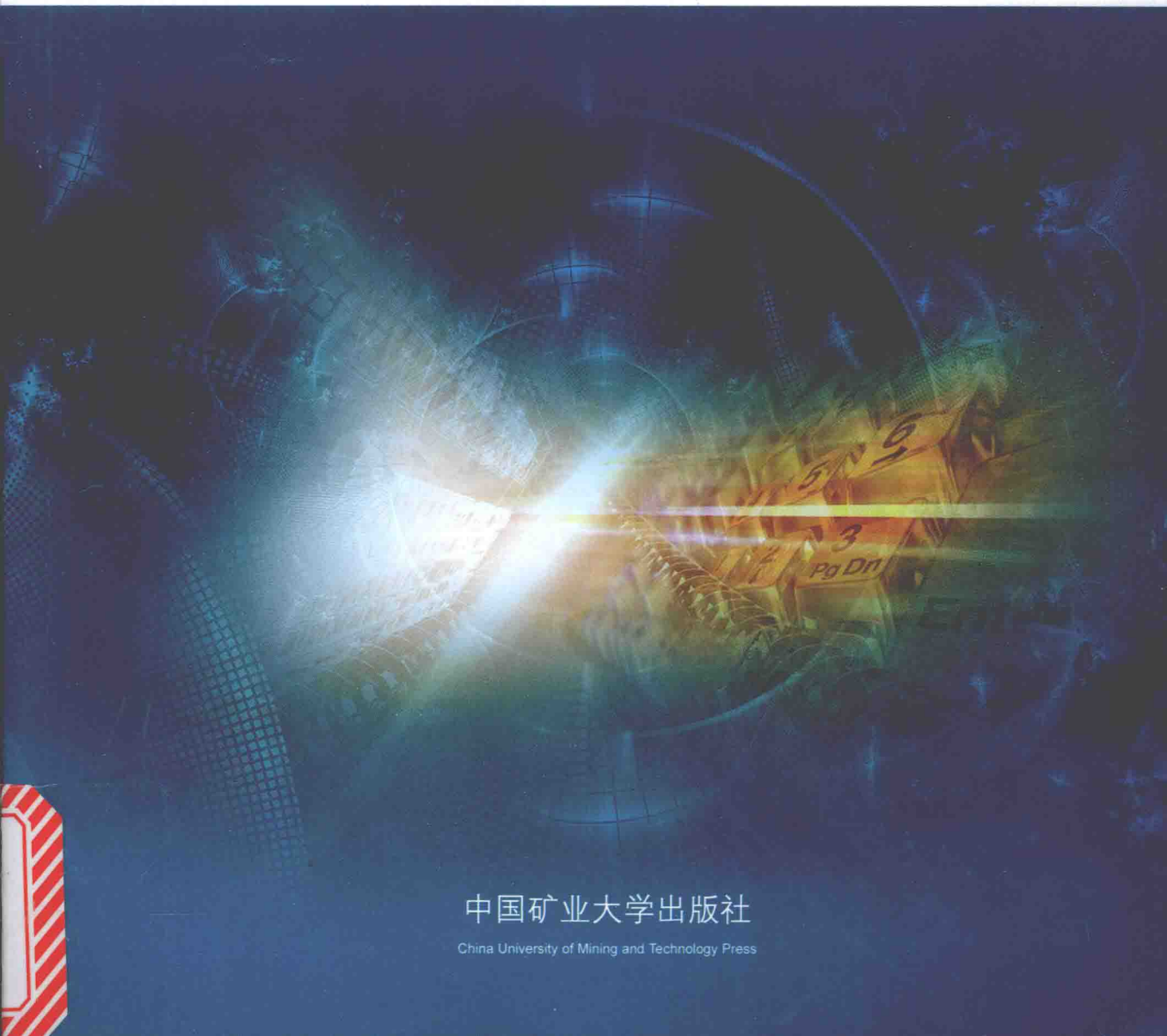


C++语言

探索发现学习教程

雷小锋 毛善君 张海荣 著



中国矿业大学出版社

China University of Mining and Technology Press

C++语言探索发现学习教程

雷小锋 毛善君 张海荣 著

中国矿业大学出版社

图书在版编目(CIP)数据

C++语言探索发现学习教程 / 雷小锋,毛善君,张海荣著.

—徐州:中国矿业大学出版社,2015.2

ISBN 978-7-5646-2642-6

I. ①C… II. ①雷…②毛…③张… III. ①C语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 042904 号

书 名 C++语言探索发现学习教程
著 者 雷小锋 毛善君 张海荣
责任编辑 周 丽 陈 慧
出版发行 中国矿业大学出版社有限责任公司
(江苏省徐州市解放南路 邮编 221008)
营销热线 (0516)83885307 83884995
出版服务 (0516)83885767 83884920
网 址 <http://www.cumtp.com> E-mail: cumtpvip@cumtp.com
印 刷 徐州中矿大印发科技有限公司
开 本 787×1092 1/16 印张 23.5 字数 586 千字
版次印次 2015 年 2 月第 1 版 2015 年 2 月第 1 次印刷
定 价 39.80 元

(图书出现印装质量问题,本社负责调换)

前 言

常听人说 C++ 语言之难, 竟如那蜀道天险般不可逾越。然而, 无论 C++ 语言如何博大精深、包罗万象, 它都只是一门语言, 一门我们用来与计算机进行沟通交流的语言, 沟通交流的目的是把我们的问题求解思路“告诉”计算机, 然后由计算机根据问题求解思路进行运算和处理, 得到问题求解结果, 最终实现问题的求解。

所谓“告诉”当然不是对不解风情的计算机倾吐衷肠, 而是用计算机能够理解的语言把我们的问题求解思路描述表达出来, 输入计算机。这里, 计算机能够理解的语言就是**计算机语言**; 而我们的问题求解思路, 用计算机语言描述出来, 就是程序(program), 描述的过程就是编写程序, 简称**编程**(programming)。接下来, 就由计算机根据程序的指示按部就班地工作, 也就是**执行程序**, 获得问题求解的结果。

可见, 程序对人而言, 就是用计算机语言描述的我们的解题思路; 对计算机而言, 就是可以被计算机理解和执行的一组指示或者命令。因此, 计算机语言, 作为编写程序的工具, 很自然就必须肩负两方面的职责: 其一一是为我们描述问题求解思路提供方便自然的语法工具(通俗而言, 就是我们想用计算机干什么, 语言就应该帮助我们描述要干什么的事情); 其二二是为计算机理解和执行程序提供足够的语义信息(关于我们要干什么的描述要能够使计算机理解和执行, 就必须提供计算机理解和执行所需的信息)。更进一步, 我们在描述问题求解思路时有什么样的需求, 计算机在理解和执行程序时有什么样的需求, 则计算机语言就应该提供满足这些需求的语法工具来为之服务。

语言服务于人和计算机, 人和计算机的需求和期望决定了语言。语言千变万化, 人机需求却持久稳定; 语言难以掌握, 人机需求却易于理解。因此, 学习 C++ 语言的首要任务不是一头扎进形式纷繁的语法中机械重复地记忆训练, 而应该首先撩开语言千变万化的面纱, 认识导致语言存在的稳定持久的人和计算机的需求, 包括我们描述问题求解思路的需求以及计算机理解执行程序的需求。

人和计算机有这样的需求, C++ 语言(包括种类繁多的其他计算机语言)就提供了那样的语法工具来满足这些需求。因此, 当我们面对一种新的语法工具茫然不知所措时, 不妨深吸一口气在心里默念“需求, 需求, 是我们的某种需求驱动了这种语法工具的产生”。然后, 运用我们的常识和理性思维, 从人机需求的角度分析推理 C++ 语言诸多语法工具存在的现实理由, 理解和把握语法形式背后的语义内涵。

倘能如此, 则我们的 C++ 语言学习即可达到一种非常理想的学习图式。学习者以思维

探索发现的方式,将人和计算机的需求与 C++语言知识映射串联起来,从而构建契合自我思维逻辑的 C++语言的整体知识结构。任何时候,我们都可以从已知的人机需求出发,借助常识和理性思维的指引,自然顺畅地推理出 C++语言所有的知识节点,遍历 C++语言的每一个角落。蜀道艰险,此时却已是通向未来的坦途。

本书组织的逻辑结构

秉承语言服务于人机需求的思想,本书力图在人机需求的框架下通过理性思维的引导对 C++语言的内容进行推理、组织以及编排。因此,本书第一章至关重要,梳理阐述人机需求,即人描述问题求解思路的需求和计算机理解执行程序的需求,然后从人机需求出发推理构建 C++语言的知识结构,进而奠定本书的整体框架。

在本章中,我们通过一个现实生活中的示例,分析人类是如何求解问题的,总结人类求解问题的三种思维模式(过程化、面向对象和泛型思维),说明在这三种思维模式下的问题求解思路有哪些步骤和环节,从而确定我们描述这三类问题求解思路的需求,当然 C++语言应该提供相应的语法工具和设施来满足这些需求。

进而,通过与工厂进行类比,说明计算机具有什么样的逻辑结构,以及在该逻辑结构下计算机是如何理解和执行程序的,即计算机的逻辑结构和工作原理。计算机的逻辑结构和工作原理,决定了计算机理解和执行程序需要哪些必要的信息,C++语言的语法工具必须能够为计算机理解和执行程序提供这些必要信息。

本章最后,我们从人和计算机的需求出发,分析推理从人机需求到语言知识的因果链路,构建 C++语言的知识结构,厘清本书的总体组织框架。

“工欲善其事,必先利其器。”因此,在接下来的第二章,我们直接切入实践环节,通过编写程序求解素数判定问题,在实践中掌握 Visual C++编程工具的使用方法,体会程序的编译链接调试等环节,并且认识一下 C++程序。

此后,从第三章开始,我们分别针对过程化、面向对象和泛型三种思维模式,深入阐述其描述问题求解思路有哪些需求,C++语言又为之提供了哪些语法工具。

关于学习的一点思考

历史是由成功者书写和打扮的。倘若追溯前尘,这些所谓的成功者也是从菜鸟过来的,也经过无穷无尽的痛苦与折磨;或许是一只翩跹飞舞的蝴蝶,拨动了沉睡已久的那一根心弦,总之有一天,他们法眼圆睁,弹铗而起,一切了然于胸。

于是,世界上出现了诸多权威和先知,他们精心炮制着各种结构精严、形式完美的 C++经书宝典,告诉我们应该如此这般,这般如此。至于为何需要如此这般,以及追求这般如此

的思维蝶化过程,在法相庄严的字里行间难觅踪影。

然而,真正的知识源于我们的内心,源于学习者自我的思维和探索,而不是葵花宝典或者世外高人的传授。在我们看来,一本合适的 C++ 语言入门教材,不应是自己养育生命的产妇,而应该是引导产妇养育生命的助产士;不应着眼于把语言知识传授灌输给学习者,而应该立足于启发和引导学习者运用常识和理性思维,探索发现 C++ 语言知识存在的因果需求,推理分析其语法工具的形式和内涵,从整体上实现 C++ 语言知识结构的自我建构。通往 C++ 语言的道路,是由我们自己的思维来探索发现的。

当然,再高明的助产士也不能代替产妇养育生命,任何一本 C++ 语言教材都不可能逾越学习的规律,代替学习者的勤奋和汗水。C++ 语言的宏大体系,决定了我们必然要经历一个长期艰苦的修炼过程。这其中,从人机需求出发推理构建 C++ 语言知识结构,领悟语法工具的形式和内涵,只是万里长征的第一步。要熟练运用 C++ 语言编写程序,实现求解问题的终极目标,还需要经过大量实践练习的磨砺。

西方谚云:“Practice makes perfect”。

卖油翁说:“无他,唯手熟尔”。

诚哉斯言!

致谢

特别的感谢要送给和我朝夕相处的同学们,他们无限的求知欲望和坚韧的求知精神,时时刻刻鼓励着我,鞭策着我,使我一路坚持下来。也要感谢我的小闹闹没有总是爬上我的膝头,认真地抠下电脑的左 shift 键,感谢妻子和父母能够忍受我的坏脾气,默默地付出。还要感谢所有支持、鼓励、帮助、批评过我的朋友们。

最后,所谓人类一思考,上帝就发笑。笔者天资驽钝,期望大方之家发笑之余不吝赐教。

版权声明

作者愿意公开本书的电子文档。版权声明如下:

- (1) 读者可以任意拷贝、修改本书的内容,但不可以篡改作者及所属单位。
- (2) 未经作者许可,不得出版或大量印发本书。

雷小锋

2014 年 8 月

目 录

前言	1
第 1 章 从人机需求到 C++ 语言	1
1.1 人类如何求解问题?	1
1.1.1 自己动手:过程化思维	1
1.1.2 求助他人:面向对象思维	2
1.1.3 发明炒菜机:泛型思维	3
1.2 计算机如何执行程序?	4
1.2.1 计算机的工作原理	4
1.2.2 存储器的逻辑结构	5
1.3 从人机需求到计算机语言	9
1.3.1 计算机母语:机器语言	9
1.3.2 引入助记符:汇编语言	10
1.3.3 满足人的需求:高级语言	11
1.4 从人机需求到 C++ 语言	11
1.4.1 C++ 过程化思维篇	11
1.4.2 C++ 面向对象思维篇	12
1.4.3 C++ 语言泛型思维篇	14
1.5 章节小结	14
习题	15
第 2 章 欲善其事 先利其器	16
2.1 编写素数判定程序	16
2.1.1 新建工程和 C++ 源文件	16
2.1.2 C++ 程序的一般结构	18
2.1.3 编写素数判定程序	19
2.1.4 程序组织和编译链接	27
2.2 C++ 程序概览	29
2.2.1 C++ 程序是字符序列	29
2.2.2 C++ 程序是单词序列	29
2.2.3 C++ 程序是语句序列	31
2.2.4 C++ 程序是函数集合	32
2.2.5 C++ 程序是一组文件	33

2.3 章节小结	34
习题	34

第一篇 过程化思维篇

第3章 简单数据的描述和处理	37
3.1 描述简单数据	37
3.1.1 描述数据的人机需求	37
3.1.2 约定数据规格:数据类型	39
3.1.3 描述字面数据:字面量	41
3.1.4 描述可变数据:变量	46
3.2 数据的基本运算处理	49
3.2.1 C++语言的表达式	50
3.2.2 C++运算符及其表达式	52
3.2.3 表达式的类型转换规则	61
3.2.4 运算符和表达式小结	63
3.3 控制数据处理流程	64
3.3.1 选择:“如果-就”	64
3.3.2 循环:“如果-就反复执行”	68
3.3.3 控制流程的跳转	71
3.4 数据输入输出处理	73
3.4.1 输入输出流的概念	73
3.4.2 键盘输入处理:cin	74
3.4.3 屏幕输出处理:cout	74
3.4.4 输入输出格式控制	76
3.4.5 文件输入输出处理	77
3.5 数据处理机器:函数	85
3.5.1 从数据处理机器到函数	85
3.5.2 制造数据处理机器:定义函数	86
3.5.3 使用数据处理机器:调用函数	87
3.5.4 函数调用的形参生成	91
3.5.5 函数嵌套和递归	93
3.5.6 函数内联(inline)	96
3.5.7 函数重载和默认参数值	97
3.6 章节小结	99
习题	100
第4章 结构化编程和程序组织	103
4.1 结构化编程思想实践	103

4.1.1	进制转换演示程序	103
4.1.2	函数就地声明的问题	106
4.1.3	头文件集中声明	108
4.2	程序的多文件组织结构	109
4.2.1	多文件组织程序	109
4.2.2	独立编译和多目标链接	110
4.3	作用域和生存期	111
4.3.1	作用域:程序实体存在的空间范围	111
4.3.2	生存期:程序实体存在的时间范围	114
4.3.3	作用域和生存期小结	116
4.3.4	多文件共享程序实体:链接性	116
4.4	避免全局冲突:名字空间	118
4.4.1	全局程序实体的冲突问题	118
4.4.2	名字空间	119
4.4.3	C++标准名字空间	122
4.5	章节小结	123
	习题	123
第5章	复合数据的描述和处理	125
5.1	描述不变的变量:常量	125
5.1.1	宏定义指令	125
5.1.2	常量	126
5.2	描述变量集合:数组变量	126
5.2.1	定义数组变量	127
5.2.2	访问数组元素	128
5.2.3	数组的数组:多维数组	129
5.3	存放地址的变量:指针变量	132
5.3.1	定义指针变量	133
5.3.2	指针访问和操作	134
5.3.3	无类型指针: void *	137
5.3.4	指针与动态变量	138
5.3.5	指针和动态数组	140
5.4	描述字符串:字符串变量	143
5.4.1	字符数组实现字符串	144
5.4.2	字符指针实现字符串	144
5.4.3	操作字符串	145
5.5	变量的别名:引用变量	145
5.5.1	定义引用	145
5.5.2	引用作为函数形参	146

5.6 嵌套复合的数据类型	147
5.6.1 常量数组与枚举	148
5.6.2 指针数组和数组指针	150
5.6.3 常量指针和指针常量	151
5.6.4 多维数组和多级指针	152
5.6.5 常量、数组、指针的引用	153
5.6.6 简化嵌套复合:typedef	154
5.7 复合数据类型与函数	155
5.7.1 数组名用作函数参数	155
5.7.2 函数如何返回数组	157
5.7.3 指针与函数	162
5.7.4 引用与函数	166
5.8 结构化编程实践	167
5.8.1 求解约瑟夫问题	167
5.8.2 矩阵转置问题	169
5.9 章节小结	171
习题	171

第二篇 面向对象思维篇

第6章 描述和使用对象	177
6.1 描述对象的人机需求	177
6.1.1 人如何描述对象:定义类	177
6.1.2 计算机的需求:定义数据类型	178
6.2 如何定义类	178
6.2.1 定义类的语法	179
6.2.2 类是定制数据类型	182
6.3 产生对象	184
6.3.1 对象和变量	184
6.3.2 访问对象成员	184
6.4 基于对象求解日期问题	187
6.4.1 main 函数:对象之间的沟通协作过程	187
6.4.2 描述日期对象:CDate 类	188
6.4.3 成员函数 input	188
6.4.4 成员函数 diff	190
6.4.5 成员函数 add	192
6.5 类成员:约定对象的属性和行为	193
6.5.1 类数据成员	193
6.5.2 类成员函数	194

6.6	约定对象生死时刻的行为	195
6.6.1	构造函数:约定对象产生的初始化行为	195
6.6.2	析构函数:约定对象死亡的善后行为	201
6.6.3	构造和析构函数的调用时机	203
6.7	约定对象运算的行为:重载操作符	211
6.7.1	重载操作符	212
6.7.2	成员重载操作符	212
6.7.3	非成员重载操作符	212
6.7.4	友元:我是你的朋友	213
6.7.5	两种操作符重载方式比较	216
6.7.6	重载 CDate 类的操作符	217
6.7.7	重载赋值操作符	219
6.8	约定类成员的常量性	221
6.8.1	常量数据成员	221
6.8.2	常量成员函数	222
6.8.3	volatile 成员	225
6.9	约定类成员的静态性	225
6.9.1	静态数据成员	226
6.9.2	静态成员函数	228
6.9.3	const、static 成员小结	230
6.10	基于循环链表求解约瑟夫问题	230
6.10.1	约瑟夫问题求解思路	230
6.10.2	实现 CRing 的成员函数	232
6.11	基于对象求解矩阵问题	234
6.11.1	设计矩阵类 CMatrix	234
6.11.2	CMatrix 类的构造和析构函数	234
6.11.3	重载 CMatrix 类的操作符	237
6.11.4	访问矩阵元素:重载函数调用	239
6.11.5	重载矩阵转置和乘法操作符	240
6.11.6	重载矩阵输入和输出操作符	241
6.11.7	组织 CMatrix 类求解问题	241
6.12	章节小结	242
习题		244
第 7 章	对象的组合和继承	246
7.1	基于组合关系定制类:组合类	246
7.1.1	组合类对象的构造和析构	247
7.1.2	子对象的显式初始化	248
7.2	基于类属关系定制类:继承与派生	249

7.2.1	继承的含义和语法	249
7.2.2	选择继承方式	251
7.2.3	改造基类成员	254
7.2.4	重写构造和析构函数	257
7.2.5	派生类定制小结	259
7.3	利用组合/继承求解线性方程组	260
7.3.1	组合出线性方程组类	260
7.3.2	求解线性方程	261
7.4	派生类对象用作基类对象	264
7.4.1	派生类和基类的赋值兼容	264
7.4.2	针对基类引用/指针进行通用编程	268
7.5	对象行为的多态:重载和虚函数	270
7.5.1	通过重载实现编译期多态	270
7.5.2	通过虚函数实现运行期多态	271
7.5.3	虚函数的运行期绑定原理	274
7.6	运行期多态的编程实践	278
7.6.1	定义形状基类 CShape	279
7.6.2	定义 CShape 的派生类	279
7.6.3	定义通用函数 display	280
7.6.4	虚析构函数的用处	282
7.7	不产生对象的类:抽象类	284
7.7.1	抽象类是包含纯虚函数的类	285
7.7.2	应用纯虚函数求解定积分	288
7.7.3	只有纯虚函数的接口类	289
7.8	继承多个基类:多重继承和虚基类	291
7.8.1	多重继承的语法及其内涵	291
7.8.2	重复继承基类的问题	293
7.8.3	多重继承的适用场合	301
7.9	运行期类型识别 RTTI	304
7.9.1	操作符 typeid	304
7.9.2	操作符 dynamic_cast	305
7.10	异常处理机制	306
7.10.1	C++异常处理机制	306
7.10.2	抛出异常:throw 语句	307
7.10.3	捕获和处理异常:try-catch 语句	307
7.11	章节小结	308
习题		309

第三篇 泛型思维篇

第 8 章 泛化数据类型:模板与泛型编程	313
8.1 数据类型泛化:定义模板	313
8.1.1 定义函数模板	314
8.1.2 定义类模板	315
8.1.3 定义通用数组类	316
8.2 应用模板:模板实例化	318
8.2.1 模板实例化的方法	318
8.2.2 模板参数推演	318
8.2.3 模板特化	319
8.3 C++标准模板库	321
8.3.1 标准模板库能干什么	321
8.3.2 利用 STL 求解问题	322
8.3.3 容器(container)	323
8.3.4 迭代器(iterator)	331
8.3.5 仿函数(function object)	335
8.3.6 算法(algorithms)	337
8.4 泛型编程实践	344
8.4.1 文本单词反转问题	344
8.4.2 多序列最小和问题	345
8.4.3 农夫过河问题	350
8.5 章节小结	359
习题	360

第 1 章

从人机需求到 C++ 语言

程序即问题求解思路,是用计算机语言描述的我们人的问题求解思路。因此,编写程序的语言,如 C++ 语言,就必须满足人和计算机两方面的需求,包括人描述问题求解思路的需求和计算机理解执行程序的需求。语言形式可以千变万化,但是,其背后的人机需求则持久稳定。因此,本章的主题就是先厘清人机需求,然后,从人机需求出发推理构建 C++ 语言的整体知识结构,主要包括:

- (1) 人描述问题求解思路的需求:人类求解问题的三种思维模式。
- (2) 计算机理解执行程序的需求:计算机的逻辑结构和工作原理。
- (3) C++ 语言如何满足人机需求:C++ 语言的语法工具和组织结构。

现在,让我们通过一个现实生活中的事例,思考人类是如何求解问题的,然后从中总结出人类求解问题的三种思维模式(过程化、面向对象和泛型思维),并说明在这三种思维模式下描述问题求解思路有哪些步骤和环节。

1.1 人类如何求解问题?

假设我们腹中饥饿,想吃番茄炒鸡蛋。求解问题有多种策略:① 自己动手;② 求助他人;③ 针对更一般的炒菜问题发明一种通用炒菜机。这三种求解策略反映了人类求解问题的三种思维模式:过程化、面向对象以及泛型思维。

1.1.1 自己动手:过程化思维

显然,自己动手是最直接的一种策略。此时,我们需要知道番茄炒鸡蛋这道菜的制作过程,不外乎是准备好鸡蛋、番茄、调味品等材料,然后对这些材料进行洗、切、煎、炒等一系列加工处理,最后出锅装盘,于是美味的番茄炒鸡蛋就大功告成。

这就是过程化的问题求解思维。在这种思维下,人们认为世界是物质状态发展变化的过程。因此,求解问题就是对物质材料进行加工处理,促使其状态发生改变,最终到达预期的目标状态,即材料加工处理的结果。对于番茄炒蛋问题,我们就需要准备鸡蛋、番茄等材料,然后对这些材料进行加工处理,促使其状态发生改变,最终到达目标状态——番茄炒蛋。可见,在过程化思维下,问题求解思路就是准备材料、加工处理材料、最终获得处理结果的过

程。描述过程化的问题求解思路,需要描述:

- (1) 问题求解需要哪些材料,这些材料具备什么样的规格;
- (2) 对材料进行什么样的加工处理过程,可以促使其达到目标状态。

对于番茄炒鸡蛋问题,描述这些信息就得到了番茄炒鸡蛋的菜谱(如图 1-1 所示)。当然,对于计算机而言,它所加工处理的材料就不再是鸡蛋、番茄这些材料,而是数据。也就是说,这里需要描述数据以及数据的加工处理过程,包括:

- (1) **描述数据**:说明求解问题需要哪些数据,这些数据的规格是什么样的。
- (2) **描述数据处理过程**:说明数据如何输入、如何加工处理以及如何输出处理结果的过程,包括输入数据、处理数据和输出结果三个环节。

获得番茄炒蛋	
❖所需材料及规格:	
鸡蛋 3 个,番茄 2 个,花生油 100 克,水淀粉 10 克……	
❖制作过程:	
1. 准备鸡蛋、番茄等材料;	//准备材料
2. 鸡蛋打散,番茄洗净切块;	//加工处理材料
3. 炒鸡蛋;	//加工处理材料
4. 炒番茄,勾芡;	//加工处理材料
5. 鸡蛋番茄混合,加调味品;	//加工处理材料
6. 出锅装盘。	//得到处理结果

图 1-1 番茄炒蛋问题的过程化求解思路

在过程化思维下编程描述问题求解思路,就是过程化编程(Procedural Programming, PP)。C++语言支持过程化编程,因此,它应该提供相应的语法工具,用于方便自然地描述数据以及数据处理过程。

1.1.2 求助他人:面向对象思维

现代社会讲究分工合作,求助他人也是不错的选择。譬如隔壁村的石大叔号称“厨神”,手艺不错,价钱合理,服务又周到,一个电话过去,立马送菜上门。

这是一种面向对象的问题求解思维。在这种思维下,人们认为世界由一系列相互联系的对象组成,每种对象都具有自身的属性和行为能力。找到一组合适的对象,通过对象之间的沟通协作发挥各自的行为能力,就可以实现问题求解。譬如,番茄炒鸡蛋问题,电话联系厨神大叔,说明菜品和送餐地址,于是大叔炒勺一挥,番茄炒蛋制作完成;接下来找个外送工,安排他把番茄炒蛋送达我们的居处,大功告成。

因此,在面向对象思维下,问题求解思路就是一组合适的对象之间的沟通协作过程。在面向对象思维下描述问题求解思路,即**面向对象编程**(Object-Oriented Programming, OOP),需要描述问题求解所需的合适对象以及对象之间的沟通协作过程:

- (1) **描述对象**:说明求解问题需要哪些对象,这些对象应该具有什么样的规格要求(包括具有哪些属性和行为能力)。
- (2) **描述沟通协作过程**:说明如何产生具体的对象,以及这些对象之间如何沟通协作发挥各自的行为能力实现问题求解的过程。

图 1-2 就是番茄炒蛋的面向对象求解思路。C++ 语言支持面向对象编程,因此,它应该提供相应的语法工具,方便自然地描述对象及其沟通协作过程。

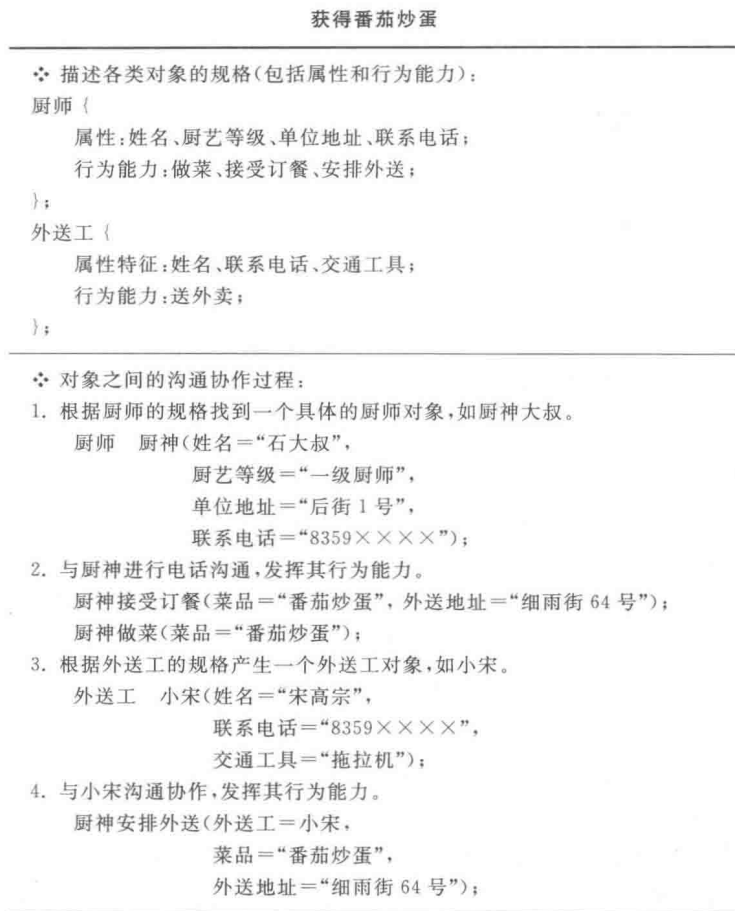


图 1-2 番茄炒蛋问题的面向对象求解思路

1.1.3 发明炒菜机:泛型思维

话说厨神大叔年复一年做着番茄炒鸡蛋,突然有所感悟转而学习机械电子和自动控制技术,随即发明了一种自动炒菜机。不论是番茄炒鸡蛋、土豆炒肉丝,还是××炒××,只要准备好材料放进去,机器很快就自动制作出美味的菜肴。

这就是泛型的问题求解思维。在这种思维下,人们认为通过将特殊问题中的某些因素设为可变参数,可以将特殊问题泛化为适用范围更广的一般问题,此后就可以借助于一般问题的求解思路来完成特殊问题的求解。譬如,将番茄炒鸡蛋问题中的番茄、鸡蛋等材料视为参变量,则问题转化为“AA 炒 BB”这样的一般问题。

对于参数泛化后的一般问题“AA 炒 BB”,求解同样可以采用自己动手的过程化思维,也可以采用求助他人的面向对象思维。总之,就是要在数据或对象是未知参变量的情况下描述出通用的过程化问题求解思路(过程泛化),或者通用的面向对象问题求解思路(面向对

象泛化)。有了参数泛化问题“AA 炒 BB”的通用求解思路,将参数 AA 设为番茄,参数 BB 设为鸡蛋,就得到番茄炒鸡蛋问题的求解思路;也可以将参数 AA 设定为土豆,BB 设定为肉丝,就得到土豆炒肉丝问题的求解思路。

通用炒菜机:获得 AA 炒 BB

✧所需材料及规格(AA 和 BB 是可变参数):

AA 3 个,BB 2 个,花生油 100 克,水淀粉 10 克……

✧制作过程:

- | | |
|---------------------|----------|
| 1. 准备 AA、准备 BB 等材料; | //准备材料 |
| 2. 炒 AA; | //加工处理材料 |
| 3. 炒 BB; | //加工处理材料 |
| 4. AA 和 BB 混合,加调味品; | //加工处理材料 |
| 5. 出锅装盘。 | //得到处理结果 |

图 1-3 AA 炒 BB 问题的过程化求解思路

在泛型思维下编写程序描述问题求解思路,即**泛型编程**(Generic Programming, GP),需要在数据或者对象参数泛化的情况下描述问题求解思路,包括:

(1) **泛化环节**:针对泛化的数据/对象类型,描述通用的问题求解思路。

(2) **应用环节**:使用通用问题求解思路,解决特定具体问题的求解过程。

C++语言支持泛型编程,它提供了函数模板和类模板的语法设施,用于支持过程化问题求解思路的泛化,以及面向对象问题求解思路的泛化。

至此,我们已经从现实生活中的实际问题出发,分析总结了人类在求解问题时常用的三种思维模式,以及在特定思维模式下描述问题求解思路的需求。其中:

(1) **过程化思维**:认为问题求解是对数据进行加工处理使之达到目标状态的过程。因此,过程化编程需要描述数据以及数据处理过程(输入、处理和输出)。

(2) **面向对象思维**:认为问题求解是通过对象之间的沟通协作来完成的。因此,面向对象编程需要描述合适的对象以及对象之间的沟通协作过程。

(3) **泛型思维**:认为通过数据和对象的参数化可以将特殊问题泛化为一般问题,特殊问题可以借助一般问题的求解思路得以解决。因此,泛型编程需要描述类型泛化的通用问题求解思路以及使用通用求解思路解决特殊问题的过程。

1.2 计算机如何执行程序?

无论采用哪种思维模式,我们所描述的问题求解思路,即程序,最终都要交由计算机理解和执行,C++语言必须满足计算机执行程序的需求。因此,C++语言必须有能力提供计算机如何执行程序,它在执行程序时需要哪些必要的信息等这些必要的信息。本节旨在说明计算机的逻辑结构及其执行程序的工作原理。

1.2.1 计算机的工作原理

我们都知道现实世界中的工厂,有指挥、生产、仓储、采购和销售五大部门。其中,采购