PH
PTR

C#精髓

- 引导程序员快速进入 C# 世界，熟练掌握 C# 编程技巧。
- 涵盖所有 C# 语言重要功能：控件语句、循环、关键字、数组、指针等等。
- 诠释所有关键性概念，提供完整示例程序。

〔美〕Chris H. Pappas
William H. Murray 著

周良忠 译

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

C#精髓

[美] Chris H. Pappas
William H. Murray 著
周良忠 译

人民邮电出版社

图书在版编目(CIP)数据

C#精髓/（美）帕帕斯（Pappas,C.H.），（美）默里（Murray,W.H.）著；周良忠译.
—北京：人民邮电出版社，2002.10
ISBN 7-115-10523-5
I. C… II. ①帕…②默…③周… III. C 语言—程序设计 IV. TP312
中国版本图书馆 CIP 数据核字(2002)第 056812 号

版 权 声 明

C# 精髓

◆ 著 [美] Chris H.Pappas William H.Murray
译 周良忠
责任编辑 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@ptpress.com.cn
网址 http://www. ptpress.com.cn
读者热线 010-67132705
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：19
字数：448 千字 2002 年 10 月第 1 版
印数：1-4 000 册 2002 年 10 月北京第 1 次印刷

著作权合同登记 图字：01 - 2002 - 3740 号
ISBN 7-115-10523-5/TP • 3019
定价：34.00 元

内 容 提 要

本书简明扼要地讲解了C#编程语言的核心知识。

全书共分12章。第1章简单介绍了C#及相关语言的起源。第2章概览了C#的一些独特功能。第3章介绍输入、编辑、保存、生成、执行以及调试一个C#控制台应用程序所需的知识。第4~5章讨论了C#语言的重要标识符、关键字、操作符和逻辑流程控制语句。第6章和第7章分别讨论了数组和指针这两个重要概念。第8章讲解了OOP编程的预备知识，包括数据类型及其转换等。第9章对OOP编程的核心概念进行了详细讲解。第10章介绍C#中的I/O。第11章讨论高级C#编程中需要注意的问题。第12章结合实例讲解了C# Windows编程的界面设计方法。

本书适合于正在学习C#的C或C++程序员，也适合于C#初学者阅读。

译 者 的 话

精彩内容——C 和 C++程序员学习 C#的必读之作

工欲善其事，必先利其器。任何人在成为程序员大家庭中的一员之前的首要任务是选择适合自己的编程语言。影响选择的因素多种多样，但最重要的是语言的易用性和功能性。程序员（或开发团队）在开发一个项目前同样需要选择一种主要的开发语言。强大的功能犹如无坚不摧的精良武器；易用性则降低了程序员进入编程领域的门槛，而且能提高开发效率。C#正是这样一种语言，它吸取了 Visual Basic 的易用性，同时集成了 C 和 C++的强大功能。毫无疑问，C#将成为当今程序员的新宠儿。

本书内容丰富，涵盖了 C#的基础知识以及高级 C#编程技巧。还结合实例讲解了 C# Windows 应用程序的界面设计。

尤其值得读者注意的是，本书时刻将 C#与 C、C++进行详细的对比讲解。这不仅使得 C#初学者能很快掌握所有知识点，而且使 C 和 C++程序员能顺利过渡到 C#的编程环境。这一特色对于读者理解 C#特有的（或与 C 和 C++有根本区别）核心概念大有帮助。例如，对于委托、类型安全、垃圾回收、装箱和拆箱等概念，只有理解了 C 和 C++在这些方面的不足（甚至是缺陷），读者才能更透彻地理解 C#在这些方面的改进（或补充）。

本书的作者是两位具有丰富编程语言教学经验的教授，全书内容编排合理，知识讲解深入浅出。

本书内容丰富，精彩纷呈，是 C 和 C++程序员学习 C#的最佳读本，也是 C#初学者的首选。

真诚奉献——准确、流畅的翻译之作

C#是一种全新的高级编程语言，本书涉及许多新的术语。为了让读者准确理解本书所讲述的知识，译者在翻译中参阅了大量微软技术文档，力求使文中术语与微软.NET 框架软件开发工具包（SDK）保持一致，这样有利于读者的学习和与其他程序员的交流。

因为目前程序员所用的 C#既有中文版本，也有英文版本。尽管原著是基于英文版本而写成的，但为了让读者能轻松上手中文版本，译者对一些主要的集成开发环境（IDE）菜单命令提供了准确的中英文对照。

翻译过程中，在尽量保持原著行文风格的同时，译者力求使译文流畅自然、符合中文表达习惯。

排版约定

本书将所有示例代码用等宽字体排版，如：

```
System.Console.WriteLine("hello Cloudcrown Studio!");
```

联系译者

由于译者水平有限，错误在所难免，望广大读者不吝指正。译者的 E-mail：webmaster@CloudCrown.com。

译者

2002 年 6 月

前　言

这是一本为具有 C 或 C++基础知识的程序员设计的书。本书的目的是向 C 和 C++程序员介绍 C#语言的精髓。本书通过与 C 和 C++的对比，详细讲解了 C#的关键功能。你将学习到如何快速开发代码以及编程策略方面的知识。

本书的目的是让程序员快速进入 C#世界，熟练掌握 C#编程技巧。本书涵盖了 C#的所有功能，还提供了相关示例代码。关键性概念以及完整的例程能让每一位读者很快掌握 C#的精髓。

欢迎进入一个全新的编程世界!

目　录

第1章 迈入C#殿堂

振奋精神，让我们迈入C#（读作C-sharp）殿堂！通过轻松学习本章内容，你很快就能理解C#代码，并开始体验编写C#代码的乐趣。不管你是编程新兵，还是开发老将，你都将最终成为微软新颖别致、功能强大、乐趣无穷的编程语言——C#的忠实信徒！

乐趣无穷？没错！通过C#可以轻松实现GUI（Graphical User Interface，图形用户界面）设计和规划，而Visual Basic程序员已经享受这种“乐趣”多年了。通过C#的这一组件，你只需点击工具栏上的控件，把控件拖到设计页面，就可以可视化地创建一个应用程序或小程序用户界面，而且可以从直观的下拉列表中设置控件的属性。

当今的更多程序越来越强烈地要求Web世界不仅能得到一定的经济效益，而且还要能提供广泛的教育性信息，C#的目的正是成为一种顺应潮流的顶级开发语言。它吸取了Java或J++、CGI、PERL、C/C++以及Visual Basic的精华，融入了程序架构独立于Java的字节代码、本地代码格式等优点，代码方案可以经历不断的改进而不必进行全部重写。

要完成一个具有Web功能的程序开发方案，我们再也不需要让一个Visual Basic程序员来设计界面，让一个C++程序员来编写纯粹、原始的数据处理引擎，再请一个Java、CGI或PERL专家将整个包放到互联网上，让全世界都可以访问它。

为了对C#与当代其他编程语言的语法相似性有一个了解，你可以先浏览第2章“独特的C#”，然后再返回来学习本章内容。下面简单介绍C#语言有趣的发展史。

1.1 从Algol起源

Algol、CPL、BCPL、B、Basic、PL/I、汇编语言、COBOL、Fortran、Pascal、Modula-2、Ada、SmallTalk、Lisp、Java、J++、CGI、PERL、Visual Basic、C、C++以及现在的C#，这一串名字真长！为什么会有这么多的语言？为什么不能创建一种语言来兼并所有的语言？我们应该学习哪一种（编程）语言呢？语言的这种发展趋势何时才是尽头？

C#易于学习和使用，是能解决所有问题的开发工具。在潜心学习这一新语言之前，让我们首先看看C#的发展历史。从中我们可以找到以上所有问题的答案。

在回顾C#的历史时，你可以发现当今许多语言的根源和发展基石。通过解释各语言的特长，我们可以通过相关信息正确应用这些新的开发工具。你可以在C#中找到许多语言的特征。通过对语言发展历史的了解，你可以预测未来编程语言发展和变革的趋势！

1.1.1 为什么在“C#”中能看到“C”

学习C的历史是值得的，因为它揭示了这一语言的成功设计思想，而且有助于理解C#为什么可能是未来数年内首选的编程语言的原因。C#语言的渊源可追溯到Algol 60。

Algol 60 诞生于 1960 年，比 FORTRAN 的问世只晚几年时间。这一具有欧洲血统的语言比较复杂，对后来的编程语言的设计有很大影响。它的设计者注重语法的规则性、模块化结构以及其他一些与高级结构化语言相关的功能。不幸的是，Algol 60 始终没有在美国真正开花结果。许多人认为，这是由于该语言的抽象性和通用性所致。

CPL（Combined Programming Language，复合程序设计语言）的创建者将 Algol 60 不切实际的意图在当时的计算机上变成了现实。不过，跟 Algol 60 一样，CPL 也难学、难以实现，因而导致了它的最终衰落。在继承 CPL 优秀思想的基础上，BCPL（Basic Combined Programming Language，基本复合程序设计语言）的创建者将 CPL 限制到只实现它基本的优秀功能。

当贝尔实验室的 Ken Thompson 为实现早期的 UNIX 而设计 B 语言时，他就试图进一步简化 CPL。他成功地创建了一种针对性非常强的语言，这种语言非常适合应用于他本人所使用的硬件（即 DEC PDP-7），这种机器只有 8 位寄存器大小。不过，BCPL 和 B 这两种语言在实现意图上太过于单纯，它们成为一种受限制的语言，仅仅适用于解决特定的问题。

例如，就在 Ken Thompson 在 DEC PDP-7 上实现了 B 语言的同时，一种新机器 PDP-11 问世了（它是 16 位字长）。PDP-11 的容量比上一代机器 PDP-7 大，当然按现在的标准来评判的话，它的容量仍然相当小。它仅有 24KB 的内存（其中系统要占用 16KB），以及一个 512KB 的固定磁盘。一些人认为要用 B 语言重写 UNIX，但 B 语言由于解释性的设计方式所以速度不快。同时还有另一个问题：B 语言是面向字节（byte-oriented）的，而 PDP-11 是面向字（word-oriented）的。因此，B 语言（取自 Basic 的第一个字母）的后代语言开始了进一步的改进工作，这就是 C 语言（意即结合了上代语言的精华，C 取自 Combine 的第一个字母）。

此时，我们有必要讨论一下 UNIX 操作系统，因为这一系统以及运行于此系统上的大部分程序都是用 C 语言写成的。UNIX OS 最先在新泽西州 Murray Hill 的贝尔实验室（Bell Laboratories）开发成功。通过设计，这一操作系统提供有用的开发工具、瘦命令以及相对开放的环境来达到“程序员友好”的意图。但这并不意味着 C 语言要依附于 UNIX 或其他操作系统或机器。UNIX/C 协作开发环境使得 C 享有“系统编程语言”之誉，因为它可用来编写编译器和操作系统。C 也用于编写许多不同领域的主要程序。

Dennis Ritchie 因为创建了 C 而闻名。C 恢复了 BCPL 和 B 语言所遗弃的一些通用性。它通过数据类型的灵活使用达到这一目的，同时保持了其简单性以及对硬件的直接访问性，这些特征均是 CPL 的设计初衷。

许多个人开发的语言（如 C、Pascal、Lisp 和 APL）具有一种“内聚性”，这是那些大型团体开发的语言（如 Ada、PL/I 和 Algol 60）所不具备的。个人编写的语言往往反映了作者的专业领域，如 Dennis Ritchie 在系统软件（计算机语言、操作系统以及程序生成器）领域的工作很杰出。

知道了 Ritchie 所擅长的专业，我们就不难理解为什么 C 是一种适合于进行系统软件设计的语言了。C 是一种相当低级的语言，它允许你在算法逻辑中通过详细的定义，来达到最高的计算机效率。但是 C 也是一种高级语言，它隐藏了计算机的结构细节，因此提高了编程效率。