



普通高等教育“十一五”国家级规划教材



21世纪大学本科
计算机专业系列教材

王挺 周会平 贾丽丽 徐锡山 编著

C++ 程序设计 (第3版)

<http://www.tup.com.cn>

- 根据教育部“高等学校计算机科学与技术专业规范”组织编写
- 与美国 ACM 和 IEEE CS *Computing Curricula* 最新进展同步



清华大学出版社



普通高等教育“十一五”国家级规划教材

21世纪大学本科计算机专业系列教材

C++ 程序设计

(第3版)

王挺 周会平 贾丽丽 徐锡山 编著

• • • • •
• • • • •
• • • • •
• • • • •

TP312C-48
521-3

清华大学出版社

开本: 185mm×260mm
版次: 2002年1月第1版
印次: 2001~2002
定价: 48.00元

内 容 简 介

本书参照 ACM 和 IEEE CS CC2013 以及教育部高等学校计算机类专业教学指导委员会编制的《高等学校计算机科学与技术专业发展战略研究报告暨专业规范(试行)》的要求,面向大学计算机专业本科教学的需要,系统地介绍 C++ 程序设计语言的语法、语义和语用,使读者掌握结构化程序设计方法和面向对象程序设计方法。全书分为程序设计基础和面向对象程序设计两大部分。第一部分主要介绍类型、运算符与表达式、输入与输出、控制结构、函数、数组、指针、结构、联合、枚举、链表等程序设计的基本概念和结构化程序设计方法;第二部分主要介绍类与对象、类和对对象的使用、操作符重载、继承、多态性、文件和流、异常和模板等面向对象程序设计的基本概念和程序设计方法。

本书注重知识的系统性和连贯性,在内容上注意与后续课程的衔接,强调严密的逻辑思维,突出程序设计方法的教学。

本书适合作为高等学校“程序设计基础”或者“高级语言程序设计”课程的教材,也可供广大自学人员学习参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C++ 程序设计/王挺等编著. —3 版. —北京:清华大学出版社,2015 (2015.3 重印)
21 世纪大学本科计算机专业系列教材
ISBN 978-7-302-39348-1

I. ①C… II. ①王… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 024973 号

责任编辑:张瑞庆

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:杨 艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市少明印务有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:28.25

字 数:688 千字

版 次:2005 年 1 月第 1 版

2015 年 2 月第 3 版

印 次:2015 年 3 月第 2 次印刷

印 数:2001~3500

定 价:49.00 元

产品编号:058874-01

21 世纪大学本科计算机专业系列教材编委会

主 任：李晓明

副 主 任：蒋宗礼 卢先和

委 员：(按姓氏笔画为序)

马华东 马殿富 王志英 王晓东 宁 洪

刘 辰 孙茂松 李仁发 李文新 杨 波

吴朝晖 何炎祥 宋方敏 张 莉 金 海

周兴社 孟祥旭 袁晓洁 钱乐秋 黄国兴

曾 明 廖明宏

秘 书：张瑞庆

本书责任编辑：钱乐秋

前 言

FOREWORD

本教材参照 ACM 和 IEEE CS CC2013 以及教育部高等学校计算机类专业教学指导委员会编制的《高等学校计算机科学与技术专业发展战略研究报告暨专业规范(试行)》(下面简称《规范》)的要求,力图通过教学,使学生掌握结构化程序设计方法和面向对象程序设计方法,掌握 C++ 程序设计语言的语法、语义和语用,能够熟练运用 C++ 语言解决一般问题。此外,通过课程学习,使学生掌握学习高级程序设计语言的一般方法,养成良好的程序设计风格,对软件工程有初步的认识。本教材虽然以讲授 C++ 为主,但并非单纯讲授一门语言,而是更加强调对程序设计方法的掌握和程序设计风格的养成,为学生今后继续学习其他高级程序设计课程打下牢固的基础。

本书的编写参照 CC2013 和《规范》的知识体系,覆盖或涉及其中程序设计基础(PF)、程序设计语言(PL)、算法和复杂性(AL)3 个领域(area)的多个知识单元(unit)。由于部分知识单元的内容将在其他课程(如数据结构、算法设计与分析、编译原理等)中详细介绍,因此本教材并未深入讲解。

本教材覆盖或涉及 CC2013 和《规范》的如下知识。

1) 程序设计基础(PF)

- 程序设计基本结构。
- 算法和问题求解。
- 基本数据结构。
- 递归。
- 事件驱动程序设计。

2) 程序设计语言(PL)

- 程序设计语言概论。
- 声明和类型。
- 抽象机制。
- 面向对象程序设计。

3) 算法和复杂性(AL)

- 算法策略。
- 基本算法。

本书参考了国外著名高校教材,结合国内高校教学的需要和学生的特点,力求形成如下特色。

(1) 注重知识的系统性和连贯性。本教材面向大学本科计算机专业的学生,参照 CC2013 和《规范》的要求,在教学内容上注意与后续课程(如数据结构、算法设计与分析、编译原理、软件工程等)的衔接。

(2) 突出程序设计方法,强调严密的逻辑思维。本教材并非单纯讲授 C++ 程序设计语言,而是在介绍 C++ 程序设计语言的基础上,系统地讲解程序设计方法,包括结构化程序设计方法和面向对象程序设计方法,特别是后者。程序设计方法是本书的讲授重点。因此,在教材的示例程序中,一般都配有分析、解释和经验总结等,帮助学生领会程序设计的方法和思想。

(3) 注重实践能力的培养。本教材提供丰富的典型例题,每章之后配有难易和综合程度各不相同的习题。教材中还适当地穿插介绍一些编程技巧和软件设计经验,使学生能够从实践的角度更好地学习和掌握 C++ 程序设计方法。

(4) 注重程序设计风格的养成。良好的程序设计风格是保证软件质量的基础。本教材除了有专门章节介绍程序设计风格外,全书始终突出对程序设计风格的要求,并介绍了从分析、设计到编程如何实现良好程序设计风格的方法。本教材的示例程序即按照这种方法实现,在变量命名、源程序格式等方面都贯彻统一的风格。

C++ 是当今最流行的一种高效实用的高级程序设计语言,应用十分广泛。它也是一门复杂的语言,与 C 语言兼容,既支持结构化的程序设计方法,也支持面向对象的程序设计方法,因而成为编程人员最广泛使用的工具。在学习 C++ 的基础上,可以进一步学习其他程序设计语言,C++ 架起了通向强大、易用、真正的应用软件开发的桥梁。本书共分为两大部分:第一部分,第 1 章至第 9 章是基础部分,主要介绍 C++ 程序设计语言、程序结构和结构化程序设计基础;第二部分,第 10 章至第 18 章是面向对象程序设计部分,它建立在 C++ 程序设计基础之上,讲述了面向对象程序设计方法。书中带 * 的章节为选讲内容,可以根据实际情况取舍。

本书是作者根据多年教学实践的经验编写而成,适合作为大学本科计算机专业和非计算机专业的“程序设计基础”或者“高级语言程序设计”课程的教材,也可供广大读者自学参考。由于作者水平有限,书中可能存在缺点和错误,恳请广大读者批评指正。

作者

国防科技大学计算机学院

2015 年 1 月

目 录

CONTENTS

第 1 章 计算机与程序设计	1
1.1 计算机系统概述	1
1.1.1 什么是计算机系统	1
1.1.2 计算机硬件	2
1.1.3 计算机软件	2
1.2 程序设计基本概念	3
1.2.1 问题求解过程	3
1.2.2 算法与程序	4
1.2.3 程序设计语言	7
1.3 程序设计方法	8
1.3.1 结构化程序设计	8
1.3.2 面向对象程序设计	9
1.4 C 语言与 C++ 语言	10
1.4.1 C 语言	10
1.4.2 C++ 语言	11
1.5 C++ 编程简介	12
1.5.1 C++ 编程的典型过程	12
1.5.2 一个简单的 C++ 程序	13
1.6 程序设计风格	14
习题 1	15
第 2 章 数据类型、运算符与表达式	17
2.1 C++ 的字符集和关键字	17
2.1.1 字符集	17
2.1.2 标识符	18
2.1.3 关键字	18
2.2 基本数据类型	19
2.2.1 整型	19



2.2.2	浮点型数据	20
2.2.3	字符型数据	20
2.2.4	布尔类型	21
2.3	常量与变量	21
2.3.1	常量	21
2.3.2	变量	24
2.3.3	符号常量	25
2.4	类型转换	26
2.5	运算符和表达式	27
2.5.1	算术运算	28
2.5.2	关系运算	30
2.5.3	逻辑运算	30
2.5.4	位运算	31
2.5.5	逗号运算	33
2.5.6	赋值运算	34
2.5.7	特殊运算符	35
习题 2	35
第 3 章	输入和输出	37
3.1	C++ 的输入和输出	37
3.2	标准输入输出函数	38
3.2.1	字符输入函数 getchar	38
3.2.2	字符输出函数 putchar	38
3.3	格式化输入输出函数	39
3.3.1	格式化输入函数 scanf	39
3.3.2	格式化输出函数 printf	41
3.3.3	格式化输入/输出函数应用举例	44
3.4	用流进行输入/输出	49
3.4.1	通过 cout 流输出数据	50
3.4.2	通过 cin 流输入数据	51
3.5	流操纵算子	52
3.5.1	设置整数基数的流操纵算子	52
3.5.2	设置浮点数精度的流操纵算子	53
3.5.3	设置域宽的流操纵算子	54
习题 3	55
第 4 章	控制结构	57
4.1	程序的语句和基本控制结构	57
4.2	表达式语句	58

4.3	顺序结构——复合语句	58
4.4	选择结构	60
4.4.1	if 选择结构	60
4.4.2	if-else 选择结构	61
4.4.3	switch 选择结构	66
4.5	循环结构	70
4.5.1	while 循环语句	70
4.5.2	do-while 循环语句	73
4.5.3	for 循环语句	74
4.6	控制转移语句	78
4.6.1	break 语句	78
4.6.2	continue 语句	80
4.6.3	goto 语句	81
4.6.4	return 语句	82
4.7	结构化程序设计方法	83
4.7.1	结构化程序设计思想	83
4.7.2	结构化程序设计原则	85
4.7.3	结构化程序设计示例	86
4.7.4	再谈程序设计风格	90
习题 4		91
第 5 章	函数	95
5.1	模块化程序设计	95
5.2	预定义函数的使用	96
5.3	函数定义与函数原型	97
5.3.1	函数定义	97
5.3.2	函数原型	98
5.4	函数的调用	99
5.4.1	函数调用的概念	99
5.4.2	参数传递	102
5.4.3	默认参数	104
5.4.4	内联函数	105
5.5	函数重载	106
5.6	存储类别	108
5.6.1	变量的存储特性	108
5.6.2	自动变量	109
5.6.3	寄存器变量	109
5.6.4	外部变量	110
5.6.5	静态变量	110

5.7	作用域	112
5.8	递归函数	114
5.8.1	递归的概念	115
5.8.2	递归函数应用举例	117
5.9	预处理指令	120
5.9.1	#include 指令	121
5.9.2	#define 指令	121
5.9.3	* #if、#else、#endif、#ifdef、#ifndef 指令	123
	习题 5	124
第 6 章 数组		127
6.1	一维数组	128
6.1.1	一维数组的声明与初始化	128
6.1.2	数组的应用	129
6.1.3	数组作为函数参数	132
6.2	多维数组	137
6.2.1	多维数组的定义与初始化	137
6.2.2	应用示例	139
6.3	字符串	142
6.3.1	字符与字符串概念	142
6.3.2	字符串与字符数组	143
6.3.3	字符串的输入和输出	145
	习题 6	146
第 7 章 指针		148
7.1	指针的定义	148
7.2	指针的运算	149
7.3	指针与数组	154
7.4	字符指针与字符数组	157
7.4.1	字符串函数	157
7.4.2	字符串函数的应用	163
7.5	指针与 const 限定符	166
7.5.1	基本概念	166
7.5.2	* 用 const 限定指针	166
7.6	指针和引用	170
7.7	动态内存分配	171
7.7.1	基本概念	171
7.7.2	动态分配内存的应用	173
7.8	* 函数指针	176

7.8.1	函数指针的定义	176
7.8.2	函数指针的使用	177
	习题 7	179
第 8 章	结构、联合、枚举	182
8.1	结构	182
8.1.1	结构的定义	183
8.1.2	结构变量成员的引用	184
8.2	结构的使用	185
8.2.1	结构与函数	185
8.2.2	结构与数组	190
8.2.3	结构与指针	193
8.2.4	位段	196
8.3	联合	198
8.3.1	联合和联合变量定义	199
8.3.2	联合的使用	200
8.4	枚举	201
8.4.1	枚举和枚举型变量的定义	202
8.4.2	枚举类型变量的赋值和使用	203
	习题 8	204
第 9 章	链表	205
9.1	链表的基本概念	205
9.2	单向链表	206
9.2.1	单向链表的定义	206
9.2.2	单向链表的操作	207
9.3	双向链表	216
9.3.1	双向链表的定义	216
9.3.2	双向链表的操作	217
	习题 9	224
第 10 章	面向对象程序设计基本概念	226
10.1	面向对象语言和面向对象方法	226
10.2	类、对象和消息	228
10.2.1	类和对象	228
10.2.2	消息	230
10.3	面向对象程序设计的特点	231
10.4	面向对象程序的结构	233
	习题 10	235

第 11 章 类与对象	237
11.1 数据抽象的概念	238
11.2 抽象数据类型	239
11.2.1 封装与信息隐藏	240
11.2.2 接口与实现的分离	240
11.2.3 用结构实现用户定义类型栈	241
11.2.4 用类实现抽象数据类型栈	245
11.3 类和对象的定义	249
11.3.1 数据成员	250
11.3.2 成员函数	251
11.3.3 访问控制	252
11.3.4 静态成员	253
11.3.5 对象的建立	257
11.4 构造函数	258
11.4.1 构造函数的作用	258
11.4.2 构造函数执行的时机	259
11.4.3 构造函数重载	260
11.4.4 默认构造函数	261
11.4.5 复制构造函数	262
11.5 析构函数	264
11.5.1 析构函数的作用	264
11.5.2 析构函数执行的时机	264
习题 11	266
第 12 章 类和对象的使用	268
12.1 类的复合	268
12.2 this 指针	276
12.3 const 特性	279
12.4 友元函数和友元类	283
12.4.1 友元函数	283
12.4.2 友元类	285
习题 12	286
第 13 章 运算符重载	288
13.1 运算符重载的概念	288
13.1.1 运算符重载的意义	289
13.1.2 运算符重载的限制	292
13.2 运算符成员函数与友元函数	293

13.3	单目运算符重载	294
13.4	重载流插入和流提取运算符	299
13.5	双目运算符重载	302
13.6	赋值运算符重载	305
13.7	类型之间的转换	309
	习题 13	317
第 14 章 继承		319
14.1	继承和派生的概念	319
14.2	继承的定义	320
14.2.1	派生类和基类	320
14.2.2	继承的方式	322
14.2.3	类的层次	323
14.2.4	在派生类中重定义基类的函数	324
14.2.5	派生类和基类的转换	329
14.3	类指针	329
14.4	继承中的构造函数和析构函数	336
14.5	多重继承	339
14.6	软件渐增式开发	343
14.6.1	复合与继承	344
14.6.2	示例	344
	习题 14	355
第 15 章 多态性		356
15.1	多态性的概念	356
15.1.1	静态绑定和动态绑定	357
15.1.2	多态性的意义	357
15.2	虚函数	358
15.3	抽象基类和纯虚函数	364
15.3.1	纯虚函数	365
15.3.2	抽象类和具体类	365
15.4	虚析构函数	369
15.5	软件渐增式开发	374
	习题 15	382
第 16 章 文件和流		383
16.1	基本概念	383
16.2	通过 FILE 结构进行文件操作	384
16.2.1	建立、打开和关闭文件	385

16.2.2	写文件	387
16.2.3	读文件	389
16.3	通过文件流进行文件操作	396
16.3.1	打开和建立文件	396
16.3.2	写文件	398
16.3.3	读文件	399
习题 16		405
第 17 章	异常	407
17.1	异常处理的意义	407
17.2	异常处理基础	408
17.3	异常的抛出和传播	411
17.4	异常的捕获和处理	415
习题 17		418
第 18 章	模板	419
18.1	类属机制	419
18.2	函数模板	420
18.2.1	函数模板的定义	420
18.2.2	使用函数模板	421
18.3	类模板	425
18.3.1	类模板的定义	426
18.3.2	使用类模板	427
习题 18		429
附录 A	C++ 运算符的优先级和结合性	431
附录 B	ASCII 字符集	434
参考文献		436

第 1 章

计算机与程序设计

【学习内容】

本章介绍计算机与程序的基本概念。主要内容包括：

- ◆ 计算机系统的基本组成和工作原理。
- ◆ 程序设计的基本过程和方法。
- ◆ 程序设计语言的基本概念及其发展历史。
- ◆ 结构化程序设计方法和面向对象程序设计方法概述。
- ◆ C 语言与 C++ 语言。
- ◆ 典型的 C++ 编程环境、开发过程和简单的 C++ 程序结构。
- ◆ 程序设计风格。

【学习目标】

- ◆ 了解计算机系统、程序设计语言、程序设计的基本概念。
- ◆ 通过简单示例掌握 C++ 程序的结构。
- ◆ 理解程序设计风格的意义。

1.1 计算机系统概述

1.1.1 什么是计算机系统

根据中国大百科全书的定义,计算机系统(computer system)是按人的要求接收和存储信息,自动进行数据处理和计算,并输出结果信息的机器系统。计算机是脑力的延伸和扩充,是近代科学的重大成就之一。计算机系统由硬件系统和软件系统组成,前者是借助电、磁、光、机械等原理构成的各种物理部件的有机组合,是系统赖以工作的实体,如 CPU、显示器、内存、硬盘和键盘等;后者是各种程序和文档,用于指挥全系统按指定的要求进行工作,程序是对计算任务的处理对象和处理规则的描述,而文档是与软件研制、维护和使用有关的资料。随着计算机技术的迅速发展,硬件的成本在不断下降,个人计算机得到了广泛应用和普及。这一趋势也为计算机在信息处理方面提出了更高的要求——设计实现更方便、功能更加强大的应用程序。但是,相对于硬件技术的发展,软件的开发技术明显滞后。现阶段的软件设计更多地依赖于人,而软件设计的方法,包括程序设计的方法,是提高软件生产率和质量的关键。

1.1.2 计算机硬件

计算机硬件系统是计算机系统快速、可靠、自动工作的基础。计算机硬件就其逻辑功能来说,主要是完成信息变换、信息存储和信息传送等处理功能,它为计算机软件提供具体实现的基础。计算机硬件系统主要由运算器、主存储器、控制器、输入设备、输出设备和辅助存储器等功能部件组成。

1. 运算器

运算器的主要功能是对数据进行算术运算和逻辑运算。整个运算过程是在控制器的控制下自动进行的。操作时,运算器从主存储器取得运算数据,经过指令指定的运算处理,所得运算结果或者留在运算器内以备下次运算时使用,或者写入主存储器。

2. 主存储器

主存储器的主要功能是存储二进制信息。它与运算器、控制器等快速部件直接交换信息。从主存储器中应能快速读出信息,并送到其他功能部件中去,或者将其他功能部件处理过的信息快速写入主存储器。

3. 控制器

控制器的功能主要是按照机器代码程序的要求,控制计算机各功能部件协调一致地动作,即从主存储器取出程序中的指令,并对该指令进行分析和解释,然后向其他功能部件发出执行该指令所需要的各种时序控制信号,再从主存储器取出下一条指令执行,如此连续运行下去,直到程序执行完为止。计算机自动工作的过程就是逐条执行程序指令的过程。控制器与运算器一起构成中央处理器,中央处理器与主存储器一起构成处理机。

4. 输入设备

输入设备主要是将用户信息(数据、程序等)变换为计算机能识别和处理的信息形式。输入设备种类很多,如纸带阅读机、软磁盘机、汉字输入设备、键盘输入设备等。其工作特点是将人工编制的程序和原始数据,在某种媒介物上以二进制编码形式表现,如纸带上穿孔和不穿孔分别表示1和0,磁表面上磁化方向不同以区别1和0等。载有信息的媒介物通过相应的输入设备,将信息变换成电信号为计算机接收,并存入存储器。

5. 输出设备

输出设备主要是将计算机中二进制信息变换为用户所需要并能识别的信息形式。输出设备种类很多,如打印机、凿孔输出机、汉字输出设备、绘图仪、显示终端、声音输出设备等。其工作特点与输入设备正好相反,是将计算机中二进制信息经过相应变换,成为用户需要的信息形式,记录在媒介物上或通过各种媒体形式表现出来供用户使用。输出的信息形式多为十进制数字、字符、图形、表格、声音等。

6. 辅助存储器

辅助存储器主要是存放主存储器难以容纳、又为程序执行所需要的大量文件信息,其特点是存储容量大,存储成本低,但存取速度较慢。它不能直接与中央处理器交换信息。辅助存储器一般为磁盘、磁带、光盘和优盘等。

1.1.3 计算机软件

光有硬件计算机还不能工作,要使计算机能解决各种实际问题,还必须要有软件的支持。

软件是用户与硬件之间的接口界面。使用计算机就必须针对待解的问题拟定算法,用计算机所能识别的语言对有关的数据和算法进行描述,即必须编写程序。用户通过软件与计算机进行交互。软件规定了计算机系统如何进行工作,包括各项计算任务内部的工作内容和 workflow,以及各项任务之间的调度和协调。软件是计算机系统结构设计的重要依据。为了方便用户,在设计计算机系统时,必须通盘考虑软件与硬件的结合,以及用户的要求和软件的要求。

计算机软件通常分为系统软件、支撑软件和应用软件。

1. 系统软件

系统软件居于计算机系统中最靠近硬件的一层。其他软件一般都通过系统软件发挥作用。系统软件与具体的应用领域无关,如操作系统和编译程序等。操作系统(operating system)负责管理系统的各种资源、控制程序的执行。编译程序(compiler)把程序员用高级语言书写的程序翻译成与之等价的、可执行的低级语言程序,如机器语言程序。在任何计算机系统的设计中,系统软件都要优先予以考虑。

2. 支撑软件

支撑软件是用于支撑其他软件开发和维护的软件。随着计算机科学技术的发展,软件的开发和维护代价在整个计算机系统中所占的比重很大,远远超过硬件。因此,支撑软件的研究具有重要意义,可直接促进软件的发展。当然,编译程序、操作系统等系统软件也可算做支撑软件。但是,20世纪70年代中后期发展起来的软件支撑环境可看成是现代支撑软件的代表,主要包括环境数据库、各种接口软件和工具组。三者形成整体,协同支撑其他软件的开发。

3. 应用软件

应用软件是特定应用领域专用的软件,如天气预报用的计算软件就是一种应用软件。对于具体的应用领域,应用软件的质量往往成为影响实际效果的决定性因素。随着计算机技术的发展,特别是互联网出现以来,计算机在整个社会中应用的深度和广度都在迅速提高,各种应用软件的复杂程度也越来越高,不断地推动软件技术的发展。

上述分类不是绝对的,而是互相交叉和变化的。有些软件,如操作系统和编译程序,既可看做是系统软件,又可看做是支撑软件。它们在一个系统中是系统软件,而在另一个系统中却是支撑软件;也可以在同一系统中既是系统软件,又是支撑软件。系统软件和应用软件之间也有类似情况。有的软件,如数据库管理系统、网络软件和图形软件,原来作为应用软件,后来又作为支撑软件。而且系统软件、支撑软件和应用软件三者的开发技术基本相同。因此,这三者既有分工,又有结合,并不截然分开。

1.2 程序设计基本概念

1.2.1 问题求解过程

在计算机中,一切信息处理都要受程序的控制,数值数据如此,非数值数据也是如此。因此任何问题求解(problem solving)最终要通过执行程序来完成。根据传统的(面向过程)程序设计方法,把计算机的应用需求转变为可在计算机上运行的程序,一般要经历问题定义、算法设计、程序编码、程序测试等步骤。