



普通高等教育创新型人才培养规划教材

# 新编 C/C++ 程序设计教程

XINBIAN C/C++ CHENGXU  
SHEJI JIAOCHENG

王晓斌 王庆军 卢颖 孙宇楠 主编



北京航空航天大学出版社  
BEIHANG UNIVERSITY PRESS



普通高等教育创新型人才培养规划教材

# 新编 C/C++ 程序设计教程

王晓斌 王庆军 主编  
卢 颖 孙宇楠

北京航空航天大学出版社

## 内 容 简 介

本书是针对 C/C++ 语言程序设计课程编写的,特别适合于学生的学习。

本书由浅入深地介绍了 C/C++ 语言中最基本、最实用的内容,主要包括:Visual C++ 6.0 开发环境、C/C++ 语言基础知识、程序控制结构、数组、指针、函数、结构体和共用体、面向对象的程序设计和文件等。书中安排了大量程序设计实例、习题、上机实践和自测题,通过实例、习题和上机实验能够帮助学生更好地掌握和运用 C/C++ 语言进行程序设计的方法和技巧;通过自测题可以检验学生对所学知识的理解和掌握程度。

本书既可作为信息管理与信息系统、电子商务和物流管理专业学生的教材,也可作为高等院校本科其他专业学生的教材。另外,也可供自学者以及参加 C/C++ 语言计算机等级考试者阅读参考。

### 图书在版编目(CIP)数据

新编 C/C++ 程序设计教程 / 王晓斌等主编. -- 北京 :  
北京航空航天大学出版社, 2015. 2  
ISBN 978-7-5124-1640-6

I. ①新… II. ①王… III. ①C 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 265546 号

版权所有,侵权必究。

### 新编 C/C++ 程序设计教程

王晓斌 王庆军 主编  
卢 颖 孙宇楠  
责任编辑 杨 昕

\*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:goodtextbook@126.com 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

\*

开本:710×1 000 1/16 印张:19.5 字数:416 千字

2015 年 2 月第 1 版 2015 年 2 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-1640-6 定价:38.00 元

---

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

# 前 言

本书是依据教育部“十二五”普通高等教育本科国家级规划教材的指导精神,结合信息管理与信息系统、电子商务、物流管理等专业的特点和培养目标编写而成的。

C/C++语言作为国际上广泛流行的通用程序设计语言,在计算机的研究和应用中已展现出其强大的生命力。C/C++语言兼顾了诸多高级语言的特点,是一种典型的结构化面向对象的程序设计语言。目前,国内大部分高等院校都把C/C++语言作为计算机和非计算机相关专业的一门程序设计语言课程。

C/C++语言涉及的概念多,规则复杂,容易出错,初学者学起来往往觉得困难。本书根据信息管理与信息系统、电子商务和物流管理三个专业的专业特点、培养目标以及教学大纲的学习要求,结合编者多年一线教学的实践经验,在充分了解学生学习C/C++语言中的难点和困惑的前提下,编写了这本适合三个专业学生培养目标的教材。新编教材重点突出、层次清晰、循序渐进、理论联系实际。另外,教材使用了大量实用的实例,使学生能轻松上手、快速掌握所学内容,全面提高学、练、用的能力。全书共10章,主要内容包括:第1章C/C++语言基础知识、第2章顺序结构程序设计、第3章选择结构程序设计、第4章循环结构程序设计、第5章数组、第6章指针、第7章函数、第8章结构体和共用体、第9章面向对象的程序设计、第10章文件。另外,本书还针对所学内容提供了上机实验题目和自测题(其中自测题3未提供参考答案),以强化和巩固所学知识,提高读者程序设计的能力。

本书可作为高等学校各专业程序设计课程的基础教学教材,尤其适合应用型本科院校计算机及非计算机专业的学生使用,同时也可作为编程人员和C/C++语言自学者的参考用书。

本书由沈阳航空航天大学王晓斌、王庆军、卢颖、魏利峰和上海宏力达信息技术有限公司的孙宇楠共同编写。

由于编者水平有限,书中难免存在一些缺点和错误,希望广大读者批评指正。

编 者

2014年11月

# 目 录

|                                 |    |
|---------------------------------|----|
| 第 1 章 C/C++ 语言基础知识 .....        | 1  |
| 学习导读 .....                      | 1  |
| 1.1 程序设计 .....                  | 1  |
| 1.1.1 程序与计算机程序 .....            | 1  |
| 1.1.2 计算机程序设计语言 .....           | 2  |
| 1.1.3 计算机程序设计 .....             | 4  |
| 1.1.4 算法及其描述 .....              | 4  |
| 1.2 C 语言和面向过程的程序设计 .....        | 6  |
| 1.2.1 C 语言简介 .....              | 6  |
| 1.2.2 C 程序的结构 .....             | 6  |
| 1.2.3 面向过程的程序设计特点 .....         | 7  |
| 1.2.4 结构化程序的三种基本结构 .....        | 7  |
| 1.3 C++ 语言和面向对象的程序设计 .....      | 10 |
| 1.3.1 C++ 语言的起源 .....           | 10 |
| 1.3.2 C++ 语言的兼容性 .....          | 11 |
| 1.3.3 C++ 语言的特点 .....           | 12 |
| 1.4 C/C++ 程序的开发过程 .....         | 13 |
| 1.4.1 编制 C/C++ 程序的步骤 .....      | 13 |
| 1.4.2 Visual C++ 6.0 上机简介 ..... | 14 |
| 1.5 数据类型 .....                  | 19 |
| 1.5.1 C/C++ 语言的词汇 .....         | 19 |
| 1.5.2 数据类型分类 .....              | 21 |
| 1.5.3 变量和常量 .....               | 22 |
| 1.5.4 数据类型之间的转换 .....           | 27 |
| 1.6 运算符和表达式 .....               | 27 |
| 1.6.1 算数运算符和算术表达式 .....         | 29 |
| 1.6.2 赋值运算符和赋值表达式 .....         | 31 |
| 1.6.3 条件运算符和条件表达式 .....         | 32 |
| 1.6.4 逗号运算符和逗号表达式 .....         | 33 |
| 本章小结 .....                      | 33 |



|                             |           |
|-----------------------------|-----------|
| 习 题 .....                   | 34        |
| <b>第 2 章 顺序结构程序设计 .....</b> | <b>37</b> |
| 学习导读 .....                  | 37        |
| 2.1 程序设计概述 .....            | 37        |
| 2.1.1 语 句 .....             | 37        |
| 2.1.2 程序的三种基本结构 .....       | 38        |
| 2.2 赋值语句 .....              | 38        |
| 2.3 C 语言的输入/输出 .....        | 39        |
| 2.3.1 字符输入函数与字符输出函数 .....   | 40        |
| 2.3.2 格式输入函数与格式输出函数 .....   | 41        |
| 2.4 C++ 的 I/O 流 .....       | 45        |
| 2.4.1 标准输出设备 cout .....     | 45        |
| 2.4.2 标准输入设备 cin .....      | 46        |
| 本章小结 .....                  | 46        |
| 习 题 .....                   | 47        |
| <b>第 3 章 选择结构程序设计 .....</b> | <b>49</b> |
| 学习导读 .....                  | 49        |
| 3.1 关系运算符与关系表达式 .....       | 49        |
| 3.1.1 关系运算符 .....           | 49        |
| 3.1.2 关系表达式 .....           | 49        |
| 3.2 逻辑运算符与逻辑表达式 .....       | 50        |
| 3.2.1 逻辑运算符 .....           | 50        |
| 3.2.2 逻辑表达式 .....           | 50        |
| 3.3 if 语句 .....             | 51        |
| 3.3.1 if 语句的几种形式 .....      | 51        |
| 3.3.2 if 语句的嵌套 .....        | 56        |
| 3.3.3 条件表达式与选择结构 .....      | 58        |
| 3.4 switch 语句 .....         | 58        |
| 本章小结 .....                  | 60        |
| 习 题 .....                   | 61        |
| <b>第 4 章 循环结构程序设计 .....</b> | <b>66</b> |
| 学习导读 .....                  | 66        |
| 4.1 for 语句 .....            | 66        |



|              |                       |            |
|--------------|-----------------------|------------|
| 4.2          | while 语句 .....        | 70         |
| 4.3          | do while 语句 .....     | 72         |
| 4.4          | 其他流程控制语句 .....        | 73         |
| 4.5          | 循环结构嵌套 .....          | 75         |
|              | 本章小结 .....            | 77         |
|              | 习 题 .....             | 78         |
| <b>第 5 章</b> | <b>数 组 .....</b>      | <b>84</b>  |
|              | 学习导读 .....            | 84         |
| 5.1          | 一维数组 .....            | 84         |
| 5.1.1        | 一维数组的定义 .....         | 84         |
| 5.1.2        | 一维数组的初始化 .....        | 86         |
| 5.1.3        | 一维数组元素的引用 .....       | 86         |
| 5.1.4        | 一维数组程序举例 .....        | 88         |
| 5.2          | 二维数组 .....            | 93         |
| 5.2.1        | 二维数组的定义 .....         | 93         |
| 5.2.2        | 二维数组的初始化 .....        | 94         |
| 5.2.3        | 二维数组元素的引用 .....       | 94         |
| 5.2.4        | 二维数组程序举例 .....        | 96         |
| 5.3          | 字符串 .....             | 99         |
| 5.3.1        | 字符数组的定义和初始化 .....     | 99         |
| 5.3.2        | 字符数组元素的引用 .....       | 101        |
| 5.3.3        | 字符数组程序举例 .....        | 102        |
| 5.3.4        | 字符串处理函数 .....         | 104        |
| 5.3.5        | C++ 的 CString 类 ..... | 105        |
|              | 本章小结 .....            | 109        |
|              | 习 题 .....             | 109        |
| <b>第 6 章</b> | <b>指 针 .....</b>      | <b>115</b> |
|              | 学习导读 .....            | 115        |
| 6.1          | 指针变量与地址 .....         | 115        |
| 6.1.1        | 指针变量的定义 .....         | 115        |
| 6.1.2        | 指针变量的初始化 .....        | 116        |
| 6.1.3        | 指针变量的引用 .....         | 116        |
| 6.2          | 指针与数组 .....           | 119        |
| 6.2.1        | 指针与一维数组 .....         | 119        |



|                            |            |
|----------------------------|------------|
| 6.2.2 指针与二维数组 .....        | 122        |
| 6.3 指针与字符串 .....           | 124        |
| 本章小结 .....                 | 127        |
| 习 题 .....                  | 127        |
| <b>第 7 章 函 数</b> .....     | <b>131</b> |
| 学习导读 .....                 | 131        |
| 7.1 函数的定义、调用和原型说明 .....    | 131        |
| 7.1.1 函数引例 .....           | 131        |
| 7.1.2 函数定义 .....           | 133        |
| 7.1.3 函数调用 .....           | 134        |
| 7.1.4 函数原型说明 .....         | 136        |
| 7.2 函数之间的参数传递 .....        | 137        |
| 7.2.1 值传递 .....            | 137        |
| 7.2.2 地址传递 .....           | 138        |
| 7.3 函数的递归调用(递归函数) .....    | 145        |
| 7.4 函数参数缺省 .....           | 146        |
| 7.5 函数重载 .....             | 147        |
| 7.5.1 重载函数应满足的条件 .....     | 147        |
| 7.5.2 匹配重载函数的规则 .....      | 147        |
| 7.6 函数模板 .....             | 149        |
| 7.7 变量的作用域和存储类别 .....      | 151        |
| 7.7.1 局部变量和全局变量 .....      | 151        |
| 7.7.2 变量的存储类别 .....        | 152        |
| 本章小结 .....                 | 153        |
| 习 题 .....                  | 154        |
| <b>第 8 章 结构体和共用体</b> ..... | <b>160</b> |
| 学习导读 .....                 | 160        |
| 8.1 结构体 .....              | 160        |
| 8.1.1 结构体类型声明(定义) .....    | 160        |
| 8.1.2 结构体类型变量的定义 .....     | 161        |
| 8.1.3 结构体类型变量的引用 .....     | 163        |
| 8.1.4 结构体类型用作函数参数 .....    | 166        |
| 8.2 链 表 .....              | 167        |
| 8.2.1 链表的概念和基本结构 .....     | 167        |



|                            |            |
|----------------------------|------------|
| 8.2.2 动态开辟和释放存储单元 .....    | 168        |
| 8.2.3 动态链表 .....           | 170        |
| 8.3 共用体 .....              | 173        |
| 8.3.1 共用体类型声明(定义) .....    | 173        |
| 8.3.2 共用体类型变量的定义 .....     | 174        |
| 8.3.3 共用体类型变量的引用 .....     | 174        |
| 本章小结 .....                 | 175        |
| 习 题 .....                  | 176        |
| <b>第9章 面向对象的程序设计</b> ..... | <b>180</b> |
| 学习导读 .....                 | 180        |
| 9.1 面向对象程序设计的概述 .....      | 180        |
| 9.1.1 面向对象的基本概念 .....      | 181        |
| 9.1.2 面向对象程序设计的特点 .....    | 182        |
| 9.2 类和对象 .....             | 183        |
| 9.2.1 类的定义 .....           | 183        |
| 9.2.2 对象的定义 .....          | 187        |
| 9.2.3 构造函数 .....           | 190        |
| 9.2.4 析构函数 .....           | 194        |
| 9.2.5 常对象和常成员 .....        | 195        |
| 9.2.6 静态成员 .....           | 198        |
| 9.2.7 友元函数 .....           | 200        |
| 9.2.8 类模板及应用 .....         | 203        |
| 9.3 继承和派生 .....            | 204        |
| 9.3.1 基类和派生类 .....         | 205        |
| 9.3.2 单继承 .....            | 206        |
| 9.4 多态性 .....              | 214        |
| 9.4.1 多态性类型 .....          | 214        |
| 9.4.2 联 编 .....            | 215        |
| 9.4.3 运算符重载 .....          | 215        |
| 9.4.4 虚函数 .....            | 220        |
| 9.4.5 抽象类 .....            | 223        |
| 本章小结 .....                 | 225        |
| 习 题 .....                  | 225        |



|                             |     |
|-----------------------------|-----|
| <b>第 10 章 文 件</b> .....     | 231 |
| 学习导读 .....                  | 231 |
| 10.1 C 中的文件 .....           | 231 |
| 10.1.1 文件概述 .....           | 231 |
| 10.1.2 文件类型指针 .....         | 232 |
| 10.1.3 文件的基本操作 .....        | 233 |
| 10.1.4 顺序文件的读/写 .....       | 235 |
| 10.1.5 随机文件的读/写 .....       | 243 |
| 10.1.6 文件操作的错误检测 .....      | 245 |
| 10.2 C++ 中的文件 .....         | 246 |
| 10.2.1 文件的打开和关闭 .....       | 246 |
| 10.2.2 文本文件的读/写 .....       | 247 |
| 10.2.3 二进制文件的读/写 .....      | 249 |
| 本章小结 .....                  | 252 |
| 习 题 .....                   | 252 |
| <b>附录 A 实 验</b> .....       | 254 |
| 实验 1 C/C++ 语言编程环境 .....     | 254 |
| 实验 2 选择分支结构程序设计 .....       | 255 |
| 实验 3 循环结构程序设计 .....         | 257 |
| 实验 4 数 组 .....              | 258 |
| 实验 5 指 针 .....              | 260 |
| 实验 6 函 数 .....              | 261 |
| 实验 7 结构体 .....              | 262 |
| 实验 8 面向对象的程序设计 .....        | 263 |
| 实验 9 文件(C/C++ 输入/输出流) ..... | 264 |
| <b>附录 B 自测题</b> .....       | 265 |
| 自测题 1 .....                 | 265 |
| 自测题 2 .....                 | 271 |
| 自测题 3 .....                 | 277 |
| 自测题 1 参考答案 .....            | 283 |
| 自测题 2 参考答案 .....            | 285 |
| <b>附录 C 关键字索引</b> .....     | 287 |



|                             |     |
|-----------------------------|-----|
| 附录 D 常用字符与 ASCII 码对照表 ..... | 288 |
| 附录 E 运算符索引 .....            | 290 |
| 附录 F 常用 C 库函数 .....         | 291 |
| 附录 G 常见错误、警告信息表 .....       | 296 |
| 参考文献 .....                  | 297 |

# 第 1 章 C/C++ 语言基础知识

## 学习导读

### 主要内容

C 语言是一种计算机程序设计语言,它既具有高级语言的特点,又具有汇编语言的特点。C++ 语言(简称 C++)是在 C 语言基础上发展起来的面向对象程序设计的语言。本章主要介绍:程序设计,C/C++ 程序的基本概念,C/C++ 语言的发展、特点、程序结构、开发过程,程序设计所涉及的数据类型、常量、变量、运算符和表达式等。

### 学习目标

- 了解 C/C++ 语言的特点;
- 了解 C/C++ 程序的结构;
- 了解 C/C++ 程序的开发过程;
- 掌握各种数据类型的常量和变量;
- 掌握各种运算符和表达式。

### 重点与难点

重点:程序设计的基本结构、语法规则。

难点:问题的算法描述(流程图表示)。

## 1.1 程序设计

### 1.1.1 程序与计算机程序

#### 1. 程 序

通常,完成一项复杂的任务,需要进行一系列的具体工作,这些按一定的顺序安排的工作即操作序列,就称为程序。

程序主要用于描述完成某项功能所涉及的对象和动作规则。

某一个学校颁奖大会的程序如下:

- 宣布大会开始;
- 介绍出席大会的领导;
- 校长讲话;
- 宣布获奖名单;



- 颁奖；
- 获奖代表发言；
- 宣布大会结束。

## 2. 计算机程序

计算机程序是为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合(语句和指令)。

计算机程序分为两类：

- 系统程序(操作系统 OS、SQL Server 数据库管理系统等)；
- 应用程序(用汇编语言、高级语言编写的可执行文件)。

计算机程序的特性：

- 目的性(程序有明确的目的)；
- 分步性(程序由一系列计算机可执行的步骤组成)；
- 有序性(不可随意改变程序步骤的执行顺序)；
- 有限性(程序所包含的步骤是有限的)；
- 操作性(有意义的程序总是对某些对象进行操作)。

计算机程序可以用机器语言、汇编语言、高级语言来编写。

### 1.1.2 计算机程序设计语言

计算机程序设计语言,即程序设计语言,通常简称为编程语言,是一组用来定义计算机程序的语法规则。人与计算机通信也需要语言,为了使计算机进行各种工作,就需要有一套用以编写计算机程序的数字、字符和语法规则,由这些字符和语法规则组成计算机的各种指令(或各种语句),这些就是计算机能接受的语言。

程序设计语言有高级语言和低级语言之分,C/C++语言是高级语言,机器语言是低级语言,汇编语言基本上是低级语言。

程序设计语言分为三类:机器语言、汇编语言和高级语言(面向过程的语言、面向问题的语言、面向对象语言)。

#### 1. 机器语言

一个机器语言程序段如下：

```
00111110  
00011010  
11111110  
00100100  
11010011  
00101111  
01110110
```

优点:能被计算机直接识别和执行,执行速度快。



缺点:程序是0和1的二进制编码,可读性非常差,编程很不方便,指令记忆难,容易出错且不易修改。

## 2. 汇编语言

汇编语言是采用记忆符号来代替机器语言的二进制编码,如用记忆符 ADD 代替加法指令,OUT 代替输出指令等。

前述的机器语言程序段,改用汇编语言可写成:

```
LD    A,26
ADD   A,36
OUT   (48),A
HALT
```

说明:汇编语言需要“翻译”后才能在计算机上执行。

优点:相对机器语言,编程较为方便。

缺点:汇编语言仍脱离不开具体机器的指令系统,它所用的指令符号与机器指令基本上是一一对应的,编程效率不高,非专业编程人员很难使用。

## 3. 高级语言

高级语言与人类自然语言和数学式子相当接近,而且不依赖于某台机器,通用性好。

用 C++ 语言编写的简单程序段如下:

```
a = 26 + 36
cout << a
```

高级语言程序也必须经过“翻译”,即把人们用高级语言编写的程序(称为源程序)翻译成机器语言的程序(称为目标程序)后才能执行。

两种翻译方式:

- 编译方式(编译程序);
- 解释方式(解释程序)。

高级语言一般采用上述两种翻译方式,如图 1-1 所示。通常情况下,学习阶段采用解释方式,应用阶段采用编译方式。

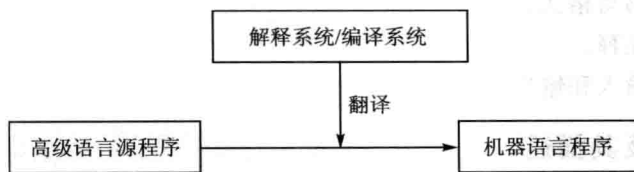


图 1-1 高级语言程序与机器语言程序转换图



### 1.1.3 计算机程序设计

#### 1. 程序设计

程序设计,即计算机程序设计,是根据系统设计文档中有关模块处理过程的描述,选择合适的程序语言,编制正确、清晰、鲁棒性强、易维护、易理解和高效率程序的过程。

#### 2. 程序设计原则

① 正确性,编制出来的程序能够严格按照规定的要求,准确无误地提供预期的全部信息。

② 可维护性,程序的应变能力强。程序执行过程中发现问题或客观条件变化时,调整和修改程序比较简便易行。

③ 可靠性,程序应具有较好的容错能力,程序不仅在正常情况下要能正确工作,而且在意外情况下,亦要能做出适当的处理,以免造成严重损失。尽管不能希望一个程序达到零缺陷,但它应当是十分可靠的。

④ 可理解性,指程序的内容清晰、明了,便于阅读和理解。对大型程序来说,要求它不仅逻辑上正确、能执行、而且应当层次清楚、简洁明了、便于阅读。

⑤ 高效率,程序的结构严谨,运算处理速度快,节省机时。程序和数据的存储、调用安排得当,节省空间,即系统运行时尽量占用较少空间,却能用较快速度完成规定功能。

#### 3. 程序设计方法

按程序开发路径有两种程序设计方法:

① 自顶向下的程序设计方法(从最顶层开始,直至实现最底层为止)。

② 自底向上的程序设计方法(从最底层开始,直至实现最顶层为止)。

#### 4. 程序设计的步骤

明确条件,分析数据,确定流程,编写程序,检查和调试,编写程序使用说明书是程序设计的主要步骤。

#### 5. 编程风格

① 标识符的命名。

② 程序的书写格式。

③ 程序的注释。

④ 程序的输入和输出。

### 1.1.4 算法及其描述

算法是学习程序设计的基础,掌握算法可以帮助读者快速理清程序设计的思路,找出问题的多种解决方法,从而选择最合适的解决方案。在程序设计中,构成算法的基本结构有 3 种:顺序结构、选择结构和循环结构。



## 1. 算 法

做任何事情都有一定的步骤,算法就是解决某个问题或处理某个事件的方法和步骤。人们使用计算机,就是利用计算机处理各种不同的问题,而要解决问题,必须事先对各类问题进行分析,确定采用的方法和步骤。此处所讲的算法是专指用计算机解决某一问题的方法和步骤。

### 2. 算法的特点

- ① 有穷性:算法必须能在有限的时间内完成问题的求解。
- ② 确定性:一个算法给出的每个计算步骤,必须是精确定义,无二义性。
- ③ 有效性:算法中的每一个步骤必须有效地执行,并能得到确定结果。
- ④ 可行性:设计的算法执行后必须有一个或多个输出结果,否则是无意义的、不可行的。

### 3. 算法设计的基本方法

算法设计的基本方法有列举法、归纳法、递推法、递归法、减半递推技术和回溯法。

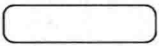
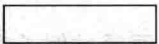
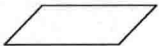
### 4. 算法复杂度

- ① 算法的时间复杂度:执行算法所需要的计算工作量(算法执行的基本运算次数)。
- ② 算法的空间复杂度:执行算法所需要的内存空间(算法程序所占空间、输入初始数据所占空间和算法执行过程中所需额外空间)。

### 5. 算法的描述方法

- ① 自然语言:日常使用的语言描述方法和步骤。其通俗易懂,但比较繁琐、冗长,并且对程序流向等描述不明了、不直观。
- ② 传统流程图:通过图形描述,具有逻辑清楚、直观形象、易于理解等特点。传统流程图的基本流程图符号及说明如表1-1所列。

表 1-1 流程图符号及说明

| 图形符号                                                                                | 名 称    | 说 明                 |
|-------------------------------------------------------------------------------------|--------|---------------------|
|  | 起止框    | 算法流程的开始和结束          |
|  | 处理框    | 完成某种操作(初始化或运算赋值等)   |
|  | 判断框    | 判断选择,根据条件满足与否选择不同路径 |
|  | 输入/输出框 | 数据的输入/输出操作          |
|  | 流程线    | 程序执行的流向             |
|  | 连接点    | 流程分支的连接             |



③ N-S 结构化流程图：将传统流程图中的流程线去掉，把全部算法写在一个矩形框内，有利于程序设计的结构化。

**注意：**当程序算法比较繁琐时，一般采用 N-S 结构化流程图，但对初学者和编写不复杂的较小程序时，建议使用传统流程图来描述算法。

## 1.2 C 语言和面向过程的程序设计

语言，是人与人进行交流沟通的工具。人与计算机通信也需要语言，为了使计算机进行各种工作，就需要有一套用以编写计算机程序的数字、字符和语法规则，由这些字符和语法规则组成计算机的各种指令（或各种语句），这些就是计算机所能接受的语言，将其称之为计算机语言。计算机语言有高级语言和低级语言之分，C/C++ 语言是高级语言，机器语言是低级语言，汇编语言基本上是低级语言。

### 1.2.1 C 语言简介

C 语言是由美国贝尔研究所的 D. M. Ritchie 于 1972 年推出。1977 年出现了《可移植 C 语言编译程序》，推动了 UNIX 在各种机器上的实现，C 语言也得到推广。1978 后，先后被移植到大、中、小及微型机上。1983 年，美国国家标准化协会（ANSI）根据 C 语言各种版本对 C 的发展和扩充，制定了新的标准 ANSI C，其比标准 C 有了很大的发展。目前，流行的 C 语言编译系统大多是以 ANSI C 为基础进行开发的。

C 语言可以作为工作系统设计语言，编写系统应用程序，也可以作为应用程序设计语言，编写不依赖计算机硬件的应用程序。它的应用范围广泛，具备很强的数据处理能力，不仅仅是在软件开发上，而且各类科研都需要用到 C 语言，适于编写系统软件、三维、二维图形和动画。具体应用如单片机及嵌入式系统开发。

C 语言是目前世界上广泛使用的高级语言，其结构紧凑、语言简洁，只有 32 个关键字，9 种控制语句；使用方便灵活，书写形式自由；数据类型完备，运算符丰富；允许直接访问物理地址，对硬件进行操作；生成目标代码质量高，程序执行效率高；可用于各种型号的计算机和操作系统。

### 1.2.2 C 程序的结构

用 C 语言编写的程序称为 C 语言源程序，简称 C 程序。下面通过例子来了解 C 程序的基本组成结构及其书写风格。程序中涉及的语法规则将在后续章节中进行介绍。

**【例 1-1】** 最简单的 C 程序。

```
/* This is the first C program */  
#include <stdio.h>           //C 标准输入/输出头文件  
void main ( )
```