

PROGRAMMING
BASED ON
COMPUTATIONAL
THINKING



“十三五” 江苏省高等学校重点教材
(编号: 2017-2-056)

计算思维之 程序设计

沈 军 · 编著



东南大学出版社
SOUTHEAST UNIVERSITY PRESS



“十三五”江苏省高等学校重点教材(编号:2017-2-056)

计算思维 之 程序设计

沈 军 编著



 东南大学出版社
SOUTHEAST UNIVERSITY PRESS

· 南京 ·

内容提要

本书以程序设计方法为主线,介绍其构建原理、基本应用和蕴含的计算思维。全书分为基础、方法和应用三篇,其中,基础篇包括第 1-3 章,主要解析程序的两个基因——数据组织和数据处理,为方法的构建奠定基础;方法篇包括第 4-9 章,主要解析目前主流的两种程序设计基本方法的构建原理及各种支持机制;应用篇包括第 10-18 章,主要解析两种程序设计基本方法的具体应用。应用篇又细分为面向 C++ 的内向应用(第 10-14 章)和面向实际应用的外向应用(第 15-18 章)两个单元,对于外向应用进一步细化为基于演绎思维(第 15-16 章)和基于归纳思维(第 17-18 章)的两个层次,实现知识和方法的学习到实际应用的思维平滑过渡和迁移。

本书可以作为普通高校程序设计课程的教材,也适合已有一定基础、需要进一步系统化提高思维能力的广大程序设计人员自学。本书可以作为各个层次程序设计课程相关教师的教学指导用书。

图书在版编目(CIP)数据

计算思维之程序设计 / 沈军编著. — 南京:东南大学出版社,2018.8

ISBN 978-7-5641-7972-4

I. ①计… II. ①沈… III. ①C++语言-程序设计-高等学校-教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 203751 号

计算思维之程序设计

出版发行	东南大学出版社
社 址	南京市玄武区四牌楼 2 号(210096)
网 址	http://www.seupress.com
出 版 人	江建中
责任编辑	张 煦
经 销	全国各地新华书店
印 刷	南京玉河印刷厂
开 本	787mm×1092mm 1/16
印 张	27
字 数	607 千字
版 次	2018 年 8 月第 1 版
印 次	2018 年 8 月第 1 次印刷
书 号	ISBN 978-7-5641-7972-4
定 价	78.00 元

东大版图书若有印装质量问题,请直接与营销部联系。电话(传真):025-83791830

前 言

随着泛计算社会的到来,程序及程序设计成为一种生活和工作习惯。然而,鉴于计算机学科的多维特征,程序设计具有显著的多维形态,主要表现为其同时涉及计算机语言、计算机环境、计算机应用和程序构造基本模型及方法几个方面。因此,如何协调好各个维度,成为程序设计课程教学必须考虑的核心。另一方面,计算机学科的多维特征形成了其自身独有的计算思维,程序设计显然也满足计算思维特征,或者说程序设计是计算思维的典型应用。因此,基于计算思维并同步多个维度来构建程序设计课程的知识体系并由此驱动教学设计和应用,有利于从根本上培养程序设计的能力。

多年来,作者一直从事计算思维原理及应用的研究与实践,并将其与认知科学相关理论融合,取得较好的教学效果。在此基础上,构建了本书的体系结构,尝试新的教学思路和策略。本书的主要特色如下:1) 将计算思维原理隐式地贯穿在全书体系、各个章节、各个维度、各个程序设计知识的解析之中;2) 正确梳理了计算机语言、计算机环境、计算机应用和程序构造基本模型及方法几个方面的同步关系,并以计算思维原理耦合各个方面;3) 以程序构造基本方法及其演化为主线,将计算机语言机制巧妙地作为基本方法的支持而展开;4) 从演化角度,系统地解析了两种主流程序设计方法的内在思维联系;5) 对 C++语言机制及其对程序设计基本方法的支持与拓展给予详细的解析。

本书的出版,期望引起各界同行对普通高校程序设计课程教学思路 and 改革方向的思考与讨论,并且,面向泛计算社会,为从根本上提高程序设计应用能力,构筑必要的思维基础。

本书的出版,我的学生们付出了大量的工作。孙竹山、周晓和崔效伟参与了案例设计并调试了大部分案例程序,徐旻昱、沈轩、张勇、黄亚彭、丁德林、袁明明、万逸、任杰完成了全书图表整理和制作工作。在此,对他(她)们付出的辛勤工作表示衷心感谢!

本书的出版得到江苏省教育厅(江苏省“十三五”高等学校重点教材立项)、东南大学(中央高校建设一流大学/学科经费)等组织机构和单位的支持,在此表示衷心致谢。

本书的出版得到了东南大学出版社的支持,在此表示衷心感谢! 特别感谢张煦编辑的大力支持。

本书中的观点都是基于作者个人的认识、理解和感悟,难免存在错误和不妥之处,希望读者来信批评与指正。作者恳切盼望各位同仁来信切磋,作者的 E-mail 是 junshen@seu.edu.cn。

作 者

2018 年春于古都金陵

目

录

第一篇 基础

第1章 走进程序设计	3
1.1 程序与程序设计	3
1.1.1 程序	3
1.1.2 程序设计	4
1.2 程序设计四要素	4
1.2.1 方法	4
1.2.2 语言	5
1.2.3 环境	5
1.2.4 应用	6
1.2.5 程序设计四要素的关系	6
1.3 程序设计的思维特征	7
1.4 程序设计的基本学习策略及其 C++映射	7
1.5 本章小结	8
第2章 数据组织基础	10
2.1 数据类型	10
2.2 单个数据的组织	12
2.3 堆叠、关联与绑定	13
2.3.1 堆叠	13
2.3.2 关联	15
2.3.3 绑定	16
2.4 数据组织中的计算思维	17
2.5 常用基本数据组织形态	17
2.5.1 线性数据组织	17
2.5.2 层次数据组织	19
2.5.3 网状数据组织	21



2.5.4 C++语言常用基本数据组织方法及其描述	23
2.6 本章小结	28
第3章 数据处理基础	30
3.1 表达式	30
3.2 基本语句	31
3.2.1 基本处理语句	31
3.2.2 流程控制语句	33
3.3 堆叠与嵌套	34
3.4 数据处理中的计算思维	36
3.4.1 表达式嵌套	36
3.4.2 语句堆叠与嵌套	36
3.5 常用基本数据处理方法	36
3.6 本章小结	52

第二篇 方法

第4章 程序设计方法概述	57
4.1 方法与模型	57
4.2 程序构造方法的认识视图	57
4.3 程序构造方法建立的基本原理	58
4.4 两种主流的程序设计方法及其思维联系	59
4.4.1 面向功能方法概述	59
4.4.2 面向对象方法概述	60
4.4.3 两者的思维联系	61
4.5 程序构造方法的进一步认识	61
4.5.1 数据组织与数据处理的关系	61
4.5.2 程序构造方法的计算思维特征	63
4.5.3 C++语言对数据组织和数据处理的统一	63
4.6 本章小结	64
第5章 面向功能方法:函数	65
5.1 基本功能模块的构造机制及其描述	65
5.1.1 基本功能模块构造机制的抽象	65
5.1.2 C++语言对基本功能模块构造机制的支持及描述	66
5.2 常用基本数据处理方法的C++语言函数定义及解析	66
5.3 基本功能模块构造机制对数据组织方法应用的具体规则	68
5.3.1 基本功能模块构造机制中数据组织方法的应用规则	68



5.3.2 C++语言对基本功能模块中数据组织方法的拓展	69
5.4 C++语言对基本功能模块表达模型的进一步拓展	71
5.4.1 空返回值	71
5.4.2 空函数与无参函数	71
5.4.3 默认参数与可变参数	72
5.4.4 多重返回	72
5.4.5 系统库函数	73
5.5 本章小结	74
第6章 面向功能方法:函数关系	76
6.1 函数之间的耦合	76
6.1.1 函数之间交互关系的实现机制	76
6.1.2 函数的调用	77
6.1.3 函数的返回	77
6.1.4 C++语言中函数调用与返回的描述	79
6.2 函数的一种特殊耦合关系——递归	80
6.3 C++语言对函数耦合关系的拓展	82
6.3.1 表达式参数与表达式返回	82
6.3.2 函数重载	83
6.3.3 函数模板	85
6.3.4 高阶函数	88
6.4 C++语言中面向功能方法的程序基本结构	88
6.4.1 多文件结构概述	88
6.4.2 编译预处理	89
6.5 深入认识面向功能方法	92
6.5.1 模型化方法的建立	93
6.5.2 存在的弊端	93
6.5.3 多维思维特征	93
6.6 本章小结	94
第7章 面向对象方法:对象	102
7.1 概述	102
7.1.1 数据类型的重要性	102
7.1.2 运用面向功能方法拓展新的数据类型——抽象数据类型(对象)	102
7.1.3 C++语言对抽象数据类型的支持机制及其拓展	104
7.2 数据类型拓展后带来的问题及其处理	107
7.2.1 实例的构造和销毁	107

7.2.2	默认构造函数与复制构造函数	111
7.2.3	初始化参数列表	112
7.2.4	同一种对象多个实例之间的数据共享	115
7.2.5	实例访问与 this 指针	116
7.2.6	如何实现新类型的基本运算	117
7.2.7	如何解决类型不一致问题	124
7.2.8	如何实现新类型的输入和输出(可流类)	126
7.3	让对象生活在面向功能方法时代	130
7.3.1	基于抽象数据类型的数据组织方法	130
7.3.2	基于抽象数据类型的数据处理方法	135
7.3.3	支持抽象数据类型的面向功能方法的程序构造	140
7.4	深入认识数据类型	142
7.5	本章小结	143
第 8 章	面向对象方法:对象关系	146
8.1	对象关系概述	146
8.2	对象嵌套关系	147
8.2.1	对象嵌套时的实例构造与析构	147
8.2.2	宿主对象的使用	149
8.3	同族对象之间的关系	152
8.3.1	继承(或普通遗传)	152
8.3.2	多态(或遗传变异)	169
8.3.3	C++语言对继承和多态的拓展	173
8.3.4	对同族关系的进一步认识	176
8.4	抽象数据类型的进一步抽象与拓展	177
8.4.1	类模板	177
8.4.2	类模板特化	177
8.4.3	类模板与继承	179
8.4.4	泛型编程	179
8.5	C++语言中面向对象方法的程序基本结构	179
8.6	深入认识面向对象的方法	180
8.7	本章小结	181
第 9 章	共享、安全与性能	191
9.1	共享	191
9.1.1	函数内的共享	191
9.1.2	抽象数据类型内的共享	192
9.1.3	单文件程序内的共享	192

9.1.4	多文件程序内的共享	193
9.1.5	共享带来的问题	193
9.2	安全	193
9.2.1	引用	193
9.2.2	const 限定	194
9.2.3	异常控制	196
9.2.4	动态类型检查	196
9.3	性能	197
9.3.1	inline 函数	197
9.3.2	类的友元	197
9.3.3	类数据成员的 mutable 限定	198
9.3.4	临时变量	199
9.3.5	初始化参数列表	201
9.4	对共享、安全与性能的综合认识	202
9.5	本章小结	202

第三篇 应用

第 10 章	程序设计应用概述	207
10.1	什么是应用	207
10.2	应用的思维特征及其 C++ 映射	208
10.2.1	应用的思维特征	208
10.2.2	应用思维特征的 C++ 映射	208
10.3	学习应用的基本策略	209
10.4	本章小结	209
第 11 章	I/O 流	211
11.1	什么是 I/O 流	211
11.2	C++ I/O 流机制的实现概述	214
11.3	C++ 标准 I/O 流	216
11.3.1	标准输入流类型及其使用	217
11.3.2	标准输出流类型及其使用	219
11.3.3	对标准 I/O 流的深入认识	219
11.4	标准 I/O 流的状态管理和格式控制	220
11.5	文件 I/O 流	224
11.5.1	操作系统文件处理的一般原理	224
11.5.2	如何创建文件输入流对象实例	225

11.5.3	如何创建文件输出流对象实例	225
11.5.4	如何关闭文件 I/O 流	226
11.5.5	如何访问文件	226
11.5.6	文件流应用示例及解析	229
11.6	对 I/O 流的深入认识	234
11.6.1	I/O 流概念的认知层次	234
11.6.2	I/O 流概念的通用性	234
11.6.3	I/O 流机制的安全性	234
11.6.4	I/O 流概念的递归性	235
11.6.5	I/O 流的模板化(模板化 I/O 流)	235
11.6.6	标准 I/O 流体系的可扩展性	235
11.7	本章小结	235
第 12 章	字符串	238
12.1	字符串的传统处理方法	238
12.1.1	通过字符数组处理符号串	238
12.1.2	通过字符型指针处理符号串	239
12.1.3	传统处理方法存在的问题	241
12.2	自己构建字符串数据类型 String	241
12.3	C++标准库的字符串数据类型 string	247
12.4	字符串流	249
12.5	进一步认识字符串	252
12.6	本章小结	254
第 13 章	异常	255
13.1	什么是异常	255
13.2	如何处理异常	255
13.3	C++异常处理机制	256
13.3.1	异常处理框架及其描述	256
13.3.2	C++异常机制使用的基本规则	259
13.3.3	异常处理时的对象实例析构	261
13.4	深入认识异常	262
13.5	本章小结	262
第 14 章	标准模板库及其应用	265
14.1	泛型程序设计及其思维本质	265
14.1.1	泛型程序设计	265
14.1.2	泛型程序设计的思维本质	265
14.2	C++标准模板库 STL	266

14.2.1	STL 的基本原理及其体系结构	266
14.2.2	对类型通用化的处理	267
14.2.3	实例解析	270
14.2.4	STL 的基本应用	273
14.2.5	深入认识 STL	277
14.3	本章小结	277
第 15 章	基于框架的程序设计	279
15.1	基于框架的程序设计基本思维	279
15.2	MFC 框架的基本原理	280
15.2.1	Windows 操作系统定义的基本程序模型	280
15.2.2	MFC 对 Windows 基本程序模型的包装	283
15.2.3	MFC 与 Visual C++ 的关系	301
15.3	MFC 框架的基本应用	302
15.3.1	MFC 框架编程概述	302
15.3.2	MFC 框架编程的基本步骤	303
15.3.3	应用示例及解析	303
15.3.4	从 Visual C++ 到 Visual Studio	325
15.4	深入认识基于框架的程序设计	325
15.4.1	框架式程序设计方法的必要性	325
15.4.2	MFC 框架的高级应用	326
15.5	本章小结	332
第 16 章	应用模式及其建构	335
16.1	基本应用模式及其建构	335
16.1.1	基本惯用法及其建构	335
16.1.2	数据组织结构中的基本应用模式及其建构	338
16.1.3	数据处理算法中的基本应用模式及其建构	345
16.2	设计模式及其建构	355
16.2.1	MVC 模式及其建构	355
16.2.2	工厂方法模式及其建构	357
16.2.3	适配器模式及其建构	361
16.3	模式及其建构应用中的计算思维	363
16.4	本章小结	363
第 17 章	广谱隐式应用	366
17.1	什么是广谱隐式应用	366
17.2	广谱隐式应用的核心与关键	366
17.3	应用示例	367

17.4	深入认识广谱隐式应用	373
17.5	本章小结	374
第 18 章	应用之道	376
18.1	应用的进化之道	376
18.2	应用的思维之道	377
18.3	应用之大道	377
18.4	本章小结	377
附录 A	ASCII 字符集	378
附录 B	C++语言定义的运算符	379
附录 C	标准库 <code>cstring</code> 的函数定义 (基于面向功能方法的字符串处理函数)	381
附录 D	标准库 <code>string</code> 类的定义	383
附录 E	典型风格 MFC 程序描述	388
附录 F	MFC 程序去框架特征的回归	396
附录 G	C++开发环境简介	413
附录 H	程序设计之计算思维准则	419
参考文献		420

第一篇
基础

第 1 章 走进程序设计

本章主要解析:什么是程序,什么是程序设计,程序和程序设计的本质区别;程序设计的四要素及其关系;程序设计的思维特征;程序设计的基本学习策略及其 C++ 映射。

本章重点:程序与程序设计的本质区别;程序设计的四要素及其关系;程序设计的思维特征。

人类发明工具的目的是为了延伸和拓展自身功能器官的能力,例如:汽车、飞机等延伸和拓展了人类双腿的能力,望远镜延伸和拓展了人类眼睛的能力,……计算机则是延伸和拓展了人类自身最高级器官——大脑的能力。

计算机工具的特殊结构及其带来的工作原理,决定了程序设计的必要性。为了有效地使用计算机工具,我们必须学会程序设计。另一方面,随着社会发展进程进入信息化时代,由 0 和 1 组成的信息如空气分子包围着我们,我们的生活、学习和工作等都需要与信息打交道!因为信息处理的核心工具就是计算机,因此,为了适应信息社会的生存需要,我们也必须懂得程序设计带来的基本思维方式和行为习惯。

1.1 程序与程序设计

1.1.1 程序

所谓“程序”(Program),广义地讲是指某件事情进行的过程安排,它取“程”字的本义“(一段)过程”(例如:一段时间、一段距离或一件事情的开始到结束等等)和“序”字的本义“次序、顺序”合并而成。引申而言,“程”表示我们需要处理的问题,该问题的处理一般需要涉及多个阶段或步骤,为了高效地处理该问题,需要给出这些阶段或步骤的顺序安排,也就是它们的“序”。可见,“程序”作为一个名词,它就是定义了某件事情进行过程各个步骤的先后次序既定安排。狭义地讲,在计算机世界中,“程序”是特指为了使用计算机工具处理某种问题,人们用计算机语言给出符合计算机环境特点的某种问题处理步骤的既定安排、定义或描述。因此,从认识论的角度,计算机世界中的“程序”是通用“程序”概念在计算机世界中的具体应用,是通用“程序”概念对计算机世界的直接投影。两种“程序”概念构成普遍性与特殊性的辩证关系。图 1-1



所示给出了这种关系的解析。

也就是说,本书讨论的“程序”,实际上是对通用“程序”概念施加了一个限制条件和约束,即计算机工具。正是这个限制条件和约束,缩小了“程序”概念的范畴(或定义域),扩大了其内涵,从而建立起一个崭新的信息世界,使人类文明经历农耕文明、工业文明,走向信息文明。

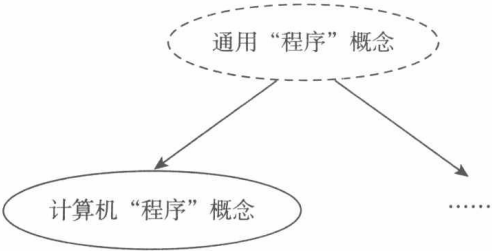


图 1-1 “程序”概念的基本解析

1.1.2 程序设计

根据“程序”概念的解析,显而易见,某件事情进行过程的步骤安排存在多种方案,每种方案都有利有弊,人们需要根据给定的约束和限制条件来进行权衡,以便为某件事情的进行及完成最终确定一种最佳的方案,这个权衡的过程就是“程序设计”(Programming,在通用世界中,更通俗地称为“安排”“计划”等)。

因此,在计算机世界中,人们为了使用计算机工具处理某种问题,用计算机语言给出符合计算机特点的某种问题处理步骤的安排、定义或描述的过程,称为“程序设计”(如不做特殊说明,本书以后的“程序”和“程序设计”都是指计算机世界中的概念)。

由此可见,“程序”是指面向计算机的工作指令执行顺序的一种安排,是一种结果,是静态的;而“程序设计”则是指“程序”的形成和产生过程,是动态的。正是“程序设计”的这种动态属性,给予了人们思维能力表现的空间和舞台,呈现了由不同的人类思维所带来的色彩缤纷的信息世界的美丽画卷及其魅力!进而,也形成了信息世界独有的思维——计算思维(Computational Thinking)。因此,“程序设计”的内涵要比“程序”的内涵更丰富。一方面,“程序设计”涉及较多的内容和方面;另一方面,“程序设计”过程呈现了人类智慧的火花。

1.2 程序设计四要素

依据“程序设计”概念的解析,程序设计涉及计算机语言(简称语言,用以描述程序)、计算机环境(简称环境,用以给出程序构造和运行的基础)、计算机应用(简称应用,用以给出符合计算机特点的问题处理方法及步骤安排)和程序构造基本方法(简称方法,用以给出程序基本结构的定义,包括基本元素及其交互关系)四个方面,称为程序设计四要素。其中,方法是程序设计的内因,语言、环境和应用都是程序设计的外因。

1.2.1 方法

程序设计方法的认识分为多个逻辑层次,一般包括面向数据组织和数据处理的基础方法、面向程序模型及结构定义的基本方法和面向应用的应用方法。基本方法是对基础方法的应用,应用方法是对基本方法的应用。本质上,基础方法和基本方法属于原理性层次,应用方法则属于原理的具体应用层次。

对数据组织和数据处理两个方面基础方法及其关系的不同认识,构成不同的程序设

计基本方法。伴随着人类自身对程序构造问题认识的不断深入,程序设计基本方法也不断地演化,经历面向功能方法、面向对象方法、面向组件方法和面向服务方法的发展,每一种基本方法都给出其程序的基本要素及其交互关系的定义,即程序基本结构模型的定义。各种应用方法都是建立在某种程序结构模型基础上。

本书主要解析原理性的基础方法和基本方法,适当涉及一些通用的、基本的应用方法(本书将应用方法归入应用要素)。

作为程序设计的内因,基本方法的演化带动了语言 and 环境的同步演化,最终又驱动了应用的发展。

1.2.2 语言

作为一种描述工具,任何语言都具备如图 1-2a 所示的基本体系,它给出了语言的各种机制及其关系。显然,作为一种描述程序的特殊语言,程序设计语言也基本满足该体系。然而,由于其作用的特殊范畴,程序设计语言又具有其自身的特殊性,主要表现为:1) 每个层次的机制数量较少,而且形式化程度高;2) 各种机制的具体表达方式具有计算机器的明显痕迹;3) 各种机制都是围绕对基础方法和基本方法的支持而展开。图 1-2b 所示给出了程序设计语言的基本体系及相应机制。

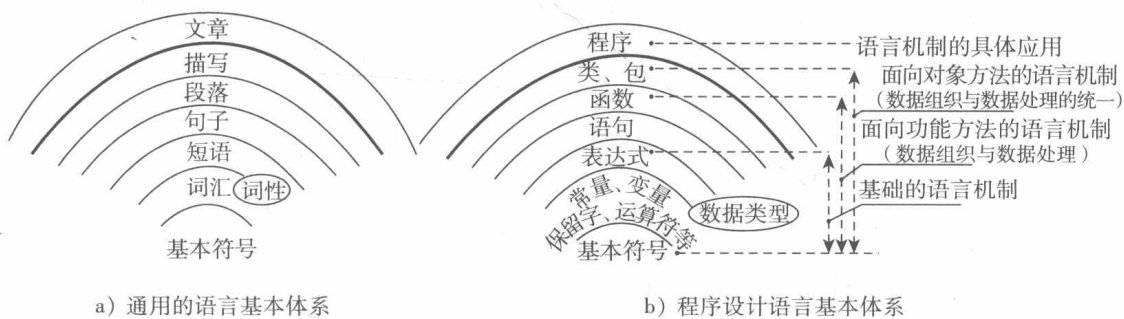


图 1-2 “语言”概念的基本解析

为了方便人类使用,程序设计语言经历了面向机器的机器语言到面向人类的高级语言的发展,其特征也与低层的机器越来越远,与人越来越近。然而,高级语言最终都必须转换为机器语言才能被计算机识别。图 1-3 所示给出了转换的原理,其中,汇编程序、解释程序和编译程序都是预先构造好的特殊程序,它们是程序开发环境的一部分。

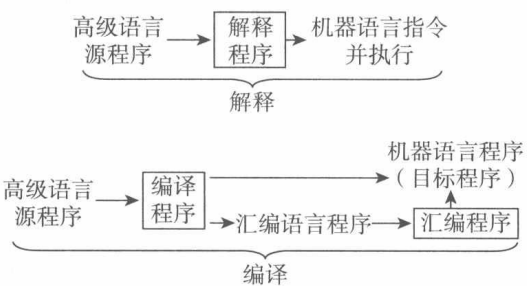


图 1-3 程序设计语言的转换

1.2.3 环境

任何语言都有其赖以作用的环境,不同的环境对语言有不同的要求,同一种语言在不同的环境中,也会有不同的调整,以适应环境的需要。程序设计语言赖以作用的环境可以分为程序运行支撑环境(或开发平台、运行平台)和程序开发(或写作、构造)环境两