

高职高专计算机实用规划教材

—— 案例驱动与项目实践



本书特色：

- 理论点拨恰到好处，精华案例贯穿全书
- 循序渐进适合初学，章节安排利于授课
- 专业选题注重实用，细排代码逻辑清晰

Visual C++ .NET 程序设计与项目实践

李 晶 刘 伟 郑建群 吴晓华 编著

■ Visual C++ .NET

■ 案例驱动

■ 项目实践



提供代码、电子教案下载
<http://www.tup.com.cn>



清华大学出版社

高职高专计算机实用规划教材——案例驱动与项目实践

Visual C++ .NET 程序设计与项目实践

李 晶 刘 伟 郑建群 吴晓华 编 著

清华大学出版社

北 京

内 容 简 介

Visual C++ .NET 是当今非常流行的程序开发工具,深受广大程序员的喜爱。

本书从 Visual C++ .NET 的基本特点开始,全面、系统地讲述基本概念、基本语法、面向对象的编程方法及常见控件的使用、数据库编程和图形图像编程等。针对初学者和自学者,本书从简单的实例入手,力求将复杂的知识用浅显易懂的语言来描述,做到深入浅出。此外,每章都提供了上机指导和一定数量的练习题,以使读者及时巩固所学的知识。

本书内容丰富,讲解透彻,不仅可以作为高等院校计算机专业和其他相关专业的教材,也可以作为广大初学者和非计算机专业人员的学习用书。

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

Visual C++ .NET 程序设计与项目实践/李晶,刘伟,郑建群,吴晓华编著. --北京:清华大学出版社,2010.10
(高职高专计算机实用规划教材——案例驱动与项目实践)

ISBN 978-7-302-23847-8

I. V… II. ①李… ②刘… ③…郑 ④吴… III. C 语言—程序设计—高等学校:技术学校——教材
IV. TP312

中国版本图书馆 CIP 数据核字(2010)第 176899 号

责任编辑:黄 飞 宋延清

装帧设计:杨玉兰

责任校对:周剑云

责任印制:何 芊

出版发行:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

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

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮 购:010-62786544



印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:21.25 字 数:511 千字

版 次:2010 年 10 月第 1 版 印 次:2010 年 10 月第 1 次印刷

印 数:1~4000

定 价:32.00 元

产品编号:035231-01

前 言

Visual Studio 2008 的发布标志着.NET 开发体验的巨大改变, 编程语言得到了很大的改进, 框架也添加了大量的内容, 但更重要的是, 开发工具得到了较大增强。现在的 Visual Studio 包含了代码段、自定义项目模板、重构、数据绑定向导、智能标签、建模工具、自动测试工具、项目和任务管理, 还有更多未列出的内容。这些工具旨在提高工作效率和成功率。本书将对 Visual C++ .NET 2008 及其功能进行详细的介绍。

在软件开发构架上, .NET Framework 将成为未来软件开发的通用平台, 由于.NET Framework 容易移植到各个不同的平台上(包括 Windows、Unix、PDA、手机等), 因此 Visual C++ 2008 程序设计师可以直接运用.NET Framework 中完整且丰富的类库设计出跨平台的软件系统; 另外通过通用语言运行时库(Common Language Runtime, CLR)的接口, 在.NET Framework 上开发的软件, 不论使用哪一种程序语言, 都可以直接相互引用, 在多人共同开发软件系统时, 程序设计师就可以使用自己擅长的编程语言来参与软件开发, 直接解决了跨语言集成的问题。

本书涵盖 Visual C++ .NET 的基础知识、数据库访问知识及数字图像处理技术等内容。在内容安排上, 每章都介绍一个相对独立的主题, 基本可以独立成章, 所以在具备某一基础之后, 读者就可以根据需要有选择地阅读相关的章节。从语言特点上来看, 书中尽量使用通俗化的语言对相关的概念进行描述和讲解, 引用的示例程序也都比较简短, 避免大段枯燥的程序, 因为那样往往会使初学者难以理解程序的内容, 不便于消化和吸收。

全书共分为 13 章。各章的主要内容简要说明如下。

第 1 章简单介绍计算机程序语言的发展、面向对象的概念、Visual C++ .NET 开发环境的安装及使用。

第 2 章讨论 Visual C++ .NET 的语言结构。了解 Visual C++ .NET 语言的基本符号、词汇和程序的基本框架。通过实例掌握 Visual C++ .NET 开发环境的使用。

第 3 章主要是对 Visual C++ .NET 的基本语法要素进行介绍, 包括基本数据类型、变量和常量、运算符和表达式及数据类型的各种转换方法。

第 4 章讲解各种控制语句的使用方法和各种语句分支结构的实现。

第 5 章讲解一维及二维数组的定义和初始化及应用, 以及字符串及字符数组的应用。需要熟练掌握指针的各种应用。

第 6 章介绍与函数有关的基本概念, 以及如何在程序中定义和使用函数等基础内容。此外, 还介绍有关作用域的概念及生存期, 进而提高对变量和函数使用的灵活性。另外, 递归函数、函数重载及内联函数也是该章的重点和难点。

第 7 章明确类和对象的概念, 掌握类的定义、对象的创建和使用。

第 8 章进一步深入了解类, 掌握派生与继承的基本概念, 熟悉类的多继承、虚基类、



虚函数和多态性的应用。

第 9 章讲解控件的基本概念，熟悉常用标准控件的添加方法及高级控件和第三方控件的使用。

第 10 章讲解 SQL 语言的组成，掌握 SQL 的数据定义，SQL 的数据查询和 SQL 的数据操作。

第 11 章讲解 Microsoft ODBC 数据库管理的基本知识，掌握应用 ODBC 编程的方法，掌握使用 ADO、DAO 和 OLE DB 连接数据库的方法以及如何灵活运用 Visual C++ .NET 进行数据库程序的开发。

第 12 章主要介绍数字图像处理编程的基本概念，包括数字图像的概念、色彩系统、调色板，还介绍位图的概念，设备相关和设备无关位图的应用。

第 13 章主要介绍异常处理，处理和避免程序运行中可能产生的错误。该章主要介绍异常处理的基本思想和 Visual C++ .NET 异常处理的实现。

本书由李晶、刘伟、郑建群、吴晓华编写。由于作者水平有限，书中的错误之处在所难免，在这里我们真诚希望广大读者就本书的各个方面提出宝贵的意见，以便我们再版时进行修改。

最后，谨向广大阅读本书的读者表示诚挚的敬意。

编 者



目 录

第 1 章 程序设计基础 1

- 1.1 计算机语言的发展 1
 - 1.1.1 汇编语言 1
 - 1.1.2 面向过程的语言 2
 - 1.1.3 面向对象的语言 2
- 1.2 Visual C++ .NET 程序集 3
- 1.3 Visual C++ .NET 安装与启动 4
 - 1.3.1 Visual C++ .NET 安装 4
 - 1.3.2 Visual C++ .NET 启动 7
- 1.4 Visual C++ .NET 开发环境 7
 - 1.4.1 配置设置 7
 - 1.4.2 Projects 选项卡 8
 - 1.4.3 菜单 9
 - 1.4.4 解决方案、项目和文件 10
 - 1.4.5 工具栏 10
 - 1.4.6 解决方案资源管理器 11
 - 1.4.7 类视图 11
 - 1.4.8 资源视图 12
 - 1.4.9 属性窗口 12
- 1.5 实践训练：创建第一个 C++ .NET 程序 12
- 1.6 习题 13

第 2 章 Visual C++ .NET 语言结构 15

- 2.1 Visual C++ .NET 语言的基本符号 15
- 2.2 Visual C++ .NET 语言的词汇 15
 - 2.2.1 标识符 16
 - 2.2.2 保留字 16
 - 2.2.3 常量 17
 - 2.2.4 变量 17
 - 2.2.5 运算符 18
 - 2.2.6 标点符号 18

- 2.2.7 注释符 19

2.3 程序的基本框架 19

- 2.3.1 结构化程序设计框架 19
- 2.3.2 面向对象程序设计框架 20

2.4 Visual C++ .NET 集成开发环境的使用 20

2.5 标识符的作用域、可见性和生存期 24

- 2.5.1 作用域与可见性 24
- 2.5.2 动态和静态生存期 26

2.6 实践训练：开发基于 Win32 的控制台程序 26

2.7 习题 28

第 3 章 数据类型、表达式和基本运算 29

3.1 Visual C++ .NET 语言的数据类型 29

- 3.1.1 基本类型 29
- 3.1.2 派生类型 30

3.2 常量 31

- 3.2.1 整型常量 31
- 3.2.2 实型常量 32
- 3.2.3 逻辑常量 32
- 3.2.4 字符常量 32
- 3.2.5 枚举型常量 33

3.3 变量 34

- 3.3.1 变量的定义 34
- 3.3.2 符号常量定义语句 36
- 3.3.3 全局变量与局部变量 38

3.4 Visual C++ .NET 的运算符与表达式 39

- 3.4.1 运算符 39
- 3.4.2 表达式 48



3.5 混合运算、赋值及类型转换	53	第5章 数组、指针及引用	88
3.5.1 数据类型的自动转换	53	5.1 一维数组	88
3.5.2 数据类型的强制转换	54	5.1.1 一维数组的定义	88
3.5.3 赋值时的类型转换	55	5.1.2 一维数组的初始化	89
3.6 实践训练: 运算符运算与 数据类型转换	56	5.1.3 一维数组的访问	90
3.7 习题	59	5.1.4 一维数组的应用举例	91
第4章 基本控制语句	61	5.2 二维及多维数组	93
4.1 Visual C++ .NET 基本语句	61	5.2.1 二维及多维数组的定义	93
4.1.1 声明语句	61	5.2.2 二维及多维数组的初始化	94
4.1.2 表达式语句	62	5.2.3 二维及多维数组的访问	95
4.1.3 选择语句	62	5.3 字符串与字符数组	97
4.1.4 循环语句	63	5.3.1 字符串的定义	97
4.1.5 复合语句	64	5.3.2 字符串的存储	97
4.1.6 赋值语句	65	5.3.3 字符数组的定义	98
4.1.7 空语句	65	5.3.4 字符数组的初始化	98
4.1.8 输入输出语句	65	5.3.5 字符数组的输入输出	99
4.2 if 语句实现的分支结构	67	5.3.6 字符串处理函数	103
4.2.1 if 语句	67	5.4 指针	107
4.2.2 if...else 语句	68	5.4.1 指针的定义	107
4.2.3 if 语句的嵌套	70	5.4.2 指针的赋值与使用	108
4.2.4 ?: 语句	72	5.4.3 指针的类型及管理	112
4.3 用 switch 语句实现的多分支 选择结构	73	5.4.4 指针与数组	116
4.4 用 for 语句实现的循环结构	75	5.4.5 指针与动态内存申请	120
4.5 while 语句	78	5.5 引用	122
4.6 do...while 语句	79	5.5.1 引用的概念	122
4.7 转向语句	81	5.5.2 引用的操作	122
4.7.1 break 语句	81	5.5.3 指针与引用的关系	124
4.7.2 continue 语句	82	5.6 实践训练: 数组与指针应用	125
4.8 实践训练: 常用控制语句的使用	83	5.6.1 数组编程实例	125
4.8.1 if 语句的使用	83	5.6.2 指针编程实例	125
4.8.2 switch 语句的使用	84	5.7 习题	126
4.8.3 for 语句的使用	85	第6章 函数	128
4.9 习题	86	6.1 函数的定义	128
		6.2 函数的调用	129
		6.3 函数的声明	131

6.4 函数的类型和返回值	132	7.3.2 默认构造函数和默认析构函数	170
6.5 函数的参数及参数传递	133	7.3.3 拷贝构造函数	171
6.5.1 形式参数和实际参数	133	7.4 对象的生存期	172
6.5.2 函数参数的求值顺序	134	7.4.1 局部对象	172
6.5.3 默认参数	135	7.4.2 静态对象	173
6.5.4 函数的参数传递	137	7.4.3 全局对象	174
6.6 数组、指针与函数	140	7.5 this 指针	174
6.6.1 数组作为函数的参数	140	7.6 静态成员和常成员	175
6.6.2 指针作为函数的参数	143	7.6.1 静态数据成员	175
6.7 变量的作用域、生命周期和存储类型	144	7.6.2 静态函数成员	176
6.7.1 变量的作用域	144	7.6.3 常成员	177
6.7.2 变量的生命周期	147	7.7 友元	178
6.7.3 变量的存储类型	148	7.7.1 友元函数	179
6.8 递归函数	150	7.7.2 友元类	179
6.9 函数重载	153	7.8 对象数组和成员对象	180
6.10 内联函数	155	7.8.1 对象数组	180
6.11 实践训练：函数调用、重载、递归和内联应用	156	7.8.2 成员对象	181
6.11.1 函数的调用编程实例	156	7.9 实践训练：编写类和对象	182
6.11.2 函数重载编程实例	157	7.10 习题	184
6.11.3 递归函数编程实例	158		
6.11.4 内联函数编程实例	158	第 8 章 继承和派生	186
6.12 习题	159	8.1 派生与继承	186
第 7 章 类和对象	161	8.1.1 派生	186
7.1 类	161	8.1.2 继承	186
7.1.1 类的概念	161	8.1.3 派生类的定义和构成	187
7.1.2 类成员的访问控制	162	8.2 派生类对基类的成员的访问	188
7.1.3 类的数据成员和函数成员	164	8.2.1 派生的方式	188
7.2 对象	165	8.2.2 派生类的访问权限	188
7.2.1 对象的概念	165	8.2.3 保护成员和保护派生	190
7.2.2 对象的成员	165	8.3 派生类的构造函数和析构函数	193
7.2.3 程序实例	166	8.3.1 派生类的构造函数	193
7.3 构造函数和析构函数	166	8.3.2 派生类的析构函数	195
7.3.1 构造函数和析构函数的定义	167	8.4 多重继承和虚基类	196
		8.4.1 多继承的含义	196
		8.4.2 多继承类的构造函数和析构函数	197





8.4.3 二义性.....	198	9.4.5 树形视图控件(TreeCtrl 类)	236
8.4.4 虚基类.....	200	9.5 第三方控件	236
8.5 运算符重载.....	203	9.6 实践训练: 编写可视化应用程序.....	237
8.5.1 运算符重载的概念.....	203	9.7 习题	240
8.5.2 重载为类的成员函数.....	204	第 10 章 数据库基础知识.....	242
8.5.3 重载为类的友元函数.....	209	10.1 SQL 语言的组成.....	242
8.6 多态性和虚函数.....	211	10.1.1 关系和表	242
8.6.1 多态性.....	211	10.1.2 SQL 数据类型.....	243
8.6.2 虚函数.....	212	10.1.3 SQL 语句类型.....	246
8.6.3 纯虚函数.....	214	10.2 SQL 的数据定义.....	248
8.6.4 抽象类.....	215	10.2.1 表的建立和删除	248
8.6.5 综合实例.....	215	10.2.2 表的扩充和修改	249
8.7 实践训练: 继承和派生应用	217	10.2.3 视图	249
8.7.1 继承和派生的实例.....	217	10.2.4 索引	251
8.7.2 多继承和虚基类.....	218	10.3 SQL 的数据查询.....	251
8.7.3 虚函数和多态性.....	219	10.3.1 单表查询	252
8.8 习题.....	220	10.3.2 多表查询	252
第 9 章 常用控件编程	223	10.3.3 表达式与函数的使用.....	253
9.1 控件的概述.....	223	10.3.4 子查询	254
9.1.1 控件的概念.....	223	10.3.5 关系代数运算	254
9.1.2 控件的使用方法.....	223	10.4 SQL 的数据操作.....	257
9.2 常用标准控件.....	224	10.4.1 修改数据	257
9.2.1 按钮(Button 类).....	224	10.4.2 插入数据	257
9.2.2 列表框控件(ListBox 类)	228	10.4.3 删除数据	258
9.2.3 文本框控件(Edit 类).....	230	10.5 实践训练: SQL 命令应用	258
9.2.4 组合框控件(ComboBox 类) ...	230	10.6 习题	260
9.2.5 静态控件(Static 类).....	231	第 11 章 基于数据库的程序开发.....	262
9.2.6 滚动条控件(ScrollBar 类).....	231	11.1 Microsoft ODBC 数据库管理.....	262
9.3 控件的动态生成.....	233	11.1.1 概述	262
9.4 高级控件.....	234	11.1.2 MFC 提供的 ODBC 数据库类	263
9.4.1 进度条控件(Progress Bar).....	234	11.1.3 应用 ODBC 编程	263
9.4.2 滑块控件(SlideCtrl 类).....	234	11.2 建立数据连接	265
9.4.3 微调器控件 (SPinButtonCtrl 类).....	235	11.2.1 使用 OLE DB 连接数据库 ...	265
9.4.4 列表视图控件(ListCtrl 类).....	235		

11.2.2 使用 DAO 连接数据库	266	12.4 绘制图形	290
11.2.3 使用 ADO 连接数据库	267	12.4.1 GDI 对象	290
11.3 用 Visual C++ .NET 2008 开发 数据库	268	12.4.2 绘制基本图形	296
11.3.1 Visual C++ .NET 2008 开发数据库技术的特点	268	12.5 位图	298
11.3.2 Visual C++ .NET 2008 开发数据库技术	269	12.5.1 位图的概念	299
11.3.3 如何选择合适的数据库 访问类型	270	12.5.2 显示 DDB 位图	299
11.4 实践训练: 连接并读取 数据库信息	270	12.5.3 DIB API 函数	300
11.4.1 建立数据源	270	12.5.4 构造 DIB 函数库	301
11.4.2 创建工程	271	12.5.5 自定义 DIB 函数库读写 BMP 文件	304
11.4.3 添加窗口组件和代码	272	12.6 实践训练: 画图与位图处理应用	305
11.4.4 访问 ADO 访问数据库的 综合实例	273	12.6.1 画图程序设计	305
11.5 习题	284	12.6.2 位图显示程序	309
第 12 章 数字图像处理编程基础	285	12.7 习题	310
12.1 数字图像的概念	285	第 13 章 异常处理	312
12.2 色彩系统	285	13.1 基本思想	312
12.2.1 RGB 色彩系统	286	13.2 Visual C++ .NET 异常处理的 实现	312
12.2.2 CMY/CMYK 色彩系统	286	13.2.1 异常处理的语法	313
12.2.3 YIQ 色彩系统	286	13.2.2 异常接口声明	315
12.2.4 HIS 色彩系统	287	13.2.3 标准 C++ 异常类	316
12.2.5 YUV 色彩系统	287	13.3 异常处理中的构造和析构	317
12.2.6 YcbCr 色彩系统	287	13.3.1 部分构造	317
12.3 调色板	287	13.3.2 多对象	319
12.3.1 调色板的基本概念	287	13.3.3 动态分配对象	320
12.3.2 调色板的分类	288	13.3.4 析构动态生成的对象	322
12.3.3 MFC 中 CPalette 类封装的 主要功能函数	289	13.3.5 利用 operator delete 函数 释放内存	323
		13.4 实践训练: 异常处理应用	325
		13.5 习题	326

第 1 章 程序设计基础

教学提示

程序设计(Programming)是指设计、编制、调试程序的方法和过程。它是目标明确的智力活动。了解计算机程序语言的发展,面向对象程序设计的概念和 Visual C++ .NET 2008 的运行环境是学好 Visual C++ .NET 2008 的基础。

教学目标

本章简单介绍计算机程序语言的发展、面向对象的概念、Visual C++ .NET 2008 的优势和安装使用。

1.1 计算机程序语言的发展

计算机程序设计语言的发展,经历了从机器语言、汇编语言到高级语言的历程。

1.1.1 汇编语言

1. 汇编语言的产生

计算机进行计算和处理信息是通过执行指令序列来完成的。指令就是使计算机执行各种操作的命令,它是由包括操作码和操作数两部分的二进制代码组成的。每种计算机都有自己的一组指令集,用来提供给用户使用。这种指令集称为计算机的指令系统。直接用计算机指令(即二进制代码)与计算机“交谈”时,计算机能够立即理解并执行。因此,人们称它为机器语言。不过,由于机器语言是二进制的,用它来编写程序,既繁琐又难以记忆,而且总是长长的一串 1 和 0,使用时很不方便。因此,就产生了汇编语言。

汇编语言(Assembly Language)是面向机器的程序设计语言。也是利用计算机所有硬件特性并能直接控制硬件的语言。

在汇编语言中,用助记符(Memori)代替操作码,用地址符号(Symbol)或标号(Label)代替地址码。这样用符号代替机器语言的二进制码,就把机器语言变成了汇编语言,因此汇编语言亦称为符号语言。

使用汇编语言编写的程序,机器不能直接识别,要由一种程序将汇编语言翻译成机器语言,这种起翻译作用的程序叫汇编程序,汇编程序是系统软件中的语言处理系统软件。汇编语言把汇编程序翻译成机器语言的过程称为汇编。

2. 汇编语言的特点

- (1) 它是面向机器的低级语言,通常是为特定的计算机或系列计算机专门设计的。
- (2) 保持了机器语言的优点,具有直接和简捷的特点。
- (3) 可有效地访问、控制计算机的各种硬件设备,如磁盘、存储器、CPU、I/O 端口等。



- (4) 目标代码简短, 占用内存少, 执行速度快, 是高效的程序设计语言。
- (5) 经常与高级语言配合使用, 应用十分广泛。

3. 汇编语言的应用

- (1) 70%以上的系统软件是用汇编语言编写的。
- (2) 某些快速处理、位处理、访问硬件设备等高效程序是用汇编语言编写的。
- (3) 某些高级绘图程序、视频游戏程序是用汇编语言编写的。

汇编语言比机器语言易于读写、易于调试和修改, 同时也具有机器语言执行速度快, 占内存空间少等优点, 但在编写复杂程序时具有明显的局限性, 汇编语言依赖于具体的机型, 不能通用, 也不能在不同机型之间移植。

1.1.2 面向过程的语言

从最初与计算机交流的痛苦经历中, 人们意识到, 应该设计一种这样的语言, 这种语言接近于数学语言或人的自然语言, 同时又不依赖于计算机硬件, 编出的程序能在所有机器上通用。经过努力, 1954 年, 第一个完全脱离机器硬件的高级语言 FORTRAN 问世了, 40 多年来, 共有几百种高级语言出现, 有重要意义的大约几十种, 影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、NOBOL、PL/1、Pascal、C、PROLOG、Ada、C++、VC、VB、Delphi、Java 等。

高级语言的发展也经历了从早期的语言到结构化程序设计语言, 从面向过程到非过程化程序语言的过程。相应地, 软件的开发也由最初的个体手工作坊式的封闭式生产, 发展为产业化、流水线式的工业化生产。

20 世纪 60 年代中后期, 软件越来越多, 规模越来越大, 而软件的生产基本上是人自为战, 缺乏科学规范的系统规划与测试、评估标准, 其恶果是有大批耗费巨资建立起来的软件系统由于含有错误而无法使用, 甚至带来巨大损失, 软件给人的感觉是越来越不可靠, 以致几乎没有不出错的软件。这一切, 极大地震动了计算机界, 史称“软件危机”。人们认识到: 大型程序的编制不同于写小程序, 它应该是一项新的技术, 应该像处理工程一样处理软件研制的全过程。程序的设计应易于保证正确性, 也便于验证正确性。1969 年, 提出了结构化程序设计方法, 1970 年, 第一个结构化程序设计语言——Pascal 语言出现, 标志着结构化程序设计时期的开始。

1.1.3 面向对象的语言

从 20 世纪 80 年代初开始, 在软件设计思想上, 又产生了一次革命, 其成果就是面向对象的程序设计。在此之前的高级语言, 几乎都是面向过程的, 程序的执行是流水线似的, 在一个模块被执行完成前, 人们不能干别的事, 也无法动态地改变程序的执行方向。这和人们日常处理事物的方式是不一致的, 对人而言是希望发生一件事就处理一件事, 也就是说, 不能面向过程, 而应是面向具体的应用功能, 也就是对象(Object)。其方法就是软件的集成化, 如同硬件的集成电路一样, 生产一些通用的、封装紧密的功能模块, 称之为软件集成块, 它与具体应用无关, 但能相互组合, 完成具体的应用功能, 同时又能重复使用。

对使用者来说,只关心它的接口(输入量、输出量)及能实现的功能,至于如何实现的,那是它内部的事,使用者完全不用关心,C++、VB、Delphi 就是典型代表。

面向对象是一种新兴的程序设计方法,或者说它是一种新的程序设计范型,其基本思想是使用对象、类、继承、封装、消息等基本概念来进行程序设计。

它是从现实世界中客观存在的事物(即对象)出发来构造软件系统,并在系统构造中尽可能运用人类的自然思维方式,强调直接以问题域(现实世界)中的事物为中心来思考问题、认识问题,并根据这些事物的本质特点,把它们抽象地表示为系统中的对象,作为系统的基本构成单位(而不是用一些与现实世界中的事物相关比较远,并且没有对应关系的其他概念来构造系统)。这可以使系统直接地映射问题域,保持问题域中事物及其相互关系的本来面貌。

从程序设计的角度来看,面向对象的程序设计语言必须有描述对象及其相互之间关系的语言成分。这些程序设计语言可以归纳为以下几类:系统中一切皆为对象;对象是属性及其操作的封装体;对象可按其性质划分为类,对象称为类的实例;实例关系和继承关系是对象之间的静态关系;消息传递是对象之间动态联系的唯一形式,也是计算的唯一形式;方法是消息的序列。

1.2 Visual C++ .NET 程序集

程序集是生成.NET 框架应用程序的基本单位,也是进行部署和版本控制的基本单位。每一个用.NET 开发的可执行程序、类库或资源文件都是程序集。每个.NET 类型都是程序集的一部分,程序集之外不存在其他的.NET 类型。

程序集是作为一个单元进行版本控制和部署的一个或多个文件的集合。程序集是.NET Framework 应用程序的主要构造块。所有托管类型和资源都包含在某个程序集内,并被标记为只能在该程序集的内部访问,或者被标记为可以从其他程序集中的代码访问。程序集在安全方面也起着重要的作用。代码访问安全系统使用程序集信息来确定为程序集中的代码授予的权限集。

程序集是.NET Framework 编程的基本组成部分。程序集执行以下功能:

- 包含公共语言运行库执行的代码。如果可移植可执行(PE)文件没有相关联的程序集清单,则将不执行该文件中的 Microsoft 中间语言(MSIL)代码。请注意,每个程序集只能有一个入口点(即 DllMain、WinMain 或 Main)。
- 程序集形成安全边界。程序集就是在其中请求和授予权限的单元。有关应用于程序集的安全边界的更多信息,请参见程序集安全注意事项。
- 程序集形成类型边界。每一类型的标识均包括该类型所驻留的程序集的名称。在一个程序集范围内加载的 MyType 类型不同于在其他程序集范围内加载的 MyType 类型。
- 程序集形成引用范围边界。程序集的清单包含用于解析类型和满足资源请求的程序集元数据。它指定在该程序集之外公开的类型和资源。该清单还枚举它所依赖的其他程序集。
- 程序集形成版本边界。程序集是公共语言运行库中最小的可版本化单元,同一程



程序集中的所有类型和资源均会被版本化为一个单元。程序集的清单描述您为任何依赖项程序集所指定的版本依赖性。

- 程序集形成部署单元。当一个应用程序启动时，只有该应用程序最初调用的程序集必须存在。其他程序集(例如本地化资源和包含实用工具类的程序集)可以按需检索。这就使应用程序在第一次下载时保持精简。
- 程序集是支持涇渭执行的单元。有关运行多个程序集版本的更多信息，请参见程序集和涇渭(side-by-side)执行。

程序集可以是静态的或动态的。静态程序集可以包括.NET Framework 类型(接口和类)，以及该程序集的资源(位图、JPEG 文件、资源文件等)。静态程序集存储在磁盘上的可移植可执行(PE)文件中。还可以使用.NET Framework 来创建动态程序集，动态程序集直接从内存运行并且在执行前不存储到磁盘上。可以在执行动态程序集后将它们保存在磁盘上。

1.3 Visual C++ .NET 安装与启动

Microsoft Visual Studio 2008 使开发人员能够快速创建高质量、用户体验丰富而又紧密联系的应用程序，充分展示了 Microsoft 开发智能客户端应用程序的构想。借助于 Visual Studio 2008，采集和分析信息将变得更为简单便捷，业务决策也会因此变得更为有效。任何规模的组织都可以使用 Visual Studio 2008 快速创建能够利用 Windows Vista 和 2007 Office System 的更安全、更易于管理并且更可靠的应用程序。

1.3.1 Visual C++ .NET 安装

- (1) 将 Visual Studio 2008 的安装盘放入光驱，即出现如图 1.1 所示的界面。



图 1.1 安装界面

- (2) 单击“安装 Visual Studio 2008”，将进入安装向导对话框，如图 1.2 所示。



图 1.2 Visual Studio 2008 安装向导

(3) 单击安装向导中的“下一步”按钮，将进入如图 1.3 所示的界面。在其中选中“我已阅读并接受许可条款”复选框，输入产品密钥和名称，然后单击“下一步”按钮。

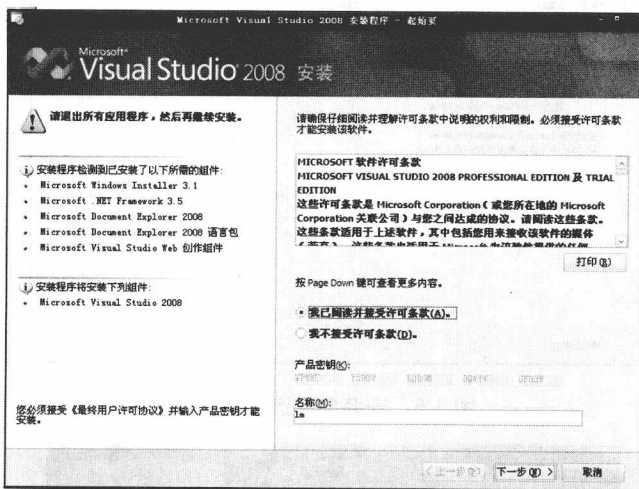


图 1.3 安装向导起始页

(4) 在随后出现的界面中，可以选择安装类型和安装路径，如图 1.4 所示。在本例中，选择的是自定义安装类型和安装路径，选择好后单击“下一步”按钮。

(5) 接下来的界面中，选择安装的功能种类，在这里全部选中即可，如图 1.5 所示。选择好后直接单击“安装”按钮。

(6) 安装过程将自动进行，安装完成后，将出现如图 1.6 所示的对话框，确认无误后单击其中的“完成”按钮即可。至此，Microsoft Visual Studio 2008 就成功地安装到用户的计算机上了。

单击“完成”按钮。安装完 Visual Studio 2008 以后，最好安装 MSDN 产品文档，因为 MSDN 中包含了大量的文档和实例，可以很方便地查询。

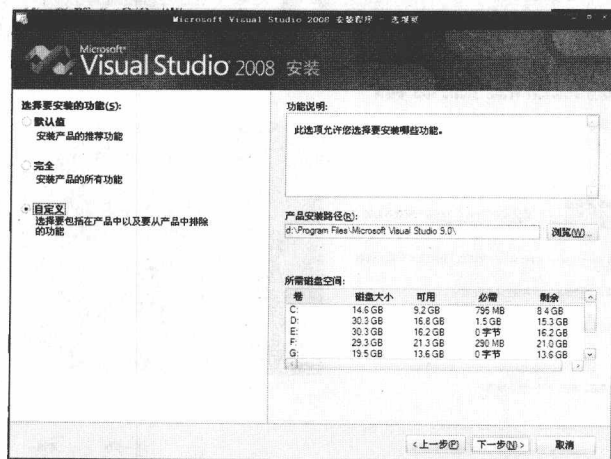


图 1.4 安装向导选项页(1)

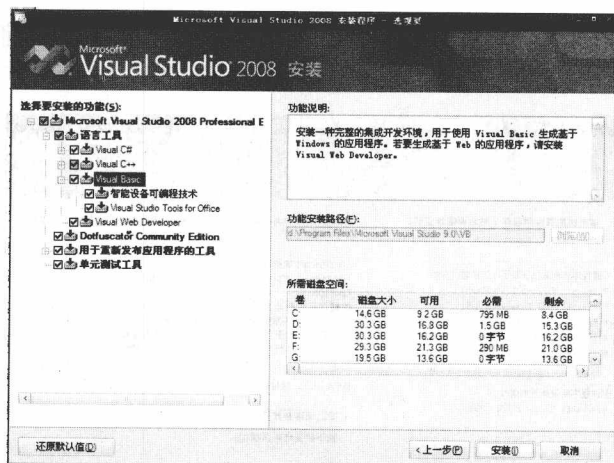


图 1.5 安装向导选项页(2)

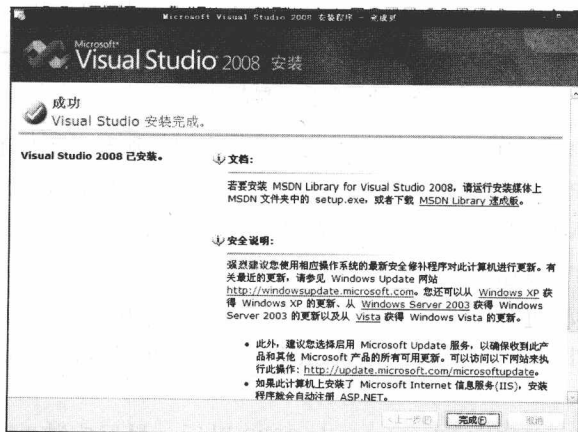


图 1.6 安装向导完成页

1.3.2 Visual C++ .NET 启动

选择 Windows XP 的“开始”→“所有程序”→“Microsoft Visual Studio2008”→“Microsoft Visual Studio 2008”命令,启动 Microsoft Visual Studio 2008。如果是首次使用,会启动选择默认环境设置对话框,选择 Visual C++开发设置,指定经常从事的开发活动类型。在以后启动的时候就不会再出现这个对话框,要想选择使用不同的设置集合,方法是从“工具”菜单中选择“导入和导出设置”,然后选择“重置所有设置”命令。

单击“启动 Visual Studio”按钮,稍等几秒钟,便会启动 Visual Studio 的起始界面,如图 1.7 所示。

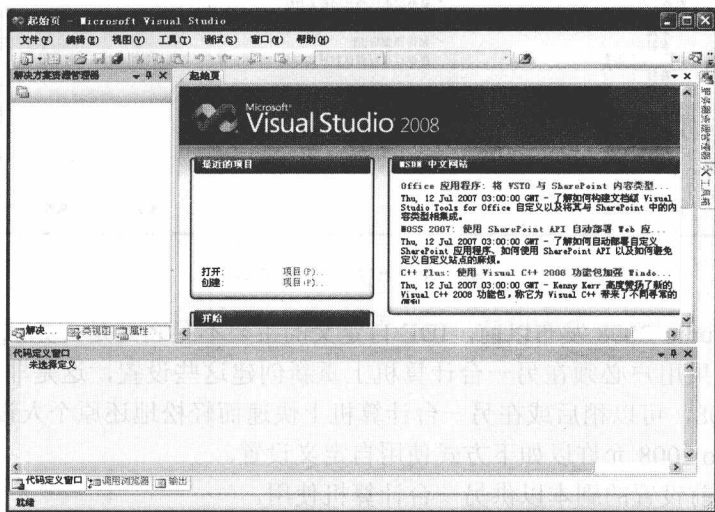


图 1.7 Visual Studio 2008 的起始界面

1.4 Visual C++ .NET 开发环境

Microsoft Visual Studio 2008 包含许多对开发环境新的增强、创新和提高,其目的在于使 Visual C++ .NET 开发人员比以往更加高效。本节探讨其中的一些功能,并且阐述开发人员如何使用它们更快、更准确地编写代码。

除其他对 IDE 的改进之外,Visual Studio 2008 再次引入了“编辑并继续”,这对经验丰富的 Visual C++开发人员而言是个好消息。

1.4.1 配置设置

如果曾经使用过 Visual Studio 的早期版本,那么用户可能会花费一些时间自定义开发环境。用户可能已经通过一些试验了解到可以指定工具窗口的位置,或者可能已经发现 Visual Studio 中的菜单和工具栏是完全可自定义的。也可以配置键盘映射、项目模板类型和帮助筛选器等。事实上,Visual Studio 使用户总是能够将 IDE 元素排列成最适合于个人的开发风格。