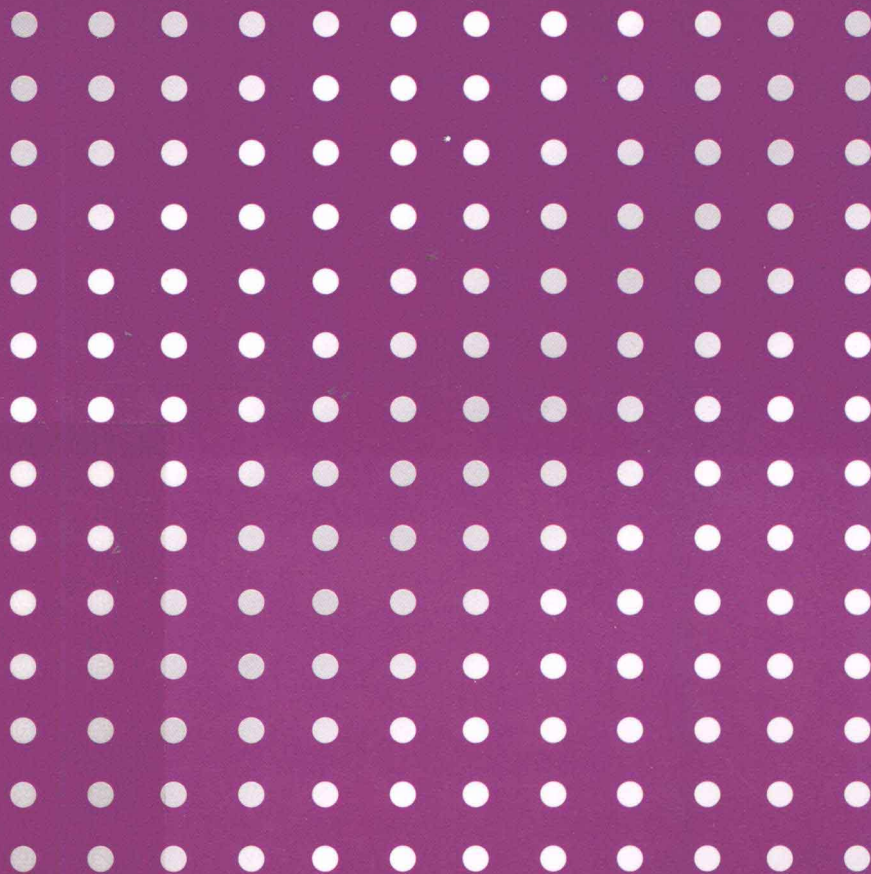


高等院校信息技术规划教材

程序设计实践教程

章小莉 周知杨 张岩 赵耿 编著



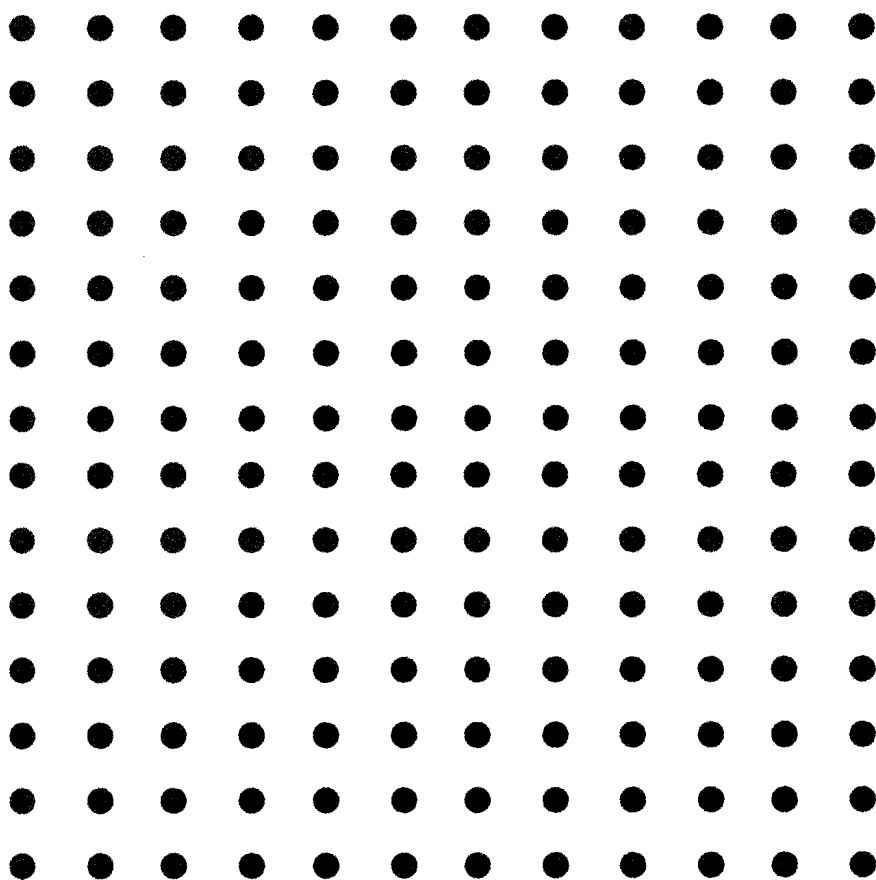
清华大学出版社



高等院校信息技术规划教材

程序设计实践教程

章小莉 周知杨 张岩 赵耿 编著



清华大学出版社
北京

内 容 简 介

本书从实用性和趣味性出发,分模块进行基础知识、信息管理与存储、字符串处理、绘图、贴图等编程训练,最终通过棋牌类游戏开发,实现模块知识综合应用的训练。

本书提供了160道描述详细、功能要求具体、开发目标明确的训练任务,意在对学生进行上千行代码编写与调试的训练,帮助他们感悟编程之道,发现编程乐趣,激发创新意识。

本书可以作为高等院校计算机类专业以及相关专业学生的教材,也可供C语言编程人员参考。

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

程序设计实践教程/章小莉等编著. --北京:清华大学出版社,2012.10

高等院校信息技术规划教材

ISBN 978-7-302-29395-8

I. ①程… II. ①章… 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(2012)第159739号

责任编辑:白立军 顾 冰

封面设计:常雪影

责任校对:时翠兰

责任印制:何 芊

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:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:26.5

字 数:611千字

版 次:2012年10月第1版

印 次:2012年10月第1次印刷

印 数:1~3000

定 价:45.00元

产品编号:045049-01

本书是 C 语言入门之后的强化训练教材,目的是帮助实现编写应用程序的梦想。教材对学过标准 C 语言的学生进行编码强化训练,提高编写和调试代码能力,学会求解实际问题、管理事务信息、设计和开发字符串处理函数。另外,本书第 4、5 章补充 MFC 的画图 and 贴图知识,第 6 章介绍棋牌类游戏开发,培养学生开发 Windows 棋牌类游戏和简单画图应用程序的能力,增加编程乐趣,激发编程兴趣。

本书共分 6 章,每章一个专题。

第 1 章 基础训练。总结标准 C 语言知识以及重点和难点,详述调试程序技巧,简介程序代码测试方法,提供求解简单实际问题的训练,巩固所学 C 语言知识。

第 2 章 信息管理与存储训练。温习文件知识,讲述文件应用难点,举例说明文件实现信息管理与存储的方法,使得在信息管理与存储训练中,克服对文件的恐惧,掌握操作文件的技能。

第 3 章 字符串处理训练。总结常用 C 语言字符串函数,详细介绍自定义字符串函数的设计、实现和测试方法,比较不同方法定义的字符串类型数据在使用中的差别,提供大量字符串实训任务,提高解决字符串相关问题的能力。

第 4 章 绘图技术训练。学习 MFC 的画图和文字显示技术,进行画图训练,为开发界面漂亮的 Windows 应用程序打基础,学会开发简单画图软件。

第 5 章 粘贴图片技术训练。了解 Windows 程序中静态和动态图片操作区别,学习 MFC 程序的粘贴图片和操作图片技术,在强化训练中,学会在程序中运用图片资源的能力。

第 6 章 综合训练。了解游戏程序内部控制方法,初步掌握基于 MFC 单文档框架的游戏程序的设计与开发,能够在游戏程序实现过程中合理地选择数据结构并设计算法,综合运用前 5 章知识,完成棋牌类游戏的制作。

每章末专题训练任务提供详细功能描述,利于专注代码编写与调试,快速提高编码能力。

附录中提供了 VC++ 6.0 基本操作说明,给出程序设计实验报告框架,以及程序设计实验报告样本,帮助学会撰写实验设计报告。

本书既可以独立使用,也可以与“C 语言程序设计”课程教学同步使用。

独立使用时,建议在学完“C 语言程序设计”课程之后,按本书划分的训练模块,让每个学生在完成 1~5 章模块训练后,再进行第 6 章综合训练,获得上千行代码编写的强化练习,确实提高程序设计能力,激发开发软件的兴趣。

与“C 语言程序设计”课程教学同步使用时,第 1~3 章可以作为 C 语言学习的实验指导书,第 4、5 章拓展程序开发知识,第 6 章综合训练,共同提高 Windows 应用程序开发能力,培养编程创新活力。

在本书编写中,周知杨编写第 1 章,章小莉编写第 2、4、5 章,张岩编写第 3、6 章,宋燕红编写第 1 章部分训练任务,徐小青编写第 2 章部分训练任务,刘翔威实现第 6 章中全部代码,赵耿审校全书。此外还得到谢婷婷、赵越、马晓阳、刘瑾、张克君、李晓东、庞岩梅等人大力帮助,在此深表感谢。虽然本书内容集编者多年工作经验及实践感悟,但是由于提笔时间匆匆,难免有错误之处,敬请大家批评指正。

技术在不断更新,教学过程在不断发展,每个教与学的人都会有很多自己的体会,敬请读者不吝赐教,以利于我们不断改进自己的工作,与大家共同进步。

致教师

今天的学生面临求职的强烈压力,学校注重培养动手能力,能极大提高他们的竞争力。本书梳理了C语言知识重点、难点,补充了绘图、贴图知识,给出160多个开发任务,营造开发氛围,为吸引学生参与编程做好准备,为培养学生的竞争力提供训练。

本书所列训练任务,有具体描述,详细功能要求,必要的设计提示。任务分模块按难易等级划分,使全体学生都有机会参与训练。

在知识方面:本书提供模块训练所需教学课件,书中例题源代码,便于教师集中进行教学辅导。

在开发任务方面:本书分5个基础模块(基础知识,信息管理与存储,字符串处理,绘图,贴图),一个综合训练(棋牌类游戏设计与开发),使学生可以经过模块练习,能力拓展,走向综合运用。

在考虑学生个性方面:每个模块训练分3到4个难易等级的开发任务,尊重学生的个性不同,提供人人参与的机会。

在实验报告方面:提供报告框架以及实验报告样本,使学生学会撰写程序开发实验报告,学会规范开发软件,并总结开发经验。

在管理方面:附有实验管理表格和评分建议,便于教师记录学生实验过程中的表现,对实验进行过程化管理。

在成绩统计方面:给出总分合计建议性公式,方便做好全方位评价学生实验的能力。

建议教师选用本书时,尽量安排一段相对集中的时间开展实验,依据本校学生的情况,把实验内容分为必做和选做两个部分,允许每个学生按能力选择开发任务,积极鼓励学生挑战难题,使学生获得上千行代码编写与调试经验,在“魔鬼”训练中“新生”。为防止学生选题过于集中,可以考虑基于学生(C语言)程序设计学习成绩分配设计任务,从而保证学习效果。

每位教师都有自己的不少经验与创意设想,敬请赐教,让我们共同进步。

致学生

学完《程序设计基础》(C语言版),很想自己动手编写一个 Windows 应用程序吧,但是真正动手,发现前面道路还很漫长,需要继续学习 Windows API 或 Windows MFC,面向对象设计思想和编程技术,于是乎,热情在漫漫长路上消失殆尽。

这本书,就是想创造编程强化训练中的乐趣,力图在学习过标准 C 语言知识之后,带领大家求解实际问题,并直接走进 VC++ 6.0 的 MFC 编程世界,基于 MFC 框架进行画图、贴图、棋牌类游戏开发,在快乐中迈入编程自由王国,激发创新思维,成为软件开发后备军。

如果 C 语言学得不太好,第 1 章帮你温习重点,解答难点,掌握基本问题求解技巧,熟练调试程序技能。第 2、3 章,对文件使用、字符串处理进行训练,找到编程感觉。

如果已经掌握 C 语言基本知识,第 4、5 章,教会 VC 画图与贴图技术,提高创作精美程序界面的能力。

第 6 章,一个棋牌类游戏的设计与开发,带给你成就感。

挽起你的袖子,走入编程训练吧,挑战与乐趣同在,创意与刻苦相伴。

目录

Contents

第 1 章 基础训练	1
1.1 C 语言知识总结	1
1.1.1 C 语言知识点归纳	1
1.1.2 难点回顾与使用注意事项	19
1.2 算法设计	30
1.3 VC 程序调试技术	36
1.3.1 Visual C++ 调试工具	36
1.3.2 调试过程中的跟踪与监测	39
1.3.3 断点设置和使用	50
1.4 基础训练举例	57
1.4.1 基本知识应用	57
1.4.2 数组应用	61
1.4.3 指针使用	68
1.4.4 结构体应用	76
1.5 程序测试	82
1.5.1 单元测试	83
1.5.2 集成测试	86
1.6 基础训练注意事项	86
1.7 基础训练任务	87
本章参考文献	98
第 2 章 信息管理与存储训练	99
2.1 文件	99
2.1.1 文件中的数据	99
2.1.2 文件分类	100
2.2 C 语言文件知识小结	102
2.2.1 标准 C 语言文件处理函数	102

2.2.2 文件处理流程	105
2.3 文件函数使用举例	105
2.4 信息管理程序开发	117
2.4.1 程序设计与实现	117
2.4.2 程序测试	121
2.5 文件使用注意事项	122
2.5.1 不同类型文件比较	122
2.5.2 常见错误及原因	124
2.6 信息管理与存储训练任务	124
2.6.1 信息管理训练任务	124
2.6.2 信息存储训练任务	141
本章参考文献	144
第3章 字符串处理训练	145
3.1 字符串函数及函数设计	145
3.1.1 C语言字符串函数	145
3.1.2 字符串处理函数设计	155
3.2 字符串处理举例	168
3.3 VC中三种字符串定义与使用注意事项	179
3.3.1 字符数组和字符指针	180
3.3.2 String类	182
3.3.3 CString类	187
3.3.4 三种类型字符串之间的转换	190
3.4 字符串处理训练任务	191
本章参考文献	199
第4章 绘图技术训练	200
4.1 VC绘图技术	200
4.1.1 VC绘图基础	200
4.1.2 绘图流程	202
4.2 绘图函数	206
4.2.1 绘制线条函数	207
4.2.2 绘制形状函数	210
4.2.3 填充函数	214
4.3 绘图类及其使用举例	216
4.3.1 矩形类	216
4.3.2 点类	217

4.3.3 画笔类	218
4.3.4 画刷类	220
4.4 绘图技术应用举例	221
4.4.1 绘图技术解题	221
4.4.2 绘图难点	239
4.5 文本显示	240
4.6 绘图技术训练任务	244
本章参考文献	251
第5章 粘贴图片技术训练	252
5.1 VC 贴图知识	252
5.1.1 粘贴静态图片	253
5.1.2 粘贴动态图片编程基础	255
5.1.3 粘贴动态图片流程	260
5.2 粘贴动态图片举例	265
5.3 粘贴图片技术训练任务	275
本章参考文献	282
第6章 综合训练	283
6.1 游戏程序开发准备	283
6.1.1 游戏程序的一般结构	284
6.1.2 游戏程序的一般流程	286
6.1.3 游戏程序开发步骤	288
6.1.4 游戏程序设计中的注意事项	290
6.2 游戏程序开发举例	294
6.2.1 图片对对碰制作	295
6.2.2 俄罗斯方块制作	312
6.3 棋牌类游戏设计任务	341
本章参考文献	359
附录A VC 集成开发环境	361
A.1 VC 可视化集成开发环境	361
A.1.1 项目工作区	361
A.1.2 应用程序向导(AppWizard)	362
A.1.3 类向导(ClassWizard)	363
A.1.4 组件库(Component Gallery)	363
A.1.5 输出窗口(Output)	363

A.2	VC 创建项目工程	363
A.2.1	创建 C++ 程序	364
A.2.2	创建 Console 应用程序	364
A.2.3	创建 Win32 应用程序	365
A.2.4	创建 MFC 应用程序	368
	本章参考文献	368
附录 B	基于 MFC 开发应用程序基础	369
B.1	程序坐标系统	369
B.2	MFC 项目工程实例	370
B.2.1	添加新类	372
B.2.2	添加新消息处理函数	373
B.2.3	添加成员变量	375
B.2.4	添加成员函数	377
B.2.5	重载虚拟函数	377
B.3	应用程序消息响应机制简介	378
B.3.1	Win32 应用程序消息响应机制	378
B.3.2	MFC 应用程序消息响应机制	381
B.4	对话框	383
B.4.1	基本概念	383
B.4.2	控件	383
B.4.3	制作对话框	384
B.5	单文档程序	388
B.5.1	单文档程序的文件结构	389
B.5.2	单文档程序的类结构	390
B.5.3	单文档程序的 MFC 接口	393
B.5.4	单文档程序的执行过程	397
	本章参考文献	398
附录 C		399
C.1	程序设计实验报告框架	399
C.2	程序设计实现报告样本	400
附录 D	程序设计实验管理表与考核建议样本	406
附录 E	实例快速索引	408

基础训练

学完“C 语言程序设计”课程后,每个人基本上都能够编写简单程序,解决一般性问题。但对一些稍有难度的问题,还是会有些无从下手,因为还缺少对所学知识的综合运用能力。

随着需要解决问题的难度加大,错误常常难以避免,想迅速捕捉和消除错误,需要充分利用调试手段和技巧,需要掌握对代码的正确测试方法。

本章总结标准 C 语言的主要知识点、重点、难点,解析疑难问题;还详细介绍 VC 开发工具的调试程序方法和操作技巧,程序代码测试技术;并配以训练任务,帮助理解和巩固所学知识。

1.1 C 语言知识总结

学习程序设计,需要准确把握许多关键点,才能减少编程错误,提高编程效率。C 语言由于书写格式很灵活,提供的标准函数多,初学者很难掌握。

只有总结所学知识,明确语言结构和组成,坚持实际动手编写代码,经过不断训练,提高调试技巧(这是非常重要的技能),才能成为一名理想的编程人员。

一名好的程序员,能够牢牢掌握编程技术,做到针对问题,进行全面深入分析,提出可行方案,设计合理算法,编写优质代码,加速调试过程,求得满意结果,并将结果以规范清晰完美的形式呈现出来。

“习惯成自然”,养成良好的编程习惯非常必要。即做到编写的程序代码简洁、可读性好;变量名表意明确,语句有注释;程序构思有说明,设计思想用流程图表述;冗余语句少见,尽可能优化代码;对程序执行结果心中有数,当程序运行结果与预期值不同时,能有的放矢地进行分析,迅速找到出错位置和原因,并用有效方案,改正错误。

编程是不断学习和成长、不断充实和满足的过程。用良好的心态,带着自信,不断练习。坚持不懈,大量编写程序,逐步理解编程技术,才能把握精髓,进入自由王国。

1.1.1 C 语言知识点归纳

“C 语言程序设计”课程详细讲述了各种基本数据类型,表达式书写格式和运算顺

序,输入输出方法,三大结构控制语句,子函数编写方法和原则,以及构造类数据类型——数组、指针、结构和文件等。下面归纳主要知识点,并举例说明常见错误,以及注意事项。

1. 基础知识

1) 基本构架

从结构上看,一个完整的 C 语言程序由一个或多个函数构成。其中必须且只能有一个主函数 main()。

从语句上看,C 语言程序由声明部分(数据描述)和执行语句(数据操作)构成。声明部分主要是指定变量类型、分配数据空间、为编译提供必要说明等,而执行语句在编译时将产生目标代码。

2) 简单的 C 程序构架如下:

```
#define ...           //预编译命令
#include ...
void main()           //主函数
{                     //函数体开始
    声明部分           //定义变量,为所用数据分配空间

    执行部分           //数据处理
}                     //函数体结束
```

执行部分的语句分为 5 类:

(1) 赋值语句;

(2) 控制语句:

- ① if () ... else...
- ② switch
- ③ for () ...
- ④ while () ...
- ⑤ do ... while()
- ⑥ continue
- ⑦ break
- ⑧ goto
- ⑨ return

(3) 函数调用语句:

函数名(参数);

(4) 空语句:

;

(5) 复合语句:

```
{ }
```

例 1-1 一个简单的 C 程序。

```
#include <stdio.h>           //包含头文件
void main()                  //程序从主函数 main 开始执行
{
    int i;                   //声明部分,定义变量 i 为整型
    i=2012;                  //执行语句,为 i 赋值 2012
    printf("Hello,%d!\n",i); //执行语句,调用输出函数,输出结果
}
```

运行结果:

Hello,2012!

3) C 语言的格式特点如下:

- 习惯上使用小写字母作为标志符,大写字母常用于常量的宏定义和其他用途。
- C 语言程序由一条语句组成,每条语句都具有规定的语法格式和特定的功能。
- C 语言程序使用分号作为语句的终止符或分隔符。
- C 语言程序不存在程序行的概念。一行中可以有多条语句(用逗号或分号隔开),一条语句也可以占用任意多行。

4) 规范清晰的书写格式要求:

- 一般情况下,每个语句占用一行。
- C 语言程序中使用 { } 表示程序的结构层次范围。不同结构层次的语句,从不同的起始位置开始,即在同一结构层次中的语句,缩进同样的字数(可用 Tab 键或空格键调整各行的起始位置)。
- 表示结构层次的大括弧,写在该结构化语句第一个字母的下方,与结构化语句对齐,可独占一行,并注意与对应的大括弧排在同一列,这样可以清晰地显示出成对出现的括弧。
- 为了增强可读性,程序中可适当地加些空格和空行。但注意,程序中各种标识符(变量名、函数名等)中间不能插入空格。名字尽量为能表明含义的单词或词汇,避免用单个字符(除非简单的循环控制变量)做标识符,以免日久遗忘,使理解困难。

2. 数据类型

C 语言是强类型语言,程序中用到的每个变量都必须先定义它的类型。原因是:

- 不同类型的数据在内存中占据不同长度的空间,而且采用不同的表示方式。
- 一种数据类型对应着一个值的范围。如 VC 的 int 类型取值范围为 $-2^{31} \sim 2^{31} - 1$, 浮点型的绝对值范围为 $10^{-38} \sim 10^{38}$ 。
- 一种数据类型对应着一组允许的操作。如模运算符 % 只能用于整型。

C 语言数据类型如图 1-1 所示,C 语言基本数据类型及取值范围如表 1-1 所示。

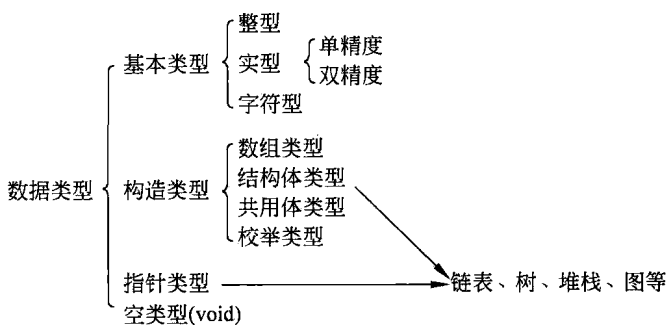


图 1-1 C 语言数据类型

表 1-1 C 语言基本数据类型及取值范围

类型标识符	类型名称	占字节数	取值范围
char	字符型	1	-128~127
unsigned char	无符号字符型	1	0~255
int	整型	2 或 4	$-2^{15} \sim 2^{15} - 1$ 或 $-2^{31} \sim 2^{31} - 1$
unsigned int	无符号整型	2 或 4	$0 \sim 2^{16} - 1$ 或 $0 \sim 2^{32} - 1$
short int	短整型	2	$-2^{31} \sim 2^{31} - 1$
unsigned short int	无符号短整型	2	$0 \sim 2^{32} - 1$
long int	长整型	4	$-2^{31} \sim 2^{31} - 1$
unsigned long int	无符号长整型	4	$0 \sim 2^{32} - 1$
float	浮点型	4	绝对值 $10^{-38} \sim 10^{38}$
double	双精度型	8	绝对值 $10^{-308} \sim 10^{308}$
long double	长双精度型	10	绝对值 $10^{-4932} \sim 10^{4932}$

这里 float、double 和 long double 型的有效数字位不同。通常 float 型的有效位是小数点后 6 位, double 型是 15 位, 而 long double 型是 19 位。

3. 常量

在程序中直接使用的数值(整型、实型、字符、字符串)叫常量。对频繁使用的某一数值, 可以起个名字(通常用大写字母), 以后就用该名字代替该数值, 这个数值在程序执行过程中不能改变。使用常量能提高程序的可读性和可维护性。

定义常量可以用宏定义实现, 方法如下:

```
#define 名字 常量值
```

常量的类型由常量值决定。常量的使用见例 1-2, 常量的表示如表 1-2 所示。

注意: define 定义语句行的末尾没有分号。如果在语句行尾写上分号, 编译程序不能准确提示“分号”错误, 而是给出其他莫名其妙的错误提示。

表 1-2 常量的表示

类型	说 明	示 例
整型	八进制, 以 0 开头	064
	十进制	52
	十六进制, 以 0x 开头	0x34
	长整型, 以 l 结尾	52l
实型	小数	123.5
	指数, 小数点左边只有一位非零数字	1.235e2
字符型	用单引号 ' 括起的字符	'C'
	\ddd 用八进制表示 ASCII 码字符	\083'
	\xhh 用十六进制表示 ASCII 码字符	\x43'
字符串	用双引号 " 括起的字符序列	"Hi,C!"

例 1-2 已知半径, 求圆面积和周长。

```
#include<stdio.h>
#define PI 3.141592654           //宏定义
#define R 5.2                   //宏定义
double area(), circum();        //函数声明
void main()
{
    printf ("area=%f \n", area ());    //输出圆的面积
    printf ("circum=%f\n", circum ()); //输出圆的周长
}
//计算圆的面积
double area ()
{
    return (PI * R * R) ;
}
//计算圆的周长
double circum()
{
    return (2.0 * PI * R) ;
}
```

运行结果如下:

```
area=84.948665
circum=32.672564
```

注意:

① 可以打印的字符直接用单引号括起字符, 不可打印的字符常用单引号括起转义字

符。常用的转义字符有\n, \r, \t, \v, \b, \\, \', \".

② 字符串常量的结尾处有一个字符串结束标志'\0',系统据此判断字符串是否结束。'\0'是一个空操作字符,既不起控制作用,也不显示,但它在存储时要占一个字节的空間。

4. 变量

程序中常用到一些需要不断调整改变的数据,例如在一个区域内变化的数值,对一批数据进行累加、累乘等运算。变量用于记录和保存变化的数值。指定一个名字为变量时,程序在进行编译链接时,就会给变量分配存储单元。此后用变量名就可以访问存储单元,改变其中存放的数值。

C 语言的变量具有三种属性:

- (1) 数据类型(int、float 等);
- (2) 存储类型(static、extern 等,默认为 auto);
- (3) 作用范围(局部变量、全局变量)。

变量的定义如下:

数据类型 变量名 1[,变量名 2,变量名 3,...];

说明:

① 程序编译时遇到变量定义会为变量分配内存单元,变量名是内存单元的符号地址。

② 变量的名字实际有两个含义,一个是左值,一个是右值。当它出现在赋值语句等号的左边时,使用的是变量的地址(左值),当它出现等号右边时,取的是变量的值(右值)。

③ 变量赋值的特点:

- 变量要先定义,后使用。如使用未定义的变量,则在编译时会查出,指明非法。
- 变量未被赋值前,赋值不确定(虽然有的系统可能默认为 0,但不能确保赋值不变)。
- 对变量赋值过程是“覆盖”过程,即用新值替换旧值。
- 读取变量的值时,并不会改变该变量的值。
- 参与表达式运算的所有变量都保持原来的值不变。

5. 数据类型转换

不同数据类型的数据(包括常量和变量)之间进行各种运算时,C 语言编译系统将自动进行数据类型的转换。

转换的原则如下:

(1) 参加运算的两个数据中,数据长度较短的数据转换为它们之中数据较长的数据类型,然后进行运算,结果为数据较长的数据类型,数据类型的自动转换如图 1-2 所示。

其转换顺序如下:

(2) 赋值运算中,C 语言自动把等号右边的数据转换成左边的

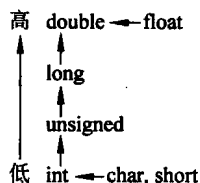


图 1-2 数据类型的自动转换