



高等学校计算机专业教材

GAODENG XUOXIAO JISUANJI ZHUANYE JIAOCAI

面向对象的程序设计语言 —— C++

◎ 陈志泊 王春玲 编



GAODENG XUOXIAO JISUANJI ZHUANYE JIAOCAI
GAODENG XUOXIAO JISUANJI ZHUANYE JIAOCAI
GAODENG XUOXIAO JISUANJI ZHUANYE JIAOCAI
GAODENG XUOXIAO JISUANJI ZHUANYE JIAOCAI

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS



高等学校计算机专业教材

面向对象的程序设计 语言——C++

陈志泊 王春玲 编

人民邮电出版社

图书在版编目 (CIP) 数据

面向对象的程序设计语言——C++/陈志泊, 王春玲编. 北京: 人民邮电出版社, 2002.3

高等学校计算机专业教材

ISBN 7-115-09374-1

I.面... II.①陈...②王... III.C 语言—程序设计—高等学校—教材 IV.TP312

中国版本图书馆 CIP 数据核字 (2002) 第 005877 号

高等学校计算机专业教材
面向对象的程序设计语言——C++

- ◆ 编 著 陈志泊 王春玲
责任编辑 滑 玉
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线:010-67180876
北京汉魂图文设计有限公司制作
北京朝阳展望印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本:787×1092 1/16
印张:20.25
字数:476 千字 2002 年 3 月第 1 版
印数:1-5 000 册 2002 年 3 月北京第 1 次印刷

ISBN 7-115-09374-1/TP·2265

定价:27.00 元

本书如有印装质量问题,请与本社联系 电话:(010)67129223

内 容 提 要

本书内容主要分成三部分，第一部分是 C++ 语言基础部分，主要讲述 Visual C++ 6.0 集成开发环境中的常用功能菜单的使用、基本应用程序的建立方法和步骤、数据类型、运算符、表达式、三种基本程序结构、数组、函数及其重载、引用、指针、结构体、共用体、编译预处理等；第二部分是面向对象的程序设计部分，详细介绍面向对象的程序设计的有关基本概念、类与对象的定义和使用、构造函数与析构函数、继承与派生、虚函数与多态性、友元函数与友元类、静态成员、模板以及文件操作等；第三部分是面向对象的程序设计应用部分，该部分主要以 Visual C++6.0 作为开发环境，利用 MFC 类库、C++ 语言和面向对象的程序设计方法，介绍设计在 Windows 环境下运行应用程序的方法和思想，并对程序的框架结构进行了深入的剖析，其中主要介绍对话框、菜单、工具栏、状态栏、文档与视图以及图形/图像处理方面的程序的建立方法，此外，还包括对有关的 MFC 类以及常用成员函数的使用。这三部分内容相互衔接，前后呼应，循序渐进。在每一部分内容的讲解中，都给出了大量的实例进行解释说明。同时，除第 1 章外，每章后面都附有相应的习题，便于读者练习。

本书可作为高等院校计算机及相关专业的本、专科程序设计语言课程的教材，也比较适合广大计算机爱好者自学和参考。

出版者的话

为了适应我国大学本科计算机专业教育发展对教材的需要，我社特邀请教育部所属的中国人民大学、中国地质大学、中国农业大学、北京科技大学、北京林业大学、北京语言文化大学（排名不分先后）6所高等学校的“计算机科学与技术系”系主任及资深教授组成专家组，规划、编写、审订了本套教材。

读者对象 本套教材的主要读者对象是普通高等院校计算机科学与技术专业的学生，兼顾信息、自动化及机电类专业学生学习计算机课程的需要。由于这些专业对学生的培养目标是：掌握计算机软件与硬件的基本理论和方法，能从事计算机应用、软件研制、技术开发和管理工作的高级技术人才，因此，本套教材的内容既注意计算机高级技术人才所应具有完整知识结构，又适当侧重主要专业知识。

教材特点 本套教材参考了美国 IEEE/CS 和 ACM 技术委员会 2001 年推荐的课程 CC2001(Computing Curricula 2001, 参见 <http://www.csab.org/~csab>, <http://cs.nju.edu.cn/~gchen/teaching/cc2001/overview-bok.html> 2001) 及我国几十所高等院校计算机专业的 2001 年教学计划进行规划。教材内容在选择国际上先进的计算机理论、技术的同时，务求符合我国目前教学环境的实际情况，有一定深度，又有较高的实用性。

根据大学教学的特点，本套教材主要包括必修课程、选修课程和辅助课程三类教材。除了每本教材内容自成体系外，还考虑了它在整个教学计划中的安排顺序，适当增加了承上启下的内容。

写作风格 本套教材根据内容需要，沿着“这是什么、有什么用、怎样用、怎样用得更好”的思路编写教材，通过讲解具体知识，传授学习方法，使学生达到掌握理论和技术的目的；同时力求文笔流畅，言简意赅。

教材每一章除基本知识外，还有本章要点、小结、思考与练习题。有些教材还附加教学大纲（包括教学重点、难点，讲授知识点的参考学时数），操作性较强的课程还配有实验教材（包括上机应具备的软硬件环境和实验内容及方法）。为了方便教师教学，本套教材提供了演示稿，可到人民邮电出版社网站（<http://www.ptpress.com.cn>）的“教材出版图书出版中心”→“教材附件”→“课件”下载。

本套教材的作者都具有多年教学经验，教材的草稿也都在各自的授课过程中多次使用。这套教材的出版，为计算机专业的师生提供了新的选择。我们希望这套以易教、易学，朴实、实用为特色的教材在培养信息化建设专业人才方面做出应有的贡献。

欢迎广大读者对本套教材的不足之处提出批评和建议。

2002 年 1 月

编者的话

随着计算机技术的飞速发展及其应用领域的扩大，快速而扎实地掌握和精通一门程序设计语言成为计算机专业人员不可回避的课题。多年来的实践和经验告诉我们，C语言以其代码效率高、灵活性强、运算符丰富、结构化程序设计方法以及与操作系统的结合紧密等特点，受到了广大计算机专业人士及计算机爱好者的欢迎。

但是，随着 Windows 操作系统功能的逐步完善以及面向对象的程序设计方法的提出，程序设计与软件开发的思维、方法、复杂程度都发生了巨大的变化，使得利用 C 语言设计 Windows 的应用程序变得力不从心。而 C++ 语言是在 C 语言的基础上发展起来的，不仅继承了 C 语言的原有特点，而且引入了面向对象的程序设计方法，从而成为计算机及相关专业程序设计语言课程的首选语言。另外，Microsoft 公司推出的 Visual C++6.0 集成开发环境提供了建立控制台应用程序的方法，为我们学习和掌握 C++ 语言提供了非常好的开发环境，同时，它又很好地利用了 C++ 语言，并提供了大量的 MFC (Microsoft Foundation Class) 基础类库、应用程序向导和“可视化”的资源编辑器，这为程序员快速而高效地开发出 Windows 应用程序提供了方便。因此，一方面，Visual C++6.0 集成开发环境使得在学习 C++ 语言和开发 Windows 下的应用程序方面都能兼顾，另一方面，也使我们在开发 Windows 下的应用程序时不再感到繁琐和困难。

然而，目前市面上能全面介绍 C++ 语言及其应用、并能将它们汇编到一本书中的实用教材并不多见。另外，一本教材不能只介绍程序设计语言本身，还必须让读者从中学会程序设计的思维、方法及其在处理实际问题中的用途。本书正是基于这种思维而编写的，全书共分为三部分内容，第一部分是 C++ 语言基础部分，第二部分是面向对象的程序设计部分，第三部分是面向对象的程序设计应用部分。前两部分重点介绍了 C++ 语言的概念、基本语法及面向对象的程序设计的方法，第三部分重点介绍了利用 C++ 语言、面向对象的程序设计方法、Visual C++ 6.0 集成开发环境及 MFC 类设计 Windows 应用程序的方法，这部分内容是对所学 C++ 语言的运用和综合。另外，全书内容相互衔接，成为一个逻辑整体。除第 1 章外，每章都附有一定数量的习题，便于读者学习。为方便读者学习和教师讲课，本书还提供了用 Power Point 制作的讲稿。需要者请到人民邮电出版社的网站中下载（选择“教材出版中心”→“教材附件”→“课件”即可）。

本书内容全面，深入浅出，不仅适合于教学，同时也适合于用 Visual C++ 开发应用程序的用户参考。如果作为教材，建议第一部分和第二部分应全讲，而对于第三部分，各院校可根据学时情况适当选讲，如果学时紧张，第 8 章可选讲。建议主讲学时为 50 学时，学时分配为：第一部分 15 学时，第二部分 20 学时，第三部分 15 学时。由于课程学时的限制，实验学时各学校可适当调整，一般为 20 学时左右；另外，除实验学时外，最好还要安排学生自由上机的时间，以加强学生的实际动手能力。

本书由陈志泊统稿和主编，全书由陈志泊、王春玲编写。另外，张惠萍、黄金燕等参加了文字编辑工作，马振龙参加了书稿校对方面的工作，在此一并表示衷心的感谢！

由于时间仓促，加之作者水平有限，书中不当之处在所难免，敬请读者批评指正。

编 者

2002年1月

目 录

第一部分 C++语言部分

第 1 章 Visual C++ 6.0 开发环境	1
1.1 开发环境概述	1
1.1.1 程序开发流程	1
1.1.2 Visual C++ 6.0 的启动和集成开发环境 (IDE)	2
1.2 开发环境的菜单功能	3
1.2.1 “File” 菜单	4
1.2.2 “Edit” 菜单	7
1.2.3 “View” 菜单	9
1.2.4 “Insert” 菜单	10
1.2.5 “Project” 菜单	10
1.2.6 “Build” 菜单	11
1.2.7 “Tools” 菜单	12
1.2.8 “Window” 菜单	13
1.2.9 “Help” 菜单	13
1.3 开发环境的工具栏及使用	14
1.3.1 “Standard” 工具栏	14
1.3.2 “Build MiniBar” 工具栏	15
1.4 项目和项目工作区	15
1.4.1 “ClassView” 选项卡	16
1.4.2 “Resource View” 选项卡	17
1.4.3 “FileView” 选项卡	17
1.5 资源	18
1.5.1 什么是资源	18
1.5.2 资源编辑器	18
1.6 开发环境的初步实践	19
1.7 小结	24
第 2 章 C++语言基础	25
2.1 从面向过程的程序设计到面向对象的程序设计	25
2.1.1 传统的结构化程序设计方法	25
2.1.2 面向对象的程序设计方法	26
2.1.3 面向对象的程序设计方法与结构化程序设计方法的比较	29
2.2 C++程序的基本组成	29
2.2.1 从 C 语言到 C++语言	29

2.2.2	C++程序的结构与基本组成	30
2.3	C++的数据类型、运算符和表达式	33
2.3.1	数据类型	33
2.3.2	运算符和表达式	39
2.4	数据的输入与输出	46
2.4.1	数据的输出 cout	46
2.4.2	数据的输入 cin	51
2.5	C++的控制语句	52
2.5.1	C++语句概述	52
2.5.2	C++程序的3种基本结构	53
2.5.3	if 语句	53
2.5.4	switch 语句	57
2.5.5	循环语句	58
2.6	数组及其使用	62
2.6.1	一维数组	62
2.6.2	二维数组	64
2.6.3	字符数组	65
2.7	函数	68
2.7.1	函数的定义	68
2.7.2	函数的调用	70
2.7.3	函数的嵌套调用	71
2.7.4	函数的递归调用	71
2.7.5	局部变量和全局变量	72
2.7.6	全局函数和静态函数	74
2.7.7	内联函数	75
2.7.8	函数重载	76
2.7.9	默认参数的函数	77
2.8	指针类型及使用	78
2.8.1	指针的概念	79
2.8.2	const 指针	80
2.8.3	指针与函数	82
2.8.4	指针与数组	85
2.8.5	指针与字符串	87
2.8.6	指针数组和指向指针的指针	88
2.9	引用	89
2.9.1	引用的概念、声明和使用	89
2.9.2	用引用作为函数的参数	91
2.9.3	如何使一个被调函数同时返回多个值	92
2.9.4	用引用返回函数值	93

2.9.5	返回引用的函数值作为赋值表达式的左值	94
2.9.6	用 <code>const</code> 限定引用	95
2.9.7	引用总结	96
2.10	结构体、共用体和枚举	96
2.10.1	结构体	96
2.10.2	堆内存的分配和释放	104
2.10.3	共用体类型	105
2.10.4	枚举类型	107
2.10.5	类型定义 <code>typedef</code> 的使用	108
2.10.6	编译预处理	108
2.11	小结	110
习题	112

第二部分 面向对象的程序设计部分

第3章	面向对象的程序设计	117
3.1	类与对象的定义	117
3.1.1	类的定义	117
3.1.2	成员函数的定义	119
3.1.3	对象及指向对象的指针	121
3.1.4	访问对象的成员	121
3.1.5	对象赋值语句	123
3.1.6	对象的作用域与生存期	124
3.2	构造函数与析构函数	125
3.2.1	构造函数	125
3.2.2	构造函数的重载	127
3.2.3	默认构造函数与缺省参数的构造函数	128
3.2.4	析构函数	129
3.2.5	拷贝构造函数	130
3.2.6	一个类的对象作为另一个类的数据成员	131
3.2.7	利用初始化表对常量数据成员或引用成员提供初值	135
3.2.8	类作用域	136
3.3	继承和派生	137
3.3.1	继承的概念	137
3.3.2	单继承	138
3.3.3	多重继承	143
3.3.4	虚基类	144
3.4	虚函数与多态性	146
3.4.1	多态性	146
3.4.2	子类型	146

3.4.3	用基类指针指向公有派生类对象	148
3.4.4	虚函数	150
3.4.5	静态联编与动态联编	151
3.4.6	纯虚函数与抽象类	152
3.5	静态成员	154
3.5.1	静态数据成员	154
3.5.2	静态成员函数	156
3.6	友元函数与友元类	157
3.6.1	友元函数	157
3.6.2	友元类	158
3.7	堆对象和对象数组	160
3.7.1	堆对象	160
3.7.2	对象数组	161
3.8	运算符的重载	162
3.9	模板与使用	165
3.9.1	类模板	165
3.9.2	函数模板	167
3.10	文件及其操作	168
3.10.1	顺序文件操作	168
3.10.2	随机文件操作	171
3.11	小结	173
习题	174

第三部分 面向对象的程序设计应用部分

第4章	利用 MFC 开发 Windows 应用程序	181
4.1	Windows 应用程序的特点与消息驱动机制	182
4.1.1	Windows 应用程序的特点与消息驱动机制	182
4.1.2	Windows 编程中常用的数据类型和句柄	184
4.2	利用 MFC AppWizard 创建 Windows 应用程序	186
4.3	MFC 应用程序的类和文件	187
4.3.1	类说明	187
4.3.2	文件说明	189
4.4	Windows 消息	189
4.4.1	标准的 Windows 消息	190
4.4.2	控件消息	191
4.4.3	命令消息	192
4.5	消息的发送和接收的基本过程和机制	193
4.6	消息映射与消息处理函数	194
4.6.1	CCmdTarget 类	194

4.6.2	消息映射与消息处理函数的概念	194
4.6.3	管理窗口消息处理函数	194
4.7	MFC 应用程序的执行过程分析	197
4.8	小结	200
	习题	201
第 5 章	对话框及常用控件	202
5.1	对话框的组成和分类	202
5.1.1	对话框的组成	202
5.1.2	对话框的分类	203
5.2	创建模式对话框	203
5.2.1	创建应用程序框架	203
5.2.2	创建对话框资源	204
5.2.3	创建对话框类	209
5.2.4	对话框的实现	212
5.3	Windows 的常用控件	215
5.3.1	常用控件及对应的 MFC 类	215
5.3.2	控件的通用属性及设置	215
5.3.3	控件的消息	216
5.3.4	几种常用控件简介	217
5.4	非模式对话框的实现	226
5.5	消息对话框和通用对话框	231
5.5.1	消息对话框	231
5.5.2	通用对话框	232
5.6	小结	233
	习题	233
第 6 章	菜单、工具栏和状态栏	235
6.1	创建菜单	235
6.1.1	创建菜单资源	235
6.1.2	建立快捷键	238
6.1.3	菜单功能的实现	239
6.1.4	CMenu 类及其常用成员函数介绍	242
6.1.5	快捷菜单的实现	245
6.1.6	动态地向菜单中添加新的菜单项	248
6.2	创建工具栏	250
6.2.1	创建工具栏的方法和步骤	250
6.2.2	如何实现工具栏的船坞化	255
6.3	创建状态栏	256
6.3.1	创建状态栏资源	256
6.3.2	实现状态栏	257

6.3.3 进一步完善状态栏	260
6.4 小结	262
习题	263
第7章 文档和视图	264
7.1 文档类和视图类间的关系	264
7.2 视图类和文档类中常用的成员函数	265
7.2.1 CView 类的 GetDocument()函数	265
7.2.2 CDocument 类的 UpdateAllViews()函数	266
7.2.3 CView 类的 OnUpdate()函数	266
7.2.4 CView 类的 OnInitialUpdate()函数	266
7.2.5 MFC 应用程序中各个类对象间的相互调用关系	266
7.3 在视图类中实现显示文档类中的数据	267
7.4 在视图类中响应键盘消息	271
7.5 SDI 文档的序列化	272
7.5.1 创建空文档	272
7.5.2 打开文档	273
7.5.3 保存文档	274
7.5.4 关闭文档	275
7.6 小结	275
习题	276
第8章 图形图像处理	277
8.1 设备环境 (DC)	277
8.1.1 设备环境类	277
8.1.2 几个设备环境类的主要差别	278
8.1.3 设备环境类的使用方法	279
8.2 绘图设备类	280
8.2.1 CGdiObject 类	280
8.2.2 CPen 类	281
8.2.3 CBrush 类	283
8.2.4 CFont 类及创建 CFont 类对象的方法	285
8.2.5 CBitmap 类	287
8.3 坐标映射方式	288
8.3.1 Windows 映像方式及其设置方法	288
8.3.2 逻辑坐标和设备坐标的转换	289
8.4 窗口和视口	290
8.4.1 窗口和视口的概念	290
8.4.2 窗口和视口有关的操作函数	290
8.4.3 窗口逻辑坐标和视口设备坐标的转换	291
8.5 设置绘图模式	294

8.6 图形元素的绘制	295
8.6.1 画点	295
8.6.2 画直线	296
8.6.3 画矩形	297
8.6.4 画圆角矩形	297
8.6.5 画扇形	298
8.6.6 画多边形	298
8.6.7 绘制文本	299
8.7 在视图中交互绘图	300
8.8 小结	303
习题	304
参考文献	305

第一部分 C++语言部分

第 1 章 Visual C++ 6.0 开发环境

本章内容是进入面向对象程序设计、使用 Visual C++ 6.0 编程的基础，在进行以后各章学习之前，应该很好地掌握本章的内容。

- (1) 熟悉 Visual C++ 6.0 的开发环境。
- (2) 熟练掌握开发环境的菜单和工具栏的功能及使用。
- (3) 掌握项目的概念以及项目工作区的组成和作用。
- (4) 掌握资源的概念及其基本的编辑。
- (5) 熟练掌握利用 AppWizard (应用程序向导) 生成简单 Windows 应用程序的步骤。

1.1 开发环境概述

Visual C++ 6.0 是 Microsoft 公司在 1998 年推出的基于 Windows 9X 和 Windows NT 的优秀集成开发环境。该开发环境为用户提供了良好的可视化编程环境，程序员可以利用该开发环境轻松地访问 C++ 源代码编辑器、资源编辑器和使用内部调试器，并且可以创建项目文件。Visual C++ 6.0 不仅包括编译器，而且它还包括了许多有用组件，通过这些组件的协同工作，可以在 Visual C++ 6.0 集成环境中轻松地完成创建源文件、编辑资源，以及对程序的编译、连接和调试等各项工作。

1.1.1 程序开发流程

在 Visual C++ 6.0 中，任何程序的开发过程均遵循图 1-1 所示的开发流程，任何一步有错误或结果不满意均需回到前一步进行修改，再顺序进行，直到所求结果满意为止。

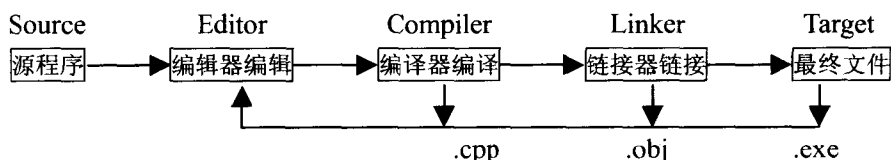


图 1-1 程序的开发过程

1.1.2 Visual C++ 6.0 的启动和集成开发环境 (IDE)

单击任务栏中的“开始”按钮，在弹出菜单中选择“程序”菜单项，单击“Microsoft Visual C++ 6.0”菜单项就可以进入 Visual C++ 6.0 的集成开发环境的主窗口，如图 1-2 所示。

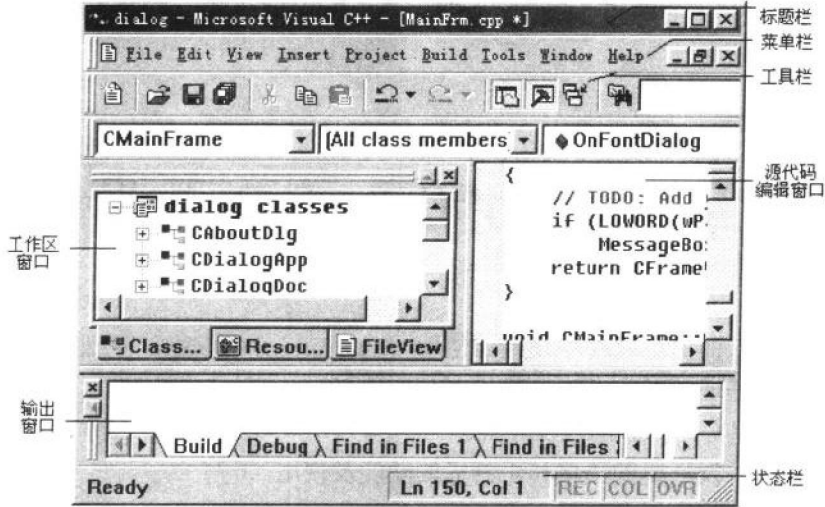


图 1-2 Visual C++ 6.0 的集成开发环境主窗口界面

由图 1-2 可以看出，该集成开发环境由标题栏、菜单栏、工具栏、工作区窗口、源代码编辑窗口、输出窗口和状态栏组成。

另外，在程序调试过程中，Visual C++ 6.0 可以为不同的调试信息创建不同的窗口，比如，观察窗口(Watch Windows)、变量窗口(Variables Windows)、寄存器窗口(Registers Windows)、存储器窗口(Memory Windows)和调用堆栈窗口(Call Stack Windows)等。

1. 标题栏

标题栏位于主窗口的最顶端，用于显示应用程序名和所打开的文件名。标题栏的颜色用于表明该窗口是否被激活。标题栏的右端有 3 个控制按钮，从左到右分别是最小化按钮、最大化/还原按钮和关闭按钮，这些按钮用于快速设置窗口大小。

2. 菜单栏

菜单栏位于标题栏下方。从左到右分别是“File”、“Edit”、“View”、“Insert”、“Project”、“Build”、“Tools”、“Window”和“Help”9 个菜单标题。其中，每个菜单标题包含一系列菜单项，每个菜单项代表一类用户操作，用户可以通过单击菜单项来执行其代表的特定功能，轻松地完成编辑程序和资源、编译、连接及调试程序等各项任务。要激活菜单栏中的菜单有两种方法：一种方法是用鼠标左键直接单击相应的菜单，另一种方法是在按住 Alt 键的同时按下相应菜单的热键，例如，要激活“File”菜单只要按下 Alt+F 键即可。

3. 工具栏

Visual C++ 6.0 也提供了功能强大的工具栏，提供对一些菜单命令的快捷访问方法。用户可以很方便地从工具栏中选择要执行的操作，而不需要每次都打开菜单再进行选择了。同时，工具栏有提示的功能，当把鼠标指向某个按钮时，状态栏都会显示出按钮的功能。Visual C++ 6.0 包含了十多种工具栏。在缺省情况下，屏幕上只显示 Standard、Build MiniBar 和 WizardBar 工具栏。其他的工具栏如 Debug、Database 等在一般情况下并不显示，在适当的时候将自动弹出，或者由用户自己来设置是否显示。

4. 工作区窗口

Visual C++ 6.0 以项目工作区 (Project Workspace) 的形式组织文件、项目和项目配置。工作区窗口中显示了所创建的项目的所有信息。通过工作区窗口可以从类、资源和文件 3 个角度来对项目进行操作。

加载或新建一个项目后，工作区将显示 3 个选项卡：“ClassView”、“ResourceView”和“FileView”。不同的选项卡分别对应一个视窗。每个视窗都显示项目的不同信息。工作区窗口如图 1-2 所示。

5. 源代码编辑窗口

源代码编辑窗口用于编辑 C++ 的头文件、源文件和文本文件等，如图 1-2 所示，是用户进行程序代码实际编写工作的场所。它实质上是一个具有特殊功能的编辑器。

源代码编辑窗口会显示出当前打开的文本文件类型，例如，是 C++ 源文件还是 C++ 头文件。用户在该窗口中单击鼠标右键，将会弹出一个快捷菜单，用于提供一些常用的文本编辑命令，如 Copy、Cut 和 Paste 等。该弹出式菜单将根据当前编辑文件类型的不同而产生相应变化。

6. 输出窗口

输出窗口位于工作区窗口和源代码编辑窗口的下方，用于显示多种类型的信息，如显示程序建立过程（编译、连接）的有关信息，或显示调试运行时的输出结果。

7. 状态栏

状态栏位于主窗口的最底端，用于显示当前操作或所选择命令的提示信息。

1.2 开发环境的菜单功能

在进行程序设计之前，先了解菜单命令的基本功能是很有必要的，因为程序员与开发界面打交道大部分是通过菜单栏中的命令来完成的。

此外，在窗口的不同地方单击鼠标右键可以弹出相应的快捷菜单，通过快捷菜单可以执行与所处环境相关的命令。