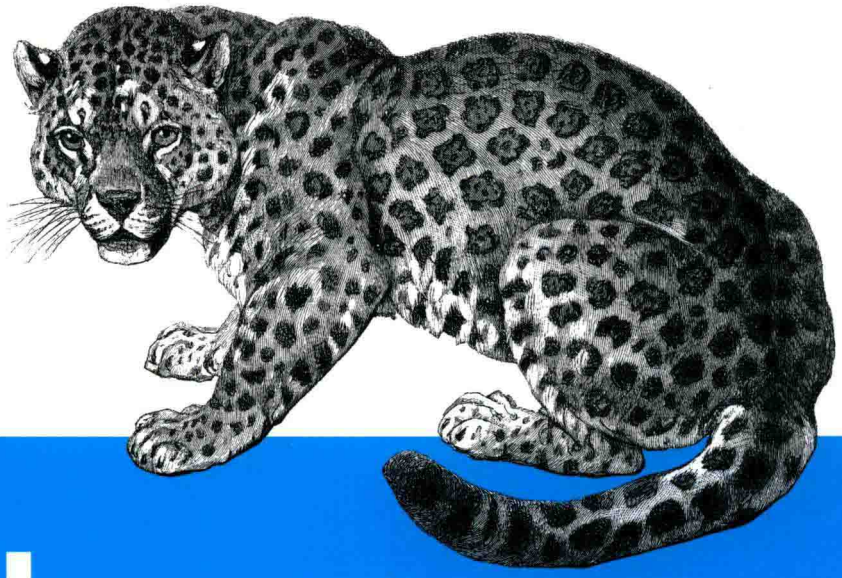
21 世纪高等教育
计算机规划教材



# C++ 程序设计 | 思想与方法

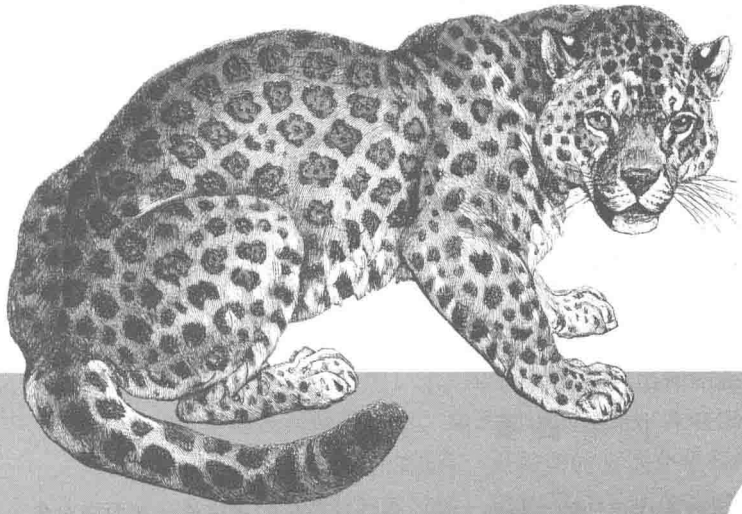
## 慕课版 第3版

◆ 翁惠玉 俞勇 编著

+ **互联网 + 教材：** 人邮学院慕课平台作支撑，买书送上海交通大学教学名师录制的全套慕课
+ **强调思想与方法：** 侧重于解决问题，强调结构化程序设计和面向对象程序设计的思想和方法
+ **讲解深入且语言通俗：** 利用计算学科中的经典问题，解释计算机内部的处理过程，让读者知其然，更知其所以然

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

21 世纪高等教育
计算机规划教材



# C++ 程序设计 | 思想与方法

慕课版 第3版

◆ 翁惠玉 俞勇 编著

人 民 邮 电 出 版 社
北 京

**图书在版编目（CIP）数据**

C++程序设计 : 思想与方法 : 慕课版 : 第3版 / 翁惠玉, 俞勇编著. -- 3版. -- 北京 : 人民邮电出版社, 2016.8
21世纪高等教育计算机规划教材
ISBN 978-7-115-42935-3

Ⅰ. ①C… Ⅱ. ①翁… ②俞… 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(2016)第151403号

## 内 容 提 要

本书以 C++语言为环境，重点讲授程序设计的思想和方法，包括过程化的程序设计和面向对象的程序设计，且本书非常强调程序设计的风格，将常用程序设计风格的要求贯穿于本书的各个章节。

本书的内容可以分为两大部分：第 1～9 章为第 1 部分，主要介绍一些基本的程序设计思想、概念、技术、良好的程序设计风格以及过程化程序设计，包括数据类型、控制结构、数据封装、过程封装以及各种常用的算法；第 10～16 章为第 2 部分，重点介绍面向对象的思想，包括如何设计及实现一个类、如何利用组合和继承实现代码的重用、如何利用多态性使程序更加灵活、如何利用抽象类制定一些工具的规范，最后为了更好地与数据结构课程衔接，介绍了容器和迭代器的概念。

本书可作为各高等院校计算机专业的教材，也可供从事计算机软件开发的科研人员参考。

◆ 编　　著　翁惠玉　俞　勇
责任编辑　税梦玲
责任印制　沈　蓉　彭志环

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：23.25　　2016 年 8 月第 3 版
字数：612 千字　　2016 年 8 月北京第 1 次印刷

定价：49.80 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315

# 改版说明

本书是作者多年讲授“程序设计”课程的经验与教训的总结。2008 年编写了本书的第 1 版，由人民邮电出版社出版，2012 年修改出版了本书的第 2 版。经过了 4 年的使用，我们又发现了第 2 版的一些不足之处，且收集了学生和授课老师对第 2 版的意见和建议，在上海交通大学和人民邮电出版社的支持下，我们对《C++程序设计：思想与方法（第 2 版）》一书进行了改版。

本书的第 3 版保持了原有的写作风格，继续秉承以程序设计方法为主、程序设计语言为辅的思想，采用以问题求解引出知识点的方法，在介绍语言要素的同时，更多地强调编程思想，强调知识的应用，并在以下几点做了较大的改进：

- 用更通俗的语言、更贴切的实例介绍程序设计；
- 跟随语言的发展，增加了 C++11 的内容；
- 加强了算法设计和问题求解过程的内容，增加了更多的实例；
- 对结构和内容进行了细致的修改，结构更加合理、内容更加易懂；
- 删除了面向对象中的多继承；
- 内容分成必讲部分和选讲部分，选讲部分与选讲部分相关的习题用*做了标注，教师可以根据学生的情况选择授课内容。

为了让学生或购买本书的广大读者能够更好地自学程序设计，在人民邮电出版社的大力支持下，我们还录制了大量的慕课视频，所有的慕课视频均放在人民邮电出版社自主开发的在线教育慕课平台——人邮学院，建议大家结合人邮学院进行学习。下面对人邮学院的使用方法做出说明。

1. 购买本书后，刮开粘贴在书封底上的刮刮卡，获取激活码（见图 1）。

2. 登录人邮学院网站（www.rymooc.com），或扫描封面上的二维码，使用手机号码完成网站注册（见图 2）。


图 1　激活码


图 2　注册人邮学院网站

3. 注册完成后，返回网站首页，单击页面右上角的“学习卡”选项（见图 3）进入“学习卡”页面（见图 4），输入激活码，即可获得课程的学习权限。


图3　单击“学习卡”选项

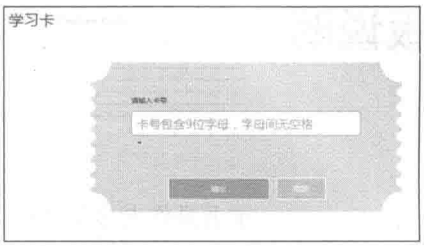



图4　在“学习卡”页面输入激活码

4. 获取权限后，学习者可根据自身情况，随时随地使用计算机、平板电脑以及手机，在课时列表（见图5）中选择课时进行学习。


图5　课时列表

5. 当在学习中遇到困难，可到讨论区（见图6）提问，导师会及时答疑解惑，本课程的其他学习者也可帮忙解答，互相交流学习心得。

6. 本书配套的PPT等资源，可在“C++程序设计”首页底部的资料区下载（见图7），也可到人邮教育社区（www.ryjiaoyu.com）下载。


图6　讨论区


图7　配套资源

人邮学院平台的使用问题，可咨询在线客服，或致电010-81055236。

希望修改以后的第3版能更好地帮助大家学习程序设计。

编者

2016年5月

# 第 3 版前言

程序设计是计算机专业十分重要的一门课程，是实践性非常强的一门课程，也应该是一门非常有趣、让学生很有成就感的课程，但在教学过程中，很多学生的反应是听懂了，但不会做，以至于最后丧失了兴趣。我们认为主要的问题是，在教学过程中教师过分重视程序设计语言本身，强调理解语言的语法，而没有把重点放在解决问题的方法上。

本书以介绍基本的程序设计思想、概念和方法为基础，强调算法、抽象等重要的程序设计思想，并选择 C++为教学语言。C++是业界非常流行的语言，它使用灵活，功能强大，既支持过程化的程序设计，又支持面向对象的程序设计，可以很好地体现程序设计的思想和方法。本书旨在强调思想，C++语言服务于这个目标，因此，对于 C++的一些特殊的成分和技巧，本书将不予重点介绍。

本书是作者根据多年来在上海交通大学讲授“程序设计”课程的经验，并参考近年来国内外经典的程序设计类图书之后编写而成的。下面介绍本书的特点。

- 相比于程序设计语言，本书更强调如何解决问题，强调程序设计的思想和方法。它以 C/C++为语言环境，全面介绍结构化程序设计和面向对象的程序设计的方法。
- 注意培养学生良好的程序设计风格，讲解了该如何进行变量/函数的命名、程序的排版、常用语句的组合等。
- 本书采用以应用引出知识点的方法，让学生先知道学习的目的，提高学生的学习兴趣。特别是尽可能地利用计算学科中的经典问题，如汉诺塔、八皇后问题等，使学生在学习程序设计的过程中不断加深对计算学科的了解。
- 本书内容丰富，覆盖面广，有一定难度。但是，本书在内容的安排上采用了模块化的结构，某些难度较大的内容用*号进行标注，学习时可以先略过，不会影响整个知识的连贯性。教师可以根据学生实际情况选讲部分内容。
- 讲解深入。例如对函数的调用、变量的作用域、递归的处理等，本书都解释了计算机内部的处理过程，使学生不仅知其然，还知其所以然。
- 强调解决问题的方法。介绍了常用的算法设计方法，包括枚举法、贪婪法、分治法、回溯法和动态规划。
- 衔接数据结构课程。对数据结构常用的工具做了详细的介绍，例如链接结构、容器和迭代器等。
- 本书每一章都设有“编程规范及常见错误”小节。“编程规范”旨在指导学生养成良好的程序设计风格，“常见错误”总结了初学者常犯的错误。
- 追随语言的发展，引入 C++11 新增加的功能。

■ 本书提供人邮学院慕课平台作支撑，操作方法见改版说明。另外，书中的重难点知识处，均放置了二维码，学习者可扫描二维码打开教学视频进行在线学习。

本书得以顺利地编写和出版，首先要感谢上海交通大学电信学院程序设计课程组的各位老师，是大家经常在一起的讨论使我们不断加深对程序设计的理解；还要感谢那些可爱的学生们，是他们与我们在课上和课后的互动，使我们了解他们的困惑，以及学习难点。

编　者

2016年5月

# 目　录

# 第 1 章 绪论

自从第一台计算机问世以来，计算机技术发展得非常迅速，功能不断扩展，性能突飞猛进。特别是微型计算机的出现，使得计算机的应用从早期单纯的数学计算发展到处理各种媒体的信息。计算机本身也从象牙塔进入了千家万户。编写程序也不再是象牙塔中的工作，而是各个专业的技术人员都需要具备的技能。

## 1.1 程序设计概述

程序设计就是教会计算机如何去完成某一特定的任务，即设计出完成某个任务的正确的程序。学习程序设计就是学习当老师，你的学生就是计算机。老师上课前先要备课，然后再去上课，最后检查学生的学习情况是否达到了预期效果。对应于这 3 个阶段，程序设计也包括 3 个过程：第一步是算法设计，第二步是编码，第三步是编译与调试。

什么是程序设计

上课前首先要知道学生的知识背景，然后才能有的放矢地去教，学习程序设计首先也要了解计算机能做什么。备课就是把所要教授的知识用学生能够理解的方式表达出来。算法设计也就是把解决问题的过程分解成一系列计算机能够完成的基本动作。上课是把备课的内容用某种学生能够理解的语言描述出来。给中国学生讲课，就把备课的内容用中文讲出来。如果给美国学生讲课，就把备课的内容用英语讲出来。编码阶段也是如此。如果你的计算机支持 C 语言，就把算法用 C 语言表示出来。如果支持 PASCAL 语言，就用 PASCAL 语言描述。算法中的每一步骤都能与程序设计语言的某个语句相对应。上完课后要检查教学的效果。如果没有达到预期的结果，需要检查备课或上课中哪个环节出了问题，修改这些问题，重新再试。同样，编码后要运行程序，检查程序的结果是否符合预期的效果，如果没有，则需要检查算法和程序代码，找出问题所在，修改程序，然后重新运行。

为此，在学习程序设计之前，我们先了解一下计算机的基本功能。

## 1.2 计算机组成

计算机系统由硬件和软件两部分组成。硬件是计算机的物理构成，是计算机的物质基础，是看得见摸得着的；软件是计算机程序及相关文档，是计算机的灵魂。

### 1.2.1　计算机硬件

经典的计算机硬件结构是由计算机的鼻祖冯·诺依曼提出的，因此被称为冯·诺依曼体系结构。冯·诺依曼体系结构主要包括以下3个方面内容。

（1）计算机的硬件由5大部分组成，即运算器、控制器、存储器、输入设备和输出设备，这些部分通过总线或其他设备互相连接，如图1-1所示。这五大部分协同完成计算任务。在现代计算机系统中，运算器和控制器通常集成在一块称为CPU（中央处理器）的芯片上。

（2）数据的存储与运算采用二进制表示。

（3）程序和数据一样，存放在存储器中。

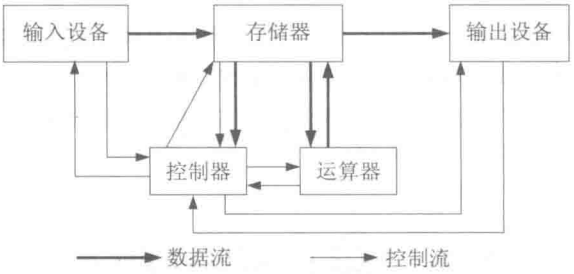



图1-1　计算机硬件系统的组成

运算器是真正执行计算的组件。它在控制器的控制下执行程序中的指令，完成算术运算、逻辑运算和移位运算等。不同厂商生产的机器，由于运算器的设计不同，能够完成的任务也不同，所能执行的指令也不完全一样。每台计算机能完成的指令集合称为这台计算机的**指令系统**或**机器语言**。运算器由算术逻辑单元（ALU）和寄存器组成。ALU完成相应的运算，寄存器用来暂存参加运算的数据和中间结果。

控制器用于协调机器其余部分的工作，是计算机的“神经中枢”。控制器依次读入程序的每条指令，分析指令，指挥各其他部分共同完成指令要求的任务。控制器由程序计数器（PC）、指令寄存器（IR）、指令译码器（ID）、时序控制电路及微操作控制电路等组成。程序计数器用来对程序中的指令进行计数，使控制器能依次读取指令；指令寄存器暂存正在执行的指令；指令译码器用来识别指令的功能，分析指令的操作要求；时序控制电路用来生成时序信号，以协调在指令执行周期中各部件的工作；控制电路用来实现各种操作命令。

存储器用来存储数据和程序。存储器可分为主存储器和外存储器。主存储器又称为**内存**，用来存放正在运行的程序和数据，具有存取速度快，可直接与运算器、控制器交换信息等特点，但其容量一般不大，而且一旦断电，信息将全部丢失。外存储器（包括硬盘、光盘、U盘等）用来存放长期保存的数据，其特点是存储容量大、成本低，但它不能直接与运算器、控制器交换信息，需要时可成批地与内存交换信息。

存储器内最小的存储单元是比特（bit）。它可以存放二进制数的一位，即一个0或一个1。bit是一个组合词，它是英语的binary digit中的字母的组合。通常8比特组成1字节（byte，缩写为B）。字节是大部分计算机分配存储器时的最小单位。存储器的容量也是用字节来表示的，如在日常中，说某台计算机的内存为1G，其实是表明这台计算机的内存是1G字节。

输入/输出设备又称外围设备，它是外部和计算机交换信息的渠道。输入设备用于输入程序、数据、操作命令、图形、图像和声音等信息。常用的输入设备有键盘、鼠标、扫描仪、光笔及语

音输入装置等。输出设备用于显示或打印程序、运算结果、文字、图形、图像等，也可以播放声音和视频等信息。常用的输出设备有显示器、打印机、绘图仪及声音播放装置等。

事实上，计算机的工作过程与我们日常生活中的工作过程完全一致。当被要求做一道四则运算题时，你会通过某个途径获取这道题并记录下来。例如，你会把这道题目抄到自己的本子上，那么本子就是主存储器，你的笔就是输入装置。要计算这道题目，你会根据先乘除后加减的原则找出里面的乘除部分，在草稿纸上计算，结果写回本子上，再执行加减得到最后结果写回本子。在此过程中，你的大脑就是 CPU，先做乘除后做加减的过程就是程序，草稿纸就是运算器中的寄存器，把答案交给老师的过程就是输出过程。

### 1.2.2　计算机软件

计算机硬件是有形的实体，可以从商店里买到。但如果计算机只有硬件，那么，它和一般的家用电器没有区别。计算机之所以有魅力，是因为它有七十二般变化，可以根据我们的要求变换不同的角色。一会儿是计算器，一会儿是字典，一会儿是 CD 播放机，一会儿又成了一架照相机。要做到这些，必须有各种软件的支持。硬件相当于人的“躯体”，能做一些最基本的动作。而软件相当于人的“思想”和处理问题的能力。一个人可能面临各种要处理的问题，他必须学习相关的知识；计算机需要解决各种问题，它需要安装各种软件。

软件可以分为系统软件和应用软件。系统软件居于计算机系统中最靠近硬件的部分，它将计算机的用户与硬件隔离，让用户可以通过软件指挥硬件工作。系统软件与具体的应用无关，但其他软件都要通过系统软件才能发挥作用。操作系统就是典型的系统软件。应用软件是为了支持某一应用而开发的软件，如字处理软件、财务软件等。

## 1.3　程序设计语言

程序设计语言是人和计算机进行交流时采用的语言。随着计算机的发展，人类与计算机交互的语言也在进步，从早期的由二进制表示的机器语言发展到了如今的类似于英语的高级语言。

### 1.3.1　机器语言

计算机刚出现时，人类和计算机交互的语言只有**机器语言**。每种计算机都有自己的机器语言。机器语言中的每个语句都是一个二进制的比特串。

机器语言是由计算机硬件识别并直接执行的语言。机器语言能够提供的功能是由计算机硬件设计所决定，因而能提供的功能非常简单，否则会导致计算机的硬件设计和制造过于复杂。不同的计算机由于硬件设计的不同，它们的机器语言也是不一样的。机器语言之所以必须由 0、1 组成是因为计算机内部的电路都是开关电路。0 和 1 正好对应于开和关两种状态。

每条机器语言的指令一般都包括操作码和操作数两个部分。操作码指出了运算的种类，如加、减、移位等。操作数指出参加运算的数据值或数据值的存储地址，如内存地址或寄存器的编号。

机器指令根据其功能一般可以分成控制指令、算术运算指令、逻辑运算指令、数据传送指令和输入输出指令。

由于机器语言是由硬件实现，提供的功能相当简单。用机器语言编制程序相当困难，就如教一个小学生做微积分一样困难。机器语言使用二进制比特串表示，因此用机器语言书写的程序很

难阅读和理解，容易出错。而且程序员在用机器语言编程序时，还必须了解机器的很多硬件细节。例如有几类寄存器，每类寄存器有多少个，每个寄存器长度是多少，内存大小是多少等。由于不同的计算机有不同的机器语言，一台计算机上的程序无法在另外一台不同类型的计算机上运行，这将引起大量的重复劳动。

机器语言通常也被称为**第一代语言**。

### 1.3.2 汇编语言

为了克服机器语言可读性差的缺点，人们采用了与机器指令意义相近的英文缩写作为助记符，于是在20世纪的50年代出现了**汇编语言**。汇编语言是符号化的机器语言，即将机器语言的每条指令符号化，采用一些带有启发性的文字串，如ADD（加）、SUB（减）、MOV（传送）、LOAD（取）。常数和地址也可以规定用一些符号写在程序中。

与机器语言相比，汇编语言的含义比较直观，使程序的阅读和理解更加容易。但计算机硬件只“认识”0、1组成的机器语言，并不“认识”由字符组成的汇编语言，不能直接理解和执行汇编语言写的程序。必须将每一条汇编语言的指令翻译成机器语言的指令后计算机才能执行。为此，人们创造了一种称为**汇编程序**的程序，让它充当汇编语言的程序到机器语言程序的翻译，将汇编语言写的程序翻译成机器语言写的程序。

汇编语言解决了机器语言的可读性的问题，但没有解决机器语言的可移植性的问题。而且汇编语言的指令与机器语言的指令基本上是一一对应的，提供的基本功能与机器语言是一致的，都是一些非常基本的功能，所以用汇编语言编写程序还是很困难的。

汇编语言通常被称为**第二代语言**。机器语言和汇编语言统称为**低级语言**。

### 1.3.3 高级语言

高级语言也被称为第三代语言。高级语言的出现是计算机程序设计语言的一大飞跃，FORTRAN、COBOL、BASIC、C++等都是高级语言。

高级语言是一种与机器的指令系统无关、表达形式更接近于科学计算的程序设计语言，从而更容易被科技工作者掌握。程序员只要熟悉简单的几个英文单词、熟悉代数表达式，以及规定的几个语句格式，就可以方便地编写程序，而且不需要知道机器的硬件环境。

由于高级语言是独立于机器硬件环境的一种语言，因而有较好的可移植性。在一种计算机上编写的程序可以在另外一台不同类型的计算机上运行，从而减少了程序员的重复劳动。高级语言提供的功能也比机器语言强得多，编写程序更加容易。

尽管每种高级语言都有自己的语法规则，但提供的功能基本类似。每种程序设计语言都允许在程序中直接写一些数字或字符串，这些被称为**常量**。对于在写程序时没有确定的值，可以给它们一个代号，称为**变量**。高级语言事先做好了很多处理不同数据的工具，称为**数据类型**，如整型、实型和字符型等。每个工具实现了一种类型的数据处理。如整型解决了整数在计算机内的如何保存、如何实现整数的各种运算问题。当程序需要处理整数时，可以直接用整型这个工具。程序设计语言提供的类型越多，功能也越强。如果程序设计语言没有提供某种类型，则当程序要处理这种类型的信息时，程序员必须自己编程解决。例如，通常的程序设计语言都没有复数这个类型，如果某个程序要处理复数，程序员必须在自己的程序中解决复数的存储和计算问题。高级语言提供了**算术运算**和**逻辑运算**的功能，使程序员可以用类似于数学中的表达式表示算术运算和逻辑运算。还提供了将一个常量或表达式计算结果与一个变量关联起来的功能，这称为**变量赋值**。也可以根据程序执行过程

中的某些中间值执行不同的语句，这称为程序设计语言的**控制结构**。对于一些复杂的大问题，直接设计出完整的算法有一定的困难，通常采用将大问题分解成一系列小问题。在设计解决大问题的算法时，可以假设这些小问题已经解决，直接调用解决小问题的程序即可。每个解决小问题的程序被称为一个**过程单元**。在程序设计语言中过程单元被称为函数、过程或子程序等。

如果解决某个问题用到的工具都是程序设计语言所提供的工具，如处理整数或实数的运算，这些程序很容易实现。如果用到了一些程序设计语言不提供的工具，则非常困难，如要处理一首歌曲、一张图片或一些复数的数据。我们希望能有这样一个工具可以播放一首歌曲或编辑一首歌曲，这时我们可以创建一个工具，即创建一个新的数据类型，如歌曲类型、图片类型和复数类型。这就是**面向对象程序设计**。面向对象的程序设计提供了我们创建工具的功能。

### 1.3.4　C++语言

C++是本书选用的程序设计语言。C++语言是从 C 语言发展演变而来。C 语言是贝尔实验室的 Dennis Ritchie 在 B 语言的基础上开发的。1972 年在 DEC PDP-11 计算机上得到了实现。C 语言作为 Unix 操作系统的开发语言而广为人知。1989 年美国国家标准协会制定了 C 语言的标准（ANCI C）。

C 语言简洁、灵活、使用方便，可直接访问内存地址，支持位操作，因而能很好地胜任操作系统及各类应用软件的开发，是一款应用非常广泛的结构化程序设计语言。

但 C 语言类型检查机制相对薄弱，几乎没有支持代码重用的机制。当程序规模越来越大时，程序员很难控制程序的复杂性。为此，贝尔实验室的 Bjarne Stroustrup 于 1980 年开始对 C 语言进行改进和扩充，增加了面向对象的功能，并于 1983 年正式取名为 C++。1984 年制定了 ANSI C++标准草案。1998 年 11 月被国际化标准组织（ISO）批准为国际标准，即 C++98，是最常用的 C++标准。2003 年对 C++98 进行了少许的改进，称为 C++03。随后 C++还在不断完善，2011 年 9 月出版了一个新的标准，称为 C++11。

C 是 C++的基础，C++包含了完整的 C 语言的特征和优点，同时又增加了对面向对象程序设计的支持。所以，学习 C++必先学习其过程化的部分，即 C 语言部分，再学习它的面向对象部分。

## 1.4　程序设计过程

要使计算机能够完成某个任务，必须有相应的软件，而软件中最主要的部分就是程序。**程序**是计算机完成某个任务所需要的指令集合。通常程序设计都是基于高级语言。

程序设计就是教会计算机去完成某一特定的任务，即设计出完成某个任务的程序。它包括 4 个过程：第一步是设想计算机如何一步一步地完成这个任务的，即将解决问题的过程分解成程序设计语言所能完成的一个个基本功能，这一阶段称为**算法设计**；第二步是用某种高级语言描述这个完成任务的过程，这一阶段称为**编码**；第三步是将高级语言写的程序翻译成硬件认识的机器语言，这个阶段称为**编译与链接**；第四步是检验编好的程序是否正确完成了给定的任务，这一阶段称为**调试**。

算法的表示

### 1.4.1　算法设计

算法设计就是要设计一个使用计算机（更确切讲是某种程序设计语言）

提供的基本动作来解决某一问题的方案，是程序设计的灵魂。算法设计的难点在于计算机提供的基本功能非常简单，而人们要它完成的任务是非常复杂的。算法设计必须将复杂的工作分解成一个个简单的、计算机能够完成的基本动作。

解决问题的方案要成为一个算法，必须用清楚的、明确的形式来表达，以使人们能够理解其中的每一个步骤，无二义性。算法中的每一个步骤必须有效，是计算机可以执行的操作。例如，若某一算法包含“用π的确切值与 $r$ 相乘”这样的操作，则这个方案就不是有效的，因为计算机无法算出π的确切值。而且，算法不能无休止地运行下去，必须在有限的时间内给出一个答案。综上所述，算法必须具有以下3个特点。

（1）表述清楚、明确，无二义性。

（2）有效性，即每一个步骤都切实可行。

（3）有限性，即可在有限步骤后得到结果。

有些问题非常简单，一下子就可以想到相应的算法，没有多大的麻烦就可以写一个解决该问题的程序；而当问题变得很复杂时，就需要更多的思考才能想出解决它的算法。与所要解决的问题一样，各种算法的复杂性也千差万别。大多数情况下，一个特定的问题可以有多个不同的解决方案（即算法），在编写程序之前需要考虑许多潜在的解决方案，最终选择一个合适的方案。

算法可以用不同的方法表示。常用的有自然语言、传统的流程图、结构化流程图、伪代码等方法。

流程图是比较早期提出的一种算法表示方法，由美国国家标准化协会 ANSI 规定。流程图用不同的图形表示程序中的各种不同的标准操作。流程图用到的图形及含义如图 1-2 所示。

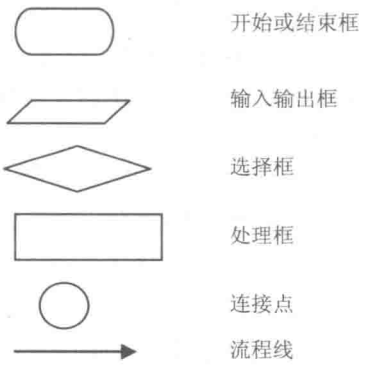



图 1-2　流程图符号

例如，求两个数相除算法的流程图表示如图 1-3 所示。

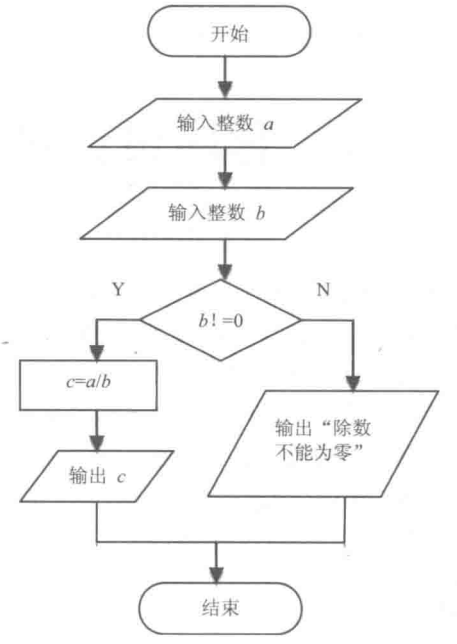



图 1-3　两个数相除算法的流程图表示