

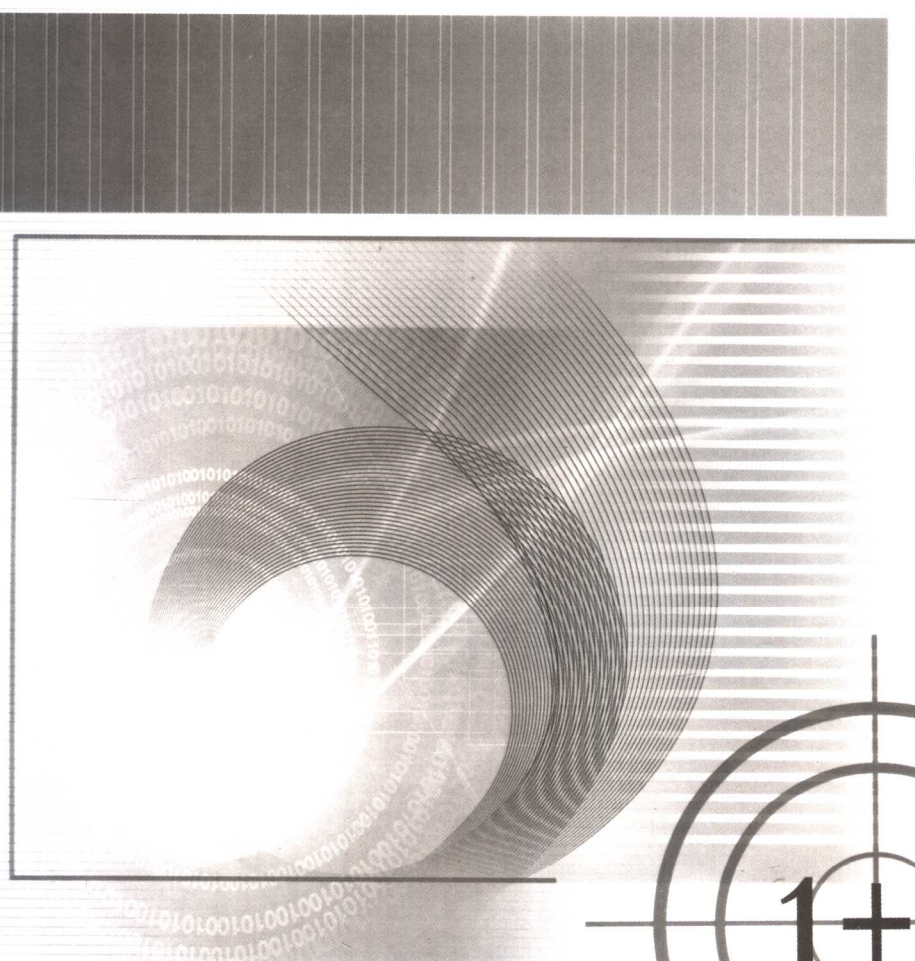


高等学校计算机基础教育规划教材

丛书主编:冯博琴

C语言程序设计

刘克成 主编 张凌晓 邵艳玲 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



高等学校计算机基础教育规划教材

丛书主编：冯博琴

C 语言程序设计

刘克成 主编

张凌晓 邵艳玲 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是根据非计算机专业计算机基础课程教学指导分委员会提出的“关于进一步加强高等学校计算机基础教学的意见”(简称白皮书)中“高等学校非计算机专业计算机基础课程教学基本要求”部分的C语言程序设计课程大纲编写而成。全书共分10章,系统地介绍了程序设计的基本概念,C语言的各种数据类型、运算符与表达式、模块化程序设计方法、流程控制结构和文件系统的基本操作,对面向对象编程基础也进行了初步阐述。

本书的内容安排循序渐进,采取通俗易懂的讲解方法,并辅以丰富的例题和大量习题强化关键知识点。另外,为帮助读者对教材内容的理解,强化动手能力,结合每章内容,还编写了配套实验指导书——《C语言程序设计实验指导与习题解答》,供学习时参考使用。

本书以应用为目的,可作为高等学校计算机程序设计教材,也可作为计算机程序设计培训教材或其他从事计算机程序设计的科技人员的参考书。

图书在版编目(CIP)数据

C语言程序设计/刘克成主编. —北京:中国铁道出版社, 2005. 12

高等院校计算机基础教育规划教材

ISBN 7-113-06792-1

I. C... II. 刘... III. C语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2005)第155471号

书 名: C语言程序设计

作 者: 刘克成 张凌晓 邵艳玲

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街8号)

策划编辑: 严晓舟 秦绪好

责任编辑: 严 力 翟玉峰 熊严飞

封面设计: 薛 为

封面制作: 白 雪

责任校对: 李 旸

印 刷: 北京新魏印刷厂

开 本: 787×1092 1/16 印张: 20.25 字数: 487千

版 本: 2006年3月第1版 2006年3月第1次印刷。

印 数: 1~5 000册

书 号: ISBN 7-113-06792-1/TP·1657

定 价: 26.00元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

高等学校计算机基础教育规划教材

主 任：冯博琴（西安交通大学 教授）

委 员：（按姓氏字母为序）

段 富	（太原理工大学	教授）
甘 勇	（郑州轻工业学院	教授）
耿国华	（西北大学	教授）
管会生	（兰州大学	教授）
李振坤	（广东工业大学	教授）
李志蜀	（四川大学	教授）
李雁翎	（东北师范大学	教授）
刘东升	（内蒙古师范大学	教授）
裴喜春	（内蒙古农业大学	教授）
石 冰	（山东大学	教授）
武 波	（西安电子科技大学	教授）
张毅坤	（西安理工大学	教授）
邹北骥	（中南大学	教授）

编
审
委
员
会

1997 年教育部高教司颁发的“加强非计算机专业计算机基础教学工作的几点意见”(简称 155 号文件)中提出的要求已经使得, 各校的计算机基础教学条件明显改善, 计算机基础教学进入了一个新阶段。

本届非计算机专业计算机基础课程教学指导分委员会按照高教司指示, 分析了当前高校遇到的计算机基础教学的新形势, 根据人才培养的基本要求, 针对计算机基础教学中普遍存在的问题, 提出了三个文件: 即“关于进一步加强高等学校计算机基础教学的意见”(俗称白皮书)、白皮书的附件“计算机基础教学内容的知识结构与课程设置”和“高等学校非计算机专业计算机基础课程教学基本要求”。在附件中提出了“1+X”的课程设置方案, 即 1 门“大学计算机基础”(必修)加上几门核心课程(必修或选修)。

白皮书及附件自 2003 年底在高校征求意见以来, 受到了普遍的关注, 在高校中引起巨大反响。特别是 2005 年 4 月~5 月教指委在全国 19 个省进行了白皮书的巡讲活动, 参会学校 641 所, 教师达 1 269 人, 从问卷调查结果数据看, 对白皮书内容同意的占 81.16%, 基本同意但有疑义的占 9.36%, 很有问题不同意的占 0.22%, 未发表意见的占 9.26%, 可见白皮书已得到大多数一线教师的认同。教指委将根据征求到的意见作进一步的修改, 不久将正式发布。无疑它将直接影响今后高校计算机基础教学的整体架构, 也将推动新一轮的计算机基础教材的面世。

中国铁道出版社是国内实力雄厚的大社, 近年对计算机教材出版颇为关注, 最近又以其敏锐的眼光和雄伟的魄力, 怀着为计算机基础教学作出贡献的责任感, 遵循白皮书提出的理念和教学基本要求, 2004 年在全国范围内邀请计算机基础教学一线教师, 组织编写“1+X”中规定的 6 门核心课程及其若干门整合课程, 争取在一二年内出版。本丛书按照白皮书对教材建设所提出的建议, 努力在以下几个方面做出特色:

- 对于核心课程的教材, 要体现课程内容的基础性和系统性; 基本概念、基本技术与方法的讲解力求准确明晰。
- 体现非计算机专业计算机基础教材特点, 重在应用。内容要激发学生学习兴趣, 通俗易懂, 理论联系实际, 每一门课都要使学生真正学到一点有用知识和技术。
- 保证教材内容的先进性, 特别对于技术性、应用性的内容更是如此。
- 重视实验教材的建设, 重点教材都要配备实验教材。

我们希望本丛书的出版对推动高校计算机基础教学有所帮助, 并在使用中不断改进, 恳望读者不吝指正。

冯博琴

2005 年 7 月

冯博琴, 西安交通大学教授, 博士生导师, 兼任教育部 2001 年~2005 年高校计算机科学与技术教学指导委员会副主任、非计算机专业计算机课程教学指导分委员会主任委员, 全国计算机基础教育研究会副会长, 陕西省计算机教育研究会理事长。

C 语言是应用广泛、最具影响力的程序设计语言之一。它概念简洁,数据类型丰富,运算符多,表达能力强,使用灵活;它既有高级语言的优点,又具有低级语言的功能(能对硬件直接进行操作),因此既适合编写应用程序,又适合编写系统程序,并且生成的目标程序执行效率高,具有良好的可移植性,是一种理想的结构化程序设计语言,赢得了广大用户的喜爱。我们根据非计算机专业计算机基础课程教学指导分委员会提出的“关于进一步加强高等学校计算机基础教学的意见”(简称白皮书)中“高等学校非计算机专业计算机基础课程教学基本要求”部分的 C 语言程序设计课程大纲编写而成,其目的是让读者在没有任何编程基础的情况下就可以很容易地学习和掌握 C 语言的基本内容。本书在体系结构内容安排上将难点分散,使读者能循序渐进地逐步深入,并且每章都安排有习题,以便加强读者对所学关键知识点的理解与掌握。

本书全面介绍了 C 语言的基本知识和结构化程序设计方法。本书共有 10 章,第 1 章介绍了程序设计的基本知识,第 2 章介绍了 C 语言的数据类型、常量与变量和简单的输入输出函数,第 3 章对 C 语言运算符和表达式进行了介绍,第 4 章综合介绍了 C 语言的顺序结构、选择结构和循环结构等进行结构化程序设计的基本方法和有关算法的知识,第 5 章阐述了模块化程序设计方法,第 6 章讨论了 C 语言数组的定义与使用,第 7 章对指针的概念与应用作了充分的阐述,第 8 章重点介绍了结构体,并对共用体和枚举类型作了简要介绍,第 9 章介绍了 C 语言的文件系统,第 10 章对面向对象编程的基本概念、特点及面向对象程序设计的基本方法进行了初步阐述。

本书是作者在多年从事 C 语言程序设计的教学实践基础上,结合讲义编写而成的。本书采用现代风格定义和声明函数,用传统流程图描述算法,所有程序都按照结构化程序设计方法编写,有助于养成良好的程序设计习惯。本书内容充实,循序渐进,选材上注意系统性、实用性,例题可以在 Turbo C 2.0 或 Visual C++ 6.0 集成开发环境下编译运行,并且为配合读者学习,我们另编写了《C 语言程序设计实验指导与习题解答》一书,作为本书的配套教材。

本书由刘克成教授主编,张凌晓、邵艳玲为副主编。第 7 章由刘克成编写;第 8 章及附录由张凌晓编写;第 5、10 章由邵艳玲编写;第 1、6 章由张晓民编写,第 4 章由杨新锋编写;第 2、3、9 章由杨彩霞编写。全书总纂工作由刘克成、张凌晓和邵艳玲负责完成。

在本书的编写过程中,承蒙西安交通大学冯博琴教授的热情支持与指导,在此表示衷心感谢!同时,又参阅了大量的网上资源和其他参考文献,在此对它们的作者和提供者一并表示感谢。

由于计算机科学技术发展迅速,程序设计的教学内容、方法和手段日新月异,加以编者水平有限,书中难免有不足之处,敬请读者批评指正,以便再版时修改完善。

编 者

2005 年 11 月

第 1 章 C 语言程序设计初步	1
1.1 程序设计的基本概念	1
1.1.1 程序与程序设计语言	1
1.1.2 问题求解与程序设计	2
1.2 程序设计的一般过程	3
1.2.1 问题描述	3
1.2.2 问题分析	3
1.2.3 计算机实现	3
1.2.4 程序运行与调试	4
1.3 程序设计的两种方法	5
1.3.1 结构化程序设计	5
1.3.2 面向对象程序设计	6
1.4 C 语言概述	7
1.4.1 为什么选择 C 语言	7
1.4.2 C 语言字符集	8
1.4.3 C 语言标识符与关键字	9
1.4.4 C 语言程序基本结构	10
1.5 C 语言编程环境	11
1.5.1 Turbo C 2.0 编程环境	11
1.5.2 Visual C++ 6.0 编程环境	14
本章小结	21
习题	22
第 2 章 数据类型与简单输入输出	23
2.1 C 语言基本数据类型	23
2.1.1 数据类型概述	23
2.1.2 整数类型	24
2.1.3 实数类型	25
2.1.4 字符类型	25
2.2 常量与变量	25
2.2.1 常量	25
2.2.2 变量	29
2.3 数据的输出与输入	31
2.3.1 数据的输出	31
2.3.2 数据的输入	36
本章小结	40
习题	40

第 3 章 运算符与表达式	43
3.1 基本运算符及其表达式	43
3.1.1 C 语言运算符概述	43
3.1.2 算术运算符及其表达式	45
3.1.3 赋值运算符及其表达式	46
3.1.4 关系运算符及其表达式	48
3.1.5 逻辑运算符及其表达式	50
3.1.6 位运算符及其表达式	52
3.1.7 条件运算符及其表达式	53
3.1.8 逗号运算符和 sizeof 运算符	54
3.2 混合运算时数据类型的转换	55
3.2.1 类型隐式转换	55
3.2.2 类型显式转换	56
本章小结	57
习题	58
第 4 章 程序流程控制	62
4.1 算法	62
4.1.1 算法的概念	62
4.1.2 算法的组成要素	63
4.1.3 算法的描述	63
4.2 C 语言的基本语句	67
4.3 顺序结构	68
4.4 选择结构	70
4.4.1 if...else 语句	71
4.4.2 if 语句	73
4.4.3 else if 语句	74
4.4.4 if 语句的嵌套	76
4.4.5 switch 语句结构	81
4.5 循环结构	83
4.5.1 穷举与迭代算法	84
4.5.2 while 语句	86
4.5.3 do...while 语句	91
4.5.4 for 语句	92
4.5.5 流程转向语句	93
4.5.6 循环结构的嵌套	95
本章小结	102
习题	103

第 5 章 模块化程序设计	109
5.1 模块化设计与函数	109
5.1.1 函数的定义	110
5.1.2 函数的调用	112
5.1.3 函数的参数传递	115
5.1.4 函数的嵌套调用	117
5.1.5 函数的递归调用	118
5.1.6 系统标准函数	121
5.2 局部变量和全局变量	122
5.2.1 局部变量	122
5.2.2 全局变量	124
5.3 变量的存储属性	126
5.3.1 自动变量	127
5.3.2 静态变量	129
5.3.3 外部变量	131
5.3.4 寄存器变量	131
5.4 内部函数和外部函数	132
5.4.1 内部函数	132
5.4.2 外部函数	132
5.5 编译预处理	133
5.5.1 文件包含	133
5.5.2 宏定义	133
5.5.3 条件编译	136
5.6 函数综合应用举例	138
本章小结	141
习题	142
第 6 章 数组	145
6.1 数组的概念	145
6.2 一维数组	145
6.2.1 一维数组的声明	146
6.2.2 一维数组的初始化	147
6.2.3 一维数组元素引用	149
6.2.4 一维数组应用举例	149
6.3 二维数组与多维数组	154
6.3.1 二维数组的声明	155
6.3.2 二维数组的初始化	155
6.3.3 二维数组元素引用	157
6.3.4 二维数组应用举例	159
6.3.5 多维数组	161

6.4 字符数组与字符串	162
6.4.1 字符数组	163
6.4.2 字符数组应用	164
6.4.3 字符串处理函数	165
6.5 数组作为函数参数	167
6.5.1 数组元素作为函数参数	168
6.5.2 数组名作为函数参数	169
6.6 数组综合应用举例	172
本章小结	177
习题	177
第 7 章 指针	181
7.1 指针的概念	181
7.1.1 地址与指针	181
7.1.2 指针变量	182
7.1.3 指针运算	183
7.2 指针与变量	185
7.3 指针与数组	186
7.3.1 指针和一维数组	186
7.3.2 指针和二维数组	189
7.3.3 指针数组	191
7.4 指针与字符串	192
7.4.1 字符指针与字符数组	192
7.4.2 字符指针数组	194
7.5 指针与函数	196
7.5.1 指针作为函数形式参数	196
7.5.2 返回指针的函数	199
7.5.3 指向函数的指针	200
7.6 指向指针的指针	202
7.7 命令行参数	204
7.8 指针综合应用举例	205
本章小结	207
习题	208
第 8 章 结构体、共用体与枚举类型	213
8.1 结构体	213
8.1.1 结构体类型的定义	213
8.1.2 结构体类型变量	215
8.1.3 结构体类型数组	219
8.1.4 结构体类型指针	222

8.1.5 结构体与函数	225
8.1.6 动态内存分配	228
8.2 共用体	231
8.3 位段结构	236
8.4 枚举类型	237
8.5 用 typedef 定义类型	240
8.6 结构体应用举例	241
本章小结	248
习题	249
第 9 章 文件	254
9.1 文件概述	254
9.1.1 文件的概念	254
9.1.2 C 语言的文件系统	255
9.2 标准文件操作	255
9.2.1 文件类型指针	256
9.2.2 文件的打开	256
9.2.3 文件的关闭	258
9.2.4 文件的顺序读写	258
9.2.5 文件的随机读写	265
9.2.6 文件检测	268
本章小结	268
习题	269
第 10 章 面向对象程序设计	273
10.1 结构化程序设计与面向对象程序设计	273
10.1.1 结构化程序设计与面向对象程序设计概述	273
10.1.2 结构化程序设计与面向对象程序设计解决问题的比较	274
10.2 面向对象程序设计的基本概念	276
10.3 面向对象程序设计的基本特点	277
10.4 类和对象	278
10.4.1 类的定义	278
10.4.2 对象的定义	279
10.4.3 对象的初始化	279
10.5 继承和派生	281
10.5.1 派生类的定义	282
10.5.2 派生类程序举例	283
10.5.3 派生类生成过程	283
10.6 多态性	284
10.6.1 函数重载	284

10.6.2 运算符重载	285
10.6.3 虚函数	286
10.7 面向对象的软件开发	287
10.7.1 面向对象的分析	287
10.7.2 面向对象的设计	288
10.7.3 面向对象的编程	288
10.7.4 面向对象的测试	288
10.7.5 面向对象的软件维护	288
本章小结	288
习题	289
附录 A ASCII 码表	291
附录 B 关键字	292
附录 C 运算符的优先级别和结合方向	293
附录 D 常用库函数	294
附录 E C 语言常见出错信息	308
参考文献	312

第1章 C 语言程序设计初步

教学目的

- ☒ 理解程序设计的基本概念
- ☒ 掌握程序设计的基本过程
- ☒ 熟悉 C 语言编程环境——TC 2.0 和 VC 6.0 的 Console 程序开发
- ☒ 了解面向过程和面向对象程序设计方法

在学习任何程序设计之前，有一些基础性的知识需要先了解。对于初次接触程序设计的人，本章可作为一个快速入门；对于有程序设计基础的人，可以略过本章，进而学习后续章节。本章主要包括程序设计的基本概念、程序设计的基本过程以及 C 语言编程环境——TC 2.0 和 VC 6.0 的介绍，并简单介绍现今流行的两种程序设计方法：结构化方法与面向对象的方法。

1.1 程序设计的基本概念

自从 20 世纪 40 年代第一台计算机诞生以来，计算机就以其独特的方式影响着人们的生活和工作，人们不仅被它几近完美的工作效率和方式所吸引，而且为它的发展投入大量的精力和劳动，创造出了一个又一个人间的神话。而这些神话大多围绕着程序设计这个中心而展开，在今天的网络和信息时代，情况更是如此。

1.1.1 程序与程序设计语言

计算机系统(Computer System)是由硬件系统(Hardware System)和软件系统(Software System)构成的，而软件系统主要是由程序(Program)组成的。离开软件，计算机几乎什么也不能做，软件的开发又离不开程序设计语言。如果将计算机比做人的话，那么硬件就是人的躯体，而软件则是人的灵魂。可见，软件在计算机中的地位是何等的重要。下面我们将介绍软件开发中的两个基本概念：程序和程序设计语言。

1. 什么是程序

在计算机尚未诞生之前，人们就有了“程序”的概念，根据《现代汉语词典》的解释，所谓“程序”就是事情进行的先后次序，比如我们平常说的“工作程序”、“会议程序”等。所谓的“违反程序”就是指做某件事情没有按照预先规定的次序和顺序进行。

什么是计算机程序(Computer Program)？根据英国《剑桥高级英语词典》的解释，“计算机程序”就是指为了让计算机完成一项任务，而在计算机中存放的一系列计算机可以识别的指令(Instruction)。打个比方，计算机就像一个优秀的士兵，他无条件服从长官的命令，为了完成一项军事任务，长官所下达的一系列命令就是“程序”。

程序可以简单，也可以复杂，简单的程序就几条指令，而复杂的程序有成千上万条指令。程序的规模越大，内容越复杂，所需要的程序指令就越多，程序的结构也就越复杂。

随着计算机科学技术的飞速发展,程序也变得越来越庞大和复杂,单单一个 Windows 操作系统就有几万行的指令代码,所以给计算机下指令已不是一两个“长官”的事了,而是一个团体集体智慧的结晶。1997 年 5 月,IBM 公司设计的名为“深蓝”(Deeper Blue)的计算机程序,以 4:2 击败了当代国际象棋大师卡斯帕罗夫,这次比赛赢得了 70 万美金的奖金,而 IBM 为开发“深蓝”花费的巨大的人力和物力远远超过了这个数字。

人们把为完成某种任务而编写的一系列指令(程序)交由计算机去执行,这个过程叫做程序设计。程序设计要求编写程序的人(程序员)首先对需要完成任务有一个比较清晰的认识,然后按照计算机可以识别的方式来组织这些指令以形成程序,最后将描述这个任务的程序交由计算机去执行,从而完成这个任务。由于任务的复杂性和多样性,这就使得程序设计不可能一次就达到要求,需要在程序的设计过程不断地修改和完善,最终满足任务的需求,这个过程叫做程序的调试(Debug)和测试(Test)。

2. 什么是程序设计语言

像长官下达命令一样,要想让士兵正确地执行命令,长官的命令必须以一种士兵可以理解的方式来表达,这就是人类的语言。要想让计算机能正确地执行人们所编写的这一系列指令(程序),程序必须以计算机能理解的方式来表达,这就是程序设计语言(Programming Language),它是人们和计算机之间进行交流和通信的工具。

人类的语言是由语法和词汇构成的,同样,计算机语言也是由语法和词汇构成的。所谓的语法就是规则的集合,规定什么是允许的,什么是不允许的,什么是正确的,什么是错误的;词汇也就是符号,它是语言的构成要素。

计算机所能直接识别的语言只有一种,这就是机器语言(Machine Language)。机器语言是由 0 和 1(即二进制, Binary)组成的指令序列。由于人们对二进制数据书写和理解都存在一定的困难,所以产生了多种所谓的高级语言,这些高级语言比较接近人们日常使用的自然语言,给人们书写和阅读程序带来了极大的方便。但高级语言计算机不能直接识别,需要将其“翻译”成机器语言,计算机才能识别和执行,这个过程叫做编译(Compile)。编译过程由相应计算机语言的编译程序自动完成,不需要手工翻译转化。

1.1.2 问题求解与程序设计

人们解决一个实际问题的基本步骤分为 4 步:首先要对问题进行充分分析;然后从分析中找到解决问题的方法与途径;接着按照这个方法与途径写出具体的解决方案;最后实施这个解决方案,使问题得以解决。

虽然说计算机是一种所谓的“智能化”工具,但它并不能代替人的思考和思维,它只是一个忠实的仆人,只是在为人类尽心地服务。所谓的“计算机解决问题”就是人们解决某一问题的方法和步骤的计算机化,或者说是通过计算机来表达人们对某一问题的解决方法。也就是说,一个问题,如果人们不知道如何去求解,那么计算机是不可能做出正确答案的。

程序设计实际就是人们用计算机语言来描述问题的解决方案,所以想让计算机帮我们解决问题,同样需要完成以下 4 个步骤:分析问题;寻找解决问题的方法和步骤;用程序语言来描述解决过程;最后是让计算机来执行这个过程以完成任务。

1.2 程序设计的一般过程

程序设计过程实际上是问题解决方案的计算机描述过程，所以程序设计过程也必须遵循人们解决问题的一般过程，即分析问题，寻找方法，描述方法，实施方法的过程。具体来说，包括以下5个步骤：问题提出与分析；问题的数据描述；问题的功能描述；数据和功能的计算机实现；编码、调试、编译、连接和运行。下面我们以一个具体的实例——“自动应答器”的设计与实现来分析以上过程和步骤。

1.2.1 问题描述

“自动应答器”的基本功能描述如下：当客户走到应答器跟前时，应答器会向客户提示：“请报上您的姓名：”，当客户回答完后，应答器会立刻回答：“XXX 客户，您好！”。

对于非语言的自动应答器其工作过程如下：计算机不能接受语音输入，也不能进行语音输出，所以：（1）当一个客户走到应答器跟前时，应答器会向客户显示信息：“请报上您的姓名：”；（2）当客户看到这个信息后，他（她）会输入本人的姓名，如“Susan”；（3）应答器接收到用户的输入后，会立刻显示：“Susan，您好！”。

1.2.2 问题分析

为了实现自动应答器的基本功能，根据问题描述的内容，它要实现：（1）向屏幕输出提示信息，即向用户屏幕输出“请报上您的姓名：”；（2）等待用户的响应，也就是等待用户输入自己的姓名，如“Susan”；（3）接收用户的输入，并能将用户输入数据连同“您好！”一起显示在用户屏幕。至此一个工作周期完成。为了便于实现，做一些简化假设：假定应答器只为一个人服务，即工作一个周期即告结束。以上的分析过程，也就是这个问题的解决方案和步骤。

1.2.3 计算机实现

为了让计算机能代替人工完成以上的工作流程或者说解决方案，我们需要将以上每一个步骤转化成计算机可以识别的指令序列（即程序）。以下是对这个应答器工作流程的一个计算机描述（其中使用上面的示例显示和输入数据）：

- （1）通过计算机指令向用户屏幕输出“请报上您的姓名：”；
- （2）通过计算机指令等待并接收用户的键盘输入，并且暂存起来以备后用；
- （3）通过计算机指令向用户屏幕输出新的数据信息：用户输入的信息加上“您好！”。

接下来将以上对这个应答器解决过程的计算机描述“翻译”成计算机语言，再通过相应的软件将这些计算机语言转换成计算机可以识别的指令，就可让其“工作”了。

以C语言为例，其翻译过程如下：

- （1）向用户屏幕输出信息可以通过函数 `printf` 来实现，具体形式如下：

```
printf("请报上您的姓名：");
```

- （2）在C语言环境中，数据是以变量的形式存储的，为了存放用户的输入信息，我们可以采用以下形式定义变量，其具体语法含义请参见后续相关章节：

```
char username[8];
```

接收用户的键盘输入并暂存数据，可以通过函数 `scanf` 来实现，具体语法形式如下：

```
scanf("%s",username);
```

(3) 向用户屏幕输出新的数据，同 (1)，仍然可以通过函数 `printf` 来实现，具体形式如下：

```
printf("%s:%s!",username, "您好");
```

将以上“翻译”的 C 语言程序按照 C 语言规定稍作整理，得到例 1.1 程序。

【例 1.1】 自动应答器的 C 语言实现。

程序如下：

```
main()
{
    char username[8];
    printf("请报上您的姓名:");
    scanf("%s",username);
    printf("%s:%s!",username, "您好");
}
```

运行结果：

请报上您的姓名: Susan

Susan: 您好!

至此整个应答器的计算机实现已告完成。

1.2.4 程序运行与调试

在完成问题的求解过程以后，就需要进行上机操作，将设计的程序输入计算机，并经过翻译后，让计算机自动执行，这个过程一般分为 4 个步骤：编辑、编译、调试、连接和运行。下面我们以 C 语言为例，用框图描述进行程序设计的整个过程，如图 1-1 所示。

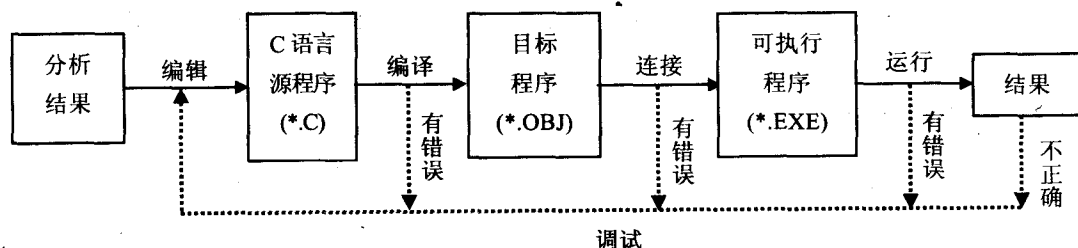


图 1-1 C 语言程序上机过程

下面我们来完成这个自动应答器的计算机处理过程：假设读者的计算机上已经有 C 语言编译系统 (TC 2.0)，并且位于 C 盘的 TC 目录下，如果没有安装，请参阅本章 1.3 节进行 TC 2.0 安装。

1. 编辑

该过程的主要作用是完成代码的基本输入和编辑工作，该过程的使用方法和普通文字编辑软件的用法相同。

在 DOS 环境下，进入 C 盘的 TC 目录，输入“TC”，按回车键，打开 TC 编程环境；输入例 1.1 程序清单，保存为“AUTORESP.C”，完成编码过程。

2. 编译与调试

该过程的作用是完成程序的语法和逻辑错误的检查与修改。该过程是一个反复的过程，即先通过语法检查，如果发现错误，则返回到编辑过程进行修改，修改完成后，再进行语法检查，直到没有语法错误为止。逻辑错误只有到连接过程和运行过程时才能发现。

选择【Compile】→【Compile to OBJ】菜单命令，对上面输入的代码进行语法检查并编译，直到没有错误为止。这时会生成一个目标文件，在此为 AUTORESP.OBJ 文件。如有错误返回到第一步进行修改。

3. 连接

该过程的作用是将目标程序与系统文件进行组装（连接），以生成可执行程序。如果在这个过程中出现错误，返回第一步进行修改，然后重复上面的步骤，直到没有错误为止。

选择【Compile】→【Link Exe file】菜单命令，该命令的功能是将 AUTORESP.OBJ 文件与系统文件相连接，生成可执行程序（EXE 文件），即 AUTORESP.EXE 文件。

4. 运行

该过程的作用是让计算机运行我们所编写的程序，如果运行的结果和我们希望的不一样，就需要返回到开始进行检查、修改，重复上面的过程，直到运行出正确结果为止。

选择【File】→【Exit】菜单命令，退出 TC 程序，在 DOS 命令行，输入程序文件名“AUTORESP”或者“AUTORESP.EXE”，按回车键后，“自动应答器”就会显示信息：“请报上您的姓名：”，用户输入姓名（如：Bill）后，按回车键，“自动应答器”就会显示“Bill，您好！”，如图 1-2 所示。至此整个任务完成。

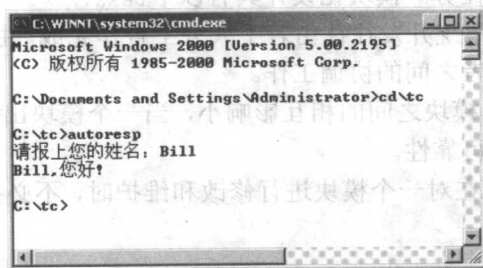


图 1-2 自动应答器运行结果

1.3 程序设计的两种方法

在程序设计语言的发展过程中，程序设计语言和方法先后经历了 3 个主要阶段：非结构化程序设计阶段、结构化程序设计阶段和面向对象程序设计阶段。非结构化程序设计方法由于其诸多的弊端，现已基本淘汰，而面向对象技术已成为程序设计的主流。但在面向对象的设计过程中，对一些细节问题的考虑又离不开结构化程序设计方法，C 语言就属于结构化程序设计语言。下面我们简要介绍结构化程序设计方法和面向对象程序设计方法的内容和特点。

1.3.1 结构化程序设计

结构化程序设计（Structured Programming）是建立在面向过程程序设计的基础之上，