程序是一组指令的组合，这组指令依据既定的逻辑，控制计算机各个部件的运行，以期得到想要的结果。

在这个定义中，有三个重要的概念。其中，谈到了程序中的“指令”，但是还没有谈到“组合”及“逻辑”是什么。

继续前面的啤酒瓶游戏：

理论上，如果场地不变、酒瓶摆放位置不变、参与人不变，那么作为指挥者，你完全可以把第一次的指挥过程记录在案，形成一套“指令的组合”，如图 2-1 所示。当再次通过该啤酒瓶雷区时，通过人就可以直接应用你前边的“指令组合”进行前进，就不需要你再重新思考来指挥他怎样通过这个啤酒瓶雷区了。

这就是程序，一组指令的组合，并且在逻辑上，它们的组织方法就是保证执行者安全通过酒瓶区。

从整体上讲，指令的组合方式，最终决定于它在逻辑目标上的要求；从细节上讲，所有程序的指令组合，都至少有一个组合要求：时序，或称为步骤，如前面的 step1 ~ step 6，听令者必须按步骤次序执行，这套指令组合才有意义。

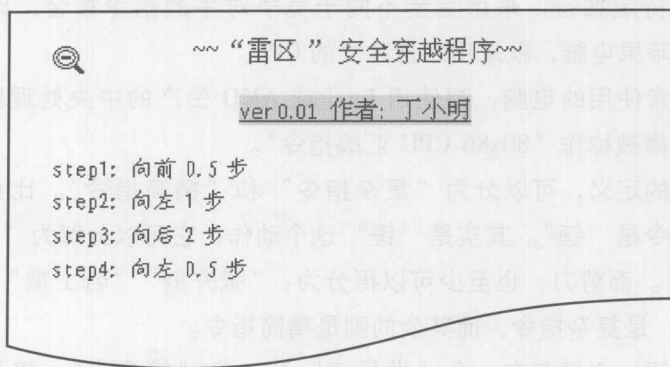


图 2-1 指令的组合