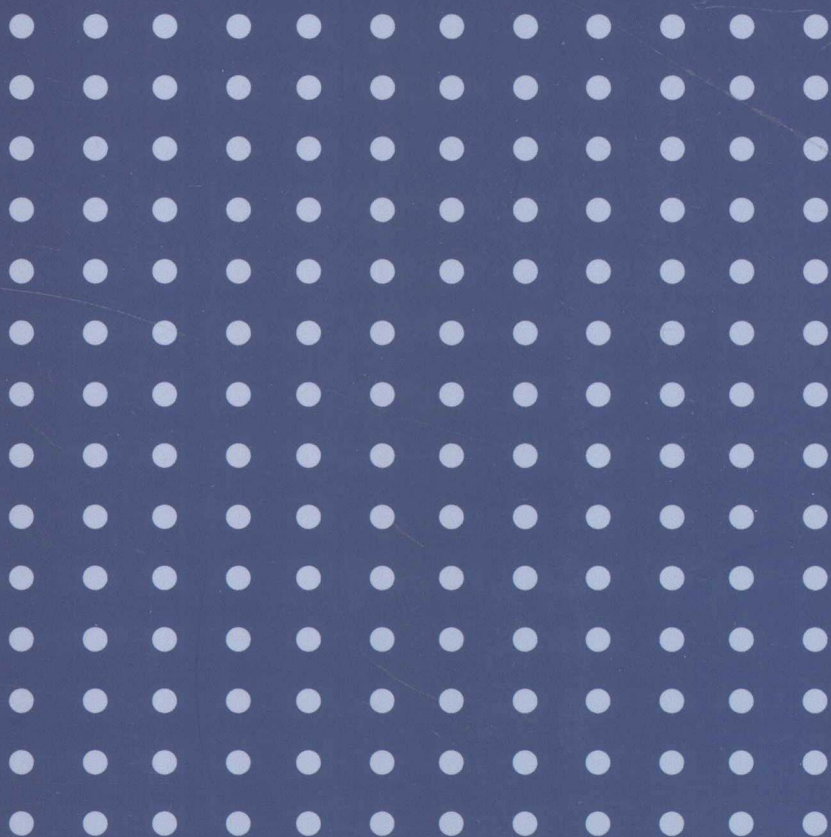


重点大学计算机专业系列教材

# C++面向对象程序设计 (第2版)

龚晓庆 付丽娜 朱新懿 李康 编著

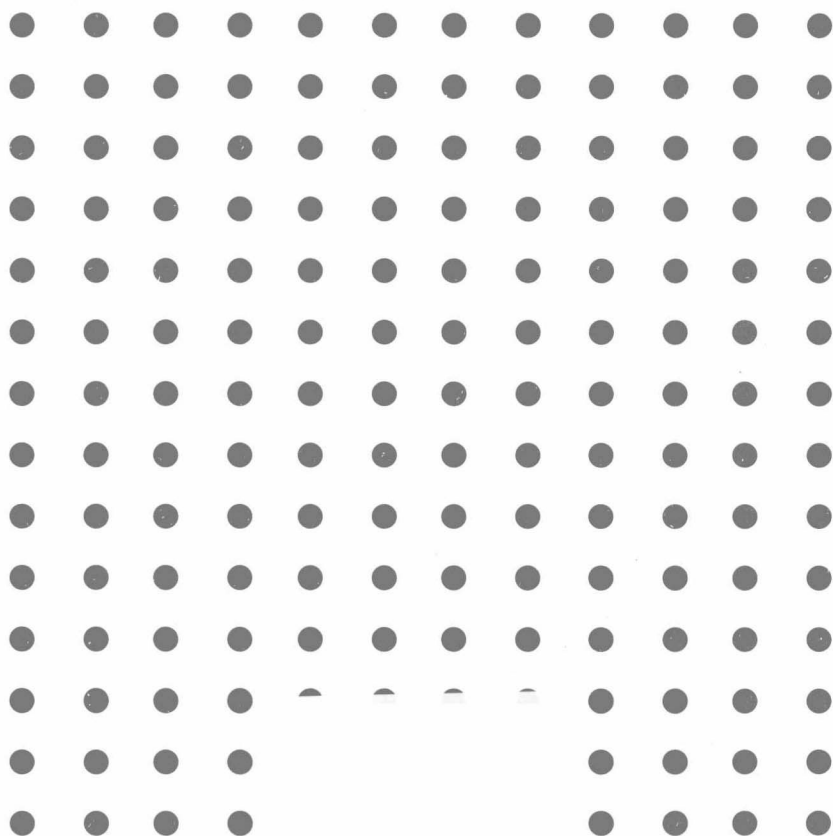


清华大学出版社

重点大学计算机专业系列教材

# C++面向对象程序设计 (第2版)

龚晓庆 付丽娜 朱新懿 李康 编著



清华大学出版社

北京

## 内 容 简 介

本书在第1版的基础上针对新的C++11标准重新撰写,讲解如何用C++11编写面向对象程序。本书以面向对象概念为主线索,内容由浅入深,主要包括:面向对象基础,C++语言概览和语言基础,复合类型,函数,类和对象,构造函数和析构函数,运算符重载,组合与继承,虚函数与多态性,模板与泛型编程,标准库容器和异常处理。

本书内容体系组织符合高校课程开设特点,适合作为高等院校计算机及相关专业本科生的C++程序设计教材,也可作为学习C++和面向对象程序设计的参考读物。

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++面向对象程序设计 / 龚晓庆等编著. —2版. —北京:清华大学出版社, 2017  
(重点大学计算机专业系列教材)  
ISBN 978-7-302-45883-8

I. ①C… II. ①龚… 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(2016)第219744号

责任编辑:郑寅堃 张爱华

封面设计:常雪影

责任校对:焦丽丽

责任印制:王静怡

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](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈: 010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

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: 185mm×260mm

印 张: 31

字 数: 752千字

版 次: 2011年3月第1版

2017年2月第2版

印 次: 2017年2月第1次印刷

印 数: 1~2000

定 价: 59.50元

产品编号: 068406-01

## 第2版前言

C++是一种通用程序设计语言，支持数据抽象、面向对象程序设计和泛型程序设计，并支持在这些风格约束之下的传统C程序设计技术。C++是目前使用最广泛的编程语言之一，尤其适用于系统程序和大型应用程序的设计。C++11标准出现后，语言机制的增强和标准库的完善为C++的编程风格带来了新的变化。

本书介绍如何用C++语言进行面向对象程序设计，在第1版的基础上针对C++11标准重新编写。内容由浅入深，适合所有对C++程序设计感兴趣的读者。如果已经学习过C语言或其他程序设计语言，对阅读和理解本书会有一定的帮助。

本书强调C++语言的实用性，提倡从语法、语义和语用这三个层面来学习和理解C++语言。除了详细解释C++的语法概念及其语义之外，更着重于C++的语用知识：在特定情况下应该使用何种语法结构，用它们来解决什么样的程序设计问题。部分章后面的习题部分还增加了一些软件公司的C++面试题作为思考题，希望能够使读者更广泛和深入地了解C++在实际中的应用。

本书共有13章，大致分为4部分：C++语言基础（第2~5章）、C++面向对象程序设计（第1章、第6~10章）、模板和泛型编程（第11~第12章）和异常处理（第13章）。具体章节的组织和内容如下（另附本书主要内容与章节组织导图）。

第1章介绍面向对象的基本概念和背景知识。

第2章介绍C++语言的特点和标准化现状，重点介绍C++程序的结构和编译方式。第3章介绍C++的内置基本数据类型、运算符、表达式和语句等基本语法结构。第4章详细介绍C++的复合类型和一些常用的标准库类型。第5章介绍C++的函数机制、命名空间和作用域。

第6~8章介绍C++的面向对象编程基础，包括类、对象、构造函数、析构函数和运算符重载等概念。第9章介绍在类基础之上的高级面向对象编程技术，讲述组合和继承的语法及应用。第10章介绍在类的继承层次中实现多态性的编程技术。

第11章介绍C++的模板机制和泛型程序设计。第12章介绍C++11的标准库容器类型和泛型算法。

第13章介绍C++的异常和其他错误处理机制。

本书由西北大学的龚晓庆、付丽娜、朱新懿和李康编写，本书的编写工作得到了西北大学信息科学与技术学院和软件学院的各位老师和同学的支持与帮助，在此表示感谢。

希望本书对读者学习C++语言有所助益，也希望有机会与各位读者一起探讨C++学习和应用中的问题。限于作者水平，书中难免有不妥之处，敬请各位读者批评指正。作者的电子邮件地址：gxq@nwu.edu.cn。

龚晓庆 付丽娜 朱新懿 李康

2016年夏于西北大学

## 本书主要内容与章节组织导图



# 第1版前言

C++是一种通用程序设计语言，支持数据抽象、面向对象程序设计和泛型程序设计，并支持在这些风格约束之下的传统C程序设计技术。C++是目前使用最广泛的编程语言之一，尤其适用于系统程序和大型应用程序的设计。

本书介绍如何用C++语言进行面向对象程序设计，内容由浅入深，适合所有对C++程序设计感兴趣的读者。如果已经学习过C或其他程序设计语言，对阅读和理解本书会有一定的帮助。

本书强调C++语言的实用性。像学习自然语言一样，可以从语法、语义和语用这3个层面来学习和理解C++语言。本书除了详细解释C++的语法概念及其语义之外，更着重于C++的运用知识：在特定情况下应该使用何种语法结构，用它们来解决什么样的程序设计问题。在各章的习题部分还增加了一些软件公司的C++面试题，使读者对C++在实际中的应用有更加广泛和深入的了解。

本书共有12章，大致分为5个部分：C++语言基础知识（第2章~第4章）、C++面向对象程序设计（第1章、第5章~第8章）、模板（第9章）、异常处理（第10章）和C++标准库（第11章~第12章）。具体章节的组织和内容如下：

第1章介绍面向对象的基本概念和背景知识。

第2章是对C++语言基本特点的概览，并介绍C++程序的结构和编译方式。第3章介绍C++的数据类型、运算符、表达式和语句等基本语法结构。第4章详细介绍C++的函数机制、名字空间和作用域。

第5章和第6章介绍C++的面向对象编程基础，包括类、对象、构造函数、析构函数和运算符重载等概念。第7章介绍在类基础之上的高级面向对象编程技术，讲述包含和继承的语法及应用。第8章介绍在类的继承层次中实现多态性的编程技术。

第9章介绍C++的模板机制和泛型程序设计。

第10章介绍C++的异常和其他错误处理机制。

第11章介绍C++标准库提供的I/O流、文件流和字符串流的概念及其

使用。第 12 章介绍 C++ 的常用标准容器、标准算法和迭代器的概念及其使用。

本书由西北大学的龚晓庆、付丽娜、朱新懿等编写。本书的初稿已连续数年作为西北大学信息学院和软件学院的“面向对象技术与 C++”课程讲义使用。本书的编写工作得到了信息学院和软件学院的各位领导与老师的大力支持与帮助，在此表示感谢。

希望本书对读者学习 C++ 有所助益，也希望有机会与各位读者一起探讨 C++ 学习和应用中的问题。限于作者水平，书中或有论述不妥甚至错误之处，敬请各位读者批评指正。

作者的电子邮件地址：gxq@nwu.edu.cn

龚晓庆 付丽娜 朱新懿  
2010 年初秋于西北大学

## 目录

|                  |    |
|------------------|----|
| 第 1 章 面向对象基础     | 1  |
| 1.1 程序设计范型       | 1  |
| 1.2 面向对象的基本概念    | 2  |
| 1.2.1 对象和类       | 3  |
| 1.2.2 封装和信息/实现隐藏 | 4  |
| 1.2.3 接口、实现和消息传递 | 6  |
| 1.2.4 继承         | 8  |
| 1.2.5 多态性        | 10 |
| 1.2.6 类之间的关系     | 12 |
| 1.3 面向对象程序的特点    | 13 |
| 1.4 面向对象方法发展简史   | 14 |
| 1.5 小结           | 16 |
| 1.6 习题           | 16 |
| 第 2 章 C++语言概览    | 17 |
| 2.1 C++语言的特点     | 17 |
| 2.1.1 C++的发展和标准化 | 17 |
| 2.1.2 C++的特点     | 18 |
| 2.2 第一个 C++程序    | 19 |
| 2.2.1 程序基本结构     | 20 |
| 2.2.2 程序的编译和运行   | 21 |
| 2.3 输入和输出        | 22 |
| 2.3.1 标准输入输出     | 23 |
| 2.3.2 注释         | 24 |

|        |                   |    |
|--------|-------------------|----|
| 2.4    | 集成开发环境的使用         | 25 |
| 2.4.1  | 使用 IDE 开发 C++ 程序  | 25 |
| 2.4.2  | Code::Blocks 使用示例 | 26 |
| 2.5    | C++ 语言特性概览        | 32 |
| 2.6    | 小结                | 33 |
| 2.7    | 习题                | 33 |
| 第3章    | C++ 语言基础          | 35 |
| 3.1    | 基本内置类型            | 36 |
| 3.1.1  | 算术类型              | 37 |
| 3.1.2  | 带符号和无符号类型         | 39 |
| 3.1.3  | 字面值常量             | 40 |
| 3.1.4  | 数据的输入和输出          | 42 |
| 3.2    | 变量和常量             | 45 |
| 3.2.1  | 变量定义              | 45 |
| 3.2.2  | 标识符               | 45 |
| 3.2.3  | 初始化               | 46 |
| 3.2.4  | 赋值                | 47 |
| 3.2.5  | 类型转换              | 48 |
| 3.2.6  | 变量声明              | 48 |
| 3.2.7  | 名字的作用域            | 49 |
| 3.2.8  | const 对象          | 51 |
| 3.2.9  | 常量表达式和 constexpr  | 52 |
| 3.2.10 | auto 和 decltype   | 53 |
| 3.3    | 运算符和表达式           | 54 |
| 3.3.1  | 基本概念              | 54 |
| 3.3.2  | 算术运算符             | 55 |
| 3.3.3  | 关系和逻辑运算符          | 55 |
| 3.3.4  | 赋值运算符             | 56 |
| 3.3.5  | 自增和自减             | 57 |
| 3.3.6  | 位运算符              | 57 |
| 3.3.7  | sizeof 运算符        | 58 |
| 3.3.8  | 条件运算符             | 58 |
| 3.3.9  | 逗号运算符             | 59 |
| 3.3.10 | 类型转换              | 59 |
| 3.4    | 语句                | 62 |
| 3.4.1  | 简单语句和复合语句         | 62 |
| 3.4.2  | 声明语句              | 63 |

|              |                           |           |
|--------------|---------------------------|-----------|
| 3.4.3        | if 语句 .....               | 63        |
| 3.4.4        | switch 语句 .....           | 65        |
| 3.4.5        | while 语句 .....            | 69        |
| 3.4.6        | for 语句 .....              | 70        |
| 3.4.7        | do-while 语句 .....         | 72        |
| 3.4.8        | break 和 continue 语句 ..... | 74        |
| 3.4.9        | goto 语句 .....             | 75        |
| 3.5          | 编程示例: 显示素数 .....          | 76        |
| 3.6          | 小结 .....                  | 78        |
| 3.7          | 习题 .....                  | 78        |
| <b>第 4 章</b> | <b>复合类型 .....</b>         | <b>83</b> |
| 4.1          | 指针和引用 .....               | 83        |
| 4.1.1        | 指针 .....                  | 84        |
| 4.1.2        | new 和 delete .....        | 87        |
| 4.1.3        | 引用 .....                  | 90        |
| 4.1.4        | 右值引用 .....                | 92        |
| 4.1.5        | const 限定指针和引用 .....       | 93        |
| 4.2          | 结构体、联合和枚举 .....           | 97        |
| 4.2.1        | 结构体 .....                 | 97        |
| 4.2.2        | 联合 .....                  | 99        |
| 4.2.3        | 枚举 .....                  | 100       |
| 4.3          | 内置数组 .....                | 101       |
| 4.3.1        | 定义和初始化数组 .....            | 101       |
| 4.3.2        | 访问数组元素 .....              | 102       |
| 4.3.3        | 数组与指针 .....               | 103       |
| 4.3.4        | 字符数组和 C 风格字符串 .....       | 104       |
| 4.4          | 标准库类型 string .....        | 105       |
| 4.4.1        | 定义和初始化 string 对象 .....    | 106       |
| 4.4.2        | string 对象上的操作 .....       | 106       |
| 4.4.3        | 处理 string 对象中的字符 .....    | 109       |
| 4.5          | 标准库类型 vector .....        | 112       |
| 4.5.1        | 定义和初始化 vector .....       | 113       |
| 4.5.2        | 向 vector 中添加元素 .....      | 114       |
| 4.5.3        | 访问 vector 中的元素 .....      | 115       |
| 4.6          | 迭代器 .....                 | 116       |
| 4.7          | 编程示例: 文件数据处理 .....        | 118       |
| 4.7.1        | 算法和数据结构 .....             | 118       |
| 4.7.2        | 文件读写 .....                | 119       |

|       |                 |     |
|-------|-----------------|-----|
| 4.7.3 | 字符串流            | 121 |
| 4.7.4 | 完成的程序           | 123 |
| 4.8   | 小结              | 124 |
| 4.9   | 习题              | 125 |
| 第5章   | 函数              | 127 |
| 5.1   | 函数基础            | 127 |
| 5.1.1 | 函数定义            | 128 |
| 5.1.2 | 函数调用            | 128 |
| 5.1.3 | 函数声明            | 130 |
| 5.1.4 | 递归函数            | 131 |
| 5.2   | 参数传递            | 131 |
| 5.2.1 | 按值传递            | 132 |
| 5.2.2 | 按引用传递           | 135 |
| 5.2.3 | 参数传递方式的选择       | 138 |
| 5.2.4 | 数组参数            | 138 |
| 5.2.5 | main()函数的参数     | 138 |
| 5.2.6 | 不定个数的参数         | 139 |
| 5.3   | 返回类型和 return 语句 | 140 |
| 5.3.1 | 返回值             | 141 |
| 5.3.2 | 返回引用            | 141 |
| 5.3.3 | 返回列表            | 143 |
| 5.3.4 | main()函数的返回值    | 143 |
| 5.3.5 | 尾置返回类型          | 144 |
| 5.4   | 函数重载            | 144 |
| 5.4.1 | 重载函数            | 144 |
| 5.4.2 | 重载函数的调用         | 145 |
| 5.4.3 | 重载函数的判断         | 146 |
| 5.4.4 | 重载函数解析          | 147 |
| 5.5   | 特殊用途的函数特征       | 151 |
| 5.5.1 | 默认实参            | 151 |
| 5.5.2 | inline 函数       | 152 |
| 5.5.3 | constexpr 函数    | 153 |
| 5.6   | 函数指针            | 155 |
| 5.6.1 | 定义函数指针          | 155 |
| 5.6.2 | 使用函数指针          | 155 |
| 5.6.3 | 函数指针的数组         | 156 |
| 5.6.4 | 函数指针形参          | 157 |
| 5.7   | 作用域和存储类别        | 158 |

|        |                  |     |
|--------|------------------|-----|
| 5.7.1  | 作用域              | 159 |
| 5.7.2  | 存储类别和存储空间分配      | 163 |
| 5.8    | namespace        | 164 |
| 5.8.1  | 命名空间的定义          | 164 |
| 5.8.2  | 命名空间成员的使用        | 166 |
| 5.8.3  | 标准命名空间 std       | 168 |
| 5.9    | 程序代码组织           | 169 |
| 5.9.1  | 声明和头文件包含         | 169 |
| 5.9.2  | 函数代码的组织          | 169 |
| 5.9.3  | 命名空间的代码组织        | 170 |
| 5.9.4  | 链接指示符: extern“C” | 171 |
| 5.10   | 设计高质量的函数         | 171 |
| 5.10.1 | 创建函数的理由          | 171 |
| 5.10.2 | 函数的命名            | 172 |
| 5.10.3 | 如何使用函数的参数        | 173 |
| 5.10.4 | 设置函数的返回值         | 174 |
| 5.11   | 小结               | 174 |
| 5.12   | 习题               | 175 |
| 第 6 章  | 类和对象             | 180 |
| 6.1    | 类的定义             | 180 |
| 6.1.1  | 基本语言定义的 ADT      | 181 |
| 6.1.2  | 数据成员与成员函数        | 182 |
| 6.1.3  | 数据成员的类内初始化       | 183 |
| 6.1.4  | 成员函数的类外定义        | 183 |
| 6.1.5  | 类代码的组织           | 184 |
| 6.1.6  | 包含守卫             | 185 |
| 6.2    | 访问控制和封装          | 186 |
| 6.2.1  | 信息隐藏的必要性         | 186 |
| 6.2.2  | 访问限定符            | 187 |
| 6.2.3  | 类和对象             | 188 |
| 6.2.4  | this 指针          | 191 |
| 6.2.5  | 访问器和修改器          | 192 |
| 6.2.6  | 友元               | 193 |
| 6.2.7  | 进一步的隐藏           | 195 |
| 6.3    | 构造函数和析构函数        | 199 |
| 6.3.1  | 构造函数             | 200 |
| 6.3.2  | 构造函数初始化列表        | 201 |
| 6.3.3  | 委托构造函数           | 202 |

|       |                |     |
|-------|----------------|-----|
| 6.3.4 | 析构函数           | 204 |
| 6.4   | const 成员       | 206 |
| 6.4.1 | const 数据成员     | 206 |
| 6.4.2 | const 成员函数     | 207 |
| 6.4.3 | mutable 成员     | 209 |
| 6.4.4 | const 用法小结     | 211 |
| 6.5   | static 成员      | 212 |
| 6.5.1 | static 数据成员    | 213 |
| 6.5.2 | static 成员函数    | 215 |
| 6.5.3 | 单件模式           | 216 |
| 6.5.4 | static 用法小结    | 217 |
| 6.6   | 指向成员的指针        | 217 |
| 6.6.1 | 数据成员的指针        | 217 |
| 6.6.2 | 成员函数的指针        | 219 |
| 6.7   | 类设计的例子         | 220 |
| 6.7.1 | 类的设计           | 221 |
| 6.7.2 | 类的 UML 表示法     | 221 |
| 6.7.3 | 动态字符栈类         | 222 |
| 6.7.4 | 字符串类           | 223 |
| 6.7.5 | 单链表类           | 226 |
| 6.8   | 小结             | 228 |
| 6.9   | 习题             | 229 |
| 第 7 章 | 对象的初始化、复制和销毁   | 235 |
| 7.1   | 对象的初始化和销毁      | 235 |
| 7.1.1 | 对象的初始化         | 238 |
| 7.1.2 | 默认构造函数         | 239 |
| 7.1.3 | 隐式类型转换构造函数     | 240 |
| 7.1.4 | 析构函数           | 241 |
| 7.1.5 | 拷贝控制成员         | 243 |
| 7.2   | 拷贝构造函数         | 245 |
| 7.3   | 拷贝赋值运算符        | 249 |
| 7.4   | 对象复制和移动        | 253 |
| 7.4.1 | 移动构造函数和移动赋值运算符 | 254 |
| 7.4.2 | 成员函数的复制和移动版本   | 258 |
| 7.4.3 | 是否要定义拷贝控制成员    | 259 |
| 7.4.4 | 交换操作           | 259 |
| 7.5   | 编译器合成的成员函数     | 264 |

|              |                  |            |
|--------------|------------------|------------|
| 7.5.1        | =delete          | 266        |
| 7.5.2        | =default         | 266        |
| 7.6          | 引用计数和写时复制技术      | 267        |
| 7.6.1        | 内置指针实现引用计数       | 268        |
| 7.6.2        | 智能指针             | 270        |
| 7.6.3        | 智能指针实现引用计数       | 274        |
| 7.7          | 小结               | 276        |
| 7.8          | 习题               | 276        |
| <b>第 8 章</b> | <b>运算符重载</b>     | <b>279</b> |
| 8.1          | 基本概念             | 279        |
| 8.1.1        | 运算符函数            | 280        |
| 8.1.2        | 运算符重载的限制         | 280        |
| 8.1.3        | 慎用运算符重载          | 281        |
| 8.2          | 常用运算符的重载         | 282        |
| 8.2.1        | 一元运算符            | 282        |
| 8.2.2        | 二元运算符            | 285        |
| 8.2.3        | 运算符函数的参数和返回类型    | 289        |
| 8.2.4        | 非成员运算符和成员运算符     | 290        |
| 8.2.5        | 重载输入输出运算符        | 290        |
| 8.2.6        | 重载赋值运算符          | 292        |
| 8.3          | 重载下标运算符          | 292        |
| 8.4          | 用户定义的类型转换        | 294        |
| 8.4.1        | 类型转换运算符          | 294        |
| 8.4.2        | 自动类型转换可能引起的二义性问题 | 297        |
| 8.5          | 函数调用运算符          | 298        |
| 8.5.1        | 函数对象             | 298        |
| 8.5.2        | lambda 函数        | 299        |
| 8.5.3        | 标准库定义的函数对象       | 302        |
| 8.5.4        | 标准库函数 bind       | 303        |
| 8.6          | 小结               | 304        |
| 8.7          | 习题               | 305        |
| <b>第 9 章</b> | <b>组合与继承</b>     | <b>307</b> |
| 9.1          | 组合——复用类的实现       | 307        |
| 9.1.1        | 对象成员与组合关系        | 308        |
| 9.1.2        | 对象成员的初始化         | 310        |
| 9.1.3        | 复用类的实现           | 311        |
| 9.1.4        | 指针成员与聚合关系        | 312        |

|        |                      |     |
|--------|----------------------|-----|
| 9.1.5  | 指针成员与关联关系            | 314 |
| 9.2    | 继承——复用类的接口           | 315 |
| 9.2.1  | 继承的语法                | 315 |
| 9.2.2  | 派生类成员的访问控制           | 316 |
| 9.2.3  | 公有继承和私有继承            | 320 |
| 9.2.4  | 派生类对象的创建和撤销          | 322 |
| 9.2.5  | 继承与特殊成员              | 326 |
| 9.3    | 派生类与基类的不同            | 327 |
| 9.3.1  | 覆盖与同名隐藏              | 327 |
| 9.3.2  | 扩充接口                 | 331 |
| 9.4    | 派生类向基类的类型转换          | 332 |
| 9.5    | 组合与继承的选择             | 335 |
| 9.5.1  | 组合的应用                | 336 |
| 9.5.2  | 继承的应用                | 337 |
| 9.5.3  | 组合的例子                | 339 |
| 9.6.4  | 继承的例子                | 342 |
| 9.6    | 多重继承                 | 347 |
| 9.6.1  | 多重继承引起的二义性           | 348 |
| 9.6.2  | 虚基类                  | 350 |
| 9.7    | 小结                   | 351 |
| 9.8    | 习题                   | 352 |
| 第10章   | 虚函数与多态性              | 358 |
| 10.1   | 派生类向基类的类型转换          | 358 |
| 10.2   | 虚函数                  | 361 |
| 10.2.1 | 声明虚函数                | 361 |
| 10.2.2 | 虚函数的覆盖规则             | 365 |
| 10.2.3 | 虚析构函数                | 368 |
| 10.2.4 | 实现多态性的步骤             | 369 |
| 10.3   | 动态绑定的实现              | 371 |
| 10.4   | 抽象类                  | 375 |
| 10.5   | RTTI                 | 378 |
| 10.5.1 | dynamic_cast 与向下类型转换 | 378 |
| 10.5.2 | typeid               | 381 |
| 10.6   | 类层次设计的例子             | 382 |
| 10.6.1 | 模仿钓鱼的例子              | 382 |
| 10.6.2 | 零件库存管理的例子            | 385 |
| 10.7   | 小结                   | 390 |
| 10.8   | 习题                   | 390 |

|        |               |     |
|--------|---------------|-----|
| 第 11 章 | 模板与泛型编程       | 394 |
| 11.1   | 函数模板          | 394 |
| 11.1.1 | 函数模板的定义       | 395 |
| 11.1.2 | 函数模板的实例化      | 397 |
| 11.1.3 | 函数模板的重载       | 398 |
| 11.2   | 类模板           | 399 |
| 11.2.1 | 类模板的定义        | 400 |
| 11.2.2 | 类模板的实例化       | 401 |
| 11.2.3 | 类模板的成员函数      | 401 |
| 11.2.4 | 模板的非类型参数      | 402 |
| 11.2.5 | 类模板的静态数据成员    | 403 |
| 11.2.6 | 类模板的友元        | 403 |
| 11.3   | 模板的编译         | 404 |
| 11.3.1 | 模板的代码组织       | 405 |
| 11.3.2 | 显式实例化         | 408 |
| 11.4   | 模板的更多特性       | 408 |
| 11.4.1 | 模板的默认实参       | 408 |
| 11.4.2 | 模板特化          | 409 |
| 11.4.3 | 可变参数模板        | 410 |
| 11.5   | 模板和代码复用       | 412 |
| 11.6   | 小结            | 413 |
| 11.7   | 习题            | 413 |
| 第 12 章 | 标准库容器和算法      | 414 |
| 12.1   | 容器和算法概览       | 414 |
| 12.1.1 | 容器概览          | 414 |
| 12.1.2 | 容器操作概览        | 415 |
| 12.1.3 | 算法概览          | 417 |
| 12.2   | 顺序容器          | 417 |
| 12.2.1 | 通用操作          | 418 |
| 12.2.2 | 特有操作          | 421 |
| 12.2.3 | 顺序容器适配器       | 424 |
| 12.2.4 | string 类的额外操作 | 425 |
| 12.3   | 迭代器           | 428 |
| 12.3.1 | 迭代器的运算        | 428 |
| 12.3.2 | 与迭代器有关的容器操作   | 430 |
| 12.3.3 | 反向迭代器         | 431 |
| 12.4   | 关联容器          | 432 |