

C++ 编程自学宝典

Beginning C++ Programming

[英] 理查德·格里姆斯 (Richard Grimes) 著 邓世超 译

开启一门超强编程语言的学习之旅

Packt



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

C++ 编程自学宝典

Beginning C++ Programming

[英] 理查德·格里姆斯 (Richard Grimes) 著 邓世超 译

人民邮电出版社
北京

图书在版编目 (C I P) 数据

C++编程自学宝典 / (英) 理查德·格里姆斯
(Richard Grimes) 著 ; 邓世超译. — 北京 : 人民邮电
出版社, 2018. 12

ISBN 978-7-115-49240-1

I. ①C… II. ①理… ②邓… III. ①C++语言—程序
设计 IV. ①TP312.8

中国版本图书馆CIP数据核字(2018)第200028号

版权声明

Copyright ©2017 Packt Publishing. First published in the English language under the title *Beginning C++ Programming*.

All rights reserved.

本书由英国 Packt Publishing 公司授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式或任何手段复制和传播。

版权所有, 侵权必究。

◆ 著 [英] 理查德·格里姆斯 (Richard Grimes)
译 邓世超
责任编辑 胡俊英
责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
固安县铭成印刷有限公司印刷

◆ 开本: 787×1092 1/16
印张: 22.25
字数: 572 千字 2018 年 12 月第 1 版
印数: 1-3 000 册 2018 年 12 月河北第 1 次印刷

著作权合同登记号 图字: 01-2017-0545 号

定价: 69.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316
反盗版热线: (010)81055315

内容提要

作为一门广为人知的编程语言，C++已经诞生 30 多年了，这期间也出现并流行过许多种编程语言，但是 C++绝对是经得起考验的。如此经典的编程语言，值得每一位编程领域的新人认真学习，也适合有经验的程序员细细品味。

本书旨在通过全面细致的内容和代码示例，带领读者更加全方位地认识 C++语言。全书分为 10 章，由浅入深地介绍了 C++的各项特性，包括 C++语法、数据类型、指针、函数、类、面向对象特性、标准库容器、字符串、诊断和调试等。本书涵盖了 C++11 规范及相关的 C++11 标准库，是全面学习 C++编程的合适之选。

本书适合 C++零基础读者，但是希望读者有一些编程经验。通过书中丰富、典型的代码示例，读者将快速把握 C++的特性和编程技巧。

作者简介

Richard Grimes 有 20 多年 C++ 编程经验，曾经致力于汽车制造业远程设备的金融分析和科学控制等多个项目。他在 70 多场微软技术（其中包含 C++ 和 C#）的国际会议上发表重要讲话，共编写了 8 本书，在编程期刊上发表了 150 多篇文章，主讲了 5 期微软培训课程。他连续 10 年（1998—2007）获得微软 MVP 认证，对微软 .net 框架和 C++ 的深刻理解以及对新技术的坦率评价使其在业内享有盛誉。

审稿人简介

Angel Hernandez 是一位备受瞩目的高级解决方案提供商、架构师，并且拥有超过 15 年的软件开发经验，擅长咨询领域。他曾经连续 11 年获得微软 Visual Studio 和开发技术（之前是 Visual C++）类别的 MVP 称号，他目前是微软 MVP 重新连接计划的成员，同时还是一名 TOGAF 从业者。他对微软和开源技术（*nix 系统）有深入的了解，也是托管和原生语言专家，最大的爱好是 C# 和 C++。

致我的妻子 Ellinor: 只有你的爱和支持能让我无往不胜。

前言

C++已经问世 30 多年了。在此期间，很多新的语言来了又走，但是 C++经得起考验。本书背后的一个大问题就是：为什么选择 C++？答案就分布于读者将要看到的本书的 10 章内容中。但作为一个“搅局者”，C++是一门灵活、强大的语言，并且拥有丰富、庞大的标准库提供支持。

C++一直是一门强大的语言，可以让用户直接访问内存，同时提供大量的高级特性，比如创建新类型和类的能力，以及重载运算符以满足用户需求。然而，更现代的 C++标准添加了不少特性：通过模板进行泛型编程，通过函数对象和 lambda 表达式进行函数式编程。用户可以根据需要充分地利用这些特性，也可以使用抽象接口指针或类 C 过程代码编写事件驱动代码。

在本书中，我们将介绍 C++11 规范以及通过该语言提供的标准库。本书使用简短的代码片段解释了如何使用这些特性，每一章包含一个实用示例来解释这些概念。在本书的最后，读者将了解该语言的所有功能以及 C++标准库可以实现的功能。假定读者是初学者，本书将引导和提示读者从零开始使用 C++。

内容概要

第 1 章“初识 C++”介绍了用于编写 C++应用程序的文件、文件引用依赖以及基本的 C++ 项目管理知识。

第 2 章“语言特性简介”涵盖了 C++语句、表达式、常量、变量和运算符，以及如何在应用程序中控制执行流程。

第 3 章“C++类型探秘”描述了 C++内置类型、聚合类型、类型别名、初始化器列表以及类型之间的转换。

第 4 章“内存、数组和指针”介绍了在 C++应用程序中如何分配和使用内存、如何使用内置类型、C++引用的角色以及如何使用 C++指针访问内存。

第 5 章“函数”解释了如何定义函数、如何使用可变数目的参数通过值和引用传递参数、创建和使用函数指针以及定义模板函数和重载运算符。

第 6 章“类”介绍了如何通过类定义新类型以及在类中使用多种专一化函数，如何将类实例化为对象以及如何将其销毁，如何通过指针访问对象以及如何编写模板类。

第 7 章“面向对象编程简介”介绍了继承和组合技术，以及它们如何影响指针、引用对象和类成员访问层级的使用，它们如何继承成员。本章还介绍了如何通过虚方法实现多态、通过抽象类实现继承编程。

第 8 章“标准库容器”介绍了 C++标准库容器类，以及如何将它们和迭代器、标准库算法搭配使用，以便用户可以访问容器中的数据。

第 9 章“字符串”介绍了标准 C++ 字符串类的特性、数字和字符串之间的转换、国际化字符串，以及如何使用正则表达式搜索和操作字符串。

第 10 章“诊断和调试”介绍了如何准备代码以便诊断和调试、如何优雅地终止应用程序以及如何使用 C++ 异常机制。

读者须知

本书涵盖了 C++11 规范以及相关的 C++ 标准库。对于本书的绝大多数内容，任何符合 C++11 规范的编译器都是适合的，这些编译器的厂家包括 Intel、IBM、Sun、Apple 和 Microsoft，以及开源的 GCC 编译器。

本书采用的开发环境是 Visual C++ 2017 社区版，因为它是一个功能齐全的编译器和开发环境，是可以免费下载的。这是作者的个人选择，不过也不会限制读者选择其他编译器。第 10 章的某些部分介绍了专属于 Microsoft 的一些特性，但是这些部分都已清楚地标记出来。

目标读者

本书适用于有一定编程经验，但还是 C++ 新手的程序员。希望读者在阅读本书之前已经知道什么是高级语言以及相关的基本概念，比如模块化代码和程序控制执行流程。

排版约定

在本书中，读者将发现一些用于区分不同信息的文本样式。以下是这些样式的一些示例及其含义的解释。

文本形式的代码、数据库表名、文件夹名、文件名、文件扩展名、路径名、简单的 URL 地址、用户输入、引用段落如下所示：“We can include other contexts through the use of the include directive”。

代码块设置如下所示：

```
class point
{
public:
    int x, y;
};
```

当我们希望某些代码片段引起读者注意时，相关的行或元素将以粗体表示：

```
class point
{
public:
    int x, y;
    point(int _x, int _y) : x(_x), y(_y) {}
};
```

任何命令行输入或输出如下所示：

```
C:\> cl /EHsc test.cpp
```

新术语或关键字都以粗体显示。读者在屏幕上看到的单词，比如在菜单或者对话框中，会以如下文本显示：“Clicking the **Next button** moves you to the next screen”。



警告

警告或需要特别注意的内容。



提示

提示或者技巧。



资源与支持

本书由异步社区出品，社区 (<https://www.epubit.com/>) 为您提供相关资源和后续服务。

配套资源

本书提供如下资源：

- 本书源代码；
- 书中彩图文件。

要获得以上配套资源，请在异步社区本书页面中点击 **配套资源**，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

如果您是教师，希望获得教学配套资源，请在社区本书页面中直接联系本书的责任编辑。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，点击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。

The screenshot shows a web form for submitting勘误 (勘误). At the top, there are three tabs: '详细信息' (Detailed Information), '写书评' (Write a Review), and '提交勘误' (Submit勘误), with the latter being the active tab. Below the tabs, there are three input fields: '页码:' (Page Number), '页内位置 (行数):' (Page Position (Line Number)), and '勘误次数:' (勘误次数). Below these fields is a rich text editor with a toolbar containing icons for bold (B), italic (I), underline (U), bulleted list, numbered list, link, and unlink. The editor area is currently empty. In the bottom right corner of the form, there is a '字数统计' (Character Count) label and a '提交' (Submit) button.

扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术 etc。



异步社区



微信服务号

目录

第 1 章 初识 C++1	
1.1 本章的主要内容.....1	
1.2 C++是什么.....1	
1.3 安装 Visual C++.....2	
1.3.1 安装配置.....2	
1.3.2 下载安装文件.....2	
1.3.3 安装 Visual Studio.....3	
1.3.4 注册 Microsoft 账号.....5	
1.4 C++项目结构简介.....6	
1.4.1 编译器.....6	
1.4.2 链接代码.....7	
1.4.3 源文件.....8	
1.4.4 预处理器和标识符.....14	
1.4.5 依赖项.....18	
1.5 编写一个简单的项目程序.....22	
1.5.1 项目结构.....22	
1.5.2 创建预编译头文件.....22	
1.5.3 创建主文件.....23	
1.5.4 输入和输出流.....24	
1.5.5 time 函数.....25	
1.5.6 构建可执行文件.....26	
1.5.7 测试代码.....27	
1.5.8 修改项目.....27	
1.6 小结.....27	
第 2 章 语言特性简介28	
2.1 编写 C++代码.....28	
2.1.1 空格.....28	
2.1.2 格式化代码.....29	
2.1.3 编写语句.....30	
2.1.4 类型和变量.....32	
2.1.5 常量和文字.....33	
2.1.6 命名空间.....37	
2.1.7 C++的变量作用域.....40	
2.2 运算符.....42	
2.3 执行流控制.....49	
2.3.1 条件语句.....49	
2.3.2 选择.....51	
2.3.3 循环.....53	
2.3.4 跳转.....57	
2.4 C++语言特性应用.....58	
2.5 小结.....62	
第 3 章 C++类型探秘63	
3.1 内置类型.....63	
3.1.1 整数类型.....64	
3.1.2 浮点类型.....68	
3.1.3 字符和字符串.....69	
3.1.4 布尔类型.....75	
3.1.5 void.....75	
3.1.6 初始化器.....76	
3.1.7 默认值.....77	
3.1.8 无类型的变量声明.....78	
3.1.9 存储类.....80	
3.1.10 类型别名.....81	
3.1.11 记录类型中的聚合 数据.....82	
3.1.12 访问运行时类型信息.....88	
3.1.13 类型取值范围.....89	
3.2 类型之间的转换.....89	
3.2.1 类型转换.....89	

3.2.2 转型	91	4.7.5 插入元素	133
3.3 C++类型应用	94	4.8 小结	135
3.4 小结	101	第5章 函数	136
第4章 内存、数组和指针	102	5.1 定义C++函数	136
4.1 C++中的使用内存	102	5.1.1 声明和定义函数	136
4.1.1 C++指针语法	102	5.1.2 声明链接	137
4.1.2 空指针	104	5.1.3 内联化	137
4.1.3 内存类型	104	5.1.4 确定返回类型	138
4.1.4 指针算术	106	5.1.5 函数命名	138
4.2 数组	107	5.1.6 函数参数	138
4.2.1 函数参数	108	5.1.7 声明异常	139
4.2.2 多维数组	109	5.1.8 函数体	139
4.2.3 将多维数组传递给 函数	110	5.2 函数参数	139
4.2.4 字符数组	111	5.2.1 传递初始化器列表	140
4.3 在C++中使用指针	113	5.2.2 默认参数	140
4.3.1 访问越界	113	5.2.3 可变参数	141
4.3.2 指针内存释放	113	5.3 函数特性	144
4.3.3 指针转型	114	5.3.1 堆栈调用	144
4.4 在代码中分配内存	115	5.3.2 声明调用规范	144
4.4.1 分配单个对象	116	5.3.3 递归	145
4.4.2 分配对象数组	116	5.3.4 函数重载	146
4.4.3 处理内存分配异常	117	5.3.5 函数和作用域	146
4.4.4 new 运算符的其他 版本	117	5.3.6 删除函数	146
4.4.5 内存的生命周期	118	5.3.7 值传递和引用传递	147
4.4.6 Windows SDK 和指针	118	5.4 函数设计	148
4.5 内存与C++标准库	120	5.4.1 前置条件和后置条件	148
4.6 引用	122	5.4.2 不变量	149
4.6.1 常量引用	123	5.5 函数指针	150
4.6.2 返回引用	123	5.5.1 留意括号	150
4.6.3 临时引用	124	5.5.2 声明函数指针	150
4.6.4 右值引用	125	5.5.3 函数指针用法	151
4.6.5 for 循环区间和引用	127	5.6 模板函数	152
4.7 指针实战	128	5.6.1 定义模板	153
4.7.1 创建项目	129	5.6.2 模板参数值	155
4.7.2 将task对象添加到 列表	130	5.6.3 专一化模板	157
4.7.3 删除任务列表	131	5.6.4 可变参数模板	157
4.7.4 遍历任务列表	132	5.7 运算符重载	159
		5.7.1 函数对象	161
		5.7.2 lambda 表达式简介	163
		5.8 函数在C++中的应用	164

5.9 小结	169	7.2.9 抽象类	222
第 6 章 类	170	7.2.10 获取类型信息	224
6.1 编写类	170	7.2.11 智能指针和虚方法	226
6.1.1 重新审视结构体	170	7.2.12 接口	228
6.1.2 定义类	171	7.2.13 类之间的关系	230
6.1.3 定义类的行为	171	7.2.14 mixin 类	231
6.1.4 创建对象	174	7.3 多态应用	233
6.1.5 声明静态成员	181	7.4 小结	244
6.1.6 嵌套类	184	第 8 章 标准库容器	245
6.1.7 访问常量对象	184	8.1 pair 和 tuple	245
6.2 对象和指针	185	8.2 容器	247
6.2.1 指向对象成员的指针	185	8.2.1 顺序容器	248
6.2.2 运算符重载	186	8.2.2 关联容器	252
6.2.3 定义函数类	188	8.2.3 专用容器	255
6.2.4 定义转换运算符	191	8.2.4 迭代器	256
6.2.5 资源管理	192	8.3 算法	261
6.2.6 编写包装器类	192	8.3.1 元素迭代	261
6.2.7 智能指针	193	8.3.2 获取信息	263
6.3 模板	198	8.3.3 容器比较	263
6.4 类的应用	200	8.3.4 修改元素	264
6.5 小结	203	8.3.5 查找元素	266
第 7 章 面向对象编程简介	204	8.3.6 元素排序	267
7.1 继承和组合	204	8.4 数值库	268
7.1.1 从一个类继承	204	8.4.1 编译期运算	268
7.1.2 方法重载和名称隐藏	206	8.4.2 复数	270
7.1.3 指针和引用	208	8.5 标准库应用	271
7.1.4 访问层级	209	8.6 小结	276
7.1.5 继承访问层级	211	第 9 章 字符串	277
7.1.6 多继承	211	9.1 把字符串类当作容器	277
7.2 多态简介	214	9.1.1 获取字符串信息	278
7.2.1 虚方法	215	9.1.2 修改字符串	278
7.2.2 虚方法表	217	9.1.3 搜索字符串	279
7.2.3 多继承和虚方法表	218	9.2 国际化	281
7.2.4 虚方法、构造函数和析构函数	219	9.3 字符串和数字	284
7.2.5 容器和虚方法	219	9.3.1 将字符串转换成数字	284
7.2.6 友元和继承	220	9.3.2 将数字转换成字符串	284
7.2.7 重载和关键字 final	221	9.4 stream 类	285
7.2.8 虚拟继承	222	9.4.1 输出浮点数	285
		9.4.2 输出整数	287

9.4.3	输出时间和货币	288	10.1.6	预处理器标识符	312
9.4.4	使用 stream 将数字转换为字符串	289	10.1.7	生成诊断信息	314
9.4.5	使用 stream 从字符串读取数字	290	10.1.8	断言	318
9.5	正则表达式	292	10.2	应用程序终止	319
9.5.1	正则表达式定义	292	10.3	异常值	320
9.5.2	标准库类	294	10.3.1	获取描述信息	321
9.6	字符串应用	298	10.3.2	消息编译器	322
9.6.1	创建项目	299	10.4	C++的异常	323
9.6.2	处理标题子元素	303	10.4.1	异常规范	323
9.7	小结	305	10.4.2	C++异常语法	323
第 10 章	诊断和调试	306	10.4.3	标准 exception 类	325
10.1	准备工作	306	10.4.4	根据类型捕获异常	326
10.1.1	不变性和条件编译	306	10.4.5	函数中的 try 语句块	328
10.1.2	条件编译	307	10.4.6	系统异常	329
10.1.3	pragma 指令	308	10.4.7	异常嵌套	330
10.1.4	添加通知消息	309	10.4.8	结构化异常处理	331
10.1.5	调试程序的编译器开关	311	10.4.9	编译器异常开关	334
			10.4.10	C++和 SEH 混合异常处理	334
			10.4.11	编写异常安全的类	336
			10.5	小结	337

第 1 章

初识 C++

为什么选择 C++? 从读者自身的实际情况来看, 原因有很多。

读者选择 C++ 可能是因为必须为一个 C++ 项目提供技术支持。在超过 30 年的生命周期中, 该项目中已经包含了数百万行 C++ 代码, 并且大部分流行的应用程序和操作系统是使用 C++ 编写的, 或者是使用了与之有关的组件和库。几乎不可能找到一台不包含 C++ 代码的电脑。

或者读者打算使用 C++ 编写新的代码。这可能是因为项目代码中将会用到一个使用 C++ 编写的程序库, 而且有成千上万的程序库可供选择: 开源的、共享的和商业软件。

或者读者可能是被 C++ 强大的功能和灵活性所吸引。现代高级程序语言的目标是将程序员从繁复的编程工作中解放出来。同时, C++ 还允许用户和机器保持尽可能紧密的联系, 使得用户可以直接访问计算机内存 (有时是比较危险的)。通过类和重载这些语言特性, C++ 是一门灵活的语言, 我们可以对它进行功能扩展, 并编写可复用的代码。

不论读者选择 C++ 的理由是什么, 这个决定都是非常明智的, 本书可以作为读者入门的起点。

1.1 本章的主要内容

本书是一本实用性的书, 读者可以对其中的代码输入、编译和运行。为了编译代码, 你将需要一个 C++ 编译器和链接器, 在本书中它们是指提供 Visual C++ 的 Visual Studio 2017 社区版程序。选择该编译器是因为我们可以免费下载它, 它符合 C++ 标准规范, 并且包含大量能够提高编程效率的工具。Visual C++ 提供了对 C++11 语言特性的支持, 并几乎兼容 C++14 和 C++17 的所有语言特性。Visual C++ 还包含了 C99 运行时库、C++11 标准库和 C++14 标准库。上述所有规范意味着读者在本书中将要学习的代码, 将能够被其他所有标准的 C++ 编译器编译。

本章将从如何获取和安装 Visual Studio 2017 社区版程序的细节开始。如果你已经拥有了一个 C++ 编译器, 那么可以跳过本小节。本书大部分内容是与编译器和链接器无关的。但是第 10 章介绍调试和诊断技术时会涉及一些专属于 Microsoft 的功能特性。Visual Studio 是一款功能齐全的代码编辑器, 所以即使你不使用它来管理项目文件, 也仍然会发现它对于编辑代码来说是非常有用的。

在介绍完程序安装之后, 读者将学习一些 C++ 的基础知识: 如何组织源码文件和项目, 以及如何管理可能存在几千个文件的项目。

最后, 本章将以一个循序渐进的结构化示例作为结尾。这里读者将学习如何使用 C++ 标准库编写简单的函数以及一种管理项目文件的机制。

1.2 C++ 是什么

C++ 的前身是 C 语言, C 语言是由 Dennis Richie 供职于贝尔实验室时设计的, 于 1973 年