



系列规划教材

深入浅出

C#程序设计

郭仁春 主编 葛晓宇 编著

清华大学出版社



深入
浅出 系列规划教材

深入浅出

C#程序设计

郭仁春 主编 葛晓宇 编著

清华大学出版社
北 京

内 容 简 介

本书从语法讲起,由浅入深,层层深入,尽量用简洁的语言、少量的代码解释复杂的C#程序;主要内容包括C#语法、面向对象编程、Windows编程、数据库、GDI+绘图、网络和线程等。本书适合作为高等院校本科生和研究生的教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

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

深入浅出C#程序设计/郭仁春主编. —北京:清华大学出版社,2017

(深入浅出系列规划教材)

ISBN 978-7-302-44969-0

I. ①深… II. ①郭… III. ①C语言—程序设计—教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2016)第213953号

责任编辑:贾 斌 薛 阳

封面设计:傅瑞学

责任校对:胡伟民

责任印制:何 芊

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

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

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

印 装 者: 三河市春园印刷有限公司

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 23.25

字 数: 566千字

版 次: 2017年2月第1版

印 次: 2017年2月第1次印刷

印 数: 1~2000

定 价: 49.00元

产品编号: 070588-01



为什么开发深入浅出系列丛书?

目的是从读者角度写书,开发出高质量的、适合阅读的图书。

“不积跬步,无以至千里;不积小流,无以成江海。”知识的学习是一个逐渐积累的过程,只有坚持系统地学习知识,深入浅出,坚持不懈,持之以恒,才能把一类技术学习好。坚持的动力源于所学内容的趣味性和讲法的新颖性。

计算机课程的学习也有一条隐含的主线,那就是“提出问题→分析问题→建立数学模型→建立计算模型→通过各种平台和工具得到最终正确的结果”,培养计算机专业学生的核心能力是“面向问题求解的能力”。由于目前大学计算机本科生培养计划的特点,以及受教学计划和课程设置的原因,计算机科学与技术专业的本科生很难精通掌握一门程序设计语言或者相关课程。各门课程设置比较孤立,培养的学生综合运用各方面的知识能力方面有欠缺。传统的教学模式以传授知识为主要目的,能力培养没有得到充分的重视。很多教材受教学模式的影响,在编写过程中,偏重概念讲解比较多,而忽略了能力培养。为了突出内容的案例性、解惑性、可读性、自学性,本书努力在以下方面做好工作。

1. 案例性

所举案例突出与本课程的关系,并且能恰当反映当前知识点。例如,在计算机专业中,很多高校都开设了高等数学、线性代数、概率论,不言而喻,这些课程对于计算机专业的学生来说是非常重要的,但就目前对不少高校而言,这些课程都是由数学系的老师讲授,教材也是由数学系的老师编写,由于学科背景不同和看待问题的角度不同,在这些教材中基本都是纯数学方面的案例,作为计算机系的学生来说,学习这样的教材缺少源动力并且比较乏味,究其原因,很多学生不清楚这些课程与计算机专业的关系是什么。基于此,在编写这方面的教材时,可以把计算机上的案例加入其中,例如,可以把计算机图形学中的三维空间物体图像在屏幕上的伸缩变换、平移变换和旋转变换在矩阵运算中进行举例;可以把双机热备份的案例融入到马尔科夫链的讲解;把密码学的案例融入到大数分解中等。

2. 解惑性

很多教材中的知识讲解注重定义的介绍,而忽略因果性、解释性介绍,往往造成知其然而不知其所以然。下面列举两个例子。

(1) 读者可能对 OSI 参考模型与 TCP/IP 参考模型的概念产生混淆,因为两种模型之

间有很多相似之处。其实,OSI 参考模型是在其协议开发之前设计出来的,也就是说,它不是针对某个协议族设计的,因而更具有通用性。而 TCP/IP 模型是在 TCP/IP 协议栈出现后出现的,也就是说,TCP/IP 模型是针对 TCP/IP 协议栈的,并且与 TCP/IP 协议栈非常吻合。但是必须注意,TCP/IP 模型描述其他协议栈并不合适,因为它具有很强的针对性。说到这里读者可能更迷惑了,既然 OSI 参考模型没有在数据通信中占有主导地位,那为什么还花费这么大的篇幅来描述它呢?其实,虽然 OSI 参考模型在协议实现方面存在很多不足,但是,OSI 参考模型在计算机网络的发展过程中起到了非常重要的作用,并且,它对未来计算机网络的标准化、规范化的发展有很重要的指导意义。

(2) 再例如,在介绍原码、反码和补码时,往往只给出其定义和举例表示,而对最后为什么在计算机中采取补码表示数值?浮点数在计算机中是如何表示的?字节类型、短整型、整型、长整型、浮点数的范围是如何确定的?下面我们来回答这些问题(以 8 位数为例),原码不能直接运算,并且 0 的原码有 +0 和 -0 两种形式,即 00000000 和 10000000,这样肯定是不行的,如果根据原码计算设计相应的门电路,由于要判断符号位,设计的复杂度会大大增加,不合算;为了解决原码不能直接运算的缺点,人们提出了反码的概念,但是 0 的反码还是有 +0 和 -0 两种形式,即 00000000 和 11111111,这样是不行的,因为计算机在计算过程中,不能判断遇到 0 是 +0 还是一-0;而补码解决了 0 表示的唯一性问题,即不会存在 +0 和 -0,因为 +0 是 00000000,它的补码是 00000000,-0 是 10000000,它的反码是 11111111,再加 1 就得到其补码是 10000000,舍去溢出量就是 00000000。知道了计算机中数用补码表示和 0 的唯一性问题后,就可以确定数据类型表示的取值范围了,仍以字节类型为例,一个字节共 8 位,有 00000000~11111111 共 256 种结果,由于 1 位表示符号位,7 位表示数据位,正数的补码好说,其范围从 00000000~01111111,即 0~127;负数的补码为 10000000~11111111,其中,11111111 为 -1 的补码,10000001 为 -127 的补码,那么到底 10000000 表示什么最合适呢?8 位二进制数中,最小数的补码形式为 10000000;它的数值绝对值应该是各位取反再加 1,即为 $01111111+1=10000000=128$,又因为它是负数,所以是 -128,即其取值范围是 -128~127。

3. 可读性

图书的内容要深入浅出,使人爱看、易懂。一本书要做到可读性好,必须做到“善用比喻,实例为王”。什么是深入浅出?就是把复杂的事物简单地描述明白。把简单事情复杂化的是哲学家,而把复杂的问题简单化的是科学家。编写教材时要以科学家的眼光去编写,把难懂的定义,要通过图形或者举例进行解释,这样能达到事半功倍的效果。例如,在数据库中,第一范式、第二范式、第三范式、BC 范式的概念非常抽象,很难理解,但是,如果以一个教务系统中的学生表、课程表、教师表之间的关系为例进行讲解,从而引出范式的概念,学生会比较容易接受。再例如,在生物学中,如果纯粹地讲解各个器官的功能会比较乏味,但是如果提出一个问题,如人的体温为什么是 37°C ?以此为引子引出各个器官的功能效果要好得多。再例如,在讲解数据结构课程时,由于定义多,表示抽象,这样达不到很好的教学效果,可以考虑在讲解数据结构及其操作时,用程序给予实现,让学生看到直接的操作结果,如压栈和出栈操作,可以把 PUSH() 和 POP() 操作实现,这样效果会好

很多,并且会激发学生的学习兴趣。

4. 自学性

一本书如果适合自学学习,对其语言要求比较高。写作风格不能枯燥无味,让人看一眼就拒人千里之外,而应该是风趣、幽默,重要知识点多举实际应用的案例,说明它们在实际生活中的应用,应该有画龙点睛的说明和知识背景介绍,对其应用需要注意哪些问题等都要有提示等。

一书在手,从第一页开始的起点到最后一页的终点,如何使读者能快乐地阅读下去并获得知识?这是非常重要的问题。在数学上,两点之间的最短距离是直线。但在知识的传播中,使读者感到“阻力最小”的书才是好书。如同自然界中没有直流的河流一样,河水在重力的作用下一定沿着阻力最小的路径向前进。知识的传播与此相同,最有效的传播方式是传播起来损耗最小,阅读起来没有阻力。

欢迎联系清华大学出版社白立军老师投稿: bailj@tup.tsinghua.edu.cn。

2014年12月15日



本书以简洁的叙事方式对 C# 语言进行讲述,如果您想高效率地学习 C#,并喜欢代码简洁,请打开本书。

对于 C# 初学者,大多数教材读起来非常吃力,这方面笔者有切身的体会,特别是那些冗长的变量命名,整页的程序代码,令人望而生畏。很多内容其实一句话可以说清,如果用很大篇幅讲解,这对初学者是不利的。本书的写作初衷就是让读者省力,节约时间,为读者扫清阅读障碍。在此方面本书做了最大努力,减少代码量,不让读者有读不下去的感觉。这也是本书不提供配套素材的根本原因。如果您觉得还是需要光盘或是源代码,说明本书做得还不够好,笔者将吸取经验,尽量做到极致。

本书前面部分介绍 VS.NET 和 C# 的基本知识,去除大量无用的专业术语,对菜单工具栏等界面进行简单介绍,让读者可以直接切入 C# 主题。

面向对象概念是 C# 编程的核心内容,“对象”这个术语过于学术化,其真正的意义其实是“物体”,C# 语言在某种程度上来说是对自然界的一种描述语言,与日常用语是一样的。本书尽量将 C# 语言与身边事物和日常语言相结合,力求让学生理解其内涵。

类似委托与事件对初学者来说很难理解,本书可以做到两三行语句让读者透彻理解,这是本书的特色之一。

Windows 和数据库程序应该是很多读者学 C# 语言的目的,特别是数据库编程应用极为广泛,数据库内容在本书中占了大部分篇幅。

用计算机绘图是笔者学习计算机技术的动力之源,C# 提供了 GDI+,可以开发绘图程序和工业实时仿真曲线等,目前很少有教材对 C# 绘图做全面的介绍,但其应用却非常广泛,本书将对 C# 绘图进行全面解析。

网络和线程是物联网、互联网+、虚拟现实、增强现实、移动通信等的核心内容,这些都是应用程序的主流发展方向,本书对此进行了全面分析。

ASP.NET 和 WPF 等内容不包含在本书之内,不是它们不重要,而是它们很复杂,不是一两章内容所能讲清楚的,若因为没有这些内容而放下本书,那是不明智的。

本书希望给 C# 初学者带来福音,能成为读者的案头字典,甚至经典。本书非常适合作为本科生,研究生的教材。也适合那些喜欢编程的人,喜欢成为编程高手的人。

编程不只是技术,同时也是艺术。

编者

2016 年 6 月



第 1 章 初识 C#	1
1.1 Visual Studio.NET 简介	1
1.2 .NET Framework 简介	1
1.3 C# 简介	2
1.4 Visual Studio .NET 开发环境	2
第 2 章 接触 C#	4
2.1 第一个 C# 程序	4
2.2 关于控制台程序	5
2.3 C# 程序基本结构	6
2.3.1 最简单的 C# 结构	6
2.3.2 控制台基本输入和输出语句	6
2.4 编程习惯	8
第 3 章 C# 语法	10
3.1 变量与常量	10
3.1.1 数值类型	10
3.1.2 变量定义	12
3.1.3 常量定义	13
3.1.4 数据类型转换	14
3.2 运算符	15
3.2.1 算术运算符	15
3.2.2 关系运算符	16
3.2.3 逻辑运算符	16
3.2.4 位运算符	17
3.2.5 赋值运算符	17
3.2.6 三元条件运算符	17
3.2.7 运算符的优先级	18
3.3 选择语句	18

3.3.1	if 语句	18
3.3.2	switch 语句	20
3.4	循环	21
3.4.1	while	21
3.4.2	do while	22
3.4.3	for	22
3.4.4	foreach 语句	23
3.4.5	跳转语句	23
3.5	异常	25
3.5.1	try catch 语句	25
3.5.2	throw 语句	26
第 4 章	字符串、时间和简单数组	27
4.1	字符串	27
4.1.1	字符串的定义	27
4.1.2	转义字符	27
4.1.3	字符串的操作	28
4.1.4	数值与字符串的相互转化	30
4.1.5	占位符	31
4.1.6	string.Format()	32
4.1.7	格式化字符串	32
4.1.8	StringBuider	34
4.1.9	字符串索引	35
4.2	简单数组	35
4.2.1	一维数组	35
4.2.2	多维数组	37
4.2.3	锯齿数组	38
4.2.4	数组应用举例	38
4.3	时间	40
4.3.1	DateTime	40
4.3.2	日期时间字符串的显示格式	41
4.3.3	DateTime 的计算	43
4.3.4	TimeSpan	43
第 5 章	面向对象编程	45
5.1	面向对象	45
5.2	类和对象	46
5.3	封装、继承、多态	47

5.4	程序结构	48
5.5	名称空间	51
5.6	部分类	52
5.7	类的成员	53
5.7.1	访问修饰符	53
5.7.2	字段	53
5.7.3	方法	54
5.7.4	属性	55
5.8	过程代码写在哪儿	57
5.9	方法重载	58
5.10	构造函数	60
5.11	析构函数	63
5.12	this	63
5.13	静态成员	64
5.14	运算符重载	67
5.15	引用类型与值类型	70
5.16	装箱与拆箱	71
5.17	函数的参数	72
5.18	索引	78
5.19	结构	81
5.20	枚举	82
第 6 章	继承与多态	83
6.1	继承	83
6.2	base 关键字	84
6.3	派生类的构造函数	85
6.4	Object 类	87
6.5	多态	87
6.6	隐藏方法 new	89
6.7	虚方法 virtual-override	90
6.8	抽象类与抽象方法	91
6.9	密封类和密封方法	92
6.10	ToString()	92
6.11	接口 interface	93
第 7 章	委托与事件	96
7.1	委托定义	96
7.2	委托参数	98



7.3	多播委托	100
7.4	匿名委托	101
7.5	委托耦合	101
7.6	关于事件	103
7.7	发布订阅	104
7.8	事件参数	106
7.9	事件句柄 EventHandler	108
7.10	事件参数 EventArgs	110
第 8 章	泛型	113
8.1	泛型概念	113
8.2	泛型类	113
8.2.1	定义泛型类	113
8.2.2	泛型类的默认值 default	115
8.2.3	泛型约束	116
8.3	泛型接口	117
8.4	泛型结构	118
8.5	泛型方法	119
第 9 章	数组与集合	121
9.1	对象数组	121
9.2	Array 类	122
9.2.1	Array 数组创建	122
9.2.2	Array 数组复制	123
9.2.3	Array 数组排序	124
9.3	元组	125
9.4	列表	126
9.4.1	ArrayList	126
9.4.2	List<T>	127
9.5	队列	132
9.6	栈	133
9.7	链表	133
9.8	有序列表	134
9.9	字典	134
9.10	有序字典	136
9.11	Lookup 类	136
9.12	集	137
9.13	可观察集合	138

9.14 位数组	139
9.14.1 BitArray	139
9.14.2 BitVector32	140
第 10 章 Windows 窗体	143
10.1 第一个窗体	143
10.2 第一个按钮	144
10.3 多个窗体	144
10.4 认识窗体类	145
10.4.1 三个文件	145
10.4.2 Form1.cs	145
10.4.3 Form1.Designer.cs	146
10.5 简易计算器	147
10.6 窗体和控件的属性	148
10.6.1 大小和位置	148
10.6.2 控件的停泊与锚点	148
10.6.3 “捕捉按钮”小游戏	148
10.6.4 颜色设置	149
10.7 鼠标键盘事件	149
10.7.1 鼠标事件	149
10.7.2 键盘事件	151
10.7.3 两个特殊的键盘事件	152
10.8 事件的一些问题	153
10.9 多文档界面	154
10.10 窗体的状态	155
10.11 窗体事件的执行顺序	156
10.12 两个窗体之间传递数据	157
第 11 章 Windows 控件	160
11.1 Windows 工具箱	160
11.2 公共控件	160
11.2.1 RadioButton	161
11.2.2 CheckBox	162
11.2.3 ListBox	163
11.2.4 ComboBox	167
11.2.5 TreeView	168
11.2.6 PictureBox 控件	170
11.3 菜单和工具栏	171

11.3.1	MenuStrip	171
11.3.2	ToolStrip	172
11.3.3	ContextMenuStrip	173
11.4	容器控件	173
11.4.1	FlowLayoutPanel	173
11.4.2	TableLayoutPanel	173
11.4.3	SplitContainer	173
11.4.4	TabControl	174
11.5	对话框	174
11.5.1	ColorDialog 和 FontDialog	174
11.5.2	FolderBrowserDialog、OpenFileDialog 和 SaveFileDialog	175
第 12 章	文件操作	177
12.1	Directory 和 DirectoryInfo	177
12.1.1	Directory	177
12.1.2	DirectoryInfo	177
12.2	File 和 FileInfo	178
12.2.1	File	178
12.2.2	FileInfo	178
12.3	Path 和 DriveInfo	178
12.3.1	Path	178
12.3.2	相对路径	179
12.3.3	DriveInfo	179
12.4	文件读写	180
12.4.1	File 读文件	180
12.4.2	写入文件	180
12.5	流	181
12.5.1	读写二进制文件	181
12.5.2	读写文本文件	183
12.6	资源管理器	184
第 13 章	数据库	187
13.1	C# 与数据库的关系	187
13.2	SQL 语言	188
13.3	ADO.NET	191
13.4	C# 调用 Access 数据库	193
13.4.1	OleDbConnection	193
13.4.2	OleDbCommand	195

13.4.3	OleDbDataReader	197
13.4.4	OleDbDataAdapter	198
13.5	DataTable	199
13.5.1	DataTable 的定义	200
13.5.2	DataColumn	200
13.5.3	DataRow	202
13.5.4	DataTable 提取数据库数据	204
13.5.5	编辑 DataTable 中的数据	205
13.5.6	DataRow 的状态	208
13.5.7	DataRow 的版本	210
13.6	DataGridView	211
13.7	DataSet	213
13.7.1	DatSet 与 DataTable	213
13.7.2	DataSet 关系	216
13.7.3	DataSet 约束	218
13.8	DataGridView	222
13.8.1	行和列	222
13.8.2	单元格	223
13.8.3	行宽和列宽	224
13.8.4	冻结	225
13.8.5	排序	225
13.8.6	隐藏	225
13.8.7	列包含控件	225
13.8.8	虚拟模式	227
13.9	服务器资源管理器	227
13.10	BindingSource	229
13.10.1	BindingSource 的简单应用	229
13.10.2	BindingSource 的定位	229
13.10.3	BindingSource 的排序过滤与查找	232
13.10.4	Binding 类	233
13.11	可视化数据控件	235
13.11.1	DataSet 控件	235
13.11.2	BindingNavigator 控件	236
13.12	窗体控件与数据库	237
13.12.1	ComboBox	237
13.12.2	窗体控件与 DataSet	239
13.12.3	控件的数据库向导	241
13.13	综合实例	242



13.13.1	创建数据库	242
13.13.2	创建主窗体	244
13.13.3	创建操作数据库的公共类	244
13.13.4	创建图书类别窗体	246
13.13.5	创建增加图书窗体	249
13.13.6	创建编辑图书窗体	251
13.13.7	创建图书查询窗体	253
13.14	C#调用 SQL Server 数据库	254
13.14.1	关于 SQL Server Express 数据库	254
13.14.2	创建 SQL Server Express 数据库	255
13.14.3	C#调用 SQL Server Express 数据库	258
第 14 章 Lambda 表达式与扩展方法		260
14.1	代码简化的历程	260
14.1.1	代码简化 1	260
14.1.2	代码简化 2	263
14.1.3	代码简化 3	264
14.1.4	代码简化 4	265
14.1.5	代码简化 5	265
14.1.6	代码简化 6	266
14.2	Lambda 表达式	266
14.3	Func, Action 和 Predicate	268
14.3.1	Func	268
14.3.2	Action	269
14.3.3	Predicate	269
14.4	委托在 List<T>中的简单应用	269
14.5	扩展方法	270
第 15 章 Linq		273
15.1	准备数据	273
15.2	投影和筛选	274
15.3	排序	276
15.4	分组	277
15.5	复合 from	278
15.6	延迟执行与立即执行	280
15.6.1	延迟执行	280
15.6.2	立即执行	282
第 16 章 GDI+ 绘图		283
16.1	GDI 与 GDI+	283

16.2	绘制简单图形	283
16.3	GDI+ 绘图原理	284
16.4	Color	285
16.4.1	RGB	285
16.4.2	ARGB	285
16.4.3	网络安全色	286
16.4.4	Color 结构	286
16.5	Pen 和 Brush	286
16.5.1	Pen	286
16.5.2	Brush	287
16.6	Graphics	287
16.7	Point、Size 和 Rectangle	288
16.7.1	Point 和 PointF	288
16.7.2	Size 和 SizeF	289
16.7.3	Rectangle 和 RectangleF	289
16.8	绘制图形	289
16.9	绘制文本	290
16.10	绘制图片	291
16.11	图形变换	292
16.12	屏幕刷新问题	294
16.13	Invalidate	295
16.14	橡皮筋	297
16.15	Image 和 BitMap	298
16.16	ColorMatrix	301
16.17	绘制函数曲线	302
16.17.1	简单函数曲线	302
16.17.2	动态曲线	303
16.18	不使用 OnPaint、Invalidate 的方法	305
第 17 章	网络编程	306
17.1	网络基本概念	306
17.1.1	TCP/IP	306
17.1.2	Socket(套接字)	307
17.1.3	其他相关概念	307
17.1.4	客户与服务器	308
17.2	Socket 程序	308
17.3	TCP 程序	312
17.4	UDP 程序	314

17.4.1	UDP 客户端	314
17.4.2	UDP 服务端	315
17.5	异步通信	316
17.5.1	同步和异步	316
17.5.2	Socket 异步连接	316
17.5.3	Socket 异步连接	319
17.6	聊天程序	320
17.6.1	客户端	321
17.6.2	服务端	322
第 18 章	线程	324
18.1	进程与线程	324
18.2	异步委托	324
18.2.1	同步委托	324
18.2.2	异步委托	325
18.2.3	IAAsyncResult	326
18.2.4	异步回调 AsyncCallback	327
18.2.5	异步回调的 Lambda 表达式	330
18.3	线程	330
18.3.1	Thread	330
18.3.2	给线程命名	332
18.3.3	给线程传递参数	332
18.3.4	线程的控制	333
18.3.5	优先级	335
18.4	线程池	336
18.5	争用	337
18.6	Windows 窗体的线程问题	338
18.6.1	Timer 控件的问题	338
18.6.2	System.Threading.Timer 类	339
18.6.3	System.Timers 类	340
18.7	BackgroundWorker	341
18.7.1	BackgroundWorker 控件	341
18.7.2	BackgroundWorker 类	344
18.8	窗体与线程交换数据的一种简易方法	345
第 19 章	关于变量命名	347
后记		350