(中国台湾)荣钦科技
杨汉玮
飞思科技产品研发中心
著
改编
监制

C/C++ 开发专家

专业人士
权威经典

C语言
开发入门与编程实践

5 STAR

理论实践并重，观念应用并行
大量完整范例，程序无痛学习
章末综合范例，活用程序语法
精选习题评测，提升学习动力

本书实例源文件请到
http://www.fecit.com.cn的
"下载专区"进行下载。

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

(中国台湾)荣钦科技 著
杨汉玮 改编
飞思科技产品研发中心 监制

ZCT

C/C++ 开发专家

C语言开发入门与编程实践

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内容简介

本书共分 11 章，循序渐进地介绍 C 语言程序设计的基本概念、基本数据、输入/输出、程序的流程控制、数组与字符串、指针、函数与预处理、结构等自定义数据类型、文件操作等内容，并在最后介绍 C 语言的常用函数库。同时在附录中介绍 Dev C++、Visual C++、C++ Builder Personal 6.0 的安装与使用，以及 C++语言的特性。书中融入大量的程序范例，并提供许多程序设计与调试的相关经验，因此，本书非常适合作为学习程序语言的教材。在每章最后一节，针对该章的语法及程序设计技巧，安排许多实用综合程序范例，以期能多给学习者更多的实践经验。

本书实例源代码可在飞思下载专区下载，以供读者参考。

本书可作为各级学校和培训机构的教材或参考书，同样，程序设计自学者或是硬件工程师也可以用其来查阅相关知识点或作为参考资料。

版权贸易合同登记号 图字：01-2007-4935

图书在版编目（CIP）数据

C 语言开发入门与编程实践 /（中国台湾）荣钦科技著；杨汉玮改编. —北京：电子工业出版社，2007.12
（C/C++开发专家）
ISBN 978-7-121-05255-2

Ⅰ. C… Ⅱ.①荣…②杨… Ⅲ.C 语言－程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2007）第 165675 号

责任编辑：王树伟　田　蕾
印　　刷：北京智力达印刷有限公司
装　　订：北京中新伟业印刷有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：787×1092　1/16　印张：30.75　字数：787.2 千字
印　　次：2007 年 12 月第 1 次印刷
印　　数：6 000 册　定价：48.80 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

出版说明

“聪明的人使用 Delphi，真正的程序员使用 C++。”
时至今日，这句曾经在业内广为流行的话语又增添了更丰富的内涵。
脚本语言、Java、.NET 等正在争夺更大的天地。
然而，C/C++仍不失为最好、最纯粹的编程语言。

—— “C/C++开发专家”引导你成为真正的程序员

C/C++的发展

作为一种结构化的中高级编程语言，C 语言具有功能齐全、适用范围广的优势，一直为很多程序员所钟爱，并被视为最佳的编程入门语言，拥有着庞大的使用和学习人群。C++是在 C 语言基础上开发的一种集面向对象编程、通用编程和传统的过程化编程于一体的编程语言，是目前业界广泛使用的一种编程语言。然而，软件产业的规模和环境发展到今天，已经发生了深刻的变化。如今企业级应用整合与开发的任务主要由 Java、基于.NET 平台的 C#及各种新型动态语言来承担。C++的应用场合有所收缩，不再像之前那样从上到下包打天下，呈现出鲜明的行业应用特色。未来 C++主要在系统级复杂应用程序，高性能、实时中间件和嵌入式领域发挥所长。随着多核 CPU 的普及和网络安全重要性的空前提升，在并发程序设计、系统安全及视频处理、嵌入式开发方面，C++将获得新的应用空间。在大规模、高性能计算，游戏开发、嵌入式实时应用开发方面，以及一些传统的客户端软件和构件开发中，C++也将继续保持其稳定的地位。

C/C++的图书现状

C++的教学和使用具有其复杂性，而传统图书和学习方法的各种弊端更加剧了这一现象，使 C++成为不少人望而生畏的难学、难用的“专家语言”。虽然国内的 C/C++图书并不缺乏，但大多只适合有一定经验的程序员提升功力之用，而内容全面准确、讲解循序渐进、学习简明易懂的原创图书并不多见。近期 C/C++图书市场存在如下特点：

1．国外经典图书全面翻新。近年来国外一些书商根据 C++所发生的变化，不断地进行版本升级或全面改写书稿，推出新的力作。

2．国内原创图书缺乏力作。近年来国内虽然有一批令人耳目一新的 C++好书面世，但在技术层面上对实践的关注略显不足，难解读者之渴。

3．关键性图书存在空白。基于组件的软件开发、复杂网络应用，以及热度尚在的 COM 开发等方面的图书有待开发。

基于上述现状，我们组织 C/C++各应用领域的作者，推出本丛书“C/C++开发专家”，力求从新的、实用的、全面的角度介绍 C/C++，使其紧密地跟踪当前国内最实用、最热门的编程技术。我们期望通过这套丛书，能够提高各位读者的 C/C++开发水平及编程的实践能力，为我国计算机产业奉献一份微薄之力。

“C/C++开发专家”助你成为真正的程序员

“C/C++开发专家”的读者定位是：C/C++初学者，需要提升应用开发能力的程序员，具有实际开发经验的中高级程序员。对阅读本丛书的读者建议如下：

➢ *面向 C/C++初学者*

本丛书通俗易懂，并自成体系。丛书全面介绍 C/C++及 Visual C++的编程技术和实践操作。通过学习，初学者可快速地掌握涉及 OOP、STL、泛型编程等标准 C/C++的内容，对 C/C++技术应用有更深刻的理解。

➢ *面向需要提升应用开发能力的程序员*

对于那些急需提升应用开发能力的程序员来说，本丛书是再好不过的专家向导。丛书除全面介绍标准 C/C++的内容外，还涉及数字图像处理、流媒体、网络通信和嵌入式开发等多个领域，可以为从事相关领域开发的程序员提供有益的帮助和参考。

➢ 面向具有实际开发经验的中高级程序员

本丛书同样适合于具有实际开发经验的中高级程序员。书中列举的大部分实例具体翔实，非常值得广大高级程序员学习和借鉴。

“C/C++开发专家”为程序员量身打造

本套丛书通过不同种类的图书来满足读者的需求。

➢ 语言入门

C/C++是一门优秀的高级语言。它绝不像一些传统图书所述是一门晦涩难懂、高深莫测的“专家语言”。本丛书的语言入门分支面向初学者，以通俗易懂的语言，介绍标准的C/C++语言知识，以及Visual C++编程技术；在保证知识体系的完整性的同时，在语言、体例上更贴近程序员的学习心理需求。

➢ 应用实践

如果脱离了具体的应用背景，任何一门计算机语言的学习都是“纸上谈兵”。如果程序员没有真正掌握面向应用的实践开发技能，那么很有可能面临来自就业的压力。本丛书的应用实践分支面向数字图像处理、流媒体、网络通信、嵌入式开发等不同的行业应用方向，介绍C/C++应用技术。目标是努力将读者培养成具有实际开发能力的从业人员。

➢ 开发详解

只让人阅读一遍的书很难说是一本好书。任何一本书在读者的眼中总会经历“厚→薄→厚”的过程。同样，C/C++语言会耐人寻味，但真正理解C/C++一般性内容需要花时间，而要做到融会贯通则更要下工夫。本丛书的开发详解分支针对C/C++语言及Visual C++中的高级特性，进行深入的剖析和讲解。C/C++程序员一旦掌握更高级的编程技巧，且对C/C++的语言内涵及开发技术有更为深入的理解，就能得心应手地运用这门语言。

➢ 技巧集锦

从大规模的并行计算到嵌入式系统开发，C/C++的应用领域非常广泛。即便是世界上最厚的一本书，也无法介绍所有的C/C++技术。针对这一特点，本丛书的技巧集锦分支对程序员经常遇到的问题进行解答和分析，并注重举一反三，启发读者思考。通过对这一话题的讨论，给正在从事或即将从事C/C++开发的程序员以最大的启迪。

“C/C++开发专家”丛书特色

本丛书具有如下特色。

➢ 由浅入深，通俗易懂

实际上，根本就不存在只面向纯粹的初学者的C/C++书籍。原因很简单：C/C++就不是初级的语言。初学者选择C/C++的时候，除了有足够的兴趣之外，还要有足够的耐心和恒心。为此，本丛书在保持完整性的同时注重语言的通俗性和知识的趣味性，避免了较为复杂的理论概念，取而代之的是常见的编程技巧和实际例子，力求由浅入深，通俗易懂，充分调动读者的阅读兴趣。

➢ 案例为主，内容生动

如果没有“案例”，C/C++的学习可能非常枯燥无趣；如果没有合适、有趣的“案例”， C/C++的学习仍会枯燥无趣。与以往的风格不同，本丛书强调编程实践，提供了大量的实例及源代码。这些案例均由作者从实际开发工作中设计的原型案例精简加工而成，形式丰富多样，具有很好的实用价值。

➢ 倡导正确的编程思想

“授之以鱼，不如授之以渔。”本丛书并非按部就班地完成知识传授，而是在介绍知识的同时倡导正确的学习思想和方法。如：倡导OOP思想、泛型编程、流行的设计模式、不断的重构理念和开源精神等。读者在阅读本书的同时，会接触到这些新的理念和方法。在某些开放性话题上，本丛书一反以往一些图书的“专家”面孔，更加贴近读者，从各个角度与读者展开交流和探讨。

飞思科技产品研发中心

序

C 语言称得上是一种历史悠久的程序语言，也往往是当今初学者最先接触的程序语言。相对于日趋复杂的各种程序语言，C 语言能以简洁的语法写出功能强大的程序。C 语言的地位持续屹立已达 30 多年，无论是后来的 C++、Java、PHP，甚至.NET 中的 C#和 Visual Basic.NET 等，都以 C 语言作为基础。C 语言中的结构化程序设计语法、函数的观念和用户自定义类型等，都是后来的程序语言参考的依据。学习完 C 语言，将来学习任何一个程序语言，都可快速上手。因此，C 语言称得上是近数十年来科技界最受欢迎的程序语言。

C 语言具备高级语言的结构化语法，也有汇编语言的高效率表现，并且拥有高度的可移植性与强大的数据处理能力。同时，C 语言兼具对内存与硬件控制的管理，可以作为结合软件设计与硬件控制的语言。正因如此，无论是程序设计师或是硬件工程师都必须学习 C 语言。

本书翔实地说明了 C 语言相关的语法，书中融入大量的程序范例，并提供许多程序设计与调试的相关经验，因此，本书非常适合作为学习程序语言的教材。除了上述特点外，本书另外一个重要特色就是，在每章的最后一节，针对该章的语法及程序设计技巧，安排许多实用综合程序范例，希望能提供给读者更多的实践经验。

另外，书中安排了大量的程序实战习题，不仅是为了让读者活用书中的语法，还可以作为程序编写磨练的机会及教师验收学生掌握情况的方法。除了教学上的考虑外，本书完全以经验为出发点，目的是希望学习程序语言的每个学生或读者，能快速进入 C 语言程序设计的领域。笔者深信，这绝对是一本学习 C 程序语言的实用教材。虽然本书校稿过程力求无误，但时间、精力有限，书中难免有疏漏之处，还望各位不吝指教！

著　者

联系方式

咨询电话：（010）68134545　　88254160

电子邮件：support@fecit.com.cn

服务网址：http://www.fecit.com.cn　　http://www.fecit.net

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

目　录

第1章

C语言与程序设计简介

C语言称得上是一种历史悠久的程序语言，也往往是当今初学者最先接触的程序语言。就运算速度而言，C语言是介于高级语言与低级语言之间的中级语言(middle-level language)。也就是说，C语言在汇编语言与高级语言间取得了一个平衡的位置，有高级语言的结构化语法也有汇编语言的高效率表现，并且具备了高度的可移植性与强大的数据处理功能。所以，它称得上是近十年来科技界广受欢迎的程序语言。

在本章中，将为您深入浅出地介绍程序设计的基本概念与C语言的相关信息，并且开始编写第一个C语言程序。“好的开始，就是成功的一半”，相信各位在正式进入C语言的程序设计领域时，能奠定一个良好的基础。

本章学习地图

- 程序语言的演进与分类
- 算法与程序设计实例
- C语言的特性与开发环境
- 第一个C语言程序

1.1 认识程序语言

“程序语言”，就是一种人类用来和计算机沟通的语言，也是用来指挥计算机运算或工作的指令集合。程序语言发展的历史已有半世纪之久，由早期的机器语言发展至今，已经迈入到第五代自然语言了。

每一代的语言都有不同的特色，并且一直朝着容易使用、除错与维护功能更强大的目标来发展。基本上，任何一种语言都有其专有语法、特性、优点及相关应用的领域。程序语言的演进过程说明如图 1-1 所示。

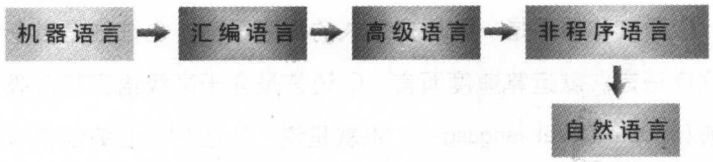

图 1-1

1.1.1 机器语言

机器语言（Machine Language）是早期的程序语言，由 1 和 0 两种符号构成，是计算机能够直接阅读与执行的基础语言，任何程序在执行前都必须被转换为机器语言。如“10111001”代表“设置变量 A”，而“00000010”代表“数值 2”。当指示计算机将变量 A 设置为数值 2 时，机器语言写法为：

```
10111001（设置变量A）
00000010（将A设置为数值2）
```

对于每一家计算机制造商而言，往往因为计算机硬件设计的不同而开发不同的机器语言。这样不但使用起来不方便、可读性低、不容易维护，并且不同的机器平台，编码方式都不尽相同。

1.1.2 汇编语言

汇编语言（Assembly Language）是一种介于高级语言及机器语言间的符号语言，与机器语言相比，汇编语言比较容易编写和学习，它将机器语言 0 和 1 的符号定义为由操作数和运算码组合而成的“指令”（Statement），只可以在特定的机型上执行，不同的 CPU 要使用不同的汇编语言。

例如，MOV 指令代表设置变量内容、ADD 指令代表加法运算和 SUB 指令代表减法运算，如下所示：

```
MOV  A , 2   (变量A的数值内容为2)
ADD  A , 2   (将变量A加上2后，将结果再存回变量A中，如A=A+2)
SUB  A , 2   (将变量A减掉2后，将结果再存回变量A中，如A=A-2)
```

汇编语言虽然比机器语言更符合人类需求，但计算机却无法直接识别。必须经过所谓的汇编器（Assembler），将汇编语言转换成机器语言，转换成的机器语言会形成一个文件，称为“可执行文件”或目标程序（Object Program），才可以在计算机上执行。

不同的 CPU 也会有不同的汇编器，常用的汇编器有 ASM、MASM 和 TASM，其中以微软推出的宏汇编器（Macro Assembler）MASM 最受欢迎。 每一种系统的汇编语言都不一样，以 PC 而言，使用的是 80x86 的汇编语言。

1.1.3 高级语言

高级语言（High-level Language） 是最接近于人类使用语言的程序语言，虽然执行较慢，但语言本身易学易用，因此被广泛应用在商业、科学、教学和军事等相关的软件开发上。它的特点是必须经过编译或解释的过程，才能转换成机器语言码。高级语言又根据转换过程可区分为以下两种。

1）编译式语言

编译式语言是一种使用编译器（Compiler）将程序代码编译为目标程序的语言。编译器可将源程序转换为机器可读的可执行文件。经过编译后，会产生“目标文件”（.obj）和“可执行文件”（.exe）两个文件其流程如图 1-2 所示。

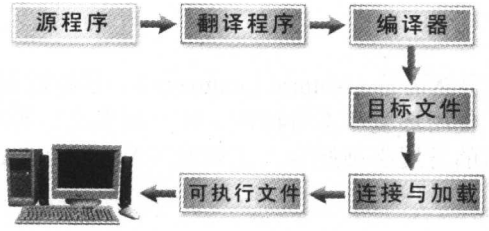

图 1-2

源程序每修改一次，就必须重新编译。经过编译后所产生的执行文件可直接对应成机器码，即可在计算机上直接执行，不需要每次执行都重新编译，执行速度自然较快。例如，C、C++、Visual C++和 Fortran 语言都属于编译式语言。

2）解释式语言

解释式语言就是利用解释器（Interpreter）对高级语言的源程序代码做逐行解释，每解释完一行程序代码后，再解释下一行。解释的过程中如果发生错误，则解释会立刻停止，其流程如图 1-3 所示。

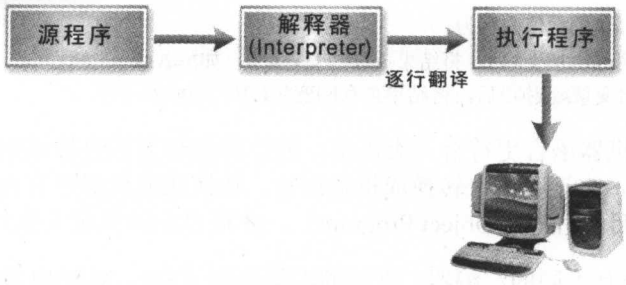

图 1-3

由于使用解释器编译的程序每次执行时都必须再解释一次，所以执行速度较慢，不过因为仅需存取源程序，不需要再转换为其他类型的文件，因此所占用的空间较少。例如，BASIC、LISP 和 Prolog 等语言都属于解释式语言。

1.1.4 非程序性语言

“非程序性语言”（Non-procedural Language）也称为第四代语言，特点是它的指令和程序真正的执行步骤没有关联性。程序设计者只需将自己打算做什么表示出来即可，而不需要去理解计算机的执行过程。例如，数据库的结构化查询语言（Structural Query Language，SQL）就是一种第四代语言。

1.1.5 人工智能语言

称为第五代语言，或是自然语言（Natural Language），其特性是语法类似一般人的对话。因为自然语言在用户口音、使用环境和本身特性（如一词多义）等方面都会造成计算机在解读时的难度，因此自然语言的发展必须配合人工智能（Artificial Intelligence，AI）理论的发展来进行。

1.2 程序设计简介

不懂计算机的人可能会把“程序”（Program）想象得十分深奥难懂，其实“程序”只是一堆合乎语法规则的指令（Statement）的集合。而“程序设计”（Programming）是通过程序编写与执行来达到用户的工作需求。

或许有人认为程序设计的主要目的是要“运行”出正确的结果，而忽略了包括执行效率与日后维护成本，其实这是不清楚程序设计的真正意义。事实上，程序设计的目的不只是讨论编写程序的能力而已，而是期望能够组织众多程序设计师来共同设计一套大型且符合用户需求的复杂程序系统。

1.2.1 算法

在程序设计领域中，除了执行效率与速度为衡量标准外，对于程序能否清楚而精确地解决问题，往往取决于算法。到底什么是“算法”呢？在韦氏辞典中定义为：在有限步骤内解决数学问题的程序。如果应用在计算机领域中，可以把算法定义为：为了解决某一个工作或问题，所需要有限次数的机械性或重复性指令与计算步骤。

算法不仅可用来解决计算机或数学问题，还可以用来描述日常生活的许多工作，例如，员工的工作报告、宠物的饲养过程和学生课程表等。算法具备如图 1-4 所示的 5 种特性，其内容与说明如表 1-1 所示。

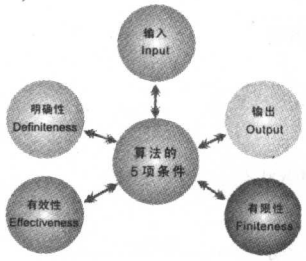

图 1-4

表 1-1

算法特性	内容与说明
输入（Input）	0 个或多个输入数据，这些输入必须有清楚的描述或定义
输出（Output）	至少会有一个输出结果，不可以没有输出结果
明确性（Definiteness）	每一个指令或步骤必须是简洁明确而不含糊的
有限性（Finiteness）	在有限步骤后一定会结束，不会产生无限循环
有效性（Effectiveness）	步骤清楚且可行，能让用户用纸笔计算而求出答案

当我们认识了算法的定义与特性后，到底该用什么方法来表达算法最为合适呢？算法的主要目标在于提供人们了解执行的工作流程与步骤，只要能清楚表现算法的 5 项特性即可。常用算法介绍如下。

- 一般文字叙述：中文、英文和数字等。文字叙述法的特色在于使用文字或语言叙述来说明演算步骤。
- 伪语言（Pseudo-Language）：接近高级程序语言的写法，也是一种计算机不能直接执行的语言。一般都需要一种特定的预处理器（Preprocessor），或者用手写转换成真正的计算机语言，经常使用的有 SPARKS、PASCAL-LIKE 等语言。例如，以下是使用 SPARKS 写成的链接串行反转算法：

```
Procedure Invert(x)
P←x;Q←Nil;
WHILE P≠NIL do
      r←q;q←p;
      p←LINK(p);
      LINK(q)←r;
END
      x←q;
END
```

- 表格或图形：如数组、树形图和矩阵图等，如图 1-5 所示。

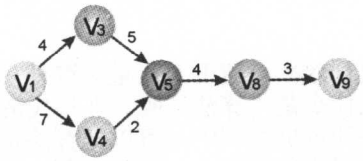

图 1-5

- 流程图：流程图（Flow Diagram）是一种通用的表示法，也有图形符号。例如，请您输入一个数值，并判别是奇数或偶数，如图 1-6 所示。

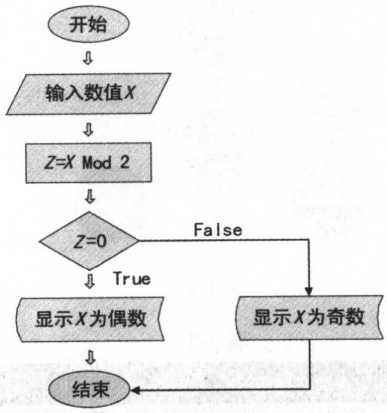

图 1-6

- 程序语言：算法也能够直接以可读性高的高级语言来表示，如 Visual Basic 语言、C 语言、C++语言和 Java 语言。

1.2.2 程序语言的选择

对于一个专业程序设计师来说，其主要目标是将精力集中在系统功能的提升，并且以更少的时间，开发出更高质量的应用程序。目前开发程序所使用的语言种类繁多，通常可根据主客观需要来考虑，并无特别规定。一般评断程序语言的 4 项标准如下（见图 1-7）。

图 1-7

- 可读性（Readability）：阅读与理解都相当容易。
- 平均成本：成本考虑不局限于编码成本，还包括了执行、编译、维护、学习、除错与日后更新等成本。
- 稳定性：所编写出来的程序代码稳定性高，不易产生副作用（Side Effect）。
- 可编写性：可针对需求编写出相对容易的程序代码。

1.2.3 程序设计流程

从程序设计的努力方向来看，无疑是以效率高、可读性好的程序设计成果为目标。一个程序产生的过程，可分为如图 1-8 所示的 5 个设计步骤。

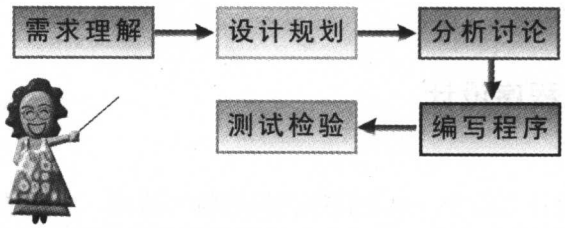

图 1-8

- 需求理解：了解程序所要解决的问题，并搜集所要提供的输入信息与期望得到的输出结果；
- 设计规划：根据需求，选择适合的数据结构，并尝试描述一个算法来解决问题；
- 分析讨论：思考其他可能适合的算法及数据结构，最后再选出最适当的一种；
- 编写程序：将分析的结论，利用程序语言写成初步的程序代码；
- 测试检验：最后必须确认程序的输出是否符合需求，该步骤需要逐步地执行程序并进行大量的测试与调试。

1.2.4 程序代码编写原则

事实上，程序代码编写的最重要原则，不只是急着达成所需的执行结果，还要考虑日后维护与修改的问题，因此程序代码的可读性就显得尤其重要。

无论是选择哪种程序语言，在一般的小程序中可能还感觉不到养成良好编程习惯的重要性，但是当程序规模越来越大，或者由团队开发程序项目时，良好的编写原则绝对有助于自己或他人来了解所编写程序的框架与内涵。以下 3 点是各位在编写时应该注意的基本原则。

1）适当的缩进

缩进的主要用途是用来区分程序的层次，使程序代码易于阅读，像在主程序中包含子区