微软.NET程序设计系列

Microsoft Press

- 欧美读者评价 ★★★★★
- 著名编程专家精心编著
- 入围欧美电脑图书排行榜
- 大量可重用的范例代码
- 赏心悦目的版式设计
- 附开发职业规划蓝图

MICROSOFT® VISUAL C++ .NET 程序设计

[美] Julian Templeman, Andy Olsen 著

张荩尹　侯天凤　李　钦　等译

Visual Studio .NET产品组　审校

Microsoft .net

清华大学出版社

微软.NET 程序设计系列

Visual C++ .NET 程序设计

[美] Julian Templeman，Andy Olsen 著
张芬尹 侯天凤 李 钦 等译
Visual Studio .NET 产品组 审校

清 华 大 学 出 版 社

(京)新登字 158 号

内 容 简 介

本书全面介绍了微软公司综合性最高、结构最复杂的软件开发工具——Visual C++ .NET。

全书共分 7 部分，内容包括 Visual C++的基本原理、面向对象编程的特性、.NET 程序设计基础知识、.NET 框架的使用、数据访问、分布式应用程序的创建、Visual C++ .NET 高级特性的其他应用等。本书从最基本的原理入手，由托管代码的编写，逐渐深入到.NET 应用程序和 XML Web 服务的运行和调试，对 Visual C++ .NET 程序设计进行了完整描述。

本书适合 Visual C++的初学者用作入门教材，也可供广大 Visual C++ .NET 程序开发人员参考使用。

图书在版编目（CIP）数据

Visual C++ .NET 程序设计 /（美）邓波曼，（美）奥尔森著；张亮尹等译. —北京：清华大学出版社，2002
（微软.NET 程序设计）
书名原文：Microsoft Visual C++ .NET Step by Step
ISBN 7-302-05643-9

Ⅰ.V… Ⅱ.①邓…②奥…③张… Ⅲ.C 语言—程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2002）第 047048 号

北京市版权局著作权合同登记号：图字 01-2001-0110 号

出 版 者：清华大学出版社(北京清华大学学研大厦，邮编 100084)
http://www.tup.tsinghua.edu.cn
责任编辑：韩宏志 向璐
印 刷 者：世界知识印刷厂
发 行 者：新华书店总店北京发行所
开 本：787×960 1/16 **印张**：34.5 **彩插页**：4 **字数**：756 千字
版 次：2002 年 7 月第 1 版 2002 年 7 月第 1 次印刷
书 号：ISBN 7-302-05643-9/TP·3325
印 数：0001～6000
定 价：58.00 元

目　录

前言 xiii

第 I 部分
Visual C++ .NET 入门篇

第 1 章　Visual C++入门 3
1.1　编写第 1 个 C++程序 4
1.1.1　main 函数 5
1.1.2　C++关键字和标识符 5
1.2　创建一个可执行程序的步骤 6
1.2.1　编辑源文件 7
1.2.2　编译源文件 7
1.2.3　链接目标文件 7
1.2.4　运行和测试程序 7
1.3　创建一个实际的可执行程序 8
1.3.1　创建项目 8
1.3.2　为项目添加 C++源文件 10
1.3.3　为源文件添加 C++代码 11
1.3.4　生成可执行文件 12
1.3.5　执行程序 12
1.4　本章小结 13

第 2 章　面向对象编程简介 14
2.1　什么是面向对象编程 14
2.2　面向对象编程语言的特性 15
2.2.1　封装 15
2.2.2　继承 16
2.2.3　多态 16
2.3　类和对象 17
2.4　面向对象编程的主要优点 17
2.5　一个简单示例 18
2.6　本章小结 23

第 3 章　变量和运算符 24
3.1　什么是变量 24
3.2　基本数据类型 24
3.3　声明变量 26
3.4　声明多个变量 27
3.5　为变量赋值 27
3.6　数组 28
3.7　指针 28
3.8　引用 29
3.9　常量 29
3.10　枚举 30
3.11　自定义类型 30
3.12　为类添加成员变量 31
3.13　.NET 框架 String 类 32
3.14　运算符和表达式 33
3.14.1　赋值运算符 33
3.14.2　算术运算符 33
3.14.3　关系运算符和逻辑运算符 34
3.14.4　按位运算符 35
3.14.5　条件运算符 36
3.14.6　sizeof 运算符 36
3.14.7　类型转换 36
3.14.8　运算符优先级和结合性 37
3.15　本章小结 38

第 4 章　函数 40
4.1　声明函数原型 40
4.1.1　声明一个简单的函数原型 41
4.1.2　声明函数原型参数 41
4.1.3　声明函数原型的返回类型 42
4.1.4　函数参数的默认值 43

4.2 定义函数体 43
4.2.1 定义一个简单的函数体....43
4.2.2 定义一个使用参数的函数体....44
4.2.3 定义一个具有返回值的函数体....46
4.3 调用函数 47
4.3.1 在示例程序中调用函数....48
4.3.2 使用调试器单步调试应用程序....49
4.3.3 局部变量和全局变量的作用域....53
4.3.4 函数重载 54
4.4 本章小结 55
第 5 章 判断语句和循环语句 58
5.1 使用 if 语句进行判断 58
5.1.1 执行单路判断 58
5.1.2 执行双路判断 62
5.1.3 执行多路判断 63
5.1.4 执行嵌套判断 66
5.2 使用 switch 语句进行判断 67
5.2.1 定义简单的 switch 语句....68
5.2.2 在 switch 语句中定义传递....70
5.2.3 在 switch 语句中使用传递....70
5.3 循环语句 71
5.3.1 使用 while 循环 71
5.3.2 使用 for 循环....73
5.3.3 使用 do-while 循环 75
5.3.4 执行无条件跳转 77
5.4 本章小结 78

第 II 部分
面向对象程序设计的基础知识

第 6 章 类和对象 85
6.1 将对象组织为头文件和源文件....85
6.1.1 在头文件中定义类....87
6.1.2 在源文件中实现类....88
6.2 创建和销毁对象....90
6.3 定义构造函数和析构函数....92
6.3.1 定义构造函数....92
6.3.2 定义析构函数....94
6.4 定义类范围成员....96
6.4.1 定义类范围数据成员....97
6.4.2 定义类范围成员函数....99
6.5 定义对象关系....101
6.5.1 定义 LoyaltyScheme 类....102
6.5.2 实现 LoyaltyScheme 类....103
6.5.3 创建、使用和销毁 LoyaltyScheme 对象....104
6.5.4 测试应用程序....107
6.6 本章小结....108
第 7 章 控制对象的生存期 111
7.1 传统 C++内存管理....111
7.1.1 创建对象 111
7.1.2 销毁对象 112
7.1.3 手工分配内存的优点和缺点 112
7.2 .NET 方法....114
7.2.1 终结器 115
7.2.2 实现终结器 116
7.2.3 有关终结器的几点说明....117
7.2.4 使用 Dispose 方法....118
7.2.5 集成 Finalize 和 Dispose....119
7.3 本章小结....121
第 8 章 继承....122
8.1 定义一个继承层次结构....122
8.2 定义一个基类....123
8.3 定义一个派生类....125
8.4 访问基类成员....127

8.5 创建对象 129
8.6 覆盖成员函数 131
8.7 定义封装类 135
8.8 定义和使用接口 135
8.9 本章小结 136

第 III 部分
.NET 编程基础

第 9 章 值类型 141
9.1 引用类型和值类型 141
9.1.1 值类型的作用 142
9.1.2 值类型的属性 143
9.2 结构体 143
9.2.1 创建和使用简单结构体 143
9.2.2 对结构体进行细致查看 144
9.2.3 结构体和类的区别 146
9.2.4 为结构体实现构造函数 146
9.2.5 结构体嵌套 146
9.2.6 复制结构体 149
9.3 枚举 149
9.3.1 创建和使用枚举 149
9.3.2 在程序中使用枚举 151
9.3.3 避免多义性 152
9.3.4 高效使用内存 152
9.4 本章小结 152
第 10 章 运算符重载 154
10.1 运算符重载 154
10.1.1 需要重载运算符的类型 155
10.1.2 重载的适用范围 155
10.1.3 重载的规则 155
10.2 托管类型中的重载运算符 156
10.2.1 重载值类型 156
10.2.2 重载算术运算符 156
10.2.3 重载运算符函数 159
10.2.4 实现逻辑运算符和相等运算符 161
10.2.5 实现 Equals 162
10.2.6 实现赋值 164
10.2.7 实现增量和减量 166
10.2.8 重载引用类型 167
10.2.9 实现引用类型的重载运算符 167
10.2.10 调用引用类型的重载运算符 167
10.3 使用重载运算符的规则 167
10.4 本章小结 168
第 11 章 异常处理 169
11.1 什么是异常 169
11.1.1 异常的工作方式 170
11.1.2 异常类型 171
11.2 引发异常 172
11.3 处理异常 174
11.3.1 使用 try 和 catch 构造 174
11.3.2 自定义异常处理 176
11.3.3 使用异常层次结构 177
11.3.4 构造函数引发的异常 177
11.3.5 嵌套和重新引发异常 179
11.3.6 __finally 块 181
11.3.7 catch(...)块 182
11.4 创建自定义异常类型 182
11.5 把__try_cast 用于动态强制类型转换 185
11.6 跨语言使用异常 186
11.7 本章小结 188
第 12 章 数组和集合 190
12.1 本地 C++数组 190
12.1.1 向函数传递数组 192
12.1.2 初始化数组 194

12.1.3 多维数组 194
12.1.4 动态分配数组....195
12.1.5 __gc 数组....196
12.1.6 使用__gc 和__nogc 关键字 197
12.1.7 数组和引用类型....197
12.1.8 多维__gc 数组....198
12.2 .NET 数组类....199
12.2.1 数组的基本操作....200
12.2.2 数组的高级操作....202
12.3 其他.NET 集合类....205
12.3.1 ArrayList 类....206
12.3.2 SortedList 类....208
12.3.3 StringCollection 类....210
12.4 本章小结....211
第 13 章 属性....212
13.1 什么是属性....212
13.2 实现标量属性....213
13.2.1 属性中的错误....214
13.2.2 只读和只写属性....215
13.3 实现索引属性....217
13.3.1 Bank 示例....217
13.3.2 实现 Bank 类....217
13.3.3 添加 Account 类....219
13.3.4 创建 Account 类属性....220
13.3.5 向 Bank 类添加 Accounts ...221
13.3.6 实现 Add 和 Remove 方法....222
13.3.7 实现索引属性以检索账户....223
13.4 本章小结....224
第 14 章 委托和事件....225
14.1 委托简介....225
14.1.1 委托的作用....226
14.1.2 定义委托....227
14.1.3 实现委托....227
14.1.4 使用委托调用静态成员函数....227
14.1.5 使用委托调用非静态成员函数....228
14.1.6 使用多播委托....229
14.2 什么是事件....232
14.2.1 实现事件源类....232
14.2.2 实现事件接收器....234
14.3 本章小结....237

第 IV 部分
使用. NET 框架

第 15 章 .NET 框架类库....241
15.1 什么是.NET 框架....241
15.1.1 公共语言运行库....242
15.1.2 中间语言....242
15.1.3 公共类型系统....242
15.1.4 公共语言规范....243
15.1.5 .NET 框架类库....243
15.1.6 元数据....244
15.1.7 程序集....245
15.2 .NET 框架命名空间....246
15.2.1 在 C++程序中使用命名空间....247
15.2.2 System 命名空间....247
15.2.3 基本类型....248
15.2.4 浮点型....249
15.2.5 集合命名空间....249
15.2.6 集合接口....250
15.2.7 Diagnostic 命名空间....251
15.2.8 IO 命名空间....251
15.2.9 绘图命名空间....252

15.2.10 窗体命名空间.....253
15.2.11 网络命名空间.....254
15.2.12 XML 命名空间.....254
15.2.13 数据命名空间.....255
15.2.14 Web 命名空间.....255
15.3 本章小结.....256
第 16 章 Windows 窗体简介.....257
16.1 什么是 Windows 窗体.....257
16.1.1 Windows 窗体和设计器.....258
16.1.2 Windows 窗体与 MFC.....258
16.1.3 ATL.....259
16.2 System.Windows.Forms 命名空间....259
16.3 创建和使用窗体.....261
16.3.1 创建一个简单窗体.....261
16.3.2 运用窗体属性.....263
16.3.3 窗体关系.....267
16.3.4 在窗体上放置控件.....267
16.3.5 处理事件.....269
16.4 使用控件.....271
16.4.1 标签.....271
16.4.2 按钮.....274
16.4.3 复选框和单选按钮.....275
16.4.4 分组框.....275
16.4.5 列表框和组合框.....277
16.4.6 文本框.....282
16.5 使用菜单.....286
16.5.1 菜单的其他功能.....289
16.5.2 显示上下文菜单.....290
16.6 本章小结.....291
第 17 章 对话框和控件.....292
17.1 使用对话框.....292
17.1.1 DialogResult 属性.....296
17.1.2 通过对话框处理数据.....297
17.1.3 Tab 次序.....301
17.2 使用通用对话框.....301
17.3 控件的补充内容.....303
17.3.1 使用 TreeView 控件.....305
17.3.2 添加目录浏览.....310
17.3.3 使用 ListView 控件.....314
17.3.4 显示目录详细资料.....318
17.3.5 使用拆分器.....321
17.3.6 使用工具栏.....323
17.3.7 使用状态栏.....328
17.4 本章小结.....331
第 18 章 图形输出.....333
18.1 GDI+的图形.....333
18.1.1 System::Drawing 命名空间.....333
18.1.2 Graphics 类.....335
18.1.3 创建 Graphics 对象.....335
18.1.4 绘图对象.....336
18.1.5 Pen.....336
18.1.6 Brush.....336
18.1.7 标准 Pen 和 Brush 对象.....337
18.1.8 绘图操作.....337
18.1.9 Paint 事件.....341
18.1.10 使用颜色.....344
18.1.11 使用字体.....345
18.2 处理图像.....348
18.3 打印.....349
18.4 本章小结.....353
第 19 章 文件处理.....354
19.1 System::IO 命名空间.....354
19.2 使用读取器和写入器进行文本输入输出.....356
19.2.1 TextWriters 的使用.....356
19.2.2 FileStream 类.....358
19.2.3 使用 TextReader.....360

19.3 处理文件和目录.....362
19.4 二进制输入输出.....372
19.4.1 BinaryWriter 类.....372
19.4.2 BinaryReader 类.....373
19.5 本章小结.....377

第 V 部分 数据访问

第 20 章 读写 XML 文档.....381
20.1 XML 和.NET.....381
20.1.1 .NET 中的 XML 命名空间.....382
20.1.2 用于处理 XML 的类.....382
20.2 用 XmlTextReader 解析 XML 文档.....383
20.2.1 验证 XML 文档的结构.....389
20.2.2 处理属性.....389
20.3 带有验证的 XML 解析.....390
20.4 用 XmlTextWriter 来编写 XML.....394
20.5 使用 XmlDocument.....399
20.6 本章小结.....409
第 21 章 转换 XML.....410
21.1 转换 XML.....410
21.2 使用 XPath.....411
21.2.1 XPathNavigator 类.....412
21.2.2 使用 XPathNavigator 类.....414
21.2.3 在 XPathNavigator 中使用 XPath.....416
21.3 使用 XSL.....418
21.4 本章小结.....423
第 22 章 使用 ADO. NET.....424
22.1 ADO.NET 概述.....425
22.1.1 ADO.NET 数据提供程序.....425
22.1.2 ADO.NET 命名空间.....425
22.1.3 ADO.NET 程序集.....426
22.2 创建连接应用程序.....426
22.2.1 连接数据库.....427
22.2.2 创建并执行命令.....429
22.2.3 执行数据修改命令.....430
22.2.4 执行查询及其处理结果.....431
22.3 创建无连接应用程序.....432
22.3.1 创建窗体.....434
22.3.2 创建并配置数据适配器.....436
22.3.3 创建并填写 DataSet.....436
22.4 本章小结.....438

第 VI 部分 创建分布式应用程序

第 23 章 创建 Web 服务.....443
23.1 Web 服务概述.....443
23.1.1 一个 Web 服务场景.....444
23.1.2 Web 服务及前景.....444
23.1.3 Web 服务体系结构.....444
23.1.4 数据格式和协议.....445
23.1.5 Web 服务的描述.....445
23.1.6 Web 服务的发现.....446
23.2 Web 服务命名空间.....447
23.3 创建一个简单的 Web 服务.....448
23.4 通过浏览器使用 Web 服务.....450
23.5 从代码中使用 Web 服务.....451
23.5.1 调试 Web 服务.....455
23.5.2 不使用 Visual Studio .NET.....455
23.6 本章小结.....456
第 24 章 ATL Server 简介.....457
24.1 ATL Server 概述.....457
24.2 用 ATL Server 创建基于 Web 的

应用程序.....459
24.2.1 ATL Server 的体系结构.....459
24.2.2 服务器响应文件(SRF)中的其他内容.....461
24.2.3 使用 ATL Server 编写 Web 应用程序.....462
24.2.4 从浏览器中使用 Web 应用程序.....465
24.3 用 ATL 创建 Web 服务.....466
24.3.1 用 ATL 编写 Web 服务.....466
24.3.2 创建代码框架.....466
24.3.3 修改接口.....467
24.3.4 使用 ATL Server.....470
24.4 本章小结.....472

第 VII 部分 高级功能

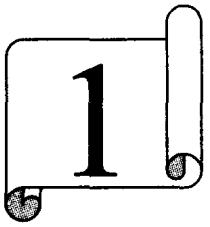
第 25 章 使用非托管代码.....475
25.1 比较托管代码和非托管代码.....475
25.1.1 混合类.....475
25.1.2 GCHandle.....476
25.2 固定和装箱.....479
25.2.1 固定指针.....479
25.2.2 装箱和拆箱.....480
25.3 使用 P/Invoke 调用 Win32 API 中的函数.....484
25.3.1 DllImportAttribute 类.....486
25.3.2 结构化数据的传递.....488
25.4 本章小结.....491
第 26 章 特征和反射.....493
26.1 元数据和特征.....493
26.2 使用预定义特征.....495
26.2.1 AssemblyInfo.cpp 文件.....496
26.2.2 使用预定义的特征类.....496
26.3 自定义特征.....501
26.3.1 特征类的属性.....503
26.3.2 特征类的标准.....503
26.3.3 编写自定义特征.....503
26.4 利用反射获取特征数据.....507
26.4.1 Type 类.....507
26.4.2 访问标准特征.....509
26.4.3 访问自定义特征数据.....510
26.5 本章小结.....512
第 27 章 使用 COM.....514
27.1 COM 组件和 COM 的交互操作.....514
27.2 在.NET 程序中使用 COM 组件.....515
27.2.1 RCW 的工作机制.....515
27.2.2 创建并使用 RCW.....517
27.2.3 处理 COM 错误.....521
27.2.4 对 COM 对象的晚绑定.....522
27.2.5 在 Windows 窗体项目中使用 ActiveX 控件.....523
27.2.6 调用控件中的方法.....527
27.3 将.NET 组件作为 COM 组件使用.....528
27.4 本章小结.....530

第 I 部分

Visual C++ .NET 入门篇

本部分将介绍 Visual C++ .NET 程序设计的基础知识，带您进入丰富多彩的 Visual C++ .NET 编程世界。

本部分首先介绍了 Visual C++编程环境，以及编写可执行程序的步骤，其中包括编辑源文件、编译源文件、链接目标文件、运行和测试程序等内容；接着介绍了 Visual C++语言的一些基本元素：变量、运算符、表达式和函数，采用理论分析与具体示例相结合的方法，阐明了这些元素的定义、特性以及用法；最后介绍了两种重要的结构化编程语句：判断语句和循环语句，讲解如何控制 Visual C++程序的执行流程。



Visual C++入门

本章学习目标

- C++特性
- C++函数
- C++关键字和标识符
- 创建一个 C++程序

欢迎进入精彩的 Microsoft Visual C++ .NET 编程世界！本章将简要介绍 C++语言，并讲述如何使用 C++标准库控制台(基于文本的)工具进行简单的输入输出(I/O)。

到底什么是 C++程序？C++程序所含的元素与其他计算机程序相同，即存储信息的数据以及操纵该数据的代码块。无论您以前是一位 Microsoft Visual Basic 开发者或是一位 40 岁的 COBOL 开发者，都一定不会对 C++包含的基本元素感到陌生。虽然很多人觉得 C++神秘莫测，但实际上这些推测并无确实根据；只要您参照本书认真进行学习，一切都可迎刃而解。

下面将首先介绍 C++的基本原则：

- **C++是一种强类型语言**
  按照一般推理，如果声明一个变量能够存储苹果，那么您只能将苹果存入其中。但 C++却放宽了这种限制，它包括很多可以在适当时候提供隐式转换的功能。因为这种强类型检查明确禁止任何可能导致数据丢失的数据类型转换，从而避免了许多常见的程序错误的出现。
- **C++是一种高效语言**
  如果您需要编写能够快速执行的代码(列表排序或执行复杂的数学运算)，C++应是您首选的语言。
- **C++是面向对象的语言**
  面向对象的编程方法深受现代程序开发者的喜爱(请参见第 2 章，以了解更多有关面向对象编程的信息)。C++是面向对象的最主要的编程语言之一。

- **C++语言以 C 语言为基础**
  C 语言是一种设计完善的语言。C++语言增加了一些奇怪的功能——专门用于与 C 语言保持兼容——如果不能与 C 语言兼容的话，C++就不可能流行起来。
- **C++是区别大小写的语言**
  如果编译器提醒您存在未声明的变量，很可能是您把本应小写的字符大写了(反之亦然)。

1.1 编写第 1 个 C++程序

现在来编写一个简单的 C++程序，让我们还是从最常用的“Hello，World!”程序开始吧。

```
#include <iostream>
using namespace std;
int main()
{
    cout <<"Hello,World!"<<endl;
    return 0;
}
```

这个短程序说明了 C++的一些基本概念：

- 第 1 行使用指令#include，通知 C++编译器在程序开头将文件 iostream(输入输出流)复制进来。C++的黄金法则是，每个元素，包括本程序后面部分使用的 cout 输出流(cout 输出流使输出进入到控制台)，在使用之前都必须进行声明。那么，这里为什么没有对 cout 进行显式声明呢？因为包含 cout 声明的 iostream 文件是一个独立单元，在通过使用#include 语句将 iostream 文件加入到程序中后，程序就隐式包含了 cout 声明。
- 第 2 行告诉编译器使用标准 C++库(std 代表 standard)，可以在单个项目中使用多个不同的库。
- 程序的其余部分是一个 C++函数的范例。在 C++中，所有的代码块都被称作函数——它不使用过程和子例程。每个 C++函数都包含函数头部分(如本程序的第 1 行)和函数体部分(如本程序中用大括号括住的部分)。函数头部分显示了函数的返回类型(本例中的 int 是 integer 的缩写)、函数的名称(main)以及圆括号中的一系列参数。本例没有使用参数，所以圆括号内部是空的；但请注意，虽然圆括号内部为空，但不能将这个括号省略掉。
- 在 C++中，所有的语句都以分号结束。

在本示例的 7 行文本中，只有两行包含 C++语句：cout 行和 return 行。cout 行将字符输出到控制台。cout 行的语法为：cout 之后紧跟一个<<运算符，然后再加上准备输出的项目(使用 endl 流操纵运算符在输出流中插入一行新字符)。

可以用单个 cout 语句输出多个项目，方法是使用多个<<运算符将各个输出项目分隔开，如下所示：

```
cout <<"Hello "<<","<<"World "<<endl;
```

使用以下多个语句，也可以取得与以上单个语句表达形式相同的效果。

```
cout <<"Hello ";
cout <<",";
cout <<"World ";
cout <<endl;
```

当然，程序员一般都倾向于使用单个语句的表达形式。

1.1.1　main 函数

为什么在本例中只使用了名为 main 的函数？这是因为，假如不将这个函数命名为 main 的话，则无法编译程序。要理解这一点，我们需要分析一下 C++语言的工作方式。

一个普通的 C++程序包含很多函数(还包括很多类，详细信息请参阅第 2 章)。那么编译器应如何判断调用哪一个函数呢？显而易见，不能允许编译器无序地调用函数。实际的调用规则是：编译器生成查找 main 函数的代码。如果省略掉 main 函数的话，编译器就会报告一个错误，并且不会生成最终可执行程序。

无格式语言

C++语言没有格式，也就是说编译器会忽略所有的空格、回车标记、换行符、制表符、换页符等等，并把所有这些字符统称为空白。只有在一个字符串内出现空格的情况下，编译器才会识别空白。

由于 C++语言没有格式，所以程序员可以使用制表符或空格缩进作为组织程序布局的方式。代码块中的语句，如 for 循环或 if 语句，通常都采取缩进方式(通常缩进 4 个字符)。这样一来，程序员可以更快地识别程序块中的内容。

但这也引起了 C++界的普遍争论(实际上也没有必要)，应该如何缩进大括号？它们是应该与代码一起缩进呢？还是应该悬挂在 for 循环或 if 语句的句首？对于这个问题而言，没有对或错的回答(可能有许多固执的 C++开发者不赞同这种说法)。但不管采取这两种方式中的哪一种，都可以增加程序的可读性。如单纯从编译器角度来考虑的话，您完全可以把整个程序写为一行！

所以编译器需要一个叫做 main 的函数。除此之外，编译器还对 main 函数的返回类型和使用的参数有具体要求；在使用 main 函数时，一些 C++规则的限制都有所放松。如：main 函数可以使用表示命令行自变量的参数。

1.1.2　C++关键字和标识符

C++**关键字**(也称为**保留字**)是指一些特殊的文本，编译器将以一种特殊的方式使用它们。在上面

的示例程序中，所使用的关键字有：using、 namespace 和 return。关键字不能被用作变量名或函数名，否则，编译器会报告一个错误。

标识符是指程序员用来代表变量和函数的名称。标识符必须以字母开头，并且只能由字母、数字、下划线组成。以下都是有效的 C++标识符：

- My_variable
- AReallyLongName

表 1.1 列举的是无效的 C++标识符。

表 1.1 无效的 C++标识符

标识符	原 因
0800Number	不能以数字开头
You+Me	只能包括字母、数字、下划线
Return	不能使用保留字

编译器错误还是链接器错误？

为了保证程序完全正确，报告错误的应是链接器。在本章后面的部分，将涉及到更多的编译器错误和链接器错误。

除了这些限制之外，任何标识符都是有效的(奇怪的是，标识符 main 不是一个保留字；从理论上讲，可以在程序中使用名为 main 的变量，但不推荐这样做)。不推荐使用表 1.2 所列的一些标识符。

表 1.2 一些有效的、但不推荐使用的标识符

标识符	不推荐使用的理由
main	可能与主函数main混淆
INT	与保留字int非常接近
B4ugotxtme	含义太模糊
_identifier1	虽然可以在开头使用下划线，但最好不要这样做

1.2 创建一个可执行程序的步骤

要生成一个可执行程序，需要经过几个步骤。Visual Studio .NET 可以自动处理这些步骤。为了帮助您了解并理解这些步骤，这里首先对它们进行简要介绍。本章后面介绍第 1 个应用程序的生成时，会再次涉及到这些步骤。

1.2.1　编辑源文件

在创建程序之前，必须编写一些代码。Visual Studio .NET 提供了一个集成的 C++编辑器，这个编辑器具有彩色语法高亮显示和智能感应(IntelliSense)功能，可以显示函数参数信息和单词完整性。

1.2.2　编译源文件

C++编译器将文本源文件转换为具有.obj 扩展名的机器码目标文件。编译器被调入到 Visual Studio .NET 环境后，这个环境将显示所有错误和警告。

但此时生成的目标文件并不是可执行文件。这些文件并不完整，因为它们没有引用不包括用于特定编译的源文件以外的任何函数。

1.2.3　链接目标文件

生成可执行文件的最后一步是链接组成特定项目的所有目标文件。这种链接既包括自编源代码生成的目标文件，也包括系统库(比如 C++标准库或 MFC[Microsoft Foundation Class 库])的目标文件。

链接错误提示的帮助作用不如编译错误信息大。编译错误将给出文件名和错误的行号。而链接器将只给出目标文件的名称，所以通常还需要做一些查找工作。

系统库和类库

系统库和类库经常有很多相关的.obj 文件。处理一个项目的所有库目标文件是一项繁重的工作，所以通常将.obj 文件合并到.lib 库文件中，以求方便。

1.2.4　运行和测试程序

虽然链接可能是创建可执行文件的最后一个步骤，但它不是开发的最后一个步骤。您还应测试和运行程序。

在很多开发环境中，这一步骤常常是程序开发周期中最困难的部分。为此，Visual Studio .NET 提供了一个高效工具——集成调试器。调试器提供的大量功能方便了运行时调试，比如设置断点和监视变量。