



21 世纪大学本科
计算机专业系列教材

石峰 编著

程序设计基础

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

- 根据教育部高教司主持评审的《中国计算机科学与技术学科教程 2002》组织编写
- 与美国 ACM 和 IEEE/CS 《Computing Curricula 2001》同步

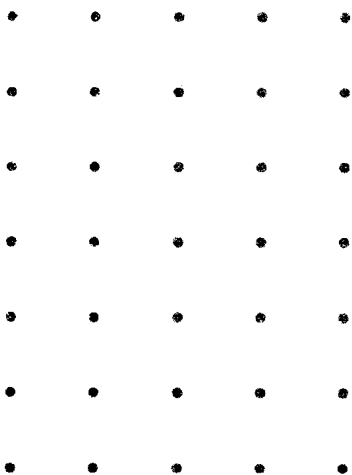


清华大学出版社

21世纪大学本科计算机专业系列教材

程序设计基础

石峰 编著



清华大学出版社

北京

内 容 简 介

本书从程序设计的一般过程与方法出发,以 C++ 语言为媒介,重点讲述结构化程序设计与面向对象程序设计的基础知识。本书集算法分析与设计、面向对象分析与设计、软件工程基础、C++ 语言编程等多方面知识于一身,详细分析了产生相关理论与技术的起因,力求使读者不但知其然,而且知其所以然,因此本书适合作为程序设计的基础教程。

在编著风格上,本书从分析人类解决问题的思维过程为开端,介绍计算机的问题求解过程和方式;从人类认识世界的规律和方法入手,讲解面向对象技术理论和方法;通过对比,使读者加深对所学知识的理解和掌握。

本书是在作者多年教学工作的基础上,结合国内外多部优秀教材,通过深入分析、提炼和加工而成,适合作为计算机及相关专业的教材,也可供从事软件开发和应用的广大科技人员参考。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

程序设计基础/石峰编著. —北京:清华大学出版社,2003

(21世纪大学本科计算机专业系列教材)

ISBN 7-302-07397-X

I. 程… II. 石… 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(2003)第 092331 号

出 版 者: 清华大学出版社

地 址: 北京清华大学学研大厦

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

邮 编: 100084

社 总 机: 010-62770175

客 户 服 务: 010-62776969

责任编辑: 张瑞庆

封面设计: 孟繁聪

印 刷 者: 清华大学印刷厂

装 订 者: 三河市新茂装订有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×230 印 张: 28 字 数: 541 千字

版 次: 2003 年 11 月第 1 版 2003 年 11 月第 1 次印刷

书 号: ISBN 7-302-07397-X/TP·5363

印 数: 1~6000

定 价: 35.00 元

21世纪大学本科计算机专业系列教材编委会

名誉主任：陈火旺

主任：李晓明

副主任：钱德沛 焦金生

委员：（按姓氏笔画为序）

马殿富 王志英 王晓东 宁洪 刘辰

孙茂松 李大友 李仲麟 吴朝晖 何炎祥

宋方敏 张大方 张长海 周兴社 侯文永

袁开榜 钱乐秋 黄国兴 蒋宗礼 曾明

廖明宏 樊孝忠

秘书：张瑞庆

本书责任编辑：张长海

序 言

PREFACE

21 世纪是知识经济的时代,是人才竞争的时代。随着 21 世纪的到来,人类已步入信息社会,信息产业正成为全球经济的主导产业。计算机科学与技术的信息产业中占据了最重要的地位,这就对培养 21 世纪高素质创新型计算机专业人才提出了迫切的要求。

为了培养高素质创新型人才,必须建立高水平的教学计划和课程体系。在 20 多年跟踪分析 ACM 和 IEEE 计算机课程体系的基础上,紧跟计算机科学与技术的发展潮流,及时制定并修正教学计划和课程体系是尤其重要的。计算机科学与技术的发展对高水平人才的要求,需要我们从总体上优化课程结构,精炼教学内容,拓宽专业基础,加强教学实践,特别注重综合素质的培养,形成“基础课程精深,专业课程宽新”的格局。

为了适应计算机科学与技术学科发展和计算机教学计划的需要,要采取多种措施鼓励长期从事计算机教学和科技前沿研究的专家教授积极参与计算机专业教材的编著和更新,在教材中及时反映学科前沿的研究成果与发展趋势,以高水平的科研促进教材建设。同时适当引进国外先进的原版教材。

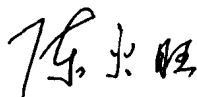
为了提高教学质量,需要不断改革教学方法与手段,倡导因材施教,强调知识的总结、梳理、推演和挖掘,通过加快教案的不断更新,使学生掌握教材中未及时反映的学科发展新动向,进一步拓广视野。教学与科研相结合是培养学生实践能力的有效途径。高水平的科研可以为教学提供最先进的高新技术平台和创造性的工作环境,使学生得以接触最先进的计算机理论、技术和环境。高水平的科研还可以为高水平人才的素质教育提供良好的物质基础。学生在课题研究中不但能了解科学研究的艰辛和科研工作者的奉献精神,而且能熏陶和培养良好的科研作风,锻炼和培养攻关能力和协作精神。

进入 21 世纪,我国高等教育进入了前所未有的大发展时期,时代的进步与发展对高等教育质量提出了更高,更新的要求。2001 年 8 月,教育部颁发了《关于加强高等学校本科教学工作,提高教学质量的若干意见》。文件指出,本科教育是高等教育的主体

和基础,抓好本科教学是提高整个高等教育质量的重点和关键。随着高等教育的普及和高等学校的扩招,在校大学本科计算机专业学生的数量将大量上升,对适合 21 世纪大学本科计算机科学与技术学科课程体系要求的,并且适合中国学生学习的计算机专业教材的需求量也将急剧增加。为此,中国计算机学会和清华大学出版社共同规划了面向全国高等院校计算机专业本科生的“21 世纪大学本科计算机专业系列教材”。本系列教材借鉴美国 ACM 和 IEEE/CS 最新制定的《Computing Curricula 2001》(简称 CC2001)课程体系,反映当代计算机科学与技术学科水平和计算机科学技术的新发展、新技术,并且结合中国计算机教育改革成果和中国国情。

中国计算机学会教育专业委员会和全国高等学校计算机教育研究会,在清华大学出版社的大力支持下,跟踪分析 CC2001,并结合中国计算机科学与技术学科的发展现状和计算机教育的改革成果,研究出了《中国计算机科学与技术学科教程 2002》(China Computing Curricula 2002,简称 CCC2002),该项研究成果对中国高等学校计算机科学与技术学科教育的改革和发展具有重要的参考价值 and 积极的推动作用。

“21 世纪大学本科计算机专业系列教材”正是借鉴美国 ACM 和 IEEE/CS CC2001 课程体系,依据 CCC2002 基本要求组织编写的计算机专业教材。相信通过这套教材的编写和出版,能够在内容和形式上显著地提高我国计算机专业教材的整体水平,继而提高我国大学本科计算机专业的教学质量,培养出符合时代发展要求的具有较强国际竞争力的高素质创新型计算机人才。



中国工程院院士

国防科学技术大学教授

21 世纪大学本科计算机专业系列教材编委会名誉主任

2002 年 7 月

前 言

FOREWORD

坐在计算机前,挥洒自如地驰骋于电脑世界,一行行程序在自己手下源源不断地流出,这是一些初学编程者的美好憧憬。然而,在学习了程序设计语言的具体语法后,真正动起手来编写程序时,往往会觉得一筹莫展、无所适从;在遇到一种新型编程语言时,即便是已经掌握了某些编程语言,学起来也常常会感到举步维艰,没有体会到已有的知识积累给学习新语言带来的便利。为什么会这样?原因在于没有掌握程序设计的一般过程,没有深入了解通用程序设计语言的本质规律。

通用程序设计语言是与人类自然语言具有一定程度相似性的人造语言,使用通用程序设计语言进行程序设计,并不能简单地等价于人们所常说的“编程”。程序设计是一个过程,通常分为3个阶段:算法分析与设计、程序编写与调试、程序测试。正像人们利用自然语言书写文章一样,与此3个阶段相对应的是构思、起草与修改、审阅。不同阶段的任务是相对独立的,如果对此没有深入的了解,就很容易将它们混为一谈。在缺乏对问题深入、全面分析的情况下,就匆匆动手编写程序,自然就会像没有经过构思而书写出来的文章一样。我们编写本书的目的,就在于使读者通过本课程的学习,从刻意地按照程序设计的准则编写程序,到自然而然地遵从程序设计准则从事程序设计。为实现这个目的,算法分析和设计自然成为本课程的主要学习任务之一。

程序设计归根到底是要使用程序设计语言将问题的解题过程描述出来,因此学习程序设计语言是本课程的另一个重要任务。本书虽然以C++语言为背景介绍命令式程序设计语言的相关知识,但是读者应该注意到,现有的通用程序设计语言的语法和风格可能迥异,但无论哪一种语言,都是以数据(类型)、操作(运算)、(操作)顺序控制和数据控制为基本内容,不同的语言都以不同的格式对这4个方面提供不同的支持,因此读者在学习本书的具体内容时,可以不管语言的具体表述格式,而是深入学习和理解这4个方面的实质,这会为学习其他程序设计语言带来便利。

与许多其他同类教材不同的是,本书强调软件开发过程的重要性,为读者介绍了有

关软件开发模型与软件测试的基本原理和技术。现代软件开发多为集体合作,在合作过程中每一个编程者所编写程序的规范性、正确性和可靠性非常重要,在学习程序设计之初就了解程序设计的规范过程,对于培养将来参与大型软件开发所需要的分工合作精神十分重要。

全书分为3个部分:第一部分包括第1章和第2章,讲述程序设计的基本概念和思想,读者在学习过程中应该以了解和掌握结构化算法设计为重点;第二部分包括第3章至第8章,以C++语言中的非面向对象语法为例,讲述一般程序设计语言所具有的基本内容:数据(类型)、操作、(操作)顺序控制、数据控制,同时重点介绍了如何进行结构化程序设计及软件开发的一般过程,对于这部分内容的学习,读者应该抛却具体的语言,从最一般的观点(通过程序)理解算法结构与解题过程描述;第三部分由第9章至第12章组成,重点介绍面向对象程序开发技术的基本方法,包括面向对象分析与设计的基本思想和主要步骤,以及C++中的面向对象特征的语法,使读者了解和掌握人类认知的一般规律,以及如何通过适当的步骤,采用面向对象技术和面向对象语言的基本语法,将自身对客观事物的认识描述出来。

本教材的内容从简到难,覆盖面广,有些内容涉及比较深入的知识(标题中带有*号),教师在使用本教材时,应该根据学生及教学的具体情况对内容进行适当裁剪。总之,学习的重点应该放在算法和程序结构的设计,注重以程序设计语言的习惯和方式表述问题的解决方案,而不在于具体语言的语法细节。

本书所有例题和习题中的程序都已在Windows XP环境Visual C++ 6.0编译系统下运行通过,读者在学习时选择Visual C++ 5.0以上的版本即可。

本书是在多年教学基础上编写的,适合作为计算机及其相关专业本科教学用书,也可用作其他专业的计算机教学参考书。对于有志深入学习程序设计的广大电脑爱好者,本书也是一个好的选择。书中所用到的部分素材来源于教学过程中的积累和其他中外文教材、资料,由于无法在此一一列举,现谨对这些教材和资料的作者表示衷心的感谢。在教材编写过程中,对于吉林大学张长海教授及编委会成员给予的关心和指导以及清华大学出版社所给予的大力支持,作者在此表示最衷心的感谢! 恳请各位专家不吝赐教。

作者

2003年10月



CONTENTS

第 1 章 程序设计概论	1
1.1 问题求解	1
1.1.1 思维与语言	1
1.1.2 算法	3
1.1.3 数据结构	5
1.1.4 程序	7
1.1.5 程序设计的一般过程	8
1.2 程序设计语言	9
1.2.1 发展历史	9
1.2.2 程序设计语言的实现	15
1.2.3 高级语言程序的基本结构	18
1.2.4 高级语言的语法规则	21
1.2.5 程序的编辑、编译和运行	30
1.3 程序设计范式的演化	32
1.3.1 结构化程序设计	32
1.3.2 函数式程序设计	33
1.3.3 面向对象程序设计	34
1.3.4 事件驱动的程序设计	35
1.3.5 逻辑式程序设计	36
习题	37
第 2 章 算法设计基础	42
2.1 算法的描述	42
2.1.1 自然语言方式	42

2.1.2	伪代码方式	43
2.1.3	程序流程图方式	46
2.1.4	N/S 盒图方式	47
2.1.5	PAD 图方式	48
2.2	结构化算法设计初步	50
2.2.1	算法描述	50
2.2.2	算法设计	53
2.3	算法的计算复杂性	58
2.4	常用算法设计策略	60
2.4.1	分治法	60
2.4.2	递归技术	61
2.4.3	贪心法*	62
2.4.4	回溯法*	64
	习题	65

第3章 基本数据类型

3.1	数据对象	67
3.1.1	数据的存储	68
3.1.2	生存期与值	69
3.2	基本数据类型	70
3.2.1	整型与浮点型	71
3.2.2	字符类型	71
3.2.3	逻辑型	72
3.2.4	空值型	72
3.3	变量与常量	73
3.3.1	常量	73
3.3.2	变量	75
3.4	指针类型	77
3.4.1	指针定义	77
3.4.2	指针的初始化	78
3.4.3	指针运算	79
3.4.4	指针与常量	80
3.5	引用类型	81
3.6	类型的意义	82

习题	83
第 4 章 操作及其控制	85
4.1 基本操作与运算符	85
4.1.1 赋值操作	85
4.1.2 算术操作	86
4.1.3 增 1 和减 1 操作	86
4.1.4 关系、逻辑操作	87
4.1.5 条件运算符	87
4.2 表达式	87
4.2.1 隐含类型转换	88
4.2.2 强制类型转换	88
4.2.3 表达式内的顺序控制	89
4.2.4 运算符的优先级	90
4.2.5 运算符的结合性	90
4.3 语句	91
4.3.1 表达式语句	91
4.3.2 语句间顺序控制	92
4.3.3 复合语句	92
4.3.4 选择语句	93
4.3.5 循环语句	97
4.3.6 直接顺序控制	103
4.4 预处理指令	106
4.4.1 文件包含指令	106
4.4.2 宏定义指令	106
4.4.3 条件编译指令	110
习题	113
第 5 章 函数与数据控制	122
5.1 函数定义与说明	122
5.1.1 函数定义	122
5.1.2 函数说明	123
5.1.3 函数的简单调用	125
5.1.4 函数的重载	126

5.2	数据控制	128
5.2.1	参数传递机制	128
5.2.2	数据对象的引用*	132
5.2.3	作用域	135
5.3	函数的顺序控制	141
5.3.1	函数执行模型*	142
5.3.2	基于栈的实现*	144
5.3.3	函数的自嵌套调用	145
5.4	相关的其他语法	146
5.4.1	数组作函数参数	146
5.4.2	函数指针	149
5.5	常用系统函数	153
5.5.1	终止程序运行	153
5.5.2	数学函数	154
5.5.3	字符串处理函数	154
5.5.4	面向对象的数据结构	154
	习题	155
第6章	构造类型	164
6.1	数组类型	164
6.1.1	数组定义	164
6.1.2	数组的赋值与使用	165
6.1.3	数组的存储	165
6.1.4	数组的初始化	166
6.1.5	字符数组	167
6.1.6	指针与数组	168
6.2	枚举类型	169
6.2.1	枚举类型定义	170
6.2.2	枚举变量定义	170
6.2.3	类型转换	171
6.3	结构类型	172
6.3.1	结构的定义	172
6.3.2	结构的使用	173
6.3.3	结构的实现	175

6.4	结构化数据	175
6.4.1	单链表	176
6.4.2	栈	178
6.4.3	存储的动态分配	181
6.5	抽象数据类型	186
6.5.1	结构的扩展	186
6.5.2	C++ 的抽象数据类型	187
6.5.3	C++ 数据类型的层次	189
6.6	自定义类型	189
	习题	189
第 7 章	结构化算法的实现	198
7.1	基本控制结构的 C++ 实现	198
7.1.1	顺序结构的 C++ 实现	199
7.1.2	分支结构的 C++ 实现	199
7.1.3	循环结构的 C++ 实现	201
7.1.4	复杂结构的 C++ 实现示例	202
7.2	子算法设计与 C++ 实现	203
7.2.1	参数为普通类型的子算法	204
7.2.2	参数为指针的子算法	205
7.2.3	参数为引用的子算法	208
7.2.4	子算法设计与 C++ 实现示例	210
7.3	递归与迭代	215
7.3.1	递归	215
7.3.2	迭代	218
7.3.3	应用示例	219
	习题	223
第 8 章	程序开发过程	225
8.1	软件开发方法概述	226
8.1.1	软件生存周期	226
8.1.2	软件开发方法	226
8.2	软件设计	228
8.2.1	概要设计	228

8.2.2	详细设计	229
8.3	软件编码	230
8.3.1	程序设计方法	230
8.3.2	程序设计语言	230
8.3.3	编码风格	230
8.4	软件测试与调试	233
8.4.1	调试工具及使用	233
8.4.2	调试过程	234
8.4.3	错误类型	235
8.4.4	异常处理*	238
8.4.5	软件测试	242
8.5	程序运行效率	244
8.5.1	适当的算法	245
8.5.2	选择快速运算	246
8.5.3	函数	249
	习题	250
第9章	面向对象程序设计	256
9.1	面向对象方法学	256
9.1.1	人类认知规律	257
9.1.2	面向对象基本原理	258
9.1.3	认知规律与面向对象	259
9.1.4	面向对象软件开发	259
9.2	基本概念	260
9.2.1	对象	260
9.2.2	类	266
9.2.3	结构与关系	267
9.3	面向对象分析	274
9.3.1	主要原则	275
9.3.2	静态结构分析与对象模型	276
9.3.3	动态行为分析	280
9.4	面向对象设计	284

习题	285
第 10 章 类与对象的实现	286
10.1 C++ 中的类	286
10.1.1 类的说明	286
10.1.2 类的定义	288
10.2 对象及其成员的使用	291
10.2.1 对象实例	291
10.2.2 对象成员的引用	292
10.2.3 生命周期	295
10.2.4 类型转换*	301
10.2.5 其他应用	305
10.3 类的特殊成员	315
10.3.1 静态成员	315
10.3.2 const 成员	319
10.3.3 内联函数	321
10.3.4 友元	322
习题	324
第 11 章 类间关系的实现	334
11.1 一般—特殊关系的实现	334
11.1.1 类的继承与派生	334
11.1.2 赋值兼容规则	344
11.1.3 两义性与作用域分辨	346
11.2 多态性与虚函数	353
11.2.1 编译时刻的多态性	354
11.2.2 运行时刻的多态性	355
11.2.3 虚函数	356
11.2.4 纯虚函数与抽象类	366
11.3 整体—部分关系的实现	370
11.4 关联关系的实现	373
11.5 关于类层次的总结	374

11.5.1	认知规律与类层次.....	374
11.5.2	构造函数的一般形式.....	375
11.5.3	成员函数的特征.....	376
	习题.....	377
第 12 章	面向对象程序设计的其他技术	382
12.1	运算符重载.....	382
12.1.1	友元运算符.....	383
12.1.2	类运算符.....	385
12.1.3	友元及类运算符的应用.....	386
12.1.4	++ 和 - 运算符的重载.....	388
12.2	流类库.....	391
12.2.1	格式化输入输出.....	393
12.2.2	运算符<< 和>> 的重载	394
12.2.3	流式文件.....	395
12.2.4	流错误.....	408
12.3	模板.....	409
12.3.1	函数模板.....	409
12.3.2	类模板.....	410
12.3.3	模板应用实例.....	412
	习题.....	416
附录 A	Microsoft C++ 的保留字	419
附录 B	C++ 中的基本数据类型	420
附录 C	扩展 ASCII 字符集	421
附录 D	C++ 的运算符	422
	词汇索引	424
	参考文献	429

第 1 章

程序设计概论

程序设计以算法设计和数据结构为基础,涉及编码(程序编制)、调试和测试等多方面的理论和技术,本章介绍程序设计的发展历史,程序设计与算法和数据结构之间的关系以及高级语言的基本语法特征。作为全书的导引,本章提前使用了某些概念,读者对此只需作一般性了解,待深入学习后续章节后将会深刻理解这些概念。

1.1 问题求解

利用计算机解决问题需要使用程序设计语言对问题的求解进行描述,按照程序设计语言的规则将问题的求解过程描述成计算机能够接受的方式——程序,这个过程就是编写程序的过程,处于运行状态的程序将控制计算机按照程序所规定的顺序对相关数据进行操作,最终给出问题的解。由于到目前为止,使用计算机解决问题的过程基本上是在模拟人类大脑的解题过程,因此在介绍如何利用计算机解决问题之前,有必要分析人类是如何解决问题的,以及在解决问题过程中人类是如何进行思维活动的。

1.1.1 思维与语言

人类的思维从本质上说分为两类:形象思维和逻辑思维。利用形象思维,人类可以从一个公理体系出发,使用从特殊到一般的归纳以及从一般到特殊的演绎进行推理,从而解决给定的问题,这是一种以推理方式为主的思维方式,深刻体现了形象思维的特点,代表了西方数学思想方法的主要方面。这种推理思维方式现在已经为人们所习惯,例如,常见的数学定理的证明过程。利用逻辑思维,人类可以从一系列具有共性的解决问题的过程中,发现解决这类问题的通用方法,这种构造解决问题算法的思维方式,深刻体现了逻辑思维的特点,代表了东方数学思想方法的主要方面。长期以来,由于西方