

<<叩响C#之门>>

图书基本信息

书名：<<叩响C#之门>>

13位ISBN编号：9787121090899

10位ISBN编号：7121090899

出版时间：2009-7

出版时间：电子工业出版社

作者：梁斌玉

页数：481

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<叩响C#之门>>

内容概要

本书写给完全没有编程经验的初学者。

它从最基本的C#语法开始，采用逐步构建的学习方法，一步一步地深入到C#编程的核心概念。

当C#语法基础打扎实后，即进入窗体编程的学习，这样读者就有了比较丰富的编程体验，此时进入委托、线程、程序集等深层次概念的学习，便水到渠成、轻松易懂。

本书针对初学者，因为不太久以前作者也是一个初学者，与初学者思维相近，心灵相通，知道初学者的困惑之处，相信学完此书之后，你不光能牢固地掌握C#的基础语法，更能深刻地体会到C#的核心思想。

<<叩响C#之门>>

书籍目录

第 部分 C#语法基础	第1章 初识编程	1.1 C#简介	1.2 Visual Studio开发环境	1.3 认识程序
	1.3.1 创建项目	1.3.2 编写代码	1.3.3 运行程序	1.4 编译
	1.5.1 二进制转换为十进制	1.5.2 十进制转换为二进制	1.6 数据和内存	1.5
第2章 变量	2.1 整型变量	2.1.1 声明整型变量	2.1.2 int型变量的取值范围	
	2.1.3 short型	2.1.4 long型	2.1.5 无符号型	2.1.6 整数类型小结
	2.1.7 溢出	2.2 实数型变量	2.3 字符	2.3.1 字符和字符串
	2.3.2 字符型变量和字符串	2.3.3 字符串的输出	2.3.4 字符串的读取	2.3.5 ASCII编码和Unicode编码
	2.3.6 转义字符	2.3.7 控制符	2.3.8 运算符	2.4 变量的格式化输出
	2.5 变量的命名规则	2.6 常量	2.7 关键字 (Keyword)	2.8 深化练习：交换两个变量的值
第3章 运算符和表达式	3.1 算术运算符	3.2 自增、自减运算符	3.3 赋值运算符	
3.4 优先级	3.5 类型转换	3.5.1 隐式转换	3.5.2 显式转换	3.5.3 字符串和数值间的转换
第4章 流程控制	4.1 算法的概念	4.2 顺序结构	4.3 逻辑表达式	
	4.3.1 关系运算符	4.3.2 逻辑运算符	4.3.3 布尔变量	4.4 选择结构
	4.4.1 if语句	4.4.2 switch语句	4.4.3 条件表达式	4.4.4 深化练习：三个变量排序
	4.5 循环结构	4.5.1 while语句	4.5.2 do-while语句	4.5.3 for语句
	4.6 break语句和continue语句	4.7 循环语句深化练习：分形图	第5章 枚举、结构体和数组	
5.1 枚举	5.2 结构体	5.3 数组	5.3.1 一维数组	5.3.2 foreach循环语句
5.3.3 二维数组	5.3.4 可变数组	第6章 函数	6.1 为什么需要函数	6.2 定义函数
6.3 返回值	6.4 参数	6.4.1 值传递	6.4.2 地址传递	6.4.3 引用型参数 (ref)
	6.4.4 输出型参数 (out)	6.4.5 参数匹配	6.5 递归调用	6.6 变量的作用域
	6.6.1 局部变量	6.6.2 程序块中的局部变量	第7章 面向对象编程：类和对象 (一)	
	7.1 背景	7.2 面向对象的基本概念	7.3 定义类	7.4 声明对象
	7.5 属性	7.6 构造函数	7.7 析构函数和垃圾回收	7.8 实例演练：Time类
第8章 面向对象编程：类和对象 (二)	8.1 以对象为成员	8.2 静态成员	8.2.1 静态变量	8.2.2 静态函数
	8.3 常量成员	8.3.1 const常量	8.3.2 readonly常量	8.4 重载 (Overload)
	8.4.1 函数重载	8.4.2 构造函数的重载	8.4.3 运算符重载	8.5 this关键字
	8.6 索引	8.7 值类型和引用类型	8.7.1 值类型变量	8.7.2 引用型变量
	8.7.3 引用型变量和垃圾回收器	8.8 引用符和对象的区别	8.9 声明对象数组	
8.10 类视图	第9章 面向对象编程：继承	9.1 继承概述	9.2 由基类创建派生类	
9.3 protected成员	9.4 虚函数的重写	9.5 普通函数的隐藏	9.6 base关键字	9.7 抽象类和抽象函数
9.8 密封类和密封函数	9.9 派生类的构造函数	9.10 万类之源	: Object	第10章 面向对象编程：多态性的概念
10.3 多态性的应用	10.4 is运算符	10.5 向下类型转换	10.6 接口	
10.7 类关系图	第11章 .NET框架基本概念	11.1 如何理解.NET框架	11.2 C#和.NET的关系	
11.3 中间语言	11.4 强数据类型	11.5 类型的判定	11.6 命名空间	11.7 装箱和拆箱
11.8 对象的相等	第12章 异常	12.1 异常概述	12.2 捕获异常——try-catch结构	
12.3 finally块——try-catch-finally结构	12.4 抛出异常——throw语句	12.5 .NET中的异常类	12.6 异常类的属性	12.7 自定义异常
第13章 Windows窗体编程 (一)	13.1 窗体 (Form)	13.2 按钮 (Button)	13.3 标签 (Label)	13.4 超链接标签 (LinkLabel)
	13.5 文本框 (TextBox)	13.6 单选按钮 (RadioButton)	13.7 复选框 (CheckBox)	13.8 数字输入框 (NumericUpDown)
	13.9 群组框 (GroupBox)	13.10 面板 (Panel)	第14章 Windows窗体编程 (二)	14.1 鼠标事件处理
	14.2 键盘事件处理	14.3 图形框 (PictureBox)	14.4 列表框 (ListBox)	14.5 复选列表框 (CheckListBox)
	14.6 下拉式列表框 (ComboBox)	14.7 列表视图 (ListView)	14.8 树状视图 (TreeView)	14.9 控件布局
	第15章 Windows窗体编程 (三)			

) 15.1 菜单 15.1.1 主菜单 (MainMenu) 15.1.2 关联菜单 (ContextMenu)
15.2 工具栏 (ToolStrip) 15.3 对话框 15.3.1 另存为对话框 (SaveFileDialog)
15.3.2 打开文件对话框 (OpenFileDialog) 15.3.3 字体对话框 (FontDialog) 15.3.4
颜色对话框 (ColorDialog) 15.3.5 自定义对话框 15.4 多文档界面 15.5 Windows窗
体类的层次结构 第 部分 .NET进阶 第16章 委托与事件 16.1 委托 16.2 多播委托
16.3 匿名函数 16.4 事件处理机制 第17章 绘图 第18章 文件和流 第19章 字符串和正
则表达式 第20章 特性、程序集和反射 第21章 多线程 第22章 集合 第23章 泛型

<<叩响C#之门>>

章节摘录

第1章 初识编程 欢迎同学们来到奇妙的编程世界！

自从1946年世界第一台电子计算机诞生以来，软件技术获得了飞速的发展，形成了众多经典的计算机语言。

2000年6月，微软推出了.NET战略（下一代互联网软件和服务战略），软件设计达到了一个全新的境界。

C#语言是.NET平台的主打语言，寄托着微软对未来的希望，C#都有些什么特点呢？

那么我们就带着这些疑问走进C#的世界吧。

1.1 C#简介 20世纪70年代，人们设计出了高效灵活的C言，后来又在C语言中加入了面向对象的技术形成C++语言。

最近几年，C++一直是最有生命力的程序设计语言，它为我们带来了强大的功能和高度的灵活性，但正是由于C++的复杂和灵活，使我们必须忍受艰苦的学习过程和漫长的开发周期，而且内存泄露和缓冲区溢出的痼疾也使很多初学者头痛不已。

因此，许多程序员一直在寻求一种新的语言，以图在开发能力和学习难度之间取得更好的平衡。

C#语言的出现解决了这种问题。

C#语言源于C++，吸取了C++的长处，摒弃了C++的缺点，增加了很多易用的新特性，同时剔除了C++中一些复杂和容易出错的语法，让碍事的东西统统走开，这使C#在功能强大的同时保持简单易学，鱼和熊掌兼得。

C#语言的特点如下。

1.简洁流畅的语法 随着C++的不断扩充和发展，语法冗余成了C++挥之不去的弊端，类似的功能往往有多种表示方法，它们之间只有细微的区别，这些难以记忆的东西不但增加学习难度，而且非常容易出错。

C#进行了大刀阔斧的改革，相似的功能只保留最常用的一种，让鸡毛蒜皮的东西走开。

简洁流畅成了C#吸引人们目光的资本，易学易用成了C#鹤立鸡群的特征。

2.精心的面向对象设计 C#是一种完全的、纯粹的面向对象语言，不但支持封装、继承、多态性等面向对象技术的关键特征，而且增加了特性（Attribute）、属性（Property）、委托（Delegation）、可容类型（Nullable Xype）等强大功能。

同时C#也是一种传统的面向对象语言，只支持标准的单一继承，避免了C++中因多继承而带来的复杂性。

3.与Web结合紧密 C#语言既适合传统编程，也适合Web编程，它与ASENET技术紧密结合，无缝支持各种分布式应用。

因此，#使创建丰富的Web网站变得容易多了，以致比尔·盖茨说：“我们的目标就是超越今天各自为营的Web站点，把Internet建成一个可以相互交组件的地方。

”

编辑推荐

技术的进步能使普通人完成，过去天才才能完成的任务。
巧妙的讲解能使人理解，过去天才才能理解的知识。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>