

21世纪高等学校计算机**专业**实用规划教材

面向对象程序设计 (C++描述)

许 薇 方修丰 编著

D

C

T

清华大学出版社

21 世纪高等学校计算机**专业**实用规划教材

面向对象程序设计 (C++描述)

许 薇 方修丰 编著

清华大学出版社
北京

内 容 简 介

本书以 C++ 语言为描述工具,描述了面向对象程序设计的主要思想和方法,并对其中的重要概念和机制做了详细论述及 C++ 技术的实现。书中知识编排脉络清晰、主旨明确,用例简洁且针对性强,突出并实现了阅读环节对学习价值的价值。

本书适合于具有结构化程序设计方法知识的读者使用,可作为计算机专业学生面向对象程序设计课程的教材,也可作为从事计算机软件开发人员的参考资料。

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:C++描述/许薇,方修丰编著. —北京:清华大学出版社,2009.9

(21 世纪高等学校计算机专业实用规划教材)

ISBN 978-7-302-20257-8

I. 面… II. ①许… ②方… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 082312 号

责任编辑:郑寅堃 林都嘉

责任校对:时翠兰

责任印制:杨 艳

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:010-62776969, c-service@tup.tsinghua.edu.cn

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

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×260 印 张:13.75 字 数:337 千字

版 次:2009 年 9 月第 1 版 印 次:2009 年 9 月第 1 次印刷

印 数:1~4000

定 价:19.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系
调换。联系电话:(010)62770177 转 3103 产品编号:032482-01

读者意见反馈

亲爱的读者：

感谢您一直以来对清华版计算机教材的支持和爱护。为了今后为您提供更优秀的教材，请您抽出宝贵的时间来填写下面的意见反馈表，以便我们更好地对本教材做进一步改进。同时如果您在使用本教材的过程中遇到了什么问题，或者有什么好的建议，也请您来信告诉我们。

地址：北京市海淀区双清路学研大厦 A 座 602 室 计算机与信息分社营销室 收
邮编：100084 电子邮箱：jsjic@tup.tsinghua.edu.cn
电话：010-62770175-4608/4409 邮购电话：010-62786544

教材名称：面向对象程序设计（C++ 描述）

ISBN 978-7-302-20257-8

个人资料

姓名：_____ 年龄：_____ 所在院校/专业：_____

文化程度：_____ 通信地址：_____

联系电话：_____ 电子信箱：_____

您使用本书是作为：☐指定教材 ☐选用教材 ☐辅导教材 ☐自学教材

您对本书封面设计的满意度：

☐很满意 ☐满意 ☐一般 ☐不满意 改进建议_____

您对本书印刷质量的满意度：

☐很满意 ☐满意 ☐一般 ☐不满意 改进建议_____

您对本书的总体满意度：

从语言质量角度看 ☐很满意 ☐满意 ☐一般 ☐不满意

从科技含量角度看 ☐很满意 ☐满意 ☐一般 ☐不满意

本书最令您满意的是：

☐指导明确 ☐内容充实 ☐讲解详尽 ☐实例丰富

您认为本书在哪些地方应进行修改？（可附页）

您希望本书在哪些方面进行改进？（可附页）

电子教案支持

敬爱的教师：

为了配合本课程的教学需要，本教材配有配套的电子教案(素材)，有需求的教师可以与我们联系，我们将向使用本教材进行教学的教师免费赠送电子教案(素材)，希望有助于教学活动的开展。相关信息请拨打电话 010-62776969 或发送电子邮件至 jsjic@tup.tsinghua.edu.cn 咨询，也可以到清华大学出版社主页(<http://www.tup.com.cn> 或 <http://www.tup.tsinghua.edu.cn>)上查询。

编审委员会成员

(按地区排序)

清华大学

周立柱 教授
覃 征 教授
王建民 教授
刘 强 副教授
冯建华 副教授

北京大学

杨冬青 教授
陈 钟 教授
陈立军 副教授

北京航空航天大学

马殿富 教授
吴超英 副教授
姚淑珍 教授

中国人民大学

王 珊 教授
孟小峰 教授
陈 红 教授

北京师范大学

周明全 教授

北京交通大学

阮秋琦 教授

北京信息工程学院

赵 宏 教授

北京科技大学

孟庆昌 教授

石油大学

杨炳儒 教授

天津大学

陈 明 教授

复旦大学

艾德才 教授

吴立德 教授

吴百锋 教授

杨卫东 副教授

同济大学

苗夺谦 教授

华东理工大学

徐 安 教授

华东师范大学

邵志清 教授

杨宗源 教授

上海大学

应吉康 教授

东华大学

陆 铭 副教授

乐嘉锦 教授

孙 莉 副教授

浙江大学	吴朝晖	教授
	李善平	教授
宁波大学	江宝钊	副教授
南京大学	骆 斌	教授
	黄 强	副教授
南京航空航天大学	黄志球	教授
	秦小麟	教授
南京理工大学	张功萱	教授
南京邮电学院	朱秀昌	教授
苏州大学	王宜怀	教授
	陈建明	副教授
江苏大学	鲍可进	教授
武汉大学	何炎祥	教授
华中科技大学	刘乐善	教授
中南财经政法大学	刘腾红	教授
华中师范大学	叶俊民	教授
	郑世珏	教授
	陈 利	教授
国防科技大学	赵克佳	教授
中南大学	刘卫国	教授
湖南大学	林亚平	教授
	邹北骥	教授
西安交通大学	沈钧毅	教授
	齐 勇	教授
长安大学	巨永峰	教授
哈尔滨工业大学	郭茂祖	教授
吉林大学	徐一平	教授
	毕 强	教授
山东大学	孟祥旭	教授
	郝兴伟	教授
中山大学	潘小轰	教授
厦门大学	冯少荣	教授
仰恩大学	张思民	副教授
云南大学	刘惟一	教授
电子科技大学	刘乃琦	教授
	罗 蕾	教授
重庆邮电学院	王国胤	教授
西南交通大学	曾华燊	教授

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机专业课程领域,以专业基础课为主、专业课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 反映计算机学科的最新发展,总结近年来计算机专业教学的最新成果。内容先进,充分吸收国外先进成果和理念。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,融合先进的教学思想、方法和手段,体现科学性、先进性和系统性,强调对学生实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。专业基础课和专业课教材配套,同一门课程可以有针对不同层次、面向不同应用的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时依靠各课程专家在调查研究本课程教材

建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机专业实用规划教材

联系人:魏江江 weijj@tup.tsinghua.edu.cn

前言

计算机问世以后,在世界范围内,几乎全部的行业和部门管理工作都使用了计算机,因而使管理工作出现了高效率、快节奏,而且节省了很多人力、物力,其经济效益是无法计量的。对于计算机本身来说,这些显著的成就更加取决于对软件的依赖。

将现实世界的情况准确地映射到计算机中,以求解决面临的问题是程序设计所追求的目标。长期以来,人们一直研究科学的方法和技术来解决由需求增加导致的需要依靠计算机来完成的各种问题。面向过程的程序设计方法在一个历史时期内适应了这种情况,成为程序设计的首选。随着问题规模的扩大,面向过程的方法和技术已经不能够适应新的发展。

面向对象方法的基本思想是从现实世界中客观存在的事物出发来构造软件系统,在系统构造中尽可能地运用人类的自然思维方式。其开发本质是应用领域内概念的鉴别和组织,而不是它们在一种程序设计语言中的最终表示。

面向对象语言是一类以对象作为基本程序结构单位的程序设计语言,用于描述的设计是以对象为核心,而对象是程序运行时刻的基本成分,语言中提供了类、继承等成分。面向对象思想的实践依赖于面向对象语言的实现。面向对象方法的发展推动了面向对象语言的不断出现和进步。目前支持面向对象的语言有很多种,其中 C++ 和 Java 最能够体现面向对象的思想。C++ 是一种广泛使用的支持面向对象技术的程序设计语言,同时也保留了 C 语言的灵活性特点。考虑到以 C 语言为代表的面向过程语言的普及性,本书采用了 C++ 作为描述工具,以求达到尽可能强的适应性。

面向对象方法的核心价值体现在类的设计和代码复用理念。运用一种面向对象语言为工具来表述面向对象思想和方法是撰写本书的目的,因此具体的 C++ 语法规则不是本书的论述重点。根据本书的编排,读者们能够掌握面向对象程序设计的基本方法,并学会利用 C++ 程序设计语言编写具有面向对象特征的程序代码,从中体会出面向对象程序设计的精髓。

全书由 8 个章节和 2 个附录构成。第 1 章 C++ 基础知识,简要介绍了 C++ 语言的部分知识,这些知识主要为后面采用 C++ 工具描述面向对象思想和方法提供了技术支持;第 2 章面向对象程序设计概述,主要介绍面向对象程序设计所涉及的基本概念和基本方法,重点表述了面向对象分析、面向对象设计、面向对象实现及面向对象的测试和软件维护;第 3 章类与对象,主要介绍对象和类的产生机制,以及采用 C++ 语言对其实现的方法;第 4 章继承和派生,主要讲述了面向对象方法中代码重用问题,并采用 C++ 语言予以实现;第 5 章多态性,主要讲述在代码复用的基础上如何使程序的表示形式与程序的表达内容尽可能无关;第 6 章模板,模板提供了数据类型的灵活性,并且支持了编译时的语法检查,极大地提高了程序的可重用性;第 7 章异常,主要介绍异常的处理思想和 C++ 的异常处理机

制；第8章流类库，流类库是C++语言利用继承组织类层次的典范，通过介绍流类库和它的使用方法来体现面向对象程序设计思想。附录中对采用Visual C++ 6.0调试程序的方法做了必要的描述。

本书中列举了大量实例，大部分的程序均已在Visual C++ 6.0下调试通过，由于Visual C++ 6.0对标准C++规范的改变，使它不能够完成全部例程的调试，所以采用了C-free4.0调试了部分程序。

与其他面向对象程序设计书籍相比较，本书有以下几个特点：

- 主旨明确，抛开语法细节，直入核心理念。
- 举例精简，力求以简明表述说明内在本质，防止陷入程序细节而妨碍对知识点的理解。
- 环节驱动，章前有导读，章后有总结并辅以思考问答使读者学前有所思学后有收获。
- 脉络清晰，知识章节编排突出循序渐进的原则，使读者能够系统地掌握面向对象技术的基础知识、基本技能。形成严密的逻辑思维能力，进而掌握面向对象程序设计的思想和方法。

本书适合于具有结构化程序设计方法知识的读者使用，可作为计算机专业学生面向对象程序设计课程的教材，也可作为从事计算机软件开发人员的参考资料。

在本书的编写过程中，我们参考和引用了国内外同行的书籍和文献，在此，向被引用文献的作者以及给予本书帮助的所有人士表示衷心的感谢！

尽管作者是多年从事面向对象程序设计教学的一线教师，有着比较丰富的教学经验，由于面向对象技术的快速发展，我们难以对面向对象程序设计的最新成果加以全面表述，更由于作者的学识水平限制，书中尚有不足和错误之处，恳请读者予以批评指正，以便再版时修订。

编 者

2009年4月

目 录

第 1 章 C++ 基础知识	1
1.1 面向对象的程序设计语言	1
1.1.1 C++ 语言	1
1.1.2 Java 语言	3
1.1.3 Visual Basic 语言	4
1.2 变量	4
1.2.1 变量的定义方法	5
1.2.2 const 修饰符	5
1.2.3 引用类型	6
1.3 函数	9
1.3.1 函数原型	9
1.3.2 内联函数	10
1.3.3 带默认参数的函数	11
1.3.4 函数的参数传递	12
1.3.5 函数重载	15
1.4 名字空间	16
1.5 动态内存分配	17
1.5.1 内存申请	17
1.5.2 内存释放	18
1.6 作用域	19
1.6.1 生存期	20
1.6.2 全局和局部变量	20
1.7 多文件结构	22
本章小结	23
思考与练习	23
第 2 章 面向对象程序设计概述	24
2.1 面向对象程序设计的基本概念	24
2.1.1 面向对象方法的产生	24

2.1.2	面向对象方法的方法与结构	25
2.1.3	类和对象的概念	26
2.1.4	消息与事件的概念	27
2.2	面向对象程序设计的特点	27
2.2.1	抽象性	27
2.2.2	封装性	28
2.2.3	继承性	28
2.2.4	多态性	29
2.3	面向对象的软件开发	30
2.3.1	面向对象分析	30
2.3.2	面向对象设计	34
2.3.3	面向对象实现	34
2.3.4	面向对象测试	35
2.3.5	面向对象软件维护	35
	本章小结	35
	思考与练习	36

第3章 类与对象 37

3.1	类	37
3.1.1	类的定义	37
3.1.2	访问控制	40
3.1.3	成员变量	41
3.1.4	成员函数	42
3.2	构造函数与析构函数	48
3.2.1	构造函数	49
3.2.2	析构函数	50
3.2.3	拷贝构造函数	53
3.2.4	浅拷贝与深拷贝	54
3.2.5	构造函数和析构函数的调用过程	57
3.3	对象	59
3.3.1	对象的定义	59
3.3.2	类成员的访问	59
3.4	子对象	62
3.4.1	组合模式	62
3.4.2	子对象和构造函数设计	64
3.4.3	内部类	66
3.5	静态成员	69
3.5.1	定义与引用	69

3.5.2	静态数据成员	69
3.5.3	静态成员函数	70
3.6	友元	71
3.6.1	友元的作用	72
3.6.2	友元函数	72
3.6.3	友元类	73
3.6.4	友元成员函数	74
3.7	设计举例	75
	本章小结	81
	思考与练习	81
第 4 章	继承与派生	82
4.1	继承和派生的概念	82
4.1.1	继承的层次结构	82
4.1.2	继承的作用	83
4.2	派生类	84
4.2.1	派生类的定义	84
4.2.2	派生类的三种继承方式	84
4.2.3	派生类对基类成员函数的继承	89
4.2.4	派生类的构造函数和析构函数	92
4.3	多重继承	94
4.3.1	多重继承的设计方法	95
4.3.2	多重继承的二义性问题	96
4.3.3	虚基类	99
4.4	赋值兼容规则	105
4.5	设计举例	107
	本章小结	120
	思考与练习	120
第 5 章	多态性	121
5.1	多态性的实现类型	121
5.2	联编	122
5.2.1	静态联编	122
5.2.2	动态联编	124
5.3	虚函数	124
5.3.1	虚函数的声明	125
5.3.2	虚函数的调用	125
5.3.3	虚析构函数	127
5.4	抽象类	129

5.4.1	纯虚函数的定义	129
5.4.2	抽象类的使用	129
5.5	运算符重载	132
5.5.1	运算符重载规则	132
5.5.2	运算符重载为类的成员函数	132
5.5.3	运算符重载为类的友元函数	134
5.6	设计举例	135
	本章小结	145
	思考与练习	146

第6章 模板 147

6.1	模板概述	147
6.1.1	参数多态性	147
6.1.2	模板	147
6.2	函数模板	148
6.2.1	函数模板的定义	149
6.2.2	函数模板的使用	149
6.2.3	模板函数的重载	150
6.3	类模板	151
6.3.1	类模板的定义	151
6.3.2	类模板的使用	152
6.3.3	类模板的继承	155
6.4	设计举例	156
	本章小结	163
	思考与练习	163

第7章 异常处理 164

7.1	异常概述	164
7.1.1	异常的基本类型	164
7.1.2	传统的异常处理方法	165
7.2	C++的异常处理方法	166
7.2.1	基本的异常处理方法	166
7.2.2	多个异常的处理方法	169
7.3	非空异常类的设计	171
7.4	异常抛出和处理的两种方式	173
7.4.1	自身处理方式	175
7.4.2	外传处理方式	175
	本章小结	175
	思考与练习	176

第 8 章 流类库..... 177

8.1 C++ 基本 I/O 流库的层次结构 177

8.1.1 I/O 流类的概念 177

8.1.2 ios 类的层次关系 178

8.1.3 streambuf 类的层次关系 178

8.2 I/O 格式控制 179

8.2.1 ios 类成员函数的格式控制 179

8.2.2 操作符的格式控制..... 181

8.3 文件的 I/O 操作 183

8.3.1 文件的打开与关闭..... 184

8.3.2 文本文件的输入输出..... 186

8.3.3 二进制文件的输入输出..... 188

8.3.4 随机文件的输入输出..... 189

8.4 自定义数据类型的 I/O 操作 191

8.4.1 输出运算符重载..... 191

8.4.2 输入运算符重载..... 191

本章小结..... 192

思考与练习..... 193

附录 A 基本 ASCII 字符集..... 194

附录 B Visual C++ 下的程序调试 195

B.1 Visual C++ 6.0(VC)用户界面 195

B.2 创建 .cpp 文件 195

B.3 编译和链接 196

B.4 完整地建立、编译和运行一个 C++ 程序 197

B.5 调试过程 197

参考文献..... 202

C++ 是一种面向对象的高级程序设计语言,它内容丰富、功能强大,具有良好的数据封装性,其核心的价值体现在对面向对象技术的支持。本章主要对书中所涉及的 C++ 语法, C++ 语言的诞生、发展及本书后继章节中用到的语法进行简单的介绍,为学生学习面向对象技术打下良好的基础。

本章导读

- 面向对象的程序设计语言
- 变量
- 函数
- 名字空间
- 动态内存分配
- 作用域
- 多文件结构

1.1 面向对象的程序设计语言

软件系统可靠地运行可以给人们带来各种效率的提高,程序设计人员根据用户要求建造出软件系统或者建造系统中的部分软件的过程称为软件开发。伴随着软件开发技术和软件开发理论的发展,进行软件开发时采用的程序设计语言也会随之同步发展,面向对象的开发理论和技术逐渐成为程序设计人员开发软件的首选。

1.1.1 C++ 语言

C++ 是一种静态数据类型检查的支持多重编程范式的通用程序设计语言。它支持过程化程序设计、数据抽象、面向对象程序设计、制作图标、泛型程序设计等多种程序设计风格,是目前使用非常广泛的一种计算机编程语言。

美国 AT&T 贝尔实验室的本贾尼·斯特劳斯特卢普(Bjarne Stroustrup)博士在 20 世纪 80 年代初期发明了 C,撰写了它的早期定义并且规范化了 C,设计了关于这门语言主要的功能,并实现了具有类开发功能的程序设计语言(最初这种语言被称做 C with Classes)。一开始 C++ 是作为 C 语言的增强版出现的,从给 C 语言增加类开始,不断地增加新特性。虚函数(virtual function)、运算符重载(operator overloading)、多重继承(multiple inheritance)、模板(template)、异常(exception)、RTTI、名字空间(name space)逐渐被加入标准。1998 年国际标准化组织(ISO)颁布了 C++ 程序设计语言的国际标准 ISO/IEC 14882-

1998。C++是具有国际标准的编程语言,通常称做 ANSI/ISO C++。1998 年是 C++ 标准委员会成立的第一年,以后每 5 年视实际需要更新一次标准,下一次标准更新将是在 2009 年,目前一般称该标准为 C++0x。遗憾的是,由于 C++ 语言过于复杂,以及它经历了长年的演变,直到 2004 年只有少数几个编译器完全符合这个标准(这么说也是不完全正确的,事实上,至今为止没有任何一款编译器完全支持 ISO C++)。

另外,就目前学习 C++ 而言,可以认为它是一门独立的语言;C++ 并不依赖 C 语言,可以完全不学 C 语言,而直接学习 C++。根据《C++ 编程思想》(*Thinking in C++*)一书所评述的,C++ 与 C 的效率往往相差在正负 5% 之间。所以有人认为在大多数场合 C++ 完全可以取代 C 语言(然而在单片机等需要谨慎利用空间、直接操作硬件的地方还是要使用 C 语言)。

C++ 语言发展大概可以分为三个阶段:第一阶段从 20 世纪 80 年代到 1995 年。这一阶段 C++ 语言基本上传统类型上的面向对象语言,并且凭借着接近 C 语言的效率,在工业界使用的开发语言中占据了相当大份额;第二阶段从 1995 年到 2000 年,这一阶段由于标准模板库(STL)和后来的 Boost 等程序库的出现,泛型程序设计在 C++ 中占据了越来越多的比重。当然,同时由于 Java、C# 等语言的出现和硬件价格的大规模下降,C++ 受到了一定的冲击;第三阶段从 2000 年至今,由于以 Loki、MPL 等程序库为代表的产生式编程和模板元编程的出现,C++ 出现了发展历史上又一个新的高峰,这些新技术的出现以及和原有技术的融合,使 C++ 已经成为当今主流程序设计语言中最复杂的一员。

C++ 语言继承了 C 语言原有的精髓,如高效性、灵活性,增加了对开发大型软件的非常有效的面向对象的特征,弥补了 C 语言不支持代码重用、不宜开发大型软件的不足,成为一种既可用于表现过程模型,又可用于表现对象模型的优秀的程序设计语言。

每个 C++ 程序都包含一个或多个函数(function),其中的一个函数的名称必须是 main。一个函数由一系列执行函数功能的语句(statement)组成。操作系统通过调用 main 函数来执行程序。这个函数执行它的组成语句并给操作系统返回一个值。

下面是一个简单的 main 函数:

```
int main()
{
    return 0;
}
```

操作系统利用 main 函数返回的值判断程序执行成功还是失败。返回 0 表明成功了。

main 函数很特殊,其中最重要的是这个函数必须存在于每个 C++ 程序中,而且它是操作系统明确调用的唯一一个函数。

定义 main 函数的方式与定义其他函数是一样的。函数定义说明了四个部分:返回类型、函数名称、括号中带有的参数列表(可能为空)和函数主体。main 函数可能包含一组参数。

上面的 main 函数需要一个 int 类型的返回类型,它表示整型。整型是内建的(built-in)类型,这意味着这种类型是语言所定义的。

函数定义的最后一部分(函数主体),是一组以波浪括号(curly brace)开始,以波浪反括号结束的语句块: