

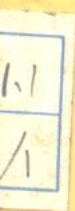
计算机应用与教学系列

用户界面程序设计

原理与实例

●〔美〕J. R. 布朗 S. 坎宁安 著

●科学出版社



TP311.1
BL/1

计算机应用与教学系列

用户界面程序设计

原理与实例

〔美〕 J. R. 布朗 S. 坎宁安 著

卞小燕 冯庆文 萧 柔 译

陈 禹 校



024511

科学出版社

1992

1024511

(京) 新登字 092 号

内 容 简 介

一个成功的软件必须具有良好的用户界面。本书阐述了设计用户界面必须遵循的原则,介绍了目前世界上先进的用户界面实现技术,并讨论了交互式程序设计的几个主要方面:输入、输出、屏幕布局、帮助信息、教学指导、错误处理,等等。书中给出了大量的例子来说明有关的工具和实现技术。

在指导实践方面,本书侧重于以下几个方面:演示各种用户界面风格的样板程序和屏幕;以C语言风格给出代码框架和工作函数;用户界面管理系统的实验性界面和新的开发技术;在视觉交互中颜色的使用和屏幕设计的原则。

本书可供从事计算机软件开发、设计和教学工作的工程技术人员使用,亦可作为大专院校计算机专业及有关专业的教师和学生的参考书。

Judith R. Brown Steve Cunningham
PROGRAMMING THE USER INTERFACE
Principles and Examples
John Wiley & Sons, Inc., 1989

计算机应用与教学系列
用户界面程序设计

原 理 与 实 例

〔美〕 J. R. 布朗 S. 坎宁安 著

卞小燕 冯庆文 萧 柔 译

陈 禹 校

责任编辑 王春晖 刘晓融

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100707

化学工业出版社印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

1992年12月第 一 版 开本 787×1092 1/16

1992年12月第一次印刷 印张: 14 ³/₄

印数: 1—3 600 字数: 330 000

ISBN 7-03-001607-6/TP·112

定价: 9.30 元

JS411/16

译者的话

应用软件要获得高效率并取得成功就必须具有良好的用户界面。计算机应用程序与用户之间的良好界面对于用户方便地使用程序，最大程度地发挥程序的功效是非常重要的。目前，计算机用户对用户界面的要求越来越高，良好的用户界面是评估软件成功与否的重要条件之一。

美国 John Wiley & Sons 公司 1989 年出版的《用户界面程序设计——原理与实例》(Programming the User Interface—Principles and Examples) 一书介绍了设计用户界面必须遵循的原则，是当前世界上实现用户界面的最先进技术的高级技术参考书。该书侧重于让程序设计人员（特别是那些目前尚没有有效的用户界面支持的程序设计人员）开发自己的用户界面实现技术。

该书主要讨论交互式程序设计的几个主要方面：输入、输出、屏幕布局、帮助信息、教学指导、错误处理等，并给出大量的例子来说明有关的工具和实现技术。

在指导实践方面，该书侧重以下几个方面：

- 演示各种用户界面风格（以及容易发生错误的地方）的样板程序和屏幕；
- 以C语言风格给出代码的框架和工作函数，包括窗口管理系统、菜单系统和命令系统；
- 介绍用户界面管理系统的实验性界面和新的开发技术；
- 阐述在视觉交互中颜色的使用及屏幕设计的原则。

成功的用户界面是保证用户有效地使用软件的重要途径，该书是一本帮助程序设计人员达到此目的的优秀工具书。

鉴于国内尚未有关于用户界面程序设计方面的书籍出版，我们把此书翻译出来，介绍给国内的读者。在翻译过程中，考虑到本书内容的实用性并限于篇幅，我们删去了原书第十四章、附录、索引以及各章的推荐读物，并改正了原书中的一些印刷错误。

由于译者水平所限，译文难免有错误和不当之处，敬请读者指正。

译者

1992 年 2 月于北京

序

本书介绍用户界面程序设计。我们的目标是帮助程序设计人员，特别是那些没有有效用户界面支持的程序设计人员，开始涉足这个领域。我们的方针是实践。我们阅读了书后列出的参考文献，并对用户界面做了研究之后，将这些文献用于本书的准备工作，我们的注意力主要集中在如何得到编写的实际用户界面代码。

如果读者是一名程序设计人员，他将对本书的代码框架和工作函数（过程）最感兴趣。这些代码和函数已经在我们的环境中进行了严格的测试，或者来自计算机工业中可信的来源。无论在什么情况下，我们的设计和代码都可作为读者在程序设计实践中开发自己的用户界面工具集的起点。当然，由于篇幅的限制，我们的例子距实际应用的代码的距离还很远，并且这些例子主要是为通用目的而不是特殊目的而设计的。

用户界面不仅仅是代码对象的集合，它还是保证用户能有效地使用软件的途径。在本书中，我们对用户界面所做的说明和所给出的用户界面设计原则是我们阅读并研究了大量的参考文献后得出的。现在的软件环境已很成熟，如果软件没有很好的界面设计，就不能算成功。

如果读者是一名学生，则可把本书当作一本入门读物，本书介绍软件用户界面能做什么，从设计和实现的角度如何构造它。我们希望读者通过给出的例子，能够实现很好的用户界面。现在已经有了实际的用户界面支持软件。本书中，我们将介绍一些用户界面和窗口系统管理工具。

我们怎么会想到写这么一本书呢？在 1986 年的 ACM SIGGRAPH 会议上，我们谈论通用教学软件，特别谈论了怎样使它与学生通讯的许多问题。我们中的一人说应该写一本书，以帮助软件开发者更好地工作，另一人说我们就可以做这项工作，之后谈话话题转移到了其它问题上。大约一个月后，我们提出了一个带注释的简短提纲。

随着我们对这个选题的深入了解，原来的提纲变动了很多。现在包括了样例界面和用户界面管理系统，这两部分对未来的用户界面来说都是很重要的，其中的一些内容在我们开始写作时还没有实现。我们在大部分代码样例中使用 C 语言，因为用 C 语言表达的算法可以比较容易地翻译成其它语言，另外，C 语言也被广泛地应用于主要的交互式软件和界面系统中。

我们衷心感谢给予我们帮助和鼓励的人们。首先要感谢我们家庭的支持和鼓励；感谢 Sara Bly (Xerox PARC)，Jon Meads (Bell Northern Research) 和 David Sealey (Iowa 大学) 的鼓励和支持，他们审读了本书；还有我们的学生，California 州立大学的 Mike Elness, Jason Fox 和 Chuck Quittmeyer，感谢他们对本书给出的算法草稿所做的工作。还有许多人士对我们列出的例子做出了贡献，在此我们向他们表示诚挚的谢意。

J. R. 布朗

S. 坎宁安

衣阿华，衣阿华市
加利福尼亚，图鲁克

1989 年 4 月

目 录

译者的话

序

第一章 绪论	1
1.1 交互式系统一例	1
1.2 交互式系统设计	3
1.3 用户界面的基本原则	4
灵活性与一致性	5
1.4 用户界面工具集	6
1.5 本书的结构	6
第二章 用户输入的原则	8
2.1 用户输入的类型	8
2.2 输入设备	8
模拟设备	11
2.3 控制输入	12
2.4 处理数据输入	15
2.5 向用户提供活动的反馈	15
2.6 实例: 带三种用户输入方式的程序	17
第三章 菜单程序设计	19
3.1 菜单驱动系统实例	19
AppleWorks	19
MacWrite	20
dBASE IV	21
Lotus 1-2-3	21
Auto CAD	22
3.2 文本菜单程序设计	23
文本菜单的设计原则	23
替换形式	25
文本菜单的实现	26
一种更通用的菜单实现方法	31
3.3 图形菜单程序设计	33
图形菜单的设计原则	33
图形菜单实现	35
第四章 命令驱动系统程序设计	39
4.1 命令驱动系统的例子	39
UNIX	39

dBASE IV	42
Tell-a-Graf	43
4.2 命令项	45
4.3 命令控制系统的总体结构	47
设计命令动作的工具	47
命令恢复	48
4.4 允许用户自定义命令	48
4.5 命令输入的编程	50
将命令分成单词表	50
将单词与命令部分匹配	52
4.6 对命令的程序动作的编程	55
第五章 数据输入程序设计	58
5.1 数据输入的不同格式	58
5.2 从程序中控制数据输入	58
5.3 文本输入技术	59
5.4 提供默认值	61
5.5 数值输入技术	61
读取整型值	62
读入实型值	63
从表达式中获取一个数值	67
5.6 “填空”输入方式的管理	74
第六章 对用户信息的反应	80
6.1 一般原则	80
计划	80
一致性	80
可读性	80
满足用户需求	81
给出信息	82
6.2 组织输出的方法	84
屏幕的主要成分	84
标题行	84
命令或状态行	85
错误或警告信息	85
屏幕体	86
不同类型屏幕的考虑	86
菜单	86
数据输入屏幕	87
问答屏幕	88
查询屏幕	88
信息屏幕	89
控制屏幕	89

强调关键部分	90
给出表格总结	91
给出图形总结	92
6.3 输出设备	92
第七章 屏幕处理技术	94
7.1 交互式屏幕	94
数据查询	94
动态介质屏幕	95
7.2 适当地显示信息	100
7.3 使用窗口技术	104
窗口的一般用途	105
活动程序操作所用的窗口	105
窗口用于暂用信息或控制	106
窗口布置	106
窗口控制	107
第八章 在输出中使用颜色	108
8.1 利用颜色增加意义	108
在应用程序中使用颜色	109
让用户自己选择颜色	110
8.2 作为一门“精确科学”的颜色	110
颜色感知的物理学	110
8.3 颜色的相互作用	111
8.4 颜色规范	111
混合颜色	112
颜色模型	112
光谱模型	112
灰度	113
Munsell 模型	113
RGB 规范	113
CMY 规范	113
HSV 规范	114
HLS 规范	114
HVC 规范	114
CNS 规范	115
实验与研究	115
颜色的复制	115
相关的颜色	115
8.5 有效地使用颜色的原则	116
颜色组合的要素	116
疲劳	117
8.6 颜色查找表	117

使屏幕产生颜色	118
与颜色进行交互	119
建立查找表	119
选择要被设置的颜色	119
设置颜色	120
交互式计算机辅助 RGB 编辑器 (ICARE)	121
8.7 颜色的意义	122
8.8 与单显的兼容性	122
第九章 直接操纵系统和特殊环境	124
9.1 直接操纵的概念	124
9.2 直接操纵的程序设计	127
9.3 数据输入的直接操纵技术	129
9.4 实验环境和高级环境	130
ARK	131
Interactive Image	132
语音识别	133
NOOBIE	133
VIDEOPLACE	134
三维虚拟环境	135
第十章 窗口和用户界面管理系统	136
10.1 窗口系统的概念	136
10.2 窗口和窗口系统	140
字符映象窗口	140
Curses	142
Microsoft Windows	144
Macintosh ToolBox	151
SUN NeWS	159
X 窗口系统	163
10.3 用户界面管理系统	168
MacApp	169
Open Look	179
TIGER	180
iCpak 201	182
Actor	183
Open Dialogue	187
NewWave	191
第十一章 帮助用户与程序交互	194
11.1 提供辅助教学功能	194
11.2 提供用户引导帮助功能	196
11.3 帮助系统的原则	197
11.4 减少和控制程序中的错误	198

第十二章 帮助系统程序设计	200
12.1 帮助系统的例子	200
AppleWorks	200
Lotus 1-2-3	201
dBASE IV	202
Prime 的 PRIMOS	203
UNIX man	204
Zenographics Mirage	205
12.2 使帮助系统正常运转	206
12.3 帮助系统选项	207
12.4 易犯的错误	208
12.5 例子	208
第十三章 错误的处理与避免	210
13.1 输入错误	210
13.2 算术错误	211
13.3 文件和设备错误	214
13.4 错误信息	217
13.5 失败的恢复	219
参考文献	220

第一章 绪 论

现在,很多人都使用计算机.每个从事写作、印刷、邮递、订货、记帐、预算、规划、记录等信息工作的人都可能在办公桌上放一台计算机,并且经常使用它.以前拥有计算机便意味着你是一个技术天才,那样的日子已经一去不复返了,现在每个人都是一个计算机用户或潜在的用户,一般人对自己想从计算机获得些什么已有了相当成熟的看法.

在个人计算机出现之前,程序主要是为了商业应用而编制的,只有经过特别训练并掌握一定的技巧以后才能使用,训练用户比使软件适应于用户更容易些.随着个人计算机的推广,用户界面的质量已成为一个大问题.从70年代中期开始,个人计算机的软件市场年销售额已上升到数十亿美元,其中包括用于开发界面的花费.

现在的计算机用户对用户界面的质量要求很高,在评价软件时也对之寄以很高的期望.欲使程序得到认可,这个程序必须能够帮助用户完成任务.一个程序想要在市场上成功,必须能使用户最大限度地发挥其技能并将成就归功于用户.良好的用户界面对一个程序的被认可与成功有很大的关系.

在按用户的思路编制程序时,需要记住的最重要的事情是没有人愿意为使用程序而使用程序.人们或许希望用计算机来完成某个任务,但该任务决非计算机与程序,编程者的工作是使得计算机与程序尽可能地无形,以便于用户工作.

本书致力于帮助程序员实现良好的用户界面.虽然我们在书中谈到了如何进行良好的设计,但并不讨论实际界面的设计,而主要讲如何利用用户界面技术来完善一个良好的设计.

1.1 交互式系统一例

本书关于用户界面的章节包括了一些软件产品的例子.现在我们先来看一个系统,了解它的各种操作模式,并讨论它们各自的含义.

该系统是 Lotus 发展公司的 Lotus 1-2-3 (以下简称 1-2-3),它可以在 IBM PC 和其它使用 MS-DOS 系统的计算机上运行.我们仅讨论一些功能不同的例子,并说明这些功能是如何提供给用户并与用户交互作用的.

Lotus 1-2-3 是一种金融模型和预算程序,是一个著名的电子报表软件.它提供了二维数组的表元,其中每一个表元中可以存放文本、数字或表达式,包括数学函数与运算符、常量以及公式.当一个公式写入一个表元时,该公式的值立刻被计算并在表元中显示出来.无论何时,改变一个表元的值,用到该值的其它表元也相应被改变了.图 1.1 显示了在 1-2-3 屏幕上填充一些表元时所进行的简单计算.

注意屏幕被分成几个区:命令菜单区、工作表元区、当前表元地址和表元输入区.第五、六、七章将讨论特定屏幕区域的功能,并说明如何实现这些功能.

B7:	Worksheet	Range	Copy	Move	File	Print	Graph	Data	System	Quit	MENU
	Global,	Insert,	Delete,	Column,	Erase,	Titles,	Window,	Status,	Page		
	A	B	C	D	E	F	G	H			
1											
2		First		125							
3		Second		8							
4											
5		Sum		133							
6											
7		Product		1000							
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											

图 1.1 普通的 1-2-3 屏幕

A1:	Worksheet	Range	Copy	Move	File	Print	Graph	Data	System	Quit	MENU
	Global,	Insert,	Delete,	Column,	Erase,	Titles,	Window,	Status,	Page		

A1:	Format	Label	Erase	Name	Justify	Protect	Unprotect	Input	Value	Transpose	MENU
	Format a cell or range of cells										

A1:	Fixed	Scientific	Currency	General	+/-	-	Percent	Date	Text	Hidden	Reset	MENU
	Currency format (\$ x,xxx.xx)											

A1:	Enter range to format: A1..H1											MENU
-----	-------------------------------	--	--	--	--	--	--	--	--	--	--	------

图 1.2 在 1-2-3 中格式化一行

在 1-2-3 中，基本操作围绕着电子报表和表元的输入值，因此需要使光标能在表元上下左右移动。在显示屏上，用反视或彩色来突出显示被选中的表元，使用户可以直接访问电子报表。当输入一个值、标号或公式时，它被重现在表元的输入区中。输入结

束后, 表元的数值、文本或公式值立即在选中的表元中显出, 其它基于该表元值的表元重新计算它们的值. 这几件事几乎是同时发生的, 用户的操作获得极快的响应, 并且在任何时刻控制着电子报表. 本书第五章说明了如何控制这种输入和显示.

除向表元输入之外, 在电子报表上还可以进行许多其它操作. 可以增删行、列, 对电子报表某一部分的内容格式化 (标号居中、左齐或右齐, 数值按指定的十进制方式或带 \$ 符号写出, 等等); 在电子报表中值或公式可以被拷贝, 电子报表可以被保存或调入, 这些全在屏顶菜单区的管理之下进行. 实际上, 1-2-3 有一个很好的层次菜单, 它允许初学者运用光标键选择操作 (使用识别记忆), 或训练用户快速键入命令字母 (使用回想记忆). 第三章说明如何编写这种菜单系统.

图 1.2 显示了在 1-2-3 菜单中如何使图中的一行格式化, 带有一个 \$ 符号和两个十进制位的三个步骤, 说明了菜单系统是如何工作的. 键入 / REC 命令, 然后指定一行可以得到同样的结果.

1-2-3 中有一个扩展的帮助系统. 在程序中的任何一个位置, 用户可以按 F1 请求帮助, 同时原工作表元中的内容被说明当前菜单选择的帮助信息所替代. 关于帮助功能的详细说明与实例请见第十二章.

在 1-2-3 中, 对出错处理得很好. 文本错误由操作系统处理. 倘若是非扩展性文本错误, 程序不会停止执行, 而仅向用户传递系统错误信息. 计算错误能被识别出来, 如果输入了一个非法的函数或一个不能完成的运算, 例如除数为 0, 该表元中包含的计算被赋以一个错误值, 并且在该表元中显示“ERR”, 其它使用该值的所有表元均得到这一错误值, 并且显示“ERR”. 程序不会中断, 并且没有错误的表元不受影响. 当该表元被改正时 (所有表元都可以被编辑), 其值被重新显示, 其它相关的表元也被重新计算. 第十三章详细讨论错误处理.

1.2 交互式系统设计

本书讨论如何实施交互式程序的许多标准特征, 本书的读者对象主要是程序员, 目的在于提高他们在交互式技术方面的造诣.

也许理想的软件应是由一组人员来设计的. 应用专家和用户专家负责了解用户需求及用户如何考虑完成其工作, 然后将这些内容转化成交互作用与屏幕设计等特征: 用户该做什么? 机器如何响应? 然后要求程序员实现这一工作. 可能的程序员已具备上述知识, 或凭直觉知道用户如何工作, 并且能通过程序设计技巧来实现它, 但这种情况很少见.

一个软件的全面设计包含许多组成部分, 其任务必须被定义并落实到输入输出级上, 这通常与用户界面分开实现. 用户界面的设计必须提供从用户到任务的输入并将输出传给用户, 这可能也是分开做的. 当然, 任务与界面均需完善并且兼容, 良好的用户界面掩饰不了任务质量的低劣, 差的用户界面却可能使一个好的产品不被接受.

软件的界面设计通常经过一个标准的途径. 第一阶段了解用户如何考虑任务, 他们希望工作怎样做. 在同应用专家协作后, 这可能与用户的观点、信息的收集或者实际技术有关. 这一阶段的目的是建立一个关于用户如何完成任务的实体模式, 它对如何在计

计算机上实现任务有一些影响。该模式用来支持软件执行任务的方式，从本质上决定如何建立用户界面。

第二阶段建立用户界面的原型 (prototype)，并让用户在这些原型上工作，以求找到用户反映最好的一个原型。在这一阶段最好请一个图形设计师来保证用户与软件之间的正确通讯。每一个原型均支持用户 / 任务模式，但它们各不相同，这些原型的测试是在不同环境由不同用户进行的，从中选择最成功的一个加以完善，形成最终界面。

在改进用户界面的各个部分以及用户通讯的支持时，我们用大量篇幅来讨论如何使用用户界面有效并实现这一技术。若想更深了解设计方面的知识，请参阅有关用户界面设计的资料。我们的目的在于描述一个良好的用户界面，并指出用户界面设计的原则是使程序员与设计者更易于共同工作。

如果程序员没有界面设计者的支持，就要记住他自己对用户界面的直感并不一定是最优的，阅读本书中的原则并尽量与自己的想法融合，多写几个样本来试验自己的想法。在软件应用领域交一些益友（那些能够提出建设性批评意见的人），请他们给你提出一些中肯的意见和新的方法。最后从未来用户和有类似经验的人身上（他们可以提供与他们自己当前使用的软件的比较）获益，尽量从同行的看法中获益。这些努力的结果是值得的。

1.3 用户界面的基本原则

有些原则似乎对所有良好的用户界面都是一致的，我们在此将它们作为一般原则阐述。当涉及到输入、输出以及用户界面的其它方面时再个别说明。

用户界面涉及所有输入输出，它必须向用户说明所能使用的指令，让用户尽可能自然地控制程序，并向用户提供程序的结果。如果界面提供了不明确的指令，或者用户不能够通过输入控制任务，好像计算机不曾参与一样，或者程序的输出含糊不清，那么这种通讯、这一界面、最终这个软件是不成功的。

原则：用户界面的主要功能是通讯。

程序必须执行给出的指令功能，允许控制、显示结果而不干扰任务进程。这意味着所有的输入应该是自由格式，或自然选择的；输出必须清楚，不能有错误的或含糊不清的显示或语法错误；运用一切适当的工具使显示屏幕快速可读。这些事情并非微不足道，它使得程序令用户满意，并防止计算机对任务的干扰。

原则：界面必须防止计算机成为用户与工作之间的障碍。

界面必须支持问题的单一模式及其解决方法，允许用户在计算机上仅以一种方法考虑任务，而无须疑惑：“计算机希望我以哪种方式做此事？”这使得任务成为习惯性的，并且增强了计算机的不可见性。

原则：界面必须始终一致。

程序的用户可能是基于信号的，也可能是基于单词的，变化很大。有的人喜欢用数字，有的人爱用字母，有的人希望不用记任何东西就可以使用程序，而有的人愿意成为程序专家，有的人乐意用菜单而有人坚持用命令。欲获得全面的成功，必须针对不同的人提供不同的最适合其要求的工作方式。

0185

原则：界面必须是灵活的，以适应广泛的用户。

或许一个紧急电话，用户必须放下手边的任务，几小时后才能继续工作。一个好的助手将会帮助他们记忆发生了什么事；程序也应能通过保持一些当前发生的事的信息来做到这一点。

原则：界面必须使用户知道任务的当前进行情况。

认为只有专家才会使用程序是没有道理的，即使专家也不可能随时记住每一件事。用户必须能获得一些帮助，这些帮助是针对其当前活动的，也是最优的。这是一个具有挑战性的通讯任务，但通过努力、思考和计划也可以完成。

原则：界面必须能够获取帮助。

所有用户都讨厌在程序崩溃时丢失或中断自己的工作，这或许不是界面的问题而是系统的问题，但由于它造成崩溃，通常使用户工作有所损失，因此在此提及。1-2-3 可以防止许多由用户引起的崩溃，这是一个值得努力的重要目标。

原则：程序不可崩溃。

程序员的主旨很简洁。

原则：程序必须努力工作，而用户无须如此。

上面的每一原则均需要程序员的努力才能避免程序不恰当的行为。自由格式化的输入、仔细地组织过的显示输出、可读的屏幕、状态及帮助信息，以及证实中止等花费了大量的工作，而这些工作在用户响应、接受，以及方便地进行工作等方面得到了报偿。

灵活性与一致性

灵活性与一致性的原则显然是相互对立的，这就需要仔细推敲。一致性包括两个方面：对某一应用程序用户的思维模式的一致，以及对用户控制应用程序的方式的一致。

如本章前面所述，用户思维方式的一致是应用专家的工作。只要用户掌握了应用程序及其控制方式，就有安排控制的灵活性，所以只要控制的性质与控制本质的结果是一样的，程序可以由命令或菜单控制，思维模式是受保护的，甚至允许用户选择其喜爱的界面风格。

有时，应用模式需要更广泛意义的一致性：与同一类应用的其它程序的一致性。除非是一个新领域的程序，否则其设计者必须密切注意在类似应用中最成功的程序里使用的技术，必须尊重用户在使用这些程序时的习惯。不然只有用非常有力的理由去说服用户在使用一个新程序时改变其习惯。商业市场经验表明：每一个电子报表程序的基本操作都类似于 VisiCalc，大多数绘图程序与 Macpaint 基本类似。这些相似并非偶然，用户与顾客决定了在选择新软件时易于学习是最重要的。

原则：使自己的程序工作时类似于同类程序。

在程序的交互式控制中一致性同样重要，在整个程序中这些控制必须使用一致的机制。例如，在一个全屏幕编辑系统中，如果用户要删除当前光标处的字符，但不想删去光标前的另一字符。若程序使用了命令，该命令需使用一致的句法及结构。如果程序使用了菜单，每一个菜单的选择方式必须相同，并对用户的选择作出同样的反应。这一点可以像第三章的菜单样例，通过在菜单中允许多个选择项做到，也可以使程序从几种可能的选择中仅选择一种。程序能生成一种结构文件并允许用户为程序设置所需的选择。

无论怎样处理,应该让用户在使用程序时方便而且有效,这种内在的一致性很重要。

原则: 使程序的所有部分相似地工作。

控制一致性的想法由一个简单程序实现.在许多不同的自动售货机和应用程序中,键对程序操作控制的一致性使 Macintosh 成功的因素之一.第十章中讨论的一些系统,如 Microsoft Windows 和许多用户界面管理系统,均对应用程序提供了一种普遍的相似性。

1.4 用户界面工具集

如上所述,用户界面程序设计工作量很大.在最终确定界面设计以前,许多程序需要积累大量的界面经验,这使得对每一个界面分别编码是不实际或不可能的.界面程序设计者需要一个界面工具集.这个工具集中包括许多普通的工具,它们完成界面的大多数任务,包括:

- 菜单显示;
- 分析命令;
- 自由格式数值输入的读入;
- 用编辑与卷滚处理文本输入;
- 在格式化屏幕上显示信息.
- 显示警告与对话框;
- 使用窗口;
- 帮助显示;
- 出错处理;

这些都是一个工具集应该完成的,对其中某部分的省略便意味着编码的重复或对程序设计技巧的省略。

本书展现了这个工具集的一部分,关于专门界面设计技巧的每一章均包括设计实例及其实现代码.这些实例很通用,读者可以在自己的工具集中采用它,但实例的重点在于清楚易懂,而不在于效率.有一些公司出售用户界面工具集,这些常常是菜单或窗口控制系统;也有一些单位出售完整的用户界面管理系统 (UIMS),第十章中对此进行讨论。

1.5 本书的结构

总之,本书的目的在于向程序员提供一些实现广泛使用的用户界面技术的工具.它分成三个部分:输入、输出和帮助/错误处理.每一部分均较详细地讨论了用户界面这个大主题的一些方面。

输入部分包括菜单、命令和数据输入,菜单(第三章)以实例说明了文本菜单和图形菜单,并说明了在用户界面中直接操纵(direct manipulation)的地位与功能.命令(第四章)包含用于程序控制的文本输入,并介绍了带参数的命令和用户自定义命令.数据输入(第五章)包含可编辑的词卷滚文本输入、自由格式的数值输入和填空技巧。

其后两章讨论包括输入和输出的界面程序设计.特殊的直接操纵系统与环境(第九章)探讨直接操纵和面向对象的程序设计,并提供了使用直接操纵技术激活软件的一些例子.用户界面管理系统与窗口(第十章)讨论窗口显示的多种系统与技术,以及一些可用的窗口系统与用户界面管理系统,这两章指出了用户界面的未来方向。

第十一到十三章讨论用户界面中时常忽略的一些问题.帮助/错误处理部分包含求

取帮助和出错处理的技术。帮助技术（第十二章）描述了如何实现帮助系统并使之工作，而出错处理（第十三章）说明了控制文件错误和计算错误的技术，对设备管理并对程序或整个计算机崩溃的恢复技术提出了一些建议。