

计算机系列教材

C++.NET 程序设计实训教程

严冬松 武建华 编著

清华大学出版社



计算机系列教材

严冬松 武建华 编著

C++.NET 程序设计实训教程



清华大学出版社
北京

内 容 简 介

本书以 Visual Studio .NET 平台的 C++ /CLI 语言为技术基础,以软件项目开发为主题,结合应用实例,将内容组织为 16 章,主要内容包括 C++ /CLI 的基础知识以及面向对象的基本概念和程序设计、基于 Windows 窗体的设计方法、常用控件的属性和事件以及事件响应函数的编写、基本界面构成及界面设计的方法、多文档界面的设计及数据传递、文件 I/O 及数据库的应用、GDI+ 基本绘图和图像处理、文本绘制、游戏编程以及综合应用系统设计实例等,使读者较全面地掌握 C++ /CLI 语言的知识体系和编程技巧。全书在内容描述上力求通俗易懂,突出实用性和操作性;在内容安排上循序渐进、深入浅出,突出趣味性和应用性。

本书可作为高等学校理工类各专业的程序设计实验教材,也可供软件开发者和编程爱好者参考。

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

C++ .NET 程序设计实训教程/严冬松,武建华编著. —北京:清华大学出版社,2018

(计算机系列教材)

ISBN 978-7-302-51258-5

I. ①C… II. ①严… ②武… III. ①C++ 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 212049 号

责任编辑:张 民 战晓雷

封面设计:常雪影

责任校对:焦丽丽

责任印制:丛怀宇

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:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:30.25

字 数:701 千字

版 次:2018 年 11 月第 1 版

印 次:2018 年 11 月第 1 次印刷

定 价:59.00 元

产品编号:080106-01

前 言

笔者从十多年的教学实践中体会到,强化学习者的编程实践,是对应用程序开发最直接、有效的教学与学习方法。这种建立在强化实践基础上的编程学习将有更为持久的生命力,也将更受企业的青睐。

程序设计是一种技能,不是看书和听课就能掌握的,一定要通过上机实践才能将编程知识转化为编程技能。学习编程是一个实践的过程,通过强化的编程训练以及积极的思考,可以很快掌握编程技术,并积累许多宝贵的编程经验。目前 Visual C++ 的实训主要是由学生上机完成教材的例题,题量小,缺少综合性的强化训练,因而本书在训练内容设计上,把一个软件项目开发要做的工作按功能分成多个模块,让学生按模块逐步学习和掌握它们的设计方法和技巧,也就是按软件项目开发的过程及内容,将项目开发的功能分成不同的单元,对各单元的用例进行优化后改成实验操作内容,对每一方面的内容进行专项技能训练,最后再综合组装成一个完整的大项目。经过这样强化训练的学生就能较好地掌握软件项目开发的全过程,能编写出一个功能齐全、界面友好的应用程序项目了。这种与实际项目开发结合的实训教材是有很大需求的。

C 和 C++ 一直是最有生命力的程序设计语言。这两种语言为程序员提供了丰富的功能、高度的灵活性和强大的底层控制能力,而这一切都不得不以牺牲学习效率作为代价,特别对 Visual C++ 来说,大部分的程序结构都被封装在 MFC 中,对于初学者来说,程序结构显得十分混乱,学习过程十分艰苦。

.NET 是微软公司未来的技术发展方向,其强大的技术优势为人们所推崇,并且在全世界掀起了学习 .NET 技术的热潮,掌握该技术,无疑在目前激烈的就业竞争中就掌握了有力武器。

Visual C++ .NET 的程序结构十分清晰,易于学习和使用,同时又不失灵活性和强大的功能,在开发能力和效率之间取得了较好的平衡。Visual C++ .NET 与 C#、Java 等语言非常接近,已成为功能强大的面向对象的编程语言。

本书编者长期从事 Visual C++ 和 Visual C++ .NET 教学与应用开发,积累了丰富的经验,了解如何学习才能提高程序设计开发能力,及以较少的时间投入获得实际应用的能力。

本书注重实用性。在参考相关教材的基础上,从实际开发经验和技术人员思维习惯出发,注重问题分析和解题方法,将重点置于问题的解决和代码调试上,使学生获得真正意义上的实践动手能力。本书中的案例已在多届教学中得到广泛的应用,给学习程序设计和软件项目开发带来很大的便利和实效。

本书还注重工程实例的巧妙运用,在实例的选择方面,考虑到本书读者主要是在校大学生,缺乏工程经验,因此主要选择学生学习生活中经常接触的领域和常见的游戏程序设计案例,以方便读者加深理解。

学好编程关键在于保持对编程的兴趣。在本书中加入了较多的游戏编程(扑克发牌模拟、英雄无敌游戏、套圈游戏、抓人游戏、贪吃蛇游戏和拼图游戏),以提高学生对编程的兴趣。本书给出的案例有一定的综合性和广泛的应用性,更具有可操作性。

本书以 Visual Studio 2013 为集成开发环境,全面介绍了利用 C++ /CLI 语言开发应用程序的相关技术。全书共 16 章:第 1~4 章以开发基于控制台的程序为主,围绕 C++ /CLI 基础知识,如函数重载、默认参数、数组与字符串、类与对象、继承和多态等内容进行程序设计训练;第 5~8 章围绕 Windows 窗体界面的开发,介绍 Windows Form 编程基础、窗体、对话框及通用控件、框条控件、容器控件的应用程序设计;第 9~12 章介绍基于 Windows Form 的基本界面设计、多文档界面设计、文件和文件夹操作、数据流技术、数据库应用开发等的应用程序设计;第 13~15 章介绍 GDI+ 的图形绘制、图像处理、文本绘制、打印与导出等核心技术的应用程序设计;第 16 章给出了多媒体播放器的完整设计与实现,并介绍了应用程序的部署与安装方法。

本书由严冬松和武建华共同执笔,在编写过程中,为确保内容的正确性,编者参阅了很多资料,参考了大量的相关书籍和网络资源,在此对相关作者表示衷心的感谢。学生邓其锋、凡梦霞、陈俭辉和张紫萱参与了本书的资料整理和程序的调试验证,在此也向他们表示感谢。

在编写过程中,尽管编者做了很大的努力,但限于水平,书中难免有疏漏之处,恳请广大读者批评指正。如有意见和建议,请联系我们,编者的邮箱是 tyands@qq.com。

编 者

2018 年 5 月

目 录

第 1 章 C++/CLI 基础训练	1
1.1 基本知识提要	1
1.1.1 程序设计方法	1
1.1.2 C++ /CLI 基本概念	2
1.1.3 数据类型转换	2
1.1.4 控制台输入输出	3
1.1.5 函数	5
1.1.6 函数的默认形参值	6
1.1.7 函数重载	6
1.2 实训操作内容	7
1.2.1 成绩计算	7
1.2.2 圆、圆球和圆柱	11
1.2.3 最大数	15
1.2.4 圆球和圆柱的体积	17
1.3 常见问题处理	19
思考与练习	20
第 2 章 C++/CLI 编程进阶	22
2.1 基本知识提要	22
2.1.1 引用、跟踪引用及函数参数引用	22
2.1.2 内存托管	23
2.1.3 CLR 数组	24
2.1.4 随机数	27
2.1.5 字符串	28
2.2 实训操作内容	31
2.2.1 放大器与交换器	31
2.2.2 新学员排队	34
2.2.3 扑克发牌模拟	37
2.2.4 分解器	42
思考与练习	44

第 3 章 类与对象	47
3.1 基本知识提要	47
3.1.1 类与对象概述	47
3.1.2 构造函数、析构函数和终结器	50
3.1.3 程序集和类库	52
3.1.4 标量属性与索引属性	52
3.1.5 常用类和结构	53
3.2 实训操作内容	54
3.2.1 立方体	54
3.2.2 类库与协作编程	62
3.2.3 标量属性	64
3.2.4 索引属性	68
思考与练习	71
第 4 章 继承与多态	73
4.1 基本知识提要	73
4.1.1 继承与派生类	73
4.1.2 多态性与虚函数	74
4.1.3 接口	76
4.1.4 委托	77
4.1.5 事件	79
4.2 实训操作内容	80
4.2.1 派生的圆桌	80
4.2.2 几何体	84
4.2.3 英雄无敌	87
4.2.4 吃水果的接口	94
4.2.5 委托与代理	98
4.2.6 按钮事件	100
思考与练习	104
第 5 章 窗体和对话框设计	108
5.1 基本知识提要	108
5.1.1 Windows 编程	108
5.1.2 窗体应用程序设计	109
5.1.3 窗体 Forms 类	110
5.1.4 鼠标事件和键盘事件	114
5.1.5 窗体与对话框	116
5.2 实训操作内容	119

5.2.1	窗体	119
5.2.2	键盘与鼠标事件	122
5.2.3	套圈游戏	128
5.2.4	五运六气	132
	思考与练习	137
第 6 章	常用控件设计	139
6.1	基本知识提要	139
6.1.1	控件的常用属性和事件	139
6.1.2	标签	141
6.1.3	按钮、单选按钮和复选框	141
6.1.4	组框	141
6.1.5	图片框	141
6.1.6	文本框	142
6.1.7	掩码文本框	142
6.1.8	数字旋钮	143
6.1.9	日期时间	143
6.1.10	月历	143
6.1.11	弹出式信息组件	144
6.1.12	链接标签	144
6.1.13	网页导航	145
6.1.14	WebRequest 类和 HttpWebRequest 类	145
6.2	实训操作内容	146
6.2.1	学生成绩	146
6.2.2	学生基本情况	148
6.2.3	用户登录窗体	154
6.2.4	天气预报	158
	思考与练习	163
第 7 章	框条控件设计	164
7.1	基本知识提要	164
7.1.1	列表框	164
7.1.2	组合框	167
7.1.3	可选列表框	168
7.1.4	进度条	168
7.1.5	滚动条	169
7.1.6	滑动条	169
7.1.7	定时器	170

7.1.8 随机数类	170
7.2 实训操作内容	171
7.2.1 学生成绩操作	171
7.2.2 我的星期都去哪了	180
7.2.3 条子的使用	184
7.2.4 抓人游戏	189
思考与练习	193
第 8 章 容器控件设计	195
8.1 基本知识提要	195
8.1.1 图像列表组件	195
8.1.2 选项卡控件	195
8.1.3 列表视图控件	197
8.1.4 树视图控件	199
8.1.5 面板控件	200
8.1.6 切分容器控件	201
8.2 实训操作内容	201
8.2.1 图像显示控制	201
8.2.2 联系人信息管理	205
8.2.3 学校专业列表	211
8.2.4 学生成绩系统	214
思考与练习	223
第 9 章 基本界面设计	225
9.1 基本知识提要	225
9.1.1 菜单	225
9.1.2 工具栏	228
9.1.3 状态栏	230
9.1.4 图标	231
9.1.5 托盘控件	232
9.1.6 串口类及虚拟串口	232
9.1.7 ActiveX 控件	232
9.2 实训操作内容	234
9.2.1 空气监控系统设计 1	234
9.2.2 空气监控系统设计 2	240
9.2.3 空气监控系统设计 3	246
9.2.4 递增数字控件	250
思考与练习	253

第 10 章 多文档界面设计	254
10.1 基本知识提要	254
10.1.1 多文档窗体	254
10.1.2 富文本框控件	257
10.1.3 系统信息的获取	258
10.2 实训操作内容	260
10.2.1 多文档界面设计	260
10.2.2 多文档编辑器	265
10.2.3 一档多视	275
思考与练习	279
第 11 章 文件与文件夹操作	281
11.1 基本知识提要	281
11.1.1 文件和流	281
11.1.2 DirectoryInfo 类	283
11.1.3 Directory 类和 DirectoryInfo 类	283
11.1.4 Path 类	285
11.1.5 File 类和 FileInfo 类	285
11.1.6 流文件读写操作	289
11.2 实训操作内容	293
11.2.1 资源管理器	293
11.2.2 文件及文件夹操作	301
11.2.3 记事本	307
11.2.4 二进制编辑器	313
思考与练习	317
第 12 章 数据库应用编程	319
12.1 基本知识提要	319
12.1.1 ADO.NET 体系结构	319
12.1.2 断开的数据库	324
12.1.3 连接的数据库	326
12.1.4 DataSet 数据操作	326
12.1.5 数据视图和数据绑定	328
12.2 实训操作内容	329
12.2.1 数据库浏览器	329
12.2.2 数据库向导操作	332
12.2.3 数据库的行列操作	344
12.2.4 数据视图和数据绑定	349

思考与练习	355
第 13 章 GDI+图形绘制	358
13.1 基本知识提要	358
13.1.1 GDI+组成	358
13.1.2 常用基本数据结构	359
13.1.3 图形绘制	361
13.1.4 画笔	362
13.1.5 画刷	364
13.1.6 二维图形绘制方法	366
13.2 实训操作内容	370
13.2.1 时钟精灵	370
13.2.2 旋转的风扇	372
13.2.3 成绩统计图	377
13.2.4 贪吃蛇游戏	380
思考与练习	389
第 14 章 GDI+图像处理	391
14.1 基本知识提要	391
14.1.1 图像处理的方法	391
14.1.2 图像的特效处理	395
14.1.3 图像几何变换	395
14.2 实训操作内容	396
14.2.1 图像处理器 1	396
14.2.2 图像处理器 2	402
14.2.3 拼图游戏	410
思考与练习	425
第 15 章 序列化、文本绘制与打印	426
15.1 基本知识提要	426
15.1.1 序列化数据存取	426
15.1.2 文本的图形绘制	428
15.1.3 打印及打印预览	430
15.2 实训操作内容	432
15.2.1 学生成绩管理系统 1	432
15.2.2 学生成绩管理系统 2	440
15.2.3 学生成绩管理系统 3	448
思考与练习	457

第 16 章 多媒体设计及应用程序部署	459
16.1 基本知识提要	459
16.1.1 多媒体文件的播放	459
16.1.2 应用程序的部署	460
16.2 实训操作内容	462
16.2.1 多媒体播放器	462
16.2.2 应用程序部署	467
思考与练习	471
参考文献	472

第1章 C++/CLI 基础训练

实训目的

- 掌握 CLI/CLR 控制台程序的创建和运行。
- 了解命名空间和头文件的作用。
- 掌握数据的输入输出方法和格式。
- 掌握常见错误提示及修改方法,学会使用调试功能。
- 掌握函数默认参数和函数重载的应用。

1.1 基本知识提要

1.1.1 程序设计方法

程序设计方法的发展主要经历了面向过程程序设计和面向对象程序设计两大阶段。

面向过程程序设计(Procedure-Oriented Programming, POP)是一种以过程为中心的编程思想,首先分析并提出解决问题的步骤,然后用函数把这些步骤一步一步实现。在面向过程程序设计中,数据和对数据的操作是分离的。

面向对象程序设计(Object-Oriented Programming, OOP)是将事物对象化,通过对象通信来解决问题。在面向对象程序设计中,数据和对数据的操作是绑定在一起的。

对象是类的实例。每一个对象都应该能够接收数据、处理数据并将数据传达给其他对象。面向对象程序设计将对象作为程序的基本单元,将程序和数据封装在其中,以提高软件的重用性、灵活性和扩展性。

面向对象的三大基本特征如下。

(1) 封装。把客观事物封装成抽象的类,隐藏属性和方法的实现细节,仅对外公开接口。

(2) 继承。子类可以使用父类的所有功能,并且对这些功能进行扩展。继承的过程就是从一般到特殊的过程。

(3) 多态。接口的多种不同的实现方式即为多态。同一操作作用于不同的对象,产生不同的执行结果。在运行时,通过指向基类的指针或引用调用派生类中的虚函数,由此实现多态。

面向对象程序设计提高了程序的灵活性和可维护性,并且在大型项目设计中广为应用。此外,面向对象程序设计要比以往的做法更加便于学习,因为它能够让人们更简单地设计并维护程序,使得程序更加便于分析、设计、理解。

1.1.2 C++ /CLI 基本概念

1. C++ /CLI

C 和 C++ 一直是最有生命力的程序设计语言,提供了丰富的功能、高度的灵活性和强大的底层控制能力。由于 C++ 从 C 中继承而来,所以 C 程序基本上也是 C++ 程序。一般认为,C++ 适合各种规模的编程;而 C 则更适合小规模和高效的编程。

CLI(Common Language Infrastructure)指的是通用语言基础结构,是一种支持动态组件编程模型的多重结构,是一个在底层操作系统与程序之间有效地运行的实时的软件层,可以支持执行程序中的活动类型及与程序相关联的下部结构的操作。C++ /CLI 中的斜线(/)代表 C++ 和 CLI 的捆绑。它是静态 C++ 对象模型到 CLI 的动态组件对象编程模型的捆绑。C++ /CLI 代表托管和本地编程的结合,通过 C++ /CLI 中的标准扩展,C++ 具有了原来没有的动态编程能力以及一系列的 .NET 特性,是 C++ 在 .NET 中的一种编程语言。

公共语言运行时(Common Language Runtime,CLR)是一个运行时环境,也是一种受控的执行环境,用它管理代码的执行并使开发过程变得更加简单。CLR 是 CLI 的 Visual C++ 在 Windows 操作系统的实现版本。

2. 命名空间

命名空间是 .NET 应用程序代码的一种容器,用于对程序代码及其内容进行分类管理。使用命名空间,还可以有效区分具有相同名称的相同代码。命名空间有两种:系统命名空间和用户自定义命名空间,后者可以解决程序中可能出现的名称冲突问题。

使用 using 引入其他命名空间到当前编辑单元,从而可以直接使用被导入命名空间的标识符,而不需要加上完整的限定名称。但是,如果在该命名空间代码外部使用命名空间的名称,就必须写出该命名空间的限定名称。限定名称包括命名空间所有的分层信息,在不同的命名空间级别之间使用双冒号符号(::)。

1.1.3 数据类型转换

数据类型转换有两种方式:隐式转换和显式转换。

1. 隐式转换

隐式转换是在多种类型数据混合运算时由系统自动完成的。转换规则是:由精度低的数据类型转换为精度高的数据类型,隐式转换一般不会失败,也不会造成信息丢失,无法实现从高精度向低精度转换。

隐式转换需要遵循以下规则。

(1) 参与运算的数据类型不一致时,先转换为同一类型,再进行运算。转换规则为从

精度低的类型到精度高的类型转换。

(2) 其他数据类型不能转换为 char 型。

(3) 不存在浮点型(float、double)与 decimal 之间的隐式转换。

(4) 表达式中出现 byte 型与 short 型数据时,必须先转换为 int 型数据。

2. 显式转换

显式转换又称为强制类型转换。显式转换需要用户明确指明要转换的类型。显式转换的一般形式为

(类型说明符)(需要转换的表达式)

显式转换包含所有的隐式转换。但显式转换不一定成功,而且转换过程中有可能出现数据丢失现象。

3. 使用方法进行数据类型的转换

除了可以使用显式转换进行数据转换外,还可以用一些方法实现特定的转换。

(1) Parse 方法。它可以将特定格式的字符串转换为数值型,使用形式为

数值类型名称::Parse(字符串表达式)

凡是数值型数据均可以调用该方法将一个字符串转换为相应的数值类型,但需要保证字符串的格式符合要转换的目的数据类型的格式。

(2) Convert 类的方法。Convert 类提供了一些常用方法,将一种基本数据类型转换为另一种基本数据类型。详见第 3 章的内容。

1.1.4 控制台输入输出

所谓控制台应用程序,就是指那些需要与传统 DOS 操作系统保持某种程序的兼容,同时又不需要为用户提供完善界面的程序,即 DOS 界面的应用程序。

1. 标准 C++ 的输入输出

标准 C++ 的输入输出都是通过对流进行操作来完成的。输入和输出并不是 C++ 语言中的正式组成成分。C 和 C++ 都没有为输入和输出提供专门的语句。输入输出不是由 C++ 本身定义的,而是在编译系统提供的 I/O 库中定义的。

尽管 cin 和 cout 不是 C++ 提供的语句,但是在不致混淆的情况下,为了叙述方便,常常把由 cin 和流提取运算符“>>”实现输入的语句称为输入语句或 cin 语句,把由 cout 和流插入运算符“<<”实现输出的语句称为输出语句或 cout 语句。

cout 语句的一般格式为

```
cout<<表达式 1<<表达式 2<<...<<表达式 n<<endl;
```

cin 语句的一般格式为

```
cin<<变量 1<<变量 2<<...变量 n;
```

C++ 的输出和输入是用流(stream)的方式实现的。

有关流对象 cin、cout 和流运算符的定义等信息是存放在 C++ 的 I/O 库中的,因此如果在程序中使用 cin、cout 和流运算符,就必须使用预处理命令把头文件 iostream 包含到本文件中,并使用命名空间 std:

```
#include<iostream>
using namespace std;
```

2. C++ /CLI 的控制台输入输出的类

在 C++ /CLI 中通过控制台窗口进行数据的输入和结果的输出是使用 System 命名空间的 Console 类的静态方法(函数)实现的。控制台输入输出都是字符流,流从键盘流向控制台窗口称为输入,而流从程序流向控制台窗口称为输出。

为了使用 CLR 类库中的 Console 类,在程序的开始应包含如下两行:

```
using namespace System
```

3. C++ /CLI 控制台输出

Console 类提供了几个静态方法用于在控制台上输出字符串或含有变量的混合文本。其中,该类的 WriteLine 方法用于在控制台上输出一行文本,并自动换行;而 Write 方法则可以输出单个值,但不会自动换行。例如:

```
Console::WriteLine(L"半径为{0}的圆面积为: {1}", 8, 3.14 * 8 * 8);
```

Write 和 WriteLine 方法都可以接收 0 个、1 个或多个参数,这些参数之间通过逗号(,)隔开,通过格式化占位符({})来分别引用所带的参数。如果需要插入更多的参数,则可以增加占位符所对应参数的编号,如{0}、{1}、{2}等,并且编号的顺序可以颠倒。例如:

```
Console::WriteLine(L"第{1}个半径为{2}的圆面积为: {0}", 3.14 * 8 * 8, 1, 8);
```

还可以在格式化占位符中指定输出的格式、对齐方式及精度等。

4. C++ /CLI 控制台输入

使用 ReadLine 和 Read 方法来获取用户在控制台输入的文本给程序,ReadLine 方法是将整行文本存入一个字符串中,当按下 Enter 键时,则结束文本的读取。该方法返回一个 String^ 类型的变量,以表示一个 String 数据类型的引用,其中包含实际读入的字符串。例如:

```
String^ line=Console::ReadLine();
```

由于 ReadLine 方法将以字符串形式返回输入的内容,所以在需要获取从控制台输入的整数和双精度浮点数时,可以通过这些基本数据类型的包装类的 Parse 方法将字符串解析成该类型的值。例如:

```
int value1=Int32::Parse(Console::ReadLine());    //将输入转换为 int 类型的值
double value2=Double::Parse(Console::ReadLine());
                                                //将输入转换为 double 类型的值
```

如果需要按照逐个字符的读取方式输入数据,那么可以调用 Read 方法读取输入的字符,并将其转换成对应的数字值。例如:

```
char ch=Console::Read();    //读取一个字符,并返回该字符的 ASCII 码
int n=Console::Read();
```

1.1.5 函数

在 C++/CLI 程序中,每一个函数的定义都是由 4 个部分组成的,即函数类型、函数名、形式参数表和函数体,其定义的格式如下:

```
<函数类型><函数名>(<形式参数表>)
{
    <若干语句>    //函数体
}
```

1. 函数类型和返回值

函数类型决定了函数的返回值类型,它可以是除数组类型之外的任何有效的 C++/CLI 数据类型,包括引用、指针等。若函数类型为 void,则表示该函数没有返回值。

在函数体中必须通过 return 语句来指定函数的返回值,且返回值的类型应与函数类型相同。return 语句的格式如下:

```
return (表达式);
```

2. 形式参数表

函数定义中,函数头的形式参数又简称为形参。形式参数表中的每一个形参都是由形参的数据类型和形参名构成的。形参个数可以是 0 个,即没有形参,但小括号不能省略。形参也可以是 1 个或多个,多个形参间要用逗号分隔。

3. 函数体

函数的函数体由在一对大括号中的若干条语句组成,用于实现这个函数执行的功能。

4. 函数的调用和声明

调用函数时,先写函数名,然后紧跟括号,括号里是实际调用该函数时所指定的参数,称为实际参数,简称实参。实参与形参相对应,即实参与形参的个数应相等,类型应一致,且按顺序一一对应来传递数据。函数调用的一般格式为

```
<函数名>(<实参表>);
```