

高等学校应用型本科创新人才培养计划指定教材
高等学校计算机类专业“十三五”课改规划教材



Visual C++

■ 程序设计及实践

曲阜师范大学
青岛英谷教育科技股份有限公司 编著



西安电子科技大学出版社
<http://www.xduph.com>

高等学校应用型本科创新人才培养计划指定教材

高等学校计算机类专业“十三五”课改规划教材

Visual C++程序设计及实践

曲阜师范大学 编著

青岛英谷教育科技有限公司

西安电子科技大学出版社

内 容 简 介

本书分为理论篇与实践篇两部分。理论篇深入浅出地讲解了 Visual C++ 的相关知识和概念,共 11 章,内容包括 C++ 编程概述、C++ 基础、函数和指针、C++ 面向对象编程、Win32 编程、MFC 基础、MFC 文档/视图、MFC 对话框、网络编程、多线程以及数据库应用程序等内容。全书从读者的角度出发,层层深入地解析了相关概念,使读者能够快速地了解 Visual C++ 相关知识,掌握 Visual C++ 编程技术。实践篇侧重于案例项目的实现。通过案例项目的提出、分析、参考解决方案等过程,透彻讲解理论篇的相关知识,并将所学的理论快速应用到项目开发中。

本书适应面广,可作为本科计算机科学与技术、软件工程、网络工程、计算机软件、计算机信息管理、电子商务和经济管理等专业程序设计课程的教材,也可供科研人员、程序员等人员参考使用。

图书在版编目(CIP)数据

Visual C++ 程序设计及实践/曲阜师范大学,青岛英谷教育科技有限公司编著.

—西安:西安电子科技大学出版社,2016.1

高等学校计算机类专业“十三五”课改规划教材

ISBN 978-7-5606-3984-0

I. ① V… II. ① 曲… ② 青… III. ① C 语言—程序设计 IV. ① TP312

中国版本图书馆 CIP 数据核字(2015)第 309612 号

策 划 毛红兵

责任编辑 毛红兵 刘炳桢

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2016 年 1 月第 1 版 2016 年 1 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 32.5

字 数 773 千字

印 数 1~3000 册

定 价 79.00 元

ISBN 978-7-5606-3984-0/TP

XDUP 4276001-1

如有印装问题可调换

高等学校计算机类专业 “十三五”课改规划教材编委会

主编 倪建成

编委 王 燕 王成端 薛庆文 孔繁之
李 丽 张 伟 李树金 高仲合
吴自库 陈龙猛 张 磊 吴海峰
郭长友 王海峰 刘 斌 禹继国
王玉锋

❖❖❖ 前 言 ❖❖❖

本科教育是我国高等教育的基础，而应用型本科教育是高等教育由精英教育向大众化教育转变的必然产物，是社会经济发展的要求，也是今后我国高等教育规模扩张的重点。应用型创新人才培养的重点在于训练学生将所学理论知识应用于解决实际问题，这主要依靠课程的优化设计以及教学内容和方法的更新。

另外，随着我国计算机技术的迅猛发展，社会对具备计算机基本能力的人才需求急剧增加，“全面贴近企业需求，无缝打造专业实用人才”是目前高校计算机专业教育的革新方向。为了适应高等教育体制改革的新形势，积极探索适应 21 世纪人才培养的教学模式，我们组织编写了高等院校计算机类专业系列课改教材。

该系列教材面向高校计算机类专业应用型本科人才的培养，强调产学研结合，经过了充分的调研和论证，并参照多所高校一线专家的意见，具有系统性、实用性等特点。旨在使读者在系统掌握软件开发知识的同时，着重培养其综合应用能力和解决问题的能力。

该系列教材具有如下几个特色。

1. 以培养应用型人才为目标

本系列教材以培养应用型软件人才为目标，在原有体制教育的基础上对课程进行了改革，强化“应用型”技术的学习，使读者在经过系统、完整的学习后能够掌握如下技能：

- ✧ 掌握软件开发所需的理论和技术体系以及软件开发过程规范体系；
- ✧ 能够熟练地进行设计和编码工作，并具备良好的自学能力；
- ✧ 具备一定的项目经验，包括代码的调试、文档编写、软件测试等内容；
- ✧ 达到软件企业的用人标准，做到学校学习与企业的无缝对接。

2. 以新颖的教材架构来引导学习

本系列教材采用的教材架构打破了传统的以知识为标准编写教材的方法，采用理论篇与实践篇相结合的组织模式，引导读者在学习理论知识的同时，加强实践动手能力的训练。

- ✧ 理论篇：学习内容的选取遵循“二八原则”，即重点内容由企业中常用的 20% 的技术组成。每个章节设有本章目标，明确本章学习重点和难点，章节内容结合示例代码，引导读者循序渐进地理解和掌握这些知识和技能，培养学生的逻辑思维能力，掌握软件开发的必备知识和技能。
- ✧ 实践篇：集多点于一线，任务驱动，以完整的具体案例贯穿始终，力求使学生在动手实践的过程中，加深对课程内容的理解，培养学生独立分析和解决

问题的能力,并配备相关知识的拓展讲解和拓展练习,拓宽学生的知识面。

另外,本系列教材借鉴了软件开发中的“低耦合,高内聚”的设计理念,组织结构上遵循软件开发中的 MVC 理念,即在保证最小教学集的前提下可以根据自身的实际情况对整个课程体系进行横向或纵向裁剪。

3. 提供全面的教辅产品来辅助教学实施

为充分体现“实境耦合”的教学模式,方便教学实施,该系列教材配备可配套使用的项目实训教材和全套教辅产品。

✧ 实训教材:集多线于一面,以辅助教材的形式,提供适应当前课程(及先行课程)的综合项目,遵循软件开发过程,进行讲解、分析、设计、指导,注重工作过程的系统性,培养读者解决实际问题的能力,是实施“实境”教学的关键环节。

✧ 立体配套:为适应教学模式和教学方法的改革,本系列教材提供完备的教辅产品,主要包括教学指导、实验指导、电子课件、习题集、实践案例等内容,并配以相应的网络教学资源。教学实施方面,本系列教材提供全方位的解决方案(课程体系解决方案、实训解决方案、教师培训解决方案和就业指导解决方案等),以适应软件开发教学过程的特殊性。

本书由曲阜师范大学、青岛英谷教育科技有限公司编写,参加编写工作的有倪建成、王燕、宁维巍、宋国强、何莉娟、杨敬熹、田波、侯方超、刘江林、方惠、莫太民、邵作伟、王千等。本书在编写期间得到了各合作院校专家及一线教师的大力支持与协作,在此衷心感谢每一位老师与同事为本书出版所付出的努力。

由于编者水平有限,书中难免有不足之处,欢迎大家批评指正!读者在阅读过程中发现问题,可以通过邮箱(yinggu@121ugrow.com)发给我们,以期进一步完善。

本书编委会
2015年8月

◆◆◆ 目 录 ◆◆◆

理 论 篇

第 1 章 C++ 编程概述	3	2.3.5 赋值运算符	29
1.1 C++ 简介	4	2.3.6 条件运算符	29
1.1.1 C++ 的发展历程	4	2.3.7 sizeof 运算符	30
1.1.2 C++ 标准	4	2.3.8 运算符优先级	30
1.1.3 C++ 特性	5	2.4 流程控制	31
1.2 C++ 编程环境	5	2.4.1 分支结构	31
1.2.1 主流开发工具	5	2.4.2 循环结构	37
1.2.2 Visual Studio	6	2.4.3 转移语句	41
1.2.3 联机帮助	7	2.5 数组	43
1.2.4 Visual C++ 项目类型	7	2.5.1 数组概述	43
1.3 第一个 C++ 程序	9	2.5.2 一维数组	44
1.4 输入/输出	10	2.5.3 二维数组	47
1.4.1 标准输入与输出对象	10	2.6 字符串	49
1.4.2 使用 cout 对象进行输出	11	2.6.1 字符串和字符集概述	49
1.4.3 使用 cin 对象进行输入	11	2.6.2 C 风格的字符串的定义	49
本章小结	12	2.6.3 常用字符串操作函数	50
本章练习	12	本章小结	51
第 2 章 C++ 基础	13	本章练习	51
2.1 常量和变量	14	第 3 章 函数和指针	53
2.1.1 标识符	14	3.1 函数	54
2.1.2 常量	14	3.1.1 函数的声明和定义	54
2.1.3 变量	15	3.1.2 函数的返回	55
2.1.4 变量作用域及生存期	16	3.1.3 形参和实参	55
2.2 数据类型	17	3.1.4 内联函数	57
2.2.1 基本内置类型	18	3.1.5 函数重载	58
2.2.2 自定义类型	19	3.1.6 静态函数	59
2.2.3 数据类型转换	20	3.1.7 C++ 中的 const 函数	59
2.3 运算符	24	3.2 指针	61
2.3.1 算术运算符	24	3.2.1 指针概述	61
2.3.2 比较运算符	25	3.2.2 指针的定义和初始化	61
2.3.3 逻辑运算符	26	3.2.3 指针的算术运算	62
2.3.4 位运算符	27	3.2.4 指针作为函数参数	64

3.2.5 指针操作动态内存(new 和 delete) ...66	5.1.6 国际化编程 119
3.2.6 函数指针67	5.1.7 Windows 应用程序文件组成 121
3.3 引用71	5.2 源程序组成结构 121
3.3.1 引用的定义和初始化71	5.2.1 入口函数 WinMain() 121
3.3.2 引用作为函数参数72	5.2.2 窗口函数 125
3.3.3 引用和指针的区别73	5.3 GDI 绘图基础 130
3.4 函数参数的传递方式73	5.3.1 GDI 概述 130
3.4.1 值传递(value)73	5.3.2 设备描述表 130
3.4.2 指针传递(address)74	5.3.3 绘图对象 131
3.4.3 引用传递(reference)75	5.3.4 GDI 绘图 131
本章小结76	本章小结 137
本章练习76	本章练习 138
第 4 章 C++ 面向对象编程77	第 6 章 MFC 基础 139
4.1 类与对象78	6.1 MFC 概述 140
4.1.1 面向对象的概念78	6.1.1 MFC 简介 140
4.1.2 类的定义79	6.1.2 一个简单的 MFC 程序 140
4.1.3 构造函数和类的实例化84	6.1.3 MFC 应用程序分析 142
4.1.4 析构函数87	6.1.4 MFC 应用程序开发方法 144
4.1.5 成员函数90	6.2 MFC 应用程序框架组成 146
4.1.6 内联成员函数93	6.2.1 MFC 全局函数 146
4.1.7 静态类成员95	6.2.2 MFC 类体系 148
4.1.8 this 指针97	6.2.3 MFC 消息映射 150
4.2 类继承97	6.2.4 MFC 应用程序类型 153
4.2.1 继承97	6.3 MFC 主要类 154
4.2.2 派生类对象的构造99	6.3.1 根类 CObject 154
4.2.3 派生类对象的析构 101	6.3.2 应用程序体系结构类 155
4.3 虚函数和多态 106	6.3.3 可视对象类 157
4.3.1 虚函数 106	6.4 MFC 通用类 159
4.3.2 虚析构函数 108	6.4.1 字符串类 CString 159
4.3.3 纯虚函数和抽象基类 111	6.4.2 文件类 CFile 162
本章小结 113	6.4.3 动态数组类 164
本章练习 114	6.4.4 链表类 165
第 5 章 Win32 编程 115	6.5 MFC 绘图 166
5.1 Windows 编程基础 116	6.5.1 MFC 对 GDI 的支持 167
5.1.1 Windows 程序的内部运行原理 116	6.5.2 CDC 类及其派生类 167
5.1.2 API 116	6.5.3 MFC 绘图对象 170
5.1.3 事件与消息 117	6.5.4 图形绘制相关数据类型 172
5.1.4 窗体与句柄 117	本章小结 173
5.1.5 常用的 Windows 数据类型 119	本章练习 174

第7章 MFC 文档/视图	175	本章练习	238
7.1 文档/视图框架的概念	176	第8章 MFC 对话框	239
7.1.1 文档/视图结构	176	8.1 对话框基础知识	240
7.1.2 文档/视图程序的类和关系	180	8.1.1 对话框种类	240
7.1.3 单文档程序中文档/视图对象的 创建过程	182	8.1.2 CDialog 类	241
7.1.4 文档类、视图类核心函数	186	8.2 对话框创建和显示	241
7.1.5 文档/视图界面设计	187	8.2.1 创建对话框模板资源	242
7.1.6 在 View 中画线	189	8.2.2 模态对话框的创建和关闭	244
7.2 菜单编程	193	8.2.3 非模态对话框的创建和销毁	247
7.2.1 菜单的结构	193	8.2.4 基于对话框的应用程序	248
7.2.2 菜单的开发步骤	194	8.3 标准控件与对话框	249
7.2.3 菜单的创建与编辑	194	8.3.1 标准控件简介	249
7.2.4 菜单消息	200	8.3.2 对话框控件的创建	250
7.2.5 菜单命令消息的处理	200	8.3.3 对话框控件成员变量	250
7.2.6 菜单更新消息的处理	203	8.3.4 访问对话框控件	252
7.2.7 为菜单项设置快捷键	205	8.3.5 对话框数据交换与验证	255
7.2.8 弹出菜单项的创建与使用	207	8.3.6 对话框控件通知消息	259
7.3 工具栏编程	210	本章小结	261
7.3.1 工具栏 CToolBar 简介	210	本章练习	262
7.3.2 编辑和创建工具栏	211	第9章 网络编程	263
7.3.3 加载工具栏	213	9.1 Windows 网络编程基础	264
7.3.4 工具栏命令处理	214	9.1.1 网络通信的基本原理	264
7.4 状态栏编程	215	9.1.2 数据封装	264
7.4.1 状态栏 CStatusBar 简介	215	9.1.3 IP 地址和端口	265
7.4.2 状态栏的创建	215	9.1.4 套接字和字节序	265
7.4.3 状态栏编程	217	9.1.5 Windows Sockets 基础	265
7.5 文本编程	219	9.1.6 客户/服务器模式	266
7.5.1 创建插入符	219	9.1.7 Socket API 编程	266
7.5.2 字符输入	222	9.1.8 MFC 对 Winsock 的支持	271
7.6 串行化	226	9.1.9 MFC 网络编程方法	272
7.6.1 串行化概述	226	9.2 UDP 通信编程	273
7.6.2 文档/视图结构对串行化的支持	226	9.2.1 CAsyncSocket 类 UDP 异步通信	273
7.6.3 串行化深入理解	230	9.2.2 创建 CAsyncSocket 类的派生类	274
7.6.4 CArchive 类使用	231	9.2.3 创建套接字	277
7.6.5 可串行化的类	235	9.2.4 UDP 接收数据	280
7.6.6 CObArray 和 CObList 类对 串行化的支持	237	9.2.5 UDP 发送数据	281
本章小结	237	9.3 TCP 通信编程	283
		9.3.1 CAsyncSocket 类服务端编程	283
		9.3.2 CSocket 类客户端编程	289

本章小结	293
本章练习	293
第 10 章 多线程	295
10.1 内核对象	296
10.2 进程和线程	296
10.3 多线程编程基础	297
10.3.1 进程地址空间	297
10.3.2 线程	297
10.3.3 跨进程边界共享内核对象	298
10.3.4 使用多线程的意义	298
10.4 多线程编程	298
10.4.1 API 多线程编程	299
10.4.2 MFC 界面线程	306
10.4.3 MFC 工作者线程	311
10.5 线程间的通信	312
10.5.1 使用全局变量	312
10.5.2 使用自定义消息	314
10.6 线程同步	319
10.6.1 MFC 线程同步类	319
10.6.2 CCriticalSection 类	320

10.6.3 CEvent 类	322
10.6.4 CSemaphore 类	326
本章小结	329
本章练习	330
第 11 章 数据库应用程序	331
11.1 数据库基础知识	332
11.1.1 数据库基本概念	332
11.1.2 数据库管理系统	332
11.1.3 结构化查询语言	333
11.1.4 数据库访问技术	335
11.2 ADO 数据库编程	336
11.2.1 ADO 技术	336
11.2.2 导入 ADO 库	339
11.2.3 连接数据库	339
11.2.4 执行 ADO 对象命令	341
11.2.5 Recordset 对象	342
本章小结	354
本章练习	354

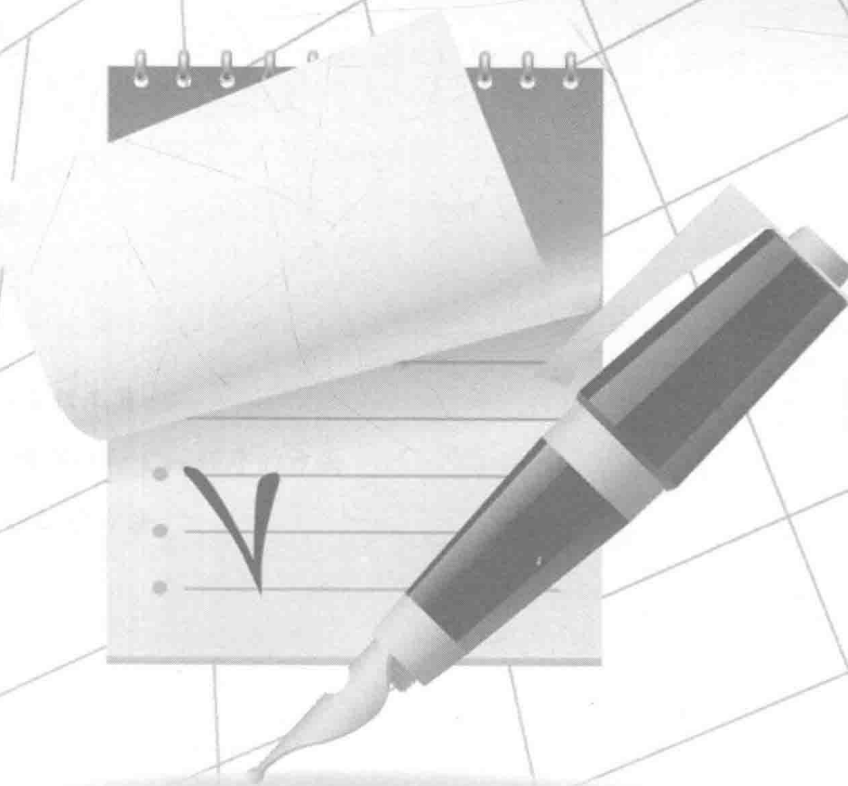
实 践 篇

实践 1 C++ 编程概述	357
实践指导	357
实践 1.1	357
实践 1.2	360
实践 1.3	362
知识拓展	365
拓展练习	368
实践 2 C++ 基础	369
实践指导	369
实践 2.1	369
实践 2.2	370
实践 2.3	371
实践 2.4	372
知识拓展	374
拓展练习	374

实践 3 C++ 面向对象	375
实践指导	375
实践 3.1	375
实践 3.2	381
知识拓展	385
拓展练习	386
实践 4 Win32 编程	387
实践指导	387
实践	387
知识拓展	395
拓展练习	397
实践 5 MFC 基础	398
实践指导	398
实践	398
知识拓展	404
拓展练习	409

实践 6 MFC 文档视图	410	实践 8.2	449
实践指导	410	知识拓展	461
实践 6.1	410	拓展练习	468
实践 6.2	414	实践 9 多线程	469
知识拓展	423	实践指导	469
拓展练习	424	实践 9.1	469
实践 7 MFC 对话框	425	实践 9.2	478
实践指导	425	知识拓展	486
实践	425	拓展练习	489
知识拓展	432	实践 10 数据库应用程序	490
拓展练习	436	实践指导	490
实践 8 网络编程	437	实践	490
实践指导	437	知识拓展	502
实践 8.1	437	拓展练习	506
附录 Visual C++ 关键字	507		

理论篇



第1章 C++编程概述



本章目标

- 了解 C++的发展历程
- 了解 C++标准的制定
- 了解 C++的特性
- 了解常见的几种 C++编程工具
- 掌握在 VS2010 下编译与运行 C++程序的方法
- 掌握基本的 C++输入/输出方法

1.1 C++ 简介

C++ 是一种使用非常广泛的编程语言，它支持过程化程序设计、数据抽象、面向对象程序设计、泛型程序设计等多种程序设计方法。

1.1.1 C++ 的发展历程

C++是由 C 语言演变而来的。20 世纪 80 年代初，贝尔实验室的 Bjarne Stroustrup 博士在 C 语言的基础上引入并扩充了面向对象的概念，他为了表达该语言与 C 语言的渊源关系，将这种语言命名为 C++。此后，C++ 语言经历了不断的完善：模板、异常处理、运行时类型识别(RTTI)、命名空间(Name Space)、虚函数、运算符重载、多重继承逐渐被加入进来。

C++ 语言支持 C 语言的所有语法和几乎所有的技术，有人把 C++ 语言看做是 C 语言的超集，同时 C++ 语言支持面向对象程序设计方法的所有概念。

C++ 语言发展大概可以分为三个阶段。

- ✧ 第一阶段(20 世纪 80 年代到 1995 年)：这一阶段 C++ 是一种早期的面向对象语言，凭借着接近 C 语言的效率，在工业界使用的开发语言中占据了相当大的份额。
- ✧ 第二阶段(1995 年到 2000 年)：这一阶段在 C++ 中增加了标准模板库(STL)、Boost 程序库，并加入了对泛型程序设计的支持。
- ✧ 第三阶段(2000 年至今)：这一阶段是 C++ 发展历史上的新高峰，出现了以 Loki、MPL 等程序库为代表的产生式编程和模板元编程，这些新技术的出现和原有技术的融合，使 C++ 成为当今主流程序设计语言中最复杂的一种。

1.1.2 C++ 标准

C++是具有国际标准的编程语言，其标准通常称做 ANSI/ISO C++。制定 ANSI/ISO C++标准的工作开始于 1990 年。

- ✧ 第一版：1998 年国际标准组织(ISO)颁布了 C++程序设计语言的国际标准 ISO/IEC 14882:1998。1998 年是 C++标准委员会成立的第一年，以后每 5 年视实际需要更新一次标准。
- ✧ 第二版：在 2003 年准时发布，即 ISO/IEC 14882:2003。
- ✧ C++0x 版：新标准的更新原定于 2009 年，目前一般称为 C++ 0x。由于各界对新特性存在很大的争议，C++ 0x 最终国际投票推迟至 2011 年 8 月 10 日进行，并获得通过。国际化标准组织于 2011 年 9 月 1 日发布 ISO/IEC 14882:2011，C++ 2011 正式成为新的 C++ 标准。



1.1.3 C++ 特性

C++语言具有如下几点特性:

- ✧ 高效: C++是与 C 语言同样高效且具有可移植性的多用途程序设计语言, 为保证语言的简洁和运行的高效, 很多特性都是以库(如 STL)或其他的形式提供的, 而没有直接添加到语言本身。
- ✧ 兼容性: 与 C 语言尽可能兼容, 借此提供一个从 C 到 C++ 的平滑过渡。
- ✧ 广泛: 支持多种程序设计方法, 结构化程序设计、面向对象程序设计、泛型程序设计等。
- ✧ 跨平台: 避免平台限定, C++ 中没有用于特定平台的限定。

1.2 C++ 编程环境

在 C++ 开发过程中, 有许多优秀的集成开发环境(IDE)可以选择, IDE 不但将程序编辑器、编译器、调试工具和其他应用程序的工具集成在一起, 还提供了大量联机帮助信息和资源, 使程序员可在不离开该环境的情况下做好开发工作。

1.2.1 主流开发工具

目前 C++ 的应用主要集中在 UNIX/Linux、Windows 和嵌入式设备三种系统环境下。不同的系统环境往往采用不同的 C++ 开发工具。

1. UNIX/Linux

在 UNIX/Linux 系统环境下进行 C++ 开发时, 主要使用以下开发工具的组合:

- ✧ 编译器: gcc。
- ✧ 编辑器: vi/vim/emacs。
- ✧ make 工具: GNU make。
- ✧ 版本管理: cvs。
- ✧ 代码阅读: cscope、ctags 等。

2. Windows

在 Windows 系统环境下, 例如 Windows XP、Windows 7, 主要有 Visual Studio 系列、C++ Builder 和 Eclipse 三种 C++ 开发工具。

3. 嵌入式设备

用于嵌入式设备的 C++ 开发工具与所采用的具体芯片有关, 其典型代表如下:

- ✧ 用于 ARM 芯片开发的 ADS、RealView MDK、ARM Realview Developer Suite、IAR Embed Workbench for ARM 等。
- ✧ 用于 51 系列单片机的 Keil C、IAR C51。
- ✧ 用于 AVR 系列单片机的 IAR for AVR、ICC AVR 等。



◇ 其他专用芯片的开发工具。



目前在 PC 上流行的主要是微软的 Visual Studio 系列和 GNU GCC, 其他的 IDE 都是衍生产品。本书所采用的 IDE 是 Visual Studio 2010。

1.2.2 Visual Studio

Visual Studio(简称 VS)是目前最流行的 Windows 应用程序开发环境, 它是集成化的开发环境, 该环境可以开发多种语言程序, 例如 Basic、C#、C++ 等。对于开发 C++ 程序的功能部分, 程序员通常称之为 Visual C++ (简称 VC)。Visual Studio 版本如表 1-1 所示。

表 1-1 Visual Studio 版本

VS 版本	年份	C++版本	说 明
Visual Studio 97	1997 年	Visual C++ 5.0	这是最早的 Visual Studio 版本, 包含有面向 Windows 开发使用的 Visual Basic 5.0、Visual C++ 5.0、Visual J++、Visual FoxPro
Visual Studio 6.0	1998 年	Visual C++ 6.0	所有开发语言的开发环境均升至 6.0 版本, 其中 Visual Basic 是最后一次发布, 从下一个版本(7.0)开始, 进化成一种新的面向对象的语言: Visual Basic.NET
Visual Studio 2002	2002 年	Visual C++ 7.0	微软引入建立在 .NET 框架上(版本 1.0)的托管代码机制以及一门新的语言 C#
Visual Studio 2003	2003 年	Visual C++ 7.1	对 Visual Studio 2002 进行部分修订, 内部版本号为 7.1
Visual Studio 2005	2005 年	Visual C++ 8.0	该版本的 Visual Studio 包含 .NET 框架的 2.0 版
Visual Studio 2008	2007 年	Visual C++ 9.0	简体中文专业版于 2008 年发布, 包含 .NET 框架 3.5
Visual Studio 2010	2010 年	Visual C++ 10.0	.NET 框架升级到 4.0 版
Visual Studio 2012	2012 年	Visual C++ 11.0	.NET 框架升级到 4.5 版, 支持 Windows 8 Metro 开发
Visual Studio 2013	2013 年	Visual C++ 12.0	.NET 框架升级到 4.5.1 版, 支持 Windows 8.1 App 开发, 增强了团队协作能力
Visual Studio 2015	2015 年	Visual C++ 14.0	.NET 框架升级到 4.6 版, 支持跨平台移动开发

Visual Studio 是一个集成的开发环境, 用于编写、调试、编译代码, 它提供了如下工具:

(1) 文本编辑器。Visual Studio 文本编辑器功能强大。例如, 输入语句时能够自动布局、实现代码缩进、匹配左右大括号、提供彩色编码关键字、执行一些语法检查、给可能产生编译错误的代码加上下划线等, 这称为“设计期间的调试”。文本编辑器提供 IntelliSense 功能。在开始输入时, IntelliSense 会自动显示类名、字段名或方法名; 输入方