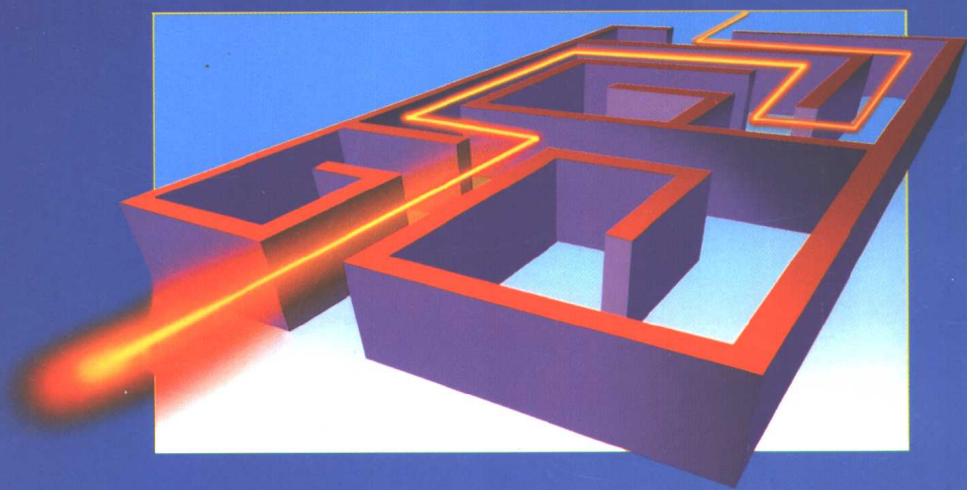


教给你编程起步的基本技能

C#

编程起步

- 创建、编译和运行 C# 程序；
- 使用控制语句、数组、字符串、类、对象和方法工作；
- 学习异常处理、I/O、代理、索引、属性和事件；
- 理解为什么 C# 是适应 .NET 发展的卓越语言。



[美] Herbert Schildt 著
长春亿特 译

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

麦格劳-希尔教育出版集团
www.mheducation.com

Mc
Graw
Hill


C#

编程起步

[美] Herbert Schildt 著

长春亿特 译

 人民邮电出版社

麦格劳-希尔教育出版集团 

图书在版编目 (CIP) 数据

C#编程起步/ (美) 希尔特 (Schildt,H.) 著; 长春亿特译.

—北京: 人民邮电出版社, 2002.4

ISBN 7-115-10069-1

I. C... II. ①希...②长... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 017202 号

版 权 声 明

Herbert Schildt

C# A Beginner's Guide

ISBN: 0-07-213329-5

Copyright © 2001 by the McGraw-Hill Companies, Inc.

Original language published by The McGraw-Hill Companies, Inc. All Rights reserved. No part of this publication may be reproduced or distributed in any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and Posts & Telecommunications Press.

本书中文简体字翻译版由人民邮电出版社和美国麦格劳-希尔教育(亚洲)出版公司合作出版。未经出版者预先书面许可, 不得以任何方式复制或抄袭本书的任何部分。

本书封底贴有 McGraw-Hill 公司激光防伪标签, 无标签者不得销售。

C# 编程起步

-
- ◆ 著 [美] Herbert Schildt
译 长春亿特
责任编辑 陈 昇

 - ◆ 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 010-67180876
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

 - ◆ 开本: 787×1092 1/16
印张: 28.5
字数: 691 千字 2002 年 4 月第 1 版
印数: 1-4 000 册 2002 年 4 月北京第 1 次印刷

著作权合同登记 图字: 01-2001-3344 号

ISBN 7-115-10069-1/TP · 2753

定价: 45.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

内容提要

本书全面系统地介绍了 C# 的各方面内容。全书采用了循序渐进的方式，每一章都以前一章为基础，共分 12 章，分别介绍了 C# 基础、数据类型与运算符、程序控制语句、类、对象、方法、运算符重载、索引、属性、继承、接口、结构、枚举、异常处理、I/O 操作、代理和事件等内容。

本书适合 C# 初学者使用，也可作为各种培训班的教材。

关于作者

Herbert Schildt 是一位世界杰出的编程类图书作者。他是 C、C++、Java 和 C#语言的权威，也是一位 Windows 编程大师。他的编程书籍在全世界范围内销售了 300 多万本，并被翻译成多种语言。他著有许多畅销书，包括《C++: The Complete Reference》、《Java 2: The Complete Reference》、《Java 2: A Beginner's Guide》、《Windows 2000 Programming from the Ground Up》及《C: The Complete Reference》。Schildt 在伊利诺斯州立大学获得计算机博士学位。

前 言

在过去的若干年中，计算技术完成了从独立系统统治的环境到在线的网络环境的飞跃。在“网络就是计算机”的时代，计算机语言、操作系统和开发工具也都争先恐后地发展。C#的诞生适应了现代编程语言的需要，符合互联网世界的需求。C#代表着编程语言演变的第二阶段。它继承了其他语言的优点，并且能和现代语言协作。例如，C#借鉴了C++和Java这两种世界上最重要的语言的特性，它还增加了代理和索引等创新。因为C#利用了.NET框架，用C#编写的代码有很高的可移植性，且允许混合语言使用。例如C#编写的软件组件可以与其他语言编码的代码兼容，只要它也是.NET框架的目标。本书的目的是通过大量例子、自测和项目循序渐进地教会你C#编程的基础，所以，你不需要具有编程经验。本书从基础讲起，比如编译和运行C#程序，然后讨论组成C#语言的关键字、特性和结构。读过之后，你会对C#编程的本质掌握得很牢固。

有必要事先声明，本书仅仅是学习C#的一个起点。C#编程不只包含定义该语言的关键字和语法，还包含类库集合，即.NET框架类库。.NET框架类库非常大，要完整地讨论它恐怕需要一本书的篇幅。尽管本书讨论了该类库定义的一部分类，但限于篇幅，其他大部分内容没有讨论。要成为一名顶尖的C#程序员，你必须掌握这个类库。在学完本书之后，你将有能力自学C#的其他方面的知识。

最后一点是：C#是一门新语言。和其他新的计算机语言一样，C#要经历一个增强和改进的阶段。你可以期待C#出现新的特性和技术，也不必为有些东西改来改去而惊讶。C#的历史才刚刚开始。

本书的组织

本书采用了循序渐进的教程方式，每一章都建立在前一章的基础上，共分12章，每章讨论C#的一个方面内容。本书的独到之处在于它包含了一些特殊部分，用来加深你对所学知识的理解。

1. 目标

每一章的开头都会告诉你学习的目的。

2. 小测验

每一章都以小测验作为总结，这是一个让你测试自己知识的机会。答案

在附录 A 中。

3. 一分钟练习

在每一个主要章节的结尾，你可以通过一分钟练习来检验你对这部分要点的理解。这些练习的答案就在该页的底部。

4. 请教专家

“请教专家”分布于全书各处，包含了附加的信息或者有趣的注释，采用问答的形式。

5. 项目

每一章都包含一个或多个项目，向你展示如何学以致用。这些都是真实的例子，可以作为自己编程的起点。

无需编程经验

本书假定你以前没有编程经验，也就是说，即使你从没有编过程序，也可以使用本书。当然，现在绝大多数读者多少都有一点编程经验，许多人可能有 C++ 或 Java 的编程经验。正如你将要学习的，C# 和它们都有联系。所以，如果你了解 C++ 或 Java，学习 C# 就会更容易。

需要的软件

为了编译和运行本书中的程序，你需要 Visual Studio.NET 7 或更新版本，并且要在你的机器上安装 .NET 框架。本书代码在 Visual Studio.NET 7.0, (Beta 2) 上测试过。

别忘了：Web 上的代码

记住，本书中所有例子和项目的源代码都可以从 Web 站点 www.osborne.com 中免费获得。

目 录

第 1 章 C#基础	1
1.1 C#的家族史	2
1.1.1 现代编程时代的开始	2
1.1.2 OOP 和 C++的诞生	3
1.1.3 因特网和 Java 的兴起	3
1.1.4 C#的诞生	4
1.2 C#与.NET 框架的关系	5
1.2.1 什么是 .NET 框架	5
1.2.2 公共语言运行环境的工作原理	6
1.2.3 受控与非受控代码	6
1.2.4 公共语言规范	6
1.3 面向对象编程	7
1.3.1 封装	8
1.3.2 多态性	8
1.3.3 继承	9
1.4 第一个简单的示例程序	10
1.4.1 使用命令行编译器 csc.exe	10
1.4.2 使用 Visual C++ 集成开发环境	11
1.4.3 逐行解释第一个示例程序	14
1.5 处理语法错误	16
1.6 小变化	17
1.7 第二个简单的示例程序	18
1.8 另一个数据类型	20
1.9 两个控制语句	22
1.9.1 if 语句	23
1.9.2 for 循环	24
1.10 使用代码块	26
1.11 分号和定位	27

1.12	缩排习惯	27
1.13	C#关键字	29
1.14	标识符	30
1.15	C#类库	31
1.16	自我测试	31
第 2 章	数据类型和运算符	33
2.1	数据类型重要的原因	34
2.2	C#的数值类型	34
2.3	整数类型	35
2.4	浮点类型	37
2.5	小数类型	37
2.6	字符	39
2.7	布尔类型	39
2.8	输出选项	41
2.9	常量	44
2.9.1	十六进制 (hexadecimal) 常量	45
2.9.2	字符转义序列	45
2.9.3	字符串常量	46
2.10	详细了解变量	48
2.10.1	变量初始化	48
2.10.2	动态初始化	49
2.11	变量的作用域和生存周期	49
2.12	运算符	52
2.13	算术运算符	52
2.14	关系运算符和逻辑运算符	54
2.15	赋值运算符	59
2.15.1	复合赋值	60
2.15.2	赋值中的类型转换	61
2.16	强制类型转换	62
2.17	运算符的优先级	64
2.18	表达式	64
2.19	表达式中的类型转换	64
2.20	空格和小括号	67
2.21	小测验	69
第 3 章	程序控制语句	71
3.1	从键盘输入字符	72
3.2	if 语句	73

3.2.1	嵌套的 if	74
3.2.2	if-else-if 阶梯式结构	75
3.3	switch 语句	76
3.4	for 循环	83
3.4.1	for 循环的一些变化	85
3.4.2	空白部分	85
3.4.3	没有循环体的循环	87
3.4.4	在 for 循环内部声明循环控制变量	88
3.5	while 循环	89
3.6	do-while 循环	90
3.7	使用 break 语句退出循环	94
3.8	使用 continue	96
3.9	goto 语句	97
3.10	嵌套循环	102
3.11	小测验	103
第 4 章	类、对象及方法简介	105
4.1	类的基础	106
4.1.1	类的一般形式	106
4.1.2	定义类	107
4.2	对象是怎样被创建的	111
4.3	引用变量与赋值	111
4.4	方法	112
4.4.1	给 Vehicle 类添加方法	113
4.4.2	从方法返回	115
4.4.3	返回值	116
4.4.4	使用参数	118
4.4.5	给 Vehicle 类加入带参数的方法	119
4.5	构造函数	126
4.5.1	带参数的构造函数	127
4.5.2	给 Vehicle 类添加构造函数	128
4.6	再次研究 new 运算符	129
4.7	垃圾回收和析构函数	130
4.8	this 关键字	133
4.9	小测验	135
第 5 章	更多的数据类型和运算符	136
5.1	数组	137
5.2	多维数组	142

5.2.1	二维数组	143
5.2.2	三维或更高维数组	144
5.2.3	初始化多维数组	144
5.3	不规则数组	145
5.4	数组引用变量赋值	147
5.5	使用长度属性	148
5.6	foreach 循环	154
5.7	字符串	156
5.7.1	构建字符串	157
5.7.2	字符串操作	157
5.7.3	字符串数组	159
5.7.4	字符串是不可变的	160
5.8	位运算符	161
5.8.1	位运算符 AND、OR、XOR 和 NOT	162
5.8.2	移位运算符	166
5.8.3	位运算符的混合赋值	167
5.9	运算符“?”	170
5.10	小测验	172
第 6 章	深入探讨方法和类	173
6.1	控制类成员的访问	174
6.2	向方法传递对象	180
6.3	使用 ref 和 out 参数	183
6.3.1	使用 ref	183
6.3.2	使用 out	185
6.4	使用可变数量的参数	187
6.5	返回对象	190
6.6	方法重载	192
6.7	重载构造函数	197
6.8	Main()方法	203
6.8.1	从 Main()返回值	203
6.8.2	向 Main()传递参数	203
6.9	递归	205
6.10	理解 static	207
6.11	小测验	212
第 7 章	运算符重载、索引和属性	214
7.1	运算符重载	215

7.1.1	运算符方法的一般形式	215
7.1.2	重载二元运算符	216
7.1.3	重载一元运算符	218
7.1.4	增加灵活性	221
7.1.5	重载关系运算符	225
7.1.6	运算符重载的技巧和限制	227
7.2	索引	228
7.3	属性	234
7.4	小测验	246
第 8 章	继承	247
8.1	继承基础	248
8.2	使用保护性访问	253
8.3	构造函数和继承性	254
8.4	继承和名字隐藏	260
8.5	创建多级层次	266
8.6	何时调用构造函数	269
8.7	基类引用及派生对象	270
8.8	虚拟方法和重载	274
8.8.1	为什么重载方法	276
8.8.2	应用虚拟方法	277
8.9	使用抽象类	280
8.10	使用封装防止继承	284
8.11	对象类	285
8.12	小测验	288
第 9 章	接口, 结构和枚举	289
9.1	接口	290
9.2	使用接口引用	294
9.3	接口属性	301
9.4	接口索引	302
9.5	接口可以被继承	304
9.6	显式实现	306
9.7	结构	308
9.8	枚举	310
9.8.1	枚举初始化	311
9.8.2	指定枚举的基本类型	312
9.9	小测验	312

第 10 章 异常处理	314
10.1 System.Exception 类	315
10.2 异常处理基础	315
10.2.1 使用 try 和 catch	316
10.2.2 一个异常处理的简单例子	316
10.2.3 第二个异常例子	317
10.3 异常未被捕获的后果	319
10.4 异常处理使你从容处理错误	320
10.5 使用多重 catch 语句	321
10.6 捕获所有异常	322
10.7 try 块可以被嵌套	323
10.8 抛出异常	324
10.9 使用 finally	326
10.10 仔细研究异常	328
10.11 派生异常类	330
10.12 捕获派生类异常	332
10.13 使用 checked 和 unchecked	336
10.14 小测验	339
第 11 章 I/O 操作	340
11.1 C#的 I/O 系统是建立在流基础之上的	341
11.1.1 字节流和字符流	341
11.1.2 预定义流	341
11.2 流类(Stream Classes)	342
11.2.1 流类	342
11.2.2 字节流类	343
11.2.3 字符流包装类	343
11.2.4 二进制流	344
11.3 控制台 I/O	345
11.3.1 读控制台输入	345
11.3.2 写控制台输出	346
11.4 文件流与面向字节的文件 I/O	347
11.4.1 打开和关闭文件	347
11.4.2 从 FileStream 读字节	349
11.4.3 写文件	350
11.5 基于字符的文件 I/O	352
11.5.1 使用 StreamWriter	352
11.5.2 使用 StreamReader	354

11.6	重定向标准流	356
11.7	读写二进制数据	359
11.7.1	BinaryWriter	360
11.7.2	BinaryReader	360
11.7.3	二进制数据的 I/O 示例	361
11.8	随机访问文件	363
11.9	把数字串转化为它们内部表达式	365
11.10	小测验	374
第 12 章	代理、事件、名字空间和高级话题	375
12.1	代理	376
12.1.1	多播	379
12.1.2	为什么要使用代理	381
12.2	事件	382
12.3	名字空间	386
12.3.1	声明名字空间	386
12.3.2	using	388
12.3.3	using 的第二种形式	390
12.3.4	名字空间是可添加的	391
12.3.5	名字空间是可嵌套的	393
12.3.6	缺省名字空间	394
12.4	转换运算符	396
12.5	预处理	401
12.6	#define	401
12.6.1	#if 和 #endif	401
12.6.2	#else 和 #elif	403
12.7	#undef	405
12.8	#error	405
12.9	#warning	405
12.10	#line	406
12.11	属性	406
12.11.1	Conditional 属性	407
12.11.2	Obsolete 属性	408
12.12	unsafe 代码	409
12.12.1	简单看一下指针	409
12.12.2	使用 unsafe	411
12.12.3	使用 fixed	411
12.13	运行期类型判别	413
12.13.1	使用 is 测试类型	413

12.13.2	使用 <code>as</code> ·····	414
12.13.3	使用 <code>typeof</code> ·····	414
12.14	其他关键字·····	415
12.14.1	内部访问修改符·····	415
12.14.2	<code>sizeof</code> ·····	415
12.14.3	<code>lock</code> ·····	415
12.14.4	<code>readonly</code> ·····	416
12.14.5	<code>stackalloc</code> ·····	416
12.14.6	<code>using</code> 声明·····	417
12.14.7	<code>const</code> 和 <code>volatile</code> ·····	417
12.15	下一步学习什么·····	417
12.16	小测验·····	418
附录 A	小测验答案·····	419

第 1 章

C#基础

本章目的:

- 了解C#的历史;
- 了解C#与.NET 框架的关系以及C#如何使用.NET 框架;
- 学习面向对象编程的三条原则;
- 创建、编译和运行C#程序;
- 使用变量;
- 使用if 和 for 语句;
- 使用代码块;
- 学习C#的关键字。

自从有了编程规则，人们就在不断地追求完美的编程语言。在这不断追求的过程中，C#成为目前标准的编程语言。由微软公司开发 C#是用来支持基于他们提出的.NET 框架下的程序开发。C#的程序经得起时间验证，这一特性正是它的过人之处。对于现代企业计算环境下的编写程序，包括 Windows 操作系统、因特网、组件等，C#都提供了高可用、有效率的解决方案。在此过程中，C#已经重新定义了编程概念。本书将会教你如何使用它来进行编程。

本章的目的是概要地介绍 C#，包括 C#的产生背景、设计思想和一些重要特性。到目前为止，学习一门编程语言最困难的事情是相互关联各个环节。也就是说，语言的各个部分共同起作用。这种相关性使得我们很难在学习 C#某一方面知识时，不涉及其他方面的知识。为了克服这个问题，本章首先概要地展示了 C#的一些特性，包括 C#程序的基本风格、一些基础控制语句和运算符。本章没有涉及太多的细节，只是着重于所有 C#程序的通用概念。

1.1 C#的家族史

计算机语言不是凭空产生的，它们之间相互关联。一门新的编程语言多多少少要受到它的前辈的影响。这更类似于异花授粉的过程，一门语言可能采用另一门语言的特性，在原有内容的基础上集成新的创新，或是去掉过时的结构。只有这样，语言才会进化，编程的艺术才会发展，C#也不例外。

C#继承了丰富的编程遗产。它是直接从世界上最成功的计算机语言 C 和 C++ 继承而来，又与 Java 紧密相关。理解 C#的关键就是理解这些关系的本质。下面，我们就通过这三种语言的对比来开始 C#的学习历程。

1.1.1 现代编程时代的开始

C 的产生标志着现代编程时代的开始。C 是 20 世纪 70 年代由 Dennis Ritchie 在基于 UNIX 操作系统的 DEC PDP-11 上创建的。尽管早期的语言，像最著名的 Pascal，已经取得了一定的成功，但是，C 语言更是个成功的典范，至今它仍旧是编程语言的主宰。

20 世纪 60 年代的结构化编程造就了 C 语言。在结构化编程语言产生之前，大型的程序是很难编写的。因为往往在编写大型程序的时候，会由于存在大量的跳转、调用和返回很难进行跟踪调试，结果程序逻辑就像意大利面条一样乱糟糟的一团。结构化的编程语言加入了优化定义的控制语句，子程序中采用了局部变量和其他的改进，使得这种问题迎刃而解。采用结构化的编程语言可以编写适度的大型程序。

尽管同一时期还有其他结构化编程语言，但是，C 语言以其强大的功能、精湛的技术和准确的表示第一个赢得了成功。它简洁易用的语法加上程序员（不是语言）主管的思想很快赢得了众多的用户。以今天的眼光我们似乎很难理解 C 的成功。但是，C 语言曾是许多程序员期待已久的令人耳目一新的语言。它最终成为 20 世纪 80 年代使用最广泛的结构化编程语言。