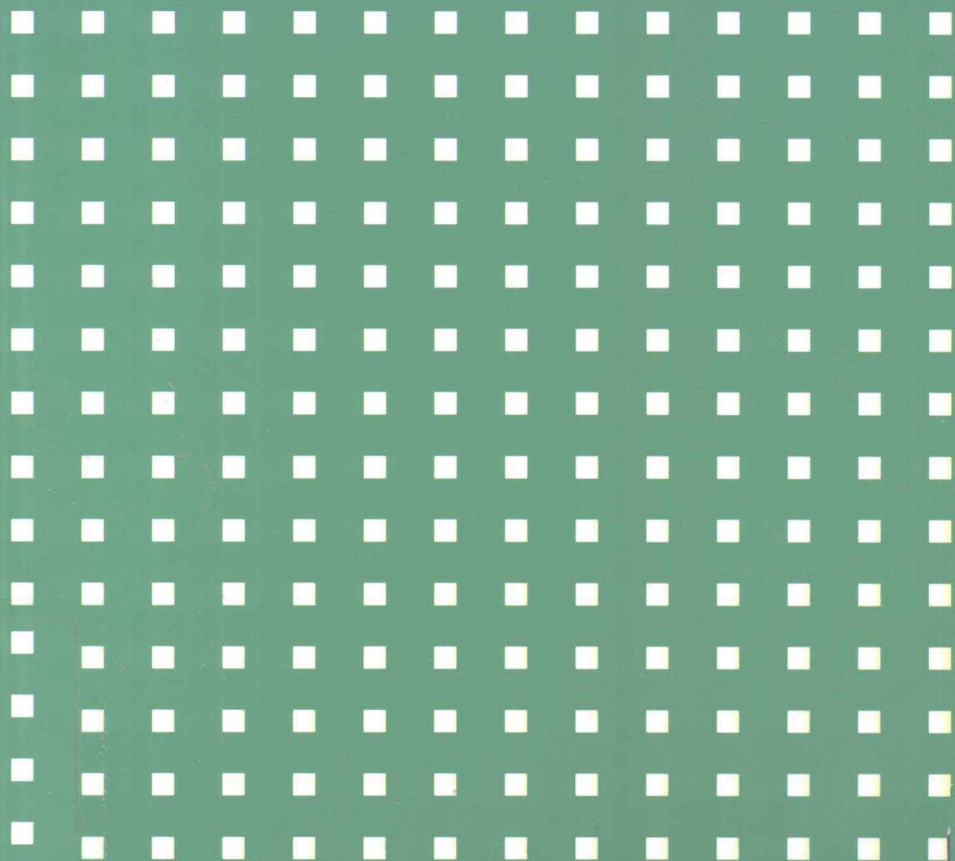


高等学校计算机专业教材精选·算法与程序设计

全国普通高等院校计算机规划教材

# 面向对象程序设计

李忠晔 刘伟东 编著



清华大学出版社

高等学校计算机专业教材精选·算法与程序设计

全国普通高等院校计算机规划教材

# 面向对象程序设计

李忠晔 刘伟东 编著

清华大学出版社

北京

## 内 容 简 介

本书从最基本的概念出发,详细地讲述了使用 Visual C++ 进行面向对象和可视化程序设计的基础、原理与方法。全书共分 14 章,前 5 章主要介绍 C++ 语言的基本编程技术,包括 C++ 基础知识、类和对象、运算符重载、继承与派生、多态性与虚函数;第 6 章~第 11 章主要介绍使用 MFC 进行可视化编程的方法,包括 MFC 的基本知识、屏幕绘图及文本显示、键盘与鼠标、Windows 控件、资源、单文档与多文档;第 12 章~第 14 章主要介绍高级编程应用,包括数据库编程、多媒体编程、Internet 编程。

本书既可以作为高等院校计算机科学与技术及相关专业“面向对象程序设计”课程的授课教材,也可以作为 C++ 程序开发人员的参考书。

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

面向对象程序设计/李忠晔,刘伟东编著. —北京:清华大学出版社,2011.9

(高等学校计算机专业教材精选·算法与程序设计)

ISBN 978-7-302-26292-3

I. ①面… II. ①李… ②刘… 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(2011)第 141685 号

责任编辑:白立军 薛阳

责任校对:梁毅

责任印制:王秀菊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

印 装 者:北京市清华园胶印厂

经 销:全国新华书店

开 本:185×260 印 张:20.25 字 数:504 千字

版 次:2011 年 9 月第 1 版 印 次:2011 年 9 月第 1 次印刷

印 数:1~3000

定 价:32.00 元

产品编号:040334-01

# 出版说明

我国高等学校计算机教育近年来迅猛发展,应用所学计算机知识解决实际问题,已经成为当代大学生的必备能力。

时代的进步与社会的发展对高等学校计算机教育的质量提出了更高、更新的要求。现在,很多高等学校都在积极探索符合自身特点的教学模式,涌现出一大批非常优秀的精品课程。

为了适应社会的需求,满足计算机教育的发展需要,清华大学出版社在进行大量调查研究的基础上,组织编写了《高等学校计算机专业教材精选》。本套教材从全国各高校的优秀计算机教材中精挑细选了一批很有代表性且特色鲜明的计算机精品教材,把作者们对各自所授计算机课程的独特理解和先进经验推荐给全国师生。

本系列教材特点如下。

(1) 编写目的明确。本套教材主要面向广大高校的计算机专业学生,使学生通过本套教材,学习计算机科学与技术方面的基本理论和基本知识,接受应用计算机解决实际问题的基本训练。

(2) 注重编写理念。本套教材作者群为各高校相应课程的主讲,有一定经验积累,且编写思路清晰,有独特的教学思路和指导思想,其教学经验具有推广价值。本套教材中不乏各类精品课配套教材,并努力把不同学校的教学特点反映到每本教材中。

(3) 理论知识与实践相结合。本套教材贯彻从实践中来到实践中去的原则,书中的许多必须掌握的理论都将结合实例来讲,同时注重培养学生分析问题、解决问题的能力,满足社会用人要求。

(4) 易教易用,合理适当。本套教材编写时注意结合教学实际的课时数,把握教材的篇幅。同时,对一些知识点按教育部教学指导委员会的最新精神进行合理取舍与难易控制。

(5) 注重教材的立体化配套。大多数教材都将配套教师用课件、习题及其解答,学生上机实验指导、教学网站等辅助教学资源,方便教学。

随着本套教材陆续出版,我们相信它能够得到广大读者的认可和支持,为我国计算机教材建设及计算机教学水平的提高,为计算机教育事业的发展做出应有的贡献。

清华大学出版社

# 前 言

目前,面向对象程序设计已经发展成一种比较成熟的编程思想,C++是目前面向对象程序设计教学的主流。Visual C++是微软公司开发的运行于Windows环境下的可视化编程工具中最重要的软件开发工具之一,特别是微软公司提供的微软基础类库(Microsoft Foundation Class Library, MFC)为用户提供了大量的标准类,大大缩短了软件的开发周期。MFC把Windows统一直观的界面风格和面向对象的编程技术结合在一起,形成一个功能强大的集成开发环境,提供了简单高效的操作方法和与设备无关的图形接口等机制,使用Visual C++开发的程序在同类编程语言里性能是最高的,学好Visual C++对计算机专业的每个学生来说都是必需的。然而目前大多数的C++教材并不介绍可视化编程,这样就不能进行实际的可视化开发。本教材力求使学生通过学习,一方面学会面向对象程序设计的基本概念和方法,另一方面能够掌握MFC可视化程序开发的技巧。

本书从C++的面向对象基本概念讲起,首先介绍VC++基础知识,包括C++基础知识、类和对象的概念、运算符重载、继承与派生、多态性与虚函数;然后介绍使用MFC进行可视化编程,包括MFC类库的基本知识、屏幕绘图及文本显示、键盘与鼠标、Windows控件、资源、单文档与多文档;最后介绍高级编程应用,包括数据库编程、多媒体编程、Internet编程。

全书的主要特色是知识点全面、语言简洁易懂、结构合理且实用、重点突出。内容由浅入深,通俗易懂,示例丰富,图示精彩。在第9章Windows控件、第10章资源、第11章单文档与多文档、第12章数据库编程、第13章多媒体编程以及第14章Internet编程中,都是以一个综合应用程序的方式,把相关知识点的內容串联起来。为了有利于学习,节省篇幅,有些知识点会通过例题来介绍,实例中附有较为详细的步骤说明、代码说明,通过案例对逐个知识点进行讲解,针对性强,以达到事半功倍的效果。本书的例题是根据教学特点精心安排的,并且所有例题都在Windows 2000及Visual C++ 6.0环境下调试运行通过。

本书面向各高等学校的本科生、研究生及从事计算机软件开发的专业人员,可作为高职、高专、本科院校或计算机培训机构的教材,也可作为计算机爱好者、程序员的自学教材或参考书。本书可作为计算机专业面向对象程序设计课程的教材,建议授课学时为50小时并先修C语言程序设计课程。

本书作者长期从事C++项目开发及本科生面向对象程序设计课程的教学工作,具有丰富的教学与程序开发经验。本书第1~4章由刘伟东编写,第5、第6章由冯贵良编写,第7~11章由李忠晔编写,第12章由朱萍编写,第13章由李静编写,第14章由刘新编写,赵喜清对本书进行了审稿。

在本书的编写过程中,编者还查阅了部分文献,文后的“参考文献”部分列出了这些文献及文献作者,在此也对上述作者表示感谢。

由于作者水平有限,书中难免会存在一些缺点和错误,恳请读者批评指正。

谢谢阅读本书的读者!

作 者

# 目 录

<b>第 1 章 C++ 基础知识</b> .....	1
1.1 语法与基本数据类型 .....	1
1.1.1 C++ 语言语法 .....	1
1.1.2 基本数据类型 .....	1
1.2 常量与变量 .....	2
1.2.1 常量 .....	2
1.2.2 变量 .....	3
1.3 运算符与表达式 .....	4
1.3.1 运算符 .....	4
1.3.2 表达式 .....	5
1.3.3 表达式类型转换 .....	5
1.4 语句 .....	6
1.4.1 控制语句 .....	6
1.4.2 其他语句 .....	10
1.5 指针 .....	10
1.5.1 指针变量的定义 .....	10
1.5.2 两个重要运算符 .....	11
1.5.3 指针变量的赋值 .....	11
1.5.4 指针变量作函数参数 .....	12
1.5.5 引用 .....	12
1.6 函数 .....	12
1.6.1 函数的分类 .....	13
1.6.2 函数的定义 .....	13
1.6.3 形参与实参 .....	14
1.6.4 函数的调用 .....	14
1.7 标准输入输出流 .....	15
1.7.1 标准输入输出流格式 .....	15
1.7.2 标准流的几点说明 .....	15
习题 .....	16
<b>第 2 章 类和对象</b> .....	18
2.1 面向对象程序设计 .....	18
2.2 类和对象 .....	19

2.2.1	类和对象的关系 .....	19
2.2.2	类的概念和定义方法 .....	20
2.2.3	对象的概念和定义方法 .....	21
2.2.4	对象成员的访问方法 .....	22
2.2.5	成员函数的定义 .....	23
2.2.6	this 指针 .....	24
2.3	构造函数和析构函数 .....	25
2.3.1	构造函数 .....	25
2.3.2	析构函数 .....	30
2.3.3	拷贝构造函数 .....	32
2.4	静态成员 .....	33
2.4.1	静态数据成员 .....	34
2.4.2	静态成员函数 .....	36
2.5	对象指针 .....	37
2.5.1	指向类对象的指针 .....	37
2.5.2	指向对象成员的指针 .....	38
2.6	友元 .....	39
2.6.1	友元函数 .....	40
2.6.2	友元类 .....	41
习题	.....	43
<b>第 3 章</b>	<b>运算符重载 .....</b>	<b>44</b>
3.1	什么是运算符重载 .....	44
3.2	重载运算符的规则 .....	44
3.3	运算符重载函数作为类的成员函数 .....	45
3.4	运算符重载函数作为类的友元函数 .....	48
3.5	重载++和--运算符 .....	52
3.6	重载流插入运算符和流提取运算符 .....	54
习题	.....	56
<b>第 4 章</b>	<b>继承与派生 .....</b>	<b>57</b>
4.1	继承与派生的概念 .....	57
4.2	单继承 .....	58
4.2.1	单继承的定义方式 .....	58
4.2.2	派生类的成员构成 .....	59
4.2.3	派生类成员的访问属性 .....	60
4.2.4	派生类的构造函数 .....	63
4.2.5	派生类的析构函数 .....	65
4.3	多重继承 .....	67

4.3.1	多重继承的定义方式 .....	67
4.3.2	多重继承派生类的构造函数与析构函数 .....	68
4.3.3	多重继承的二义性 .....	69
4.3.4	虚基类 .....	70
习题	.....	73
<b>第 5 章</b>	<b>多态性与虚函数</b> .....	<b>74</b>
5.1	多态性 .....	74
5.1.1	多态的类型 .....	74
5.1.2	函数重载 .....	74
5.1.3	联编 .....	77
5.2	虚函数 .....	77
5.3	纯虚函数和抽象类 .....	81
5.3.1	纯虚函数 .....	81
5.3.2	抽象类 .....	82
习题	.....	82
<b>第 6 章</b>	<b>MFC 编程</b> .....	<b>83</b>
6.1	Windows 编程基础知识 .....	83
6.1.1	事件驱动原理 .....	83
6.1.2	基本概念 .....	85
6.1.3	常用的数据结构 .....	86
6.2	MFC 概述 .....	88
6.3	MFC 类的层次结构及类简介 .....	88
6.3.1	程序结构类 .....	88
6.3.2	窗口类 .....	89
6.3.3	图形类 .....	91
6.3.4	部分非 CObject 派生类 .....	92
6.4	用 AppWizard 建立 MFC 应用程序 .....	92
6.4.1	MFC 应用程序类型 .....	92
6.4.2	MFC 的 AppWizard 创建应用程序的步骤 .....	93
6.4.3	应用程序向导生成的类和文件 .....	98
6.4.4	项目工作区窗口 .....	99
6.5	消息和消息映射 .....	100
6.5.1	消息的结构 .....	100
6.5.2	消息的种类 .....	101
6.5.3	消息映射 .....	102
6.6	常用的类的操作 .....	107
习题	.....	110



<b>第 7 章 屏幕绘图及文本显示</b> .....	111
7.1 GDI 与设备描述表 .....	111
7.1.1 GDI 的基本概念 .....	111
7.1.2 设备描述环境 .....	111
7.2 CDC 类 .....	112
7.3 映射模式 .....	114
7.3.1 设备坐标 .....	114
7.3.2 逻辑坐标 .....	114
7.3.3 映射模式的相关设置 .....	115
7.4 绘图工具的使用 .....	116
7.4.1 GDI 绘图工具介绍 .....	116
7.4.2 画笔 .....	117
7.4.3 画刷 .....	119
7.5 绘图模式 .....	122
7.6 常用 CDC 类绘图函数 .....	123
7.7 绘图过程及应用实例 .....	126
7.8 文本输出 .....	129
习题 .....	132
<b>第 8 章 键盘与鼠标</b> .....	133
8.1 鼠标 .....	133
8.1.1 鼠标事件和鼠标消息 .....	133
8.1.2 鼠标消息处理函数 .....	134
8.1.3 鼠标应用实例 .....	135
8.2 键盘 .....	145
8.2.1 键盘消息与虚拟键代码 .....	145
8.2.2 键盘消息处理函数 .....	146
8.2.3 键盘应用实例 .....	149
习题 .....	152
<b>第 9 章 Windows 控件</b> .....	154
9.1 Windows 标准控件 .....	154
9.1.1 控件的创建 .....	155
9.1.2 控件的管理 .....	155
9.1.3 控件的通用属性 .....	157
9.1.4 常用控件窗口操作函数 .....	159
9.2 静态控件及其应用 .....	160
9.2.1 静态文本控件 .....	161

9.2.2	组框控件	162
9.2.3	静态图像控件	162
9.3	按钮控件	163
9.3.1	按钮控件的种类及消息	163
9.3.2	按钮控件的成员函数	164
9.3.3	命令按钮	165
9.3.4	单选按钮	167
9.3.5	复选按钮	169
9.4	编辑框控件	170
9.4.1	编辑框属性	171
9.4.2	编辑框类的成员函数	172
9.4.3	编辑框的主要消息	173
9.4.4	编辑框控件的使用	173
9.4.5	编辑框编程实例	175
9.5	列表框控件	182
9.5.1	列表框的属性	182
9.5.2	列表框类的成员函数	183
9.5.3	列表框的消息	185
9.5.4	列表框的使用	185
9.5.5	列表框应用举例	186
9.6	组合框控件	187
9.6.1	组合框属性	188
9.6.2	组合框类的成员函数	188
9.6.3	组合框的消息	189
9.6.4	组合框的使用	189
9.6.5	组合框应用举例	190
9.7	滚动条控件	191
9.7.1	滚动条类的成员函数	191
9.7.2	滚动条控件的使用	192
9.7.3	滚动条类应用举例	193
9.8	滑动条控件	195
9.8.1	滑动条的属性	196
9.8.2	滑动条类的成员函数	196
9.8.3	滑动条的使用	197
9.8.4	滑动条类应用举例	197
9.9	翻动按钮控件	199
9.9.1	翻动按钮的属性	199
9.9.2	翻动按钮类的成员函数	200
9.9.3	翻动按钮应用举例	200

9.10	进度条控件	201
9.11	通用对话框	203
9.12	消息对话框	205
9.13	综合应用	206
	习题	211
<b>第 10 章</b>	<b>资源</b>	<b>213</b>
10.1	菜单	213
10.1.1	创建和编辑菜单	214
10.1.2	菜单消息	215
10.1.3	菜单应用实例	216
10.2	加速键	223
10.3	工具栏	224
10.3.1	增加工具栏按钮	224
10.3.2	工具栏应用实例	225
10.4	状态栏	227
10.4.1	状态栏的实现	227
10.4.2	状态栏应用实例	228
10.5	图标资源	231
10.6	位图资源	233
10.7	对话框资源	234
10.7.1	模态对话框	235
10.7.2	非模态对话框	238
	习题	238
<b>第 11 章</b>	<b>单文档与多文档</b>	<b>240</b>
11.1	概述	240
11.1.1	文档/视图结构	240
11.1.2	SDI 程序的创建过程及相关函数	241
11.1.3	SDI 程序的命令消息传递过程	243
11.2	文档/视图结构的 5 个类	243
11.3	文档串行化操作	245
11.4	设置窗口样式	246
11.5	SDI 编程实例	249
11.6	MDI 简介	252
11.7	多文档编程实例	253
11.8	类的序列化	260
	习题	264

<b>第 12 章 数据库编程</b> .....	266
12.1 ODBC 简介 .....	266
12.1.1 ODBC 的基本构成 .....	266
12.1.2 配置 ODBC 数据源 .....	267
12.1.3 MFC 提供的有关 ODBC 的常用类 .....	268
12.2 简单的 MFC ODBC 数据库应用编程 .....	271
12.3 ActiveX 控件 .....	281
12.3.1 使用 MSFlexGrid 控件 .....	281
12.3.2 使用 RemoteData 和 DBGrid 控件 .....	283
12.4 多表连接 .....	285
习题 .....	288
<b>第 13 章 多媒体编程</b> .....	289
13.1 Windows Media Player 控件 .....	289
13.2 利用 Windows Media Player 控件实现多媒体编程 .....	293
习题 .....	298
<b>第 14 章 Internet 编程</b> .....	299
14.1 WinInet 简介 .....	299
14.1.1 CInternetSession 类 .....	299
14.1.2 CInternetConnection 类 .....	300
14.1.3 CInternetFile 类 .....	301
14.1.4 CFileFind 类 .....	301
14.1.5 用 WinInet 类开发应用程序的一般步骤 .....	301
14.2 Internet 编程实例 .....	301
习题 .....	306
<b>参考文献</b> .....	307

# 第 1 章 C++ 基础知识

## 1.1 语法与基本数据类型

C++ 语言制定了严格的语法规则、字符集和数据类型。程序设计中不能违反这些规则,不能使用字符集以外的字符,程序中用到的数据也都必须是其指定的数据类型。

### 1.1.1 C++ 语言语法

#### 1. 基本符号集

字符是组成语言的最基本的元素。C++ 语言的基本符号集由字母、数字、空格、标点和特殊字符组成,在字符、字符串和注释中还可以使用汉字或其他可表示的图形符号。

(1) 字母: 小写字母 a~z 共 26 个,大写字母 A~Z 共 26 个。

(2) 数字: 0~9 共 10 个。

(3) 空格: 多用于语句各单词之间,做间隔符。

(4) 标点和特殊字符: 部分符号有特殊的用途,如逗号用在类型说明和函数参数列表中分隔各个变量,又是一个分隔符。

#### 2. 关键字

关键字是由 C++ 语言规定的具有特定意义的字符串,通常也称为保留字。用户定义的标识符不应与关键字相同。C++ 语言的关键字分为以下几类:

##### 1) 类型说明符

用于定义和说明变量、函数或其他数据结构的类型,如 int, double 等。

##### 2) 语句定义符

用于表示一个语句的功能。if-else 就是条件语句的语句定义符。

##### 3) 预处理命令字

用于表示一个预处理命令,如 include。

#### 3. 标识符

在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外,其余函数名都由用户自定义。

标识符的命名有一定的规则:

(1) 标识符只能由字母、数字和下划线 3 类字符组成。

(2) 第一个字符必须是字母或下划线,不能是数字。

(3) 标识符不能是 C++ 的关键字。

注意: 标识符的大小写是有区别的。例如 BOOK 和 book 是两个不同的标识符。

### 1.1.2 基本数据类型

在 C++ 语言中,只有 3 种基本的数据类型,即整数类型(int)、字符类型(char)和实数类

型(浮点类型)。

### 1. 整数类型

基本的整数类型只有一种,用 int 表示,但可以使用修饰符 short 和 long 来构造两种新的类型,也就是短整型和长整型。还可以用修饰符 signed 和 unsigned 分别把各种整数类型设定为有符号类型和无符号类型。

### 2. 实数类型

实数类型又称为浮点类型,可以分为单精度(float)、双精度(double)和长双精度(long double)3类。系统把一个实型数据分成小数部分和指数部分分别存放。

### 3. 字符类型

将字符进行编码,从而为每个字符指定一个确定的编码,作为识别与使用这些字符的依据。字符类型也可以用修饰符 signed 和 unsigned 来进行修饰。注意:可将字符类型在它的取值范围内当成一个整数类型来用。

在 C++ 语言中,基本数据类型的具体说明如表 1-1 所示。

表 1-1 基本数据类型(以 Turbo C++ 为例)

类 型	字节数	数的取值范围
[signed] char	1	$-2^7 \sim (2^7 - 1)$
unsigned char	1	$0 \sim (2^8 - 1)$
[signed] short [int]	2	$-2^{15} \sim (2^{15} - 1)$
unsigned short [int]	2	$0 \sim (2^{16} - 1)$
[signed] int	2	$-2^{15} \sim (2^{15} - 1)$
unsigned int	2	$0 \sim (2^{16} - 1)$
long [int]	4	$-2^{31} \sim (2^{31} - 1)$
unsigned long [int]	4	$0 \sim (2^{32} - 1)$
float	4	$-3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$
double	8	$-1.7 \times 10^{-308} \sim 1.7 \times 10^{308}$
long double	16	$-1.2 \times 10^{-4932} \sim 1.2 \times 10^{4932}$

## 1.2 常量与变量

按基本数据类型量的取值是否可以改变,可将基本数据类型量分为常量和变量。

### 1.2.1 常量

所谓常量,就是在程序运行过程中,其值不会被改变的量。不同的数据类型,有不同的常量。根据常量的表示方法,可以把常量分为如下两类:

(1) 字面常量(也称直接常量)。这一类常量可以从它们的字面形式来判别,如 12、4.6、'a'等。

(2) 符号常量。即用一个标识符来代表一个常量。如

```
#define PI 3.1415926
```

符号常量的定义格式:

#define 符号常量名 表达式 (符号常量名常用大写,以区别变量)

使用符号常量的好处是含义清楚、见名知义、修改方便、一改全改。

### 1. 整型常量

在 C++ 语言中,整型常量有以下 3 种表示形式:

- (1) 十进制形式。也是最常用的一种表示形式,如 123、-456。
- (2) 八进制形式。以 0 开头的整数是八进制数,如 013、-057。
- (3) 十六进制形式。以 0x 开头的数是十六进制数,如 0x9f、-0xef。

### 2. 实型常量

实型常量即实数,它有以下两种表示形式:

(1) 十进制小数形式。它由数字和小数点组成,如 0.124、124.0、0.0。注意,如果小数点的左边或右边为 0,那么这个 0 可以省略,但小数点一定不能省略,例如,.124 其实就是 0.124。124. 其实就是 124.0。

(2) 指数形式。由 3 个部分组成,第一部分是一个实数,第二部分是字母 e 或 E,第三部分是一个指数。例如,321.54e6,它的含义就是  $321.54 \times 10^6$ 。

### 3. 字符常量

字符常量就是用单引号括起来的一个字符,如 'a'、'A' 和 '#' 等。另外有一种字符形式比较特殊,即转义字符。转义字符是一种特殊的字符常量。一个转义字符用一个反斜杠 (\) 加一个字母来表示,表 1-2 列出了常见的一些转义字符。

表 1-2 转义字符

种类	转义字符	转义字符的意义
1	\n	回车换行
	\t	跳到下一制表位置
	\b	退格
	\r	回车
2	\\	反斜线符“\”
	\'	单引号符
	\"	双引号符
3	\ddd	1~3 位八进制数所代表的字符
	\xhh	1、2 位十六进制数所代表的字符

## 1.2.2 变量

### 1. 基本概念

所谓变量,顾名思义,就是其值可变的量,如图 1-1 所示,一个变量首先要有名字,这个名字是由程序员给出的。每个变量都会在内存中占用一小段存储空间,它是由编译程序来分配的,用于保存该变量的值。在程序中,可以通过变量的名字来找到相应的存储空间,存取或修改其中的变

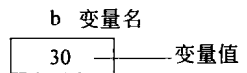


图 1-1

量值。

## 2. 变量的定义

在 C++ 语言中,所有的变量在使用之前都必须先定义,也就是“先定义,后使用”。变量的命名需要使用合法的标识符。

变量的定义方法为:

数据类型 变量 1,变量 2,⋯,变量 n;

例如:

```
int a,b;           //定义了两个整型变量
char c;           //定义了一个字符变量
```

## 3. 变量的初始化

C++ 语言允许在定义变量的同时就给变量赋值,这个过程叫变量的初始化。

例如:

```
char c='A';       //定义一个字符变量,初始值为'A'
int a=3,b,c=3;   //定义3个整型变量,只初始化其中的a和c
```

# 1.3 运算符与表达式

C++ 语言的运算符范围很宽,除了控制语句和输入输出以外,几乎所有的基本操作都作为运算符来处理。丰富的运算符和表达式使 C++ 语言功能非常完善。

## 1.3.1 运算符

C++ 提供了以下运算符。

- (1) 算术运算符: +, -, \*, /, %, ++, --;
- (2) 关系运算符: <, <=, >, >=, ==, !=;
- (3) 逻辑运算符: &&, ||, !;
- (4) 位运算符: <<, >>, &, |, ^, ~;
- (5) 赋值运算符: =, +=, -=, \*=, /=, %=, &.=, ^=, |=, <<=, >>=;
- (6) 条件运算符: ? : ;
- (7) 指针运算符: \* ;
- (8) 取地址运算符: &;
- (9) 求字节数运算符: sizeof;
- (10) 强制类型转换运算符: (类型);
- (11) 成员运算符: . ;
- (12) 指向成员的运算符: ->;
- (13) 下标运算符: [];
- (14) 圆括号: ()。

优先级和结合性是运算符的两个重要特性。优先级是指这些运算符计算时都有一定的



顺序,例如需要先算乘除后算加减。结合性又称为结合方向,它决定组成表达式的各个部分是否参与计算以及什么时候参与计算。

### 1.3.2 表达式

表达式由常量、变量、运算符组合而成,类似数学表达式,计算以后返回一个结果值。C++ 语言的表达式可以从不同的角度分类,根据运算符的不同可分为算术运算表达式、关系运算表达式、逻辑运算表达式、赋值表达式等;根据运算符的运算对象个数可分为一元表达式、二元表达式和条件表达式。

#### 1. 一元表达式

带有一元运算符的表达式称为一元表达式。它自右向左组合。例如:  $++x$ 。

#### 2. 二元表达式

二元表达式就是两个表达式在二元运算符的作用下所形成的式子。例如:  $a+3$ 。

#### 3. 条件表达式

由条件运算符构成的表达式。例如:  $(b>0)? b: -b$ 。

### 1.3.3 表达式类型转换

#### 1. 隐式转换

表达式的返回值是有类型的,在表达式中出现不同类型的操作数时,要按规则将其转换成相同的类型,表达式隐含的数据类型取决于组成表达式的变量和常量的类型。因此,表达式的返回值有可能是某种整型,或者某种精度的浮点型,或者某种指针类型。

类型转化的原则是从低级向高级自动转换,转换顺序基本如下所示:

字符型→整型→长整型→浮点型→单精度型→双精度型

也就是说当字符型和整型在一起运算时,结果为整型,如果整型和浮点型在一起运算,所得的结果就是浮点型,如果有双精度型参与运算,那么答案就是双精度型。

#### 2. 赋值语句的转换

赋值语句的转换是将赋值号右边的值转换成左边的类型,其结果的类型与左端变量的类型一致。

例如:

```
int i;
char c;
c=i;
```

整型 int 转换成字符型 char 是容易的,只要将超出的高位丢掉就行。

例如:

```
float x;
int i;
x=i;
i=x;
```

浮点型 float 转换成整型 int 时将截去小数部分,双精度型 double 转换成浮点型 float