

计算机技术

译林

精选系列

Addison
Wesley

标准 C++

输入输出流 与本地化

Angelika Langer

Klaus Kreft

何渝 孙悦红

刘宏志 武剑

著

译

人民邮电出版社

www.pptph.com.cn

00115254



TP312 C
206

计算机技术译林精选系列

标准 C++ 输入输出流与本地化

Angelika Langer

Klaus Kreft 著

何渝 孙悦红 刘宏志 武剑 译

人民邮电出版社

图书在版编目 (CIP) 数据

标准 C++ 输入输出流与本地化 / () 兰格 (Langer, A.) () 克雷菲特 (Kreft, K.)
编著; 何渝等译. —北京: 人民邮电出版社, 2001.4

(计算机技术译林精选系列)

ISBN 7-115-09052-1

I. 标… II. ①… ②克… ③何… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2000) 第 73224 号

JS443/28

计算机技术译林精选系列

标准 C++ 输入输出流与本地化

- ◆ 著 Angelika Langer Klaus Kreft
- 译 何渝 孙悦红 刘宏志 武剑
- 责任编辑 万东旭 陈冀康
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
- 邮编 100061 电子函件 315@pptph.com.cn
- 网址 <http://www.pptph.com.cn>
- 读者热线: 010-67129212 010-67129211(传真)
- 北京汉魂图文设计有限公司制作
- 北京鸿佳印刷厂印刷
- 新华书店总店北京发行所经销
- ◆ 开本: 787 × 1092 1/16
- 印张: 34.5
- 字数: 859 千字 2001 年 4 月第 1 版
- 印数: 1 - 5 000 册 2001 年 4 月北京第 1 次印刷

著作权合同登记 图字: 01 - 2000 - 0659 号

ISBN 7-115-09052-1/TP·2021

定价: 57.00 元

内容提要

本书是一本介绍 C++ 标准 IO 流 (Stream) 和本地化的程序员手册, 同时书后还附有完整的相关类、函数、模板和头文件等说明。

本书主要由三部分组成: 第一部分为“输入与输出流”, 包括第一章 IO 流基础, 第二章 IO 流的结构, 第三章输入/出流的高级用法。分别从最基本的什么是输入输出流, 到流的格式、流的本地化等, 一直介绍到流的各种高级用法。第二部分称为国际化, 但主要介绍的是本地化, 包括国际化与本地化的关系和重要性, 为本地化而建立的 Local 和相关的类, 以及如何正确合理地使用这些类。第三部分为参考指南, 包括各个相关类、函数、模板和头文件的详细说明及用法。

本书特别适用于已经学过 C++ 教程或有一定实际编程经验、并打算经常使用 IO 流和本地化的 C++ 编程人员。对于那些经常使用 C++ 作为编程工具的编程人员来说, 本书是一本很有价值的常用参考书。

版权声明

Angelika Langer, Klaus Kreft: Standard C++ IOStreams
and Locales

Copyright © 2000 by Addison Wesley Longman, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise without the prior consent of Addison-Wesley.

Published by arrangement with Addison Wesley Longman, Inc. All Rights Reserved.

版权所有。未经出版者书面许可，对本书任何部分不得以任何方式或任何手段复制和传播。

人民邮电出版社经 Addison Wesley Longman 公司授权出版。版权所有，侵权必究。

由于 C++ 编程语言自身的特点，加上企业界的大量采用，近年来，学习、使用 C++ 编程语言日益流行。当使用 C++ 编程语言来编写程序时，很大的一部分工作是在与它的各种库，或者各种类打交道，在这当中又常常涉及到其中的两个重要领域——IOStream（IO 流，输入输出流）和本地化。

随着计算机在世界范围的日益流行，计算机软件需要解决的问题是对其国际化和本地化的处理。其实，当人们在使用或开发一个计算机软件产品时，一定会自觉或不自觉地向有这样的感受：一个计算机软件产品不仅仅是一个单纯的技术产品，它本身常常带有非常浓厚的地理区域和文化习俗色彩。一般说来，软件都离不开输入输出，而这些输入输出避不开各种地理区域和文化色彩的影响。一个很简单的例子就是，全世界有成千上万种语言文字，最常用的就有几十种，计算机软件的使用者当然最喜欢使用自己的母语。很容易理解，使用中文的用户，一般都希望软件也是中文的。因此，要开发满足这种要求的软件就一定会融入中国的文化习俗。比如：软件中要使用中国语言文字字符，要使用到相应的语言文字字符的标准编码，软件中要用中国表示日期和时间的方法，要用表示中国货币的表示方式……，等等。

不难理解，当软件开发人员要开发一个软件产品时，该软件就可能要涉及到需要处理多种特定的语言文字、特定的文化习俗信息。如何在技术上实现？如何在满足其通用性的同时又满足其本地性的要求？这些问题都是软件开发人员需要仔细考虑周到，并且给予适当解决。

当我们使用 C++ 编程语言作为开发软件的工具时，同样会碰到上面提到的类似问题。为了方便用户了解它、使用它，人民邮电出版社组织引进翻译了《Standard C++ IOStreams and Locales》一书，希望对有关人员有所帮助。

需要说明一点的是，此书是一本介绍标准 IO 流和本地化的程序员手册，附有完整的相关类、函数、模板和头文件等的说明，它的目的并不是教给读者有关 C++ 或 IO 流的基本知识。因此，这本书并不适合初学者，而适用于那些已经学过 C++ 教程，或者

有一定实际编程经验、并打算经常使用 IO 流和本地化的 C++ 编程人员。

参加本书翻译工作的还有许永凯、宋兆松、朱小艳等，杨放鸣参加了部分文字录入和校阅的工作。

由于时间仓促，加之译者水平有限，本书难免有不妥之处，敬请读者批评指正。

译者

2000 年 7 月

我在 C++ 的输入/输出库上的工作开始于 1986 年左右，那时我开始修改贝尔实验室 C++ 内部发行版中的 `stream` 库部分程序上的一个漏洞。同时，我还决定将 C 的 `stdio` 库所具有、而这个 `stream` 库所缺乏的一些功能添加进去。当我进一步审视这个 `stream` 库时，我突然意识到 C++ 中输入输出库的体系结构引发了有关 C++ 设计的一些有趣问题，而且明白了是什么构成了现在的这个 `IOStream` 库。起初，我认为这只是一个内部用于试验而使用的库，没打算用它来替换 `stream` 库。但后来，Bjarne Stroustrup 鼓励 C++ 开发产品部门将 `stream` 库换成 `IOStream`（IO 流）库，而他们确实也这么做了。这使得我起初以为只需花费几个月的一件小事变成了后来持续了 10 多年的一项艰苦工作，其中包括 ANSI/ISO 标准化的工作。

我最初设计的一个主要目标是使 IO 流库可以以多种有趣的方式扩展。具体来说，在 `stream` 库中，`streambuf` 类只是一个实现上的细节，但是在 IO 流库中，我打算让它本身成为一个有用的类。我希望编写出多个功能不同的 `streambuf` 类，但除了我自己写了几个之外，几乎没有人这么做。我回答的问题多半是诸如“我怎样使我的数字看起来像这样”，而不是“我怎样写一个 `streambuf` 类”。就连教科书的作者也将 `streambuf` 忽略不谈。很显然，没有人同意我的观点，即输入/输出库的体系结构是一个值得研究的有趣问题。

我常常被问及的另一个问题是“为什么这些 `stream` 类的成员不是虚函数”，进一步讨论的结果往往是，提问者需要编写一个具有某种独特功能的 `streambuf` 类，或者有时候他要做的是另写一个用 `streambuf` 类进行传送的顶层类。而当我愿意就此进行详细解释时，他们又往往失去了兴趣。

所有这些都让我产生了一种对这个库没有物尽其用的沮丧和失望。好几个朋友都劝我通过写书来消除这种沮丧情绪。他们的理由是，可扩展性的特点很难理解，但是一本讲解清晰的书可以鼓励人们去使用它。随着 C++ 标准化工作的深入和这个库复杂程度的增加，写这样一本书的必要性与日俱增。我好几次都打算开

始这项写作工作，但由于时间和精力关系，始终未能如愿。

在此期间，每当我听说有关于 IO 流库的书时，我的心情都十分复杂。一方面，我非常高兴看到 IO 流库得到越来越多的关注，但另一方面，我又担心这些书会比我捷足先登。但我的担心完全是多余的，这些书都没有涉及到我所关心的主要问题，即库体系结构和 C++ 设计。

也就是说，本书是第一本涉及到上面两个问题的书。它不仅回答了我所关心的问题，而且还提供了具体的例子，以使读者能快速地将这些思想应用到自己的需求中去。由于我现在已经不用再考虑写书的问题，所以我对这本书的感觉就都是正面的了。

Jerry Schwarz

1999 年 11 月

自 1998 年起, C++ 编程语言就已经以 ISO/IEC 国际标准 14882 的形式得到了规范, 但由于历史原因, 这个标准常常被称为 ANSI C++ 标准。在这个标准中, 有大量的被称为“C++ 标准库”的摘要集。实际上, 有半数的标准是有关库的。这本书讨论了标准库中的两个主要领域: `IOStream` (IO 流, 输入输出流) 和本地化。

在从 1989 到 1998 年的标准化过程中, 这门新语言的特点及其标准库引起了 C++ 语言界的极大兴趣。为了更好地论述对这方面信息的需要, 在标准化过程中及之后出版了好几本书, 有些是对标准 C++ 的概述, 一般都包括对一些库摘要的简介; 有一本书是专门讨论标准库。然而到目前为止, 该库中只有一个部分的讨论较为深入, 那就是 STL, 它是由 Hewlett-Packard 独立于标准化工作而开发的一个算法集, 后来也被集成到了 C++ 标准库中。虽然 STL 毫无疑问是标准库中用得最多的部分, 但它在整个标准库中所占的比例不足三分之一 (以占标准化文件中的页数和委员会讨论的时间为依据), 而 IO 流和本地化就占了标准库的另外三分之一。

在 1993 年, 当我们以专业人员的身份接触到 C++ 标准化的时候, 几乎还没有 IO 流方面的出版物, 而 C++ 的本地化也还没有被提出。仅有的几本关于 IO 流的书都是过时的, 其内容都是介绍经典的、标准化前的 IO 流。C++ 方面的教程提供的信息仅限于 IO 流, 而很少提及本地化。正是因为这个原因, 我们觉得非常有必要写一本书来专门讨论那些其他指南所忽略的内容。

适合此书的读者

此书是一本介绍标准 IO 流和本地化的程序员手册, 同时还附有完整的相关类、函数、模板和头文件等的说明。它既非指南, 也非教程, 它的目的并不是教给读者有关 C++ 或基本 IO 流的知识。我们假设读者知道, 至少大概知道, 当他们敲入诸如下面这行语句时会发生什么。

```
cout << "Hello world" << endl;
```

因此，这本书并不适合初学者，而适用于已经学过 C++ 教程或有一定实际编程经验、并打算经常使用 IO 流和本地化的 C++ 编程人员。

与已有 10 多年历史的 IO 流不同的是，本地化是 C++ 中的一个新概念，因此我们对它的介绍将从最基本的部分开始。对 C 语言中本地化的一些知识将有助于对新内容的理解，但这并非是必须的。我们不打算对国际化做太详细的介绍，因为国际化是一个太广泛的题目，对它的详尽讨论可以另成一本书。然而，由于 IO 流和本地化这两方面结合得非常紧密，所以本书将解释 C++ 中本地化的概念，而侧重点在于介绍本地化结合 IO 流的使用。

至于 IO 流，我们都知道经典的 IO 流库自从 C++ 诞生的早期就已经存在了。由于在每本 C++ 教程中都有关于它的介绍，所以我们假定读者对它的基本特点都已经非常熟悉，对此不再赘述，而将做其他更深层次的讨论。例如，我们将介绍它的一些高级特性——如用户自定义的移位运算符和操作符，通过使用 `word / pword` 来扩展流，以及新的流和流缓冲类的派生——同时还将介绍一些较为浅显的内容，如格式控制和错误处理。

总之，本书的目的是尽可能全面地介绍各种原理，以使读者学会利用 IO 流和本地化来完成他们具体的编程任务；重点是介绍基本概念以及 IO 流和本地化所支持的较为高级的编程技巧，而不是停留于各种接口的细枝末节。基于这个原因，我们没有列出详尽而冗长的具体实例及其代码。由于 IO 流和本地化是一种应用非常广泛的工具，可以用来解决各式各样的问题，所以不可能找到一套所谓的典型实例。相反，我们将重点介绍几个经过精练的例子，来解释编程技术和概念；并且为了避免过多地分散读者的注意，我们有意地没有把它们扩充成完整的应用程序。我们相信，凭借读者的聪明智慧，一旦他们明白了基本原理之后，是完全可以自己写出具体的应用程序的。

关于标准

我们收到过大量的询问，如“这个和那个为什么是如此和这般？”等等，以期得到 IO 流和本地化为什么设计成现在这个样子的解释。对于一些有基本原理的问题，我们就做出相应解释；而对于那些存在不一致之处及“有趣的”设计决定，只能用“历史原因”或“委员会的设计”来回答；对于某些可能导致误区的地方，我们都加以指出，以使读者可以避免它们。除此之外，我们既不打算替这个标准做任何辩护，也不准备讨论其他的设计可能，我们只是进行如实描述而已。

致谢

这本书的写作历时 3 年，在这个不短的时期内，我们得到了许多人的大力支持和帮助。和任何书一样，作者只是故事的一部分，我们要感谢所有那些给予过我们这样或那样帮助的人们。

在 Addison-Wesley 出版公司方面，要感谢 Mike Hendrickson, Deborah Lafferty, 尤其是 Marina Lang。他们对本书的价值深信不疑，从提议到印刷的整个过程中，自始至终地提供了大量的帮助。还要感谢 Beth Burleigh Fuller 和 John Fuller，谢谢他们在出书的过程中所给予的支持。

我们还要感谢所有那些知识渊博而且富有耐心的标准委员会的人们，以及那些回答了我们的无数的、有时甚至是愚蠢的问题的人们，他们是：Nathan Myers，是他设计了 C++ 本地化并且推荐给标准委员会，他告诉了我们本地化的所有细节，并帮助我们理解他的设计思想及其后收益匪浅的讨论；Jerry Schwarz，“IO 流之父”，IO 流（或 C++1.0 版中所称的“streams”）第一版的作者，他在我们理解 IO 流许多特点的设计意图上给予了宝贵的指点，同时也感谢他的耐心和支持；Bill Plauger，C++ 标准库之微软版的作者，他帮助我们理解程序实现上的错误同我们自己的误解区分开来，同时，他还在我们如何正确地理解和解释标准的问题上提供了宝贵的帮助；Philippe LeMouel，我以前在 RogueWaveSoftware 公司的同事，他实现了 IO 流并向我们做出了实现的解释。还有 Joe Delaney，他在 RogueWave 的本地化实现上做了大量工作。Dietmar Kühl 替 gnu 编译器完成了 IO 流和本地化的实现工作，并且回答了大量问题。Edison 设计集团的 John Spicer 和西门子的 Erwin Unruh 回答了有关模板和其他语言特点的问题。Beman Dawes 替标准委员会维护了有关库的问题清单，并协助澄清了标准中无数的公开问题。

同时，还要感谢我们的审阅人，他们当中的一些人花了相当多的时间和精力编写了全面而有益的评语，他们是：（按姓氏字母顺序）Chuck Allison，Stephen Clamage，Mary Dageforde，Amelia Lewis，Stan Lippman，Dietmar Kühl，Werner Mossner 和 Patrick Thompson，以及那些不愿留名的人们。

——Angelika Langer
Klaus Kreft

我要感谢 Bernd Eggink，他用德语写了一本关于经典 IO 流的书，我们之间关于 IO 流的 E-mail 往来激发了共同写一本有关标准 IO 流的书的念头。最初的计划是将他的书译成英文，并将标准的内容更新。但他的突然重病使得这个计划流产了。

我还要感谢 RogueWaveSoftware 公司的创始人 Thomas Keffer，是他建议我自己写一本书，并自此不断地给我鼓励和支持。当他提出这个建议时，我正在他的公司中谋事。当他提议我写一本关于 C++ 国际化的书时，我只是一个在美国公司里工作的德国人。我还要感谢 Roland Hartinger，我的前任上司、西门子 Nixdorf 的 C++ 编译器建设集团的主席，是他令我开始了编写库的工作，并鼓励我加入了标准委员会。

最后，也是最重要的，我要感谢 Klaus Kreft 加入了我的写书工作，没有他的参与，这本书就不可能完成，是他做了大量的有价值的工作。

——Angelika Langer

首先，我要感谢我的父母，是他们发掘并培养了我对数学和自然科学的兴趣和天赋。没有他们的支持，就不会有我的今天。

其次，我要感谢两位我通过职业关系结识并成为了我灵感和思想源泉的人：Gerhard Draxler，自他退休后我就对他万分思念；以及 Werner Mossner，正是和他一起，我第一次

接触了 IO 流。我们俩一起完成了一个从 IO 流类中派生出来的登录系统。

最后，我要感谢 **Angelika Langer**，她始终都是我思想的源泉和十分牢固的合作者。没有她的付出，这本书就不会面世；但她所做贡献的重要性远远比不上她在我私人生活中所占的地位。

——**Klaus Kref**

本书的结构

本书对 C++ 标准库中国际化和流输入/输出内容的介绍分为以下几个部分：

用户指南包括第一部分：输入与输出流；第二部分：国际化。

第三部分为参考指南。

最后是附录。

用户指南和参考指南都以一种方便检索的方式加以组织。用户指南的组织以论题、体系结构概念和特定用法为线索，而不是依次讨论各个类和接口。例如，在用户指南中你会看到诸如“IO 流中的错误提示”和“本地化对象的创建”。参考指南则正好相反，它以类和接口为主线，对每个头文件、类和函数等都设立相应的条目。

用户指南的组织结构

第一部分和第二部分分别包括下面几部分：

- 问题域的总体介绍。
- API 的用法和描述。
- 体系结构：构体、设计概念和组件的习惯用法。
- 框架的扩展。

这只是一些“虚拟”的节，每节都又可能包含几个小节。另外，这些节的标题也不同于上面所列的这些摘要的标题。例如，在第二部分关于国际化的部分中，API 的用法和描述一节就包括了“本地化”和“标准化的方面”两个小节，而高级用法一节叫作“用户定义的方面”。在第一部分关于流输入/输出的部分中，API 的用法和描述又包括好几个小节，如格式控制、错误提示、操作符等等。然而，总体结构还是遵从上面的模式。

总体介绍部分是针对那些对问题域还不太了解的读者。有关流输入/输出的介绍比对国际化的介绍要简洁得多，因为我们假定许多读者已经非常熟悉流输入/输出，但在国际化问题上还没有太多经验，因此需要更多这方面的信息。

IO 流和本地化可以从以下两个角度来看：

- ① 它们可以作为一个基础库来提供大量现成可用的接口。
- ② 它们也可作为框架而加以定制和扩展。

关于用法一节的重点是介绍基础库，它描述了各个组件 API 的最普通最直接用法。每节从原则上讲都可以分开阅读。

关于体系结构一节介绍了每个组件的内核，即其内部构造的各个方面。在这一节中，读者会看到类的示意图以及对一些有趣的细节和某些设计问题的讨论。读者可以在阅读后面有关高级用法和组件扩展的内容之前温习一下这部分，也可以先跳过这一节，在需要更好地理解高级用法部分时回到相应小节来查阅。

高级用法一节探讨了扩展组件的方法，重点在于介绍每个组件框架方面的内容。同样，这些小节都可以单独阅读。为了说明扩展 IO 流的技术，我们在这几个小节中使用了同一个例子——数据对象的插入和抽取。在最低程度上，后继章节的源代码都是在前面章节的基础上扩充而成，但读者并不必为了理解后继章节的程序而通读前面所有的内容。程序中有许多标注分别指向前面章节的相应部分，因此读者可以根据需要在前后章节中来回查阅。

参考指南的组织结构

参考指南用来查找函数定义和类接口，主要分为以下五个部分：

- ① 本地化。
- ② 字符特征。
- ③ IOStreams (IO 流)。
- ④ 流迭代程序。
- ⑤ 其他 I/O 操作。

每个部分都包含以下条目：

- 头文件。
- 全局类型定义。
- 全局对象。
- 全局函数。
- 类的条目（以字母顺序排列）。

类示意图

我们使用标准的对象建模语言 UML 来作为类示意图的符号系统。对于那些还不太熟悉 UML 符号系统的读者，下面是我们在本书中使用的符号总览。

类	继承
参数化类	聚合
关联要素	合成

翻译成 C++ 的术语则是：

类 = 类

参数化类= 类模板

关联要素= 模板类，即类模板的实例化

继承 = 继承，即“是”一种关系（虚拟继承和非虚拟继承无区别）

聚合 = 对象的引用或指针

合成 = 包含一个对象，即“有”一种关系

代码示例

需要注意的是，本书中示例的源代码可能在所有的开发环境中都不能编译。实际上，在写这本书的时候，几乎还没有任何 C++ 编译器能理解 C++ 标准规定的所有语言特点，也没有任何编译器带有完全符合标准的 C++ 库。

我们曾试着用微软开发的 MVC 6.0 编译器来编译本书中示例的代码，但在意料之中的是，我们的一些例程在这个开发环境中不能被编译和成功测试。尽管如此，我们还是尽了最大努力来向读者展示标准 C++ 程序在一个标准 C++ 编译器上运行的情况。

虽然这不是很理想，而且我们也确实希望能验证所有的示例代码，但我们还是决定将未经运行检验的例程包括进来。不足之处是这些例程在你的环境中也许不能编译，因为你的编译器同我们的一样，并不遵从标准。另一个缺点是一些小错误可能会未被查觉而混入了例程。另一个安全的作法是完全省略某些技术，因为它们现在还不能被编译。然而，我们想向读者介绍标准 C++ 所支持的全部技术，因此，这本书的写作着眼于将来（希望是不久之后）的标准 C++ 开发环境。编译器和库总会跟上发展的，那些在你目前的编译器上还不能执行的技术会在你一旦拥有一个真正符合标准的 C++ 编译器后得到实现。因此，我们觉得将这些技术包括进来将有助于提高本书的实用性。

第一部分 输入与输出流

第一章 IO 流基础	3
1.1 输入与输出	3
1.1.1 什么是输入输出流	3
1.1.2 用 IO 流进行文本流 IO 的步骤	4
1.1.3 IO 流的层次	6
1.1.4 IO 流类中流的概念	7
1.1.5 IO 流类中的类	7
1.1.6 IO 流作为框架	9
1.2 格式化输入/输出	9
1.2.1 预定义的全局流	10
1.2.2 输入输出运算符	10
1.2.3 流的格式参数	13
1.2.4 操纵符	18
1.2.5 流的本地化	22
1.2.6 有格式的输入与输出的比较	22
1.2.7 格式输入的特性	23
1.3 流状态	25
1.3.1 流状态标志	25
1.3.2 测试流状态	27
1.3.3 捕获流异常情况	29
1.3.4 复位流状态	31
1.4 文件输入与输出	32
1.4.1 创建、打开、关闭和析构文件流对象	32
1.4.2 打开方式	34
1.4.3 双向文件流	37
1.5 内存内部的输入/输出	39
1.5.1 串流内容的获取和替换	39
1.5.2 打开方式	40