

高等学校计算机基础教育教材精选

程序设计基础

赵 宏 主 编
王 恺 副主编

清华大学出版社



高等学校计算机基础教育教材精选

程序设计基础

赵 宏 主 编
王 恺 副主编

清华大学出版社
北 京

内 容 简 介

本书是专门为高等学校理工类特别是新工科学生编写的教材。全书共 16 章,内容涵盖结构化程序设计方法、面向对象程序设计方法以及基本的数据结构和算法 3 部分。本书通过对一些精选问题求解思路和方法的分析,以及针对初学者容易出现错误和困惑的地方提供的大量提示,帮助读者更好地理解使用计算机解决问题的基本原理和方法,提高计算思维能力,初步具备使用 C++ 程序设计语言解决实际问题的能力。

本书面向初学者,不要求读者具备计算机高级程序设计语言方面的背景知识。本书也是学堂在线的“程序设计基础(上)”和“程序设计基础(下)”MOOC 课程使用的教材,同时配套有《程序设计基础——上机实习及习题集》。本书也适合结合 MOOC 课程自主学习的读者学习参考。

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

程序设计基础/赵宏主编. —北京:清华大学出版社,2019

(高等学校计算机基础教育教材精选)

ISBN 978-7-302-53215-6

I. ①程… II. ①赵… III. ①程序设计—高等学校—教材 IV. ①TP311.1

中国版本图书馆 CIP 数据核字(2019)第 129404 号

责任编辑:张瑞庆

封面设计:何凤霞

责任校对:胡伟民

责任印制:李红英

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:三河市君旺印务有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:27

字 数:677 千字

版 次:2019 年 9 月第 1 版

印 次:2019 年 9 月第 1 次印刷

定 价:66.90 元

产品编号:083947-01

前言

程序设计基础

21 世纪人类步入信息社会,大数据、人工智能、互联网+、物联网、区块链等已经融入人们日常生活中,正在影响和改变着人们工作、学习和生活的方式,而这些都离不开计算机。

“计算”本身是一门学科,它在发展的同时也促进了其他学科的发展。21 世纪科学上最重要的、经济上最有前途的研究前沿都有可能通过与计算科学进行学科融合而得到解决。计算机不仅为不同专业提供了解决专业问题的有效方法和手段,而且还提供了一种独特的处理问题的思维方式——计算思维。逻辑思维、实证思维和计算思维三大科学思维构成了现代科技创新的三大支柱。计算思维不仅仅属于计算机科学家所特有,它已经成为每个人应该具备的基本能力。因此,在培养学生的解析能力时,不仅要掌握阅读、写作和算术(Reading, wRiting, and aRithmetic——3R)能力,还要使学生接触计算的方法和模型,学会计算思维。

众所周知,计算机可以进行数值计算(科学计算)问题的求解,例如解方程(组)、函数求值、概率统计等,用来解决如气象预报、石油探测等问题。更多的时候,人们利用计算机进行非数值计算问题的求解,例如字符、图形、图像、声音、动画等,来解决文字处理、飞机售票、学生信息管理、道路交通管理等问题。对于计算机的处理对象,特别是非数值计算的问题求解,需要研究计算机的操作对象(数据元素)以及它们之间的关系和运算。

人类使用计算机求解实际问题的基本步骤是:首先将实际问题抽象成数学模型,即分析问题,从中抽象出操作的对象和相应的操作,找出这些操作对象之间的关系,并用数学的语言加以描述;其次设计实现这些操作的算法,并编写程序实现相应的算法;最后才是运行程序对实际问题进行求解。如何表示信息和如何处理信息,这正是计算机科学研究的主要问题。人类进行抽象和形式化,就需要学习和掌握常用的计算思维方式。

程序设计方法、数据结构和算法是计算机科学与工程的基础性领域知识,是开发高效计算机程序、解决各领域应用问题的核心。在寻求和实现数学模型的过程中,计算机算法与数据的结构密切相关。学习程序设计方法、数据结构和算法课程,不仅可以使学生掌握计算机基础课程的基本方法,更是训练学生计算思维的有效途径。

本书面向高等院校理工类特别是新工科学生掌握如何使用计算机求解问题、具有主动使用计算机解决生活和学科问题的意识和能力的需求,针对计算机学科最基础性的问题编写的教材。全书共分 3 部分:①结构化程序设计方法;②面向对象程序设计方法;③基本数据结构和算法。书中除了讲解计算的基本概念、方法,还给出了完整的实现代码。几乎每一章都给出了拓展学习的内容,读者可以通过扫描二维码进一步学习和提高。同时在配套教材《程序设计基础——上机实习及习题集》中,还为每一章配套了“课程实习”“课后习题”

和“课后习题参考答案”等内容。

全书共 16 章,主要内容如下。

第 1~6 章为结构化程序设计方法。其中:

第 1 章 如何让计算机进行计算。首先介绍计算思维和程序流程图的基本画法;然后介绍程序设计的基本概念、步骤和方法;最后介绍 C++ 源程序的基本结构和组成元素以及 Visual studio 2010 集成开发环境。

第 2 章 计算机如何表示与处理数据。首先介绍二进制数及几种基本数据类型的二进制数据表示方法,包括不同数制数据之间的转换方法,整数、实数、字符和逻辑型数据的二进制表示方法等;然后介绍如何通过 C++ 语言实现这些基本数据类型在计算机中的存储,以及如何对这些基本数据类型的数据进行处理的方法。

第 3 章 选择与迭代算法。介绍处理问题时的选择算法和迭代算法,以及如何使用 C++ 语言实现选择和迭代算法。

第 4 章 结构化数据的处理。介绍多记录数据和多属性数据的存储方法,以及如何使用 C++ 语言实现这些数据的存储和处理。

第 5 章 模块化。介绍模块化的思想,以及如何使用 C++ 语言编写模块化程序。

第 6 章 数据存储。重点介绍计算机中数据存储的基本原理,以及如何使用 C++ 语言编写程序去操作内存中的数据。

第 7~10 章为面向对象程序设计方法。其中:

第 7 章 面向对象方法。介绍面向对象方法的基本概念,以及用 C++ 语言实现面向对象程序设计的基本方法。

第 8 章 继承与多态。介绍如何使用 C++ 语言来实现面向对象程序设计的两个重要特性——继承和多态。

第 9 章 输入输出流。介绍标准输入输出的基本方法,即输入输出流和文件输入输出流两方面的内容。

第 10 章 模板。介绍模板的基本概念,以及 C++ 中函数模板和类模板的定义及使用等方法。

第 11~16 章为基本的数据结构和算法。其中:

第 11 章 数据结构与算法的基本概念。首先介绍数据结构的基本术语、抽象数据类型、数据结构的存储结构和逻辑结构,然后介绍算法的基本概念和算法分析方法,最后介绍算法设计基本方法与常用的算法设计策略。

第 12 章 线性表。介绍线性表的逻辑结构,并给出线性表的抽象数据类型;还介绍线性表的顺序存储和链式存储的表示和实现方法等。

第 13 章 栈和队列。介绍栈的逻辑结构和抽象数据类型,并分别给出栈的顺序存储和链式存储的表示和实现方法;介绍队列的逻辑结构和抽象数据类型,并分别给出队列的顺序存储和链式存储的表示及实现方法等。

第 14 章 树和二叉树。介绍树的基本概念、二叉树的基本特性;二叉树的顺序表示、链式表示,二叉树的遍历和其他常用操作及其实现方法;哈夫曼树和哈夫曼码等。

第 15 章 图。介绍图的基本概念;图在计算机中常用的 3 种表示方法、图的遍历方法及其实现方法;结合具体应用问题,讲解最小生成树和最短路径的问题等。

第 16 章 算法设计策略及应用实例。介绍分治、贪心、动态规划、回溯和分支限界 5 种算法设计策略,并给出相应的应用实例。启发读者在遇到实际问题时,使用合理的策略设计出理想的算法。

为了便于练习,本书不仅给出算法的描述,还给出完整的程序代码。读者可直接或稍加改动就可以复用这些代码来解决自己的实际问题。

本书是由南开大学计算机学院公共计算机基础教学部的教师结合多年的教学经验及目前理工类和新工科大学生对计算机基础知识的需要编写的,教材综合考虑了 MOOC 和 SPOC 课程知识碎片化的特点,方便教师、学生和其他读者使用。赵宏负责第 1~3 章、第 7 章、第 10~13 章和第 16 章的编写并统编全书,王恺负责第 4~6 章、第 8 章、第 9 章、第 14 章和第 15 章的编写。

在本书的编写过程中,得到了清华大学出版社张瑞庆编审的大力支持,在此表示真诚的感谢!

本书还参考了国内外的一些程序设计方面的开放课程网站和书籍,力求有所突破和创新。由于编者能力和时间的限制,书中难免有不妥之处,恳请同行和读者指正,在此表示真诚谢意!

编 者

2019 年 4 月于南开园

目录

第 1 章 如何让计算机进行计算	1
1.1 计算思维和程序流程图	1
1.1.1 计算思维.....	1
1.1.2 程序流程图.....	2
1.2 程序设计的基本概念	4
1.2.1 用计算机求解问题的过程.....	4
1.2.2 程序设计方法.....	6
1.3 高级程序设计语言——C++	7
1.4 初识 C++ 程序	8
1.4.1 简单 C++ 程序实例	9
1.4.2 C++ 源程序的组成	9
1.4.3 C++ 源程序的组成元素	12
1.5 集成开发环境——VS 2010	13
第 2 章 计算机如何表示与处理数据	16
2.1 常用数制及不同数制数值之间的转换.....	17
2.1.1 数制	17
2.1.2 不同数制之间的转换	18
2.2 整数在计算机中的表示.....	20
2.2.1 数据的单位	20
2.2.2 整数的表示方法	20
2.3 实数在计算机中的表示.....	24
2.4 非数值数据在计算机中的表示.....	25
2.4.1 字符型数据在计算机中的表示	25
2.4.2 字符串	27
2.4.3 逻辑型数据	27
2.5 C++ 中的基本数据类型和转义字符	27
2.5.1 C++ 的基本数据类型	27
2.5.2 C++ 中的转义字符	28

2.6	变量和常量	29
2.6.1	常量	29
2.6.2	变量	30
2.7	基本数据的处理	31
2.7.1	运算符和表达式	31
2.7.2	算术运算符与算术表达式	31
2.7.3	赋值运算符与赋值表达式	32
2.7.4	关系运算符与关系表达式	33
2.7.5	逻辑运算符与逻辑表达式	34
2.7.6	基本数据类型之间的转换	35
2.8	C++ 中的基本语句	38
2.8.1	定义/声明语句	38
2.8.2	表达式语句	41
2.8.3	复合语句和空语句	41
2.8.4	输入输出语句	42
2.9	C++ 中的几个特殊运算符	42
2.9.1	++和--	42
2.9.2	条件运算符	43
2.9.3	逗号运算符	45
2.9.4	sizeof 运算符	45
2.10	更多关于 C++ 的运算符和表达式	46
2.10.1	运算符的优先级和结合性	46
2.10.2	有副作用的表达式和无副作用的表达式	48
2.10.3	表达式的求值顺序	49
第 3 章	选择与迭代算法	50
3.1	单路选择算法及其 C++ 实现	50
3.1.1	单路选择问题	50
3.1.2	用 C++ 的 if 语句编程解决单路选择问题	51
3.2	双路选择算法及其 C++ 实现	52
3.2.1	双路选择问题	52
3.2.2	用 C++ 提供的 if...else 语句编程解决双路选择问题	53
3.3	嵌套选择及其 C++ 实现	54
3.4	多路选择算法及其 C++ 实现	56
3.4.1	多路选择问题	56
3.4.2	用 C++ 提供的 switch 语句编程解决多路选择问题	56
3.5	迭代算法及其 for 语句的实现	58
3.5.1	迭代算法	59
3.5.2	用 C++ 提供的 for 语句实现迭代算法	59
3.6	迭代算法及其 while 语句的实现	60

3.6.1	用 C++ 提供的 while 语句实现迭代算法	60
3.6.2	用 C++ 提供的 do...while 语句实现迭代算法	61
3.7	迭代嵌套及其 C++ 实现	62
3.8	迭代与选择嵌套及其 C++ 实现	64
3.8.1	迭代与选择嵌套及其 C++ 实现	64
3.8.2	选择与迭代嵌套及其 C++ 实现	65
3.9	C++ 中的转向语句	65
3.9.1	break 语句	66
3.9.2	continue 语句	66
3.9.3	return 语句	67
3.9.4	goto 语句	67
第 4 章	结构化数据的处理	69
4.1	一维数据及其 C++ 实现	69
4.1.1	一维数据问题	69
4.1.2	用 C++ 提供的一维数组存储一维数据	71
4.2	二维数据及其 C++ 实现	73
4.2.1	二维数据问题	73
4.2.2	C++ 提供的一维数组或二维数组存储二维数据	74
4.3	字符串及其 C++ 实现	77
4.3.1	字符串问题	77
4.3.2	用 C++ 提供的一维数组存储字符串	78
4.4	多个字符串的处理	79
4.4.1	多个字符串问题	79
4.4.2	用 C++ 提供的二维数组存储来多个字符串	80
4.5	多属性数据及其 C++ 实现	81
4.5.1	多属性数据问题	81
4.5.2	用 C++ 提供的结构体存储多属性数据	81
4.6	一组多属性数据的处理	84
4.6.1	一组多属性数据的问题	84
4.6.2	使用结构体数组对一组多属性数据进行存储和处理	84
4.7	C++ 中的枚举数据类型	85
4.7.1	枚举类型的定义	85
4.7.2	枚举变量的定义	86
4.7.3	枚举变量的使用	86
4.8	数组的应用——选择排序	87
4.8.1	选择排序算法	87
4.8.2	用 C++ 实现选择排序算法	88

第 5 章 模块化	90
5.1 模块化及其 C++ 实现	90
5.1.1 采用模块化思想处理问题	91
5.1.2 用 C++ 实现结构化程序设计	91
5.1.3 函数的调用机制及内联函数	94
5.1.4 调用库函数	95
5.2 递归算法及其 C++ 实现	95
5.2.1 递归算法	95
5.2.2 递归算法实例	96
5.3 默认形参值	98
5.3.1 指定默认形参值的位置	98
5.3.2 默认形参值的指定顺序	99
5.4 函数重载	99
5.5 编译预处理	101
5.5.1 文件包含	101
5.5.2 宏定义	102
5.5.3 条件编译	103
5.6 多文件结构	105
5.6.1 头文件	105
5.6.2 源文件	106
5.6.3 多文件结构程序实例	106
5.6.4 避免头文件被重复包含	108
5.7 变量和函数的作用域与生存期	109
5.7.1 全局变量的作用域与生存期	109
5.7.2 局部变量的作用域与生存期	110
5.7.3 函数的作用域	112
5.8 模块化应用实例——二分查找法	114
5.8.1 二分查找法	114
5.8.2 二分查找法应用实例	115
第 6 章 数据存储	117
6.1 数据存储的基本原理	117
6.2 地址与 C++ 中的指针	118
6.2.1 指针变量的定义	119
6.2.2 指针变量的初始化	119
6.2.3 使用指针访问内存中的数据	120
6.3 指针与数组	123
6.3.1 数组在内存中的存储方式	123
6.3.2 使用指针操作数组	124
6.3.3 数组名与指针变量的区别	125

6.3.4 指向行的指针变量	126
6.4 指针与字符串	127
6.5 动态使用内存空间	129
6.6 二级指针	133
6.7 指针与函数	134
6.7.1 指针作为函数参数	134
6.7.2 指针作为函数返回值	140
6.8 引用与函数	141
6.8.1 引用的概念和声明	141
6.8.2 函数的传值调用	142
6.8.3 函数的引用调用	143
6.8.4 返回引用的函数	144
第7章 面向对象方法	146
7.1 面向对象方法的基本概念	147
7.2 C++ 中的类和对象	150
7.2.1 类的定义	150
7.2.2 构造函数	152
7.2.3 对象的定义和对象的访问	153
7.3 类成员的访问控制	156
7.4 析构函数	158
7.5 拷贝构造函数	160
7.6 类声明与实现的分离	162
7.7 类的静态成员	164
7.7.1 静态数据成员	164
7.7.2 静态成员函数	166
7.8 类的常量成员	168
7.8.1 常量数据成员	168
7.8.2 常量成员函数	168
7.9 this 指针	169
7.10 类的友元	170
7.11 类的对象成员	174
7.12 自定义类的运算符重载	177
7.12.1 类成员函数形式的运算符重载	177
7.12.2 类友元形式的运算符重载	179
第8章 继承与多态	184
8.1 继承	184
8.1.1 继承概述	184
8.1.2 派生类的定义	185



8.1.3	访问控制方式和派生类的继承方式	187
8.1.4	成员函数重定义	189
8.1.5	派生类的构造函数和析构函数	190
8.1.6	多继承	192
8.2	多态	199
8.2.1	类型兼容和多态性的概念	199
8.2.2	多态性的实现	202
8.3	抽象类	204
8.3.1	抽象类的作用	204
8.3.2	抽象类的实现	205
第9章	输入输出流	207
9.1	输入输出流概述	207
9.2	cout 和 cin 对象以及插入和提取运算符	208
9.2.1	标准流对象	208
9.2.2	>>和<<运算符与标准输入输出	208
9.3	使用成员函数进行标准输出和输入	210
9.3.1	使用 put()函数进行标准输出	210
9.3.2	使用 get()函数进行标准输入	210
9.3.3	getline()函数进行标准输入	212
9.4	文件流对象以及插入和提取运算符	213
9.4.1	文件流对象	213
9.4.2	<<和>>运算符与文件输入输出	216
9.5	使用成员函数进行文件的输出和输入	217
9.5.1	使用 put()函数进行文本文件输出	217
9.5.2	使用 get()函数进行文本文件输入	218
9.5.3	使用 getline()函数进行文本文件输入	218
9.6	按数据块进行输出和输入	220
9.6.1	使用 write()函数按数据块进行输出	220
9.6.2	使用 read()函数按数据块进行输入	221
9.7	文件的随机读写	225
9.8	自定义数据类型的输入输出	227
第10章	模板	231
10.1	函数模板	231
10.1.1	函数模板的定义	232
10.1.2	函数模板的使用	232
10.2	类模板	234
10.2.1	类模板的定义	235
10.2.2	类模板的使用	236

10.2.3	类模板的静态成员和友元	238
第 11 章	数据结构和算法的基本概念	240
11.1	数据结构的基本概念	240
11.1.1	基本术语	241
11.1.2	数据的逻辑结构	242
11.1.3	数据的存储结构	244
11.1.4	数据的操作	245
11.2	抽象数据类型	246
11.3	算法设计与算法分析基础	248
11.3.1	算法的基本概念	248
11.3.2	算法分析	249
11.3.3	算法分析实例	254
11.4	算法设计基本方法与策略基础	256
11.4.1	算法设计的方法	256
11.4.2	算法设计策略	260
第 12 章	线性表	269
12.1	线性表及其抽象数据类型	269
12.1.1	线性表的基本概念	270
12.1.2	线性表的抽象数据类型	270
12.2	线性表的顺序存储结构及其实现	271
12.2.1	线性表的顺序表示	271
12.2.2	顺序表的实现	273
12.2.3	顺序表代码复用实例	277
12.3	线性表的链式表示方法及实现	281
12.3.1	链式存储结构	281
12.3.2	单向链表及其基本操作	281
12.3.3	单向链表代码复用实例	288
12.3.4	线性表的顺序存储与链式存储的比较	291
12.3.5	循环链表及其基本操作	292
12.3.6	双向链表及其基本操作	294
第 13 章	栈和队列	297
13.1	栈的基本概念	297
13.1.1	栈的基本概念	297
13.1.2	栈的抽象数据类型	298
13.2	栈的表示及实现	299
13.2.1	栈的顺序表示及实现	299
13.2.2	顺序栈代码复用实例	303
13.2.3	栈的链式表示及实现	304

13.3	队列的基本概念	307
13.3.1	队列的基本概念	307
13.3.2	队列的抽象数据类型	308
13.4	队列的表示及实现	308
13.4.1	队列的顺序表示及实现	309
13.4.2	循环队列代码复用实例	313
13.4.3	队列的链式表示及实现	315
第 14 章	树和二叉树	319
14.1	树的基本概念	319
14.1.1	树的定义	321
14.1.2	树的表示形式	321
14.1.3	树的基本术语	322
14.2	二叉树及其基本性质	324
14.2.1	二叉树的定义	324
14.2.2	二叉树的基本性质	325
14.3	二叉树的抽象数据类型和表示方式	327
14.3.1	二叉树的顺序表示及实现	328
14.3.2	二叉树的链式表示及实现	333
14.4	二叉树的遍历及常用操作	339
14.4.1	二叉树的遍历及其实现	339
14.4.2	二叉树其他常用操作的实现	345
14.5	二叉排序树	350
14.5.1	二叉排序树的定义	350
14.5.2	二叉排序树的生成	350
14.5.3	二叉排序树的查找	353
14.6	二叉树排序树应用示例	355
14.7	哈夫曼树和哈夫曼码	356
14.7.1	基本术语	356
14.7.2	哈夫曼树及其构造方法	357
14.7.3	哈夫曼码及其编解码方法	358
第 15 章	图	360
15.1	图的基本概念及特性	360
15.2	图的抽象数据类型和表示方式	364
15.2.1	图的抽象数据类型	364
15.2.2	图的表示法	365
15.2.3	图的邻接矩阵表示法的实现	367
15.3	图的遍历	370
15.3.1	广度优先遍历及其实现	371

15.3.2	深度优先遍历及其实现	373
15.4	应用实例	376
15.4.1	图的应用	376
15.4.2	用图来描述和求解实际问题	377
第 16 章	算法设计策略及应用实例	380
16.1	分治策略	380
16.1.1	分治策略概述	380
16.1.2	分治策略的算法设计步骤和程序模式	381
16.1.3	分治策略应用实例	382
16.2	贪心策略	385
16.2.1	最优化问题与最优化原理	385
16.2.2	贪心策略概述	385
16.2.3	贪心策略的算法设计步骤及程序模式	386
16.2.4	贪心策略应用实例	387
16.3	动态规划策略	389
16.3.1	动态规划策略概述	390
16.3.2	动态规划策略的相关概念	392
16.3.3	动态规划策略算法设计步骤及程序模式	394
16.3.4	动态规划策略应用实例	395
16.4	回溯策略	398
16.4.1	回溯策略概述	398
16.4.2	回溯策略算法设计步骤及程序模式	399
16.4.3	回溯策略应用实例	400
16.5	分支限界策略	401
16.5.1	堆	401
16.5.2	分支限界策略概述	404
16.5.3	分支限界策略算法设计步骤及程序模式	405
16.5.4	分支限界策略应用实例	405

第 1 章

如何让计算机进行计算

导学

【主要内容】

本章首先介绍计算思维和程序流程图的基本画法；然后介绍程序设计的基本概念、步骤和方法；接着简单介绍 C++ 高级程序设计语言，并通过两个简单的 C++ 程序实例，介绍 C++ 源程序的基本结构和组成元素；最后介绍 Visual Studio 2010 集成开发环境。

【重点】

- 计算思维的基本概念。
- 程序流程图的画法。
- C++ 程序的基本组成。
- 在 Visual Studio 2010 下编辑和运行 C++ 程序。

【难点】

培养主动使用计算机去解决生活和学科中问题的意识。



1.1 计算思维和程序流程图

1.1.1 计算思维

随着计算机技术的发展和应用，大数据、人工智能、互联网+、物联网、区块链等都已经融入人们日常的生活，正在影响和改变着人们工作、学习和生活的方式，而这些都离不开计算机，离不开程序设计。计算思维直面机器智能的不解之谜：在什么方面人类比计算机做得好？在什么方面计算机比人类做得好？最基本的问题是：什么是可计算的？

要使用计算机解决问题，就要具备计算思维。周以真教授在“Computational Thinking”这篇论文中明确地定义了计算思维：计算思维是运用计算机科学的基本理念，进行问题求解、系统设计以及理解人类行为。也就是说，计算思维是一种利用计算机解决问题的思维方式，而不是具体的学科知识。计算思维已成为世界公认的与理论思维、实验思维并列的三大思维之一。

计算思维作为利用计算机解决问题的思维方式,应当在所有领域被应用,而不是局限于计算机学科领域。计算思维是每个人的基本技能,而不仅仅属于计算机科学家。我们已见证了计算思维在其他学科中的影响。例如,机器学习已经改变了统计学,就数学尺度和维数而言,统计学习用于各类问题的规模在几年前还是不可想象的。人类已进入人工智能时代,人工智能的基础和核心都是机器学习。

当下及未来,计算思维是每个人都要具备的基本能力,而学习程序设计是培养计算思维的最直接有效的方法。本书将使用计算机进行计算的基本原理和方法以及如何使用 C++ 语言去实现这些方法等内容有机融合,培养人们主动运用“计算”的思想去思考和解决问题的意识及能力,为将来使用计算机解决问题打下良好的基础。

1.1.2 程序流程图






“程序流程图”简称“流程图”,是人们对解决问题的方法、思路或算法的一种描述。流程图利用图形化的符号框来表示不同的操作,并用流程线来连接这些操作。通过画流程图,可以清楚地描述算法,方便交流,并为后边编写程序实现算法打好基础。

流程图具有如下优点:

- 采用简单规范的符号,画法简单。
- 结构清晰,逻辑性强。
- 便于描述,容易理解。

流程图的基本符号及其含义见表 1-1。

表 1-1 流程图的基本符号和含义

符号名称	图形符号	含 义
圆角矩形		开始框或结束框,描述算法的开始或结束
矩形		处理框,描述处理
平行四边形		输入输出框,描述数据的输入或结果的输出
菱形		判断框,根据某个条件是否满足,作进一步相应的处理
流程线		流程线,描述各种处理之间的顺序关系

任何复杂的算法都可以由顺序结构、选择(分支)结构和循环(迭代)结构这 3 种基本结构组成,因此,只要掌握 3 种基本结构的流程图的画法,就可以画出任何算法的流程图。

【例 1-1】 图 1-1 是求任意两个整数和的算法流程图,采用顺序结构。

通过流程图可以看出,求任意两个整数和的算法如下:

处理开始→输入任意两个整数到 x 和 y 中→计算 x 和 y 的和并放到 result 中→输出问题求解结果 result→处理结束。

【例 1-2】 图 1-2 是判断某年是否是闰年的算法流程图,采用选择结构。

通过流程图可以很容易就看出判断某年是否是闰年的算法,表 1-2 是描述这个算法的表格。