

➤ 高等学校应用型本科“十三五”规划教材



C++ 程序设计实验指导 与习题解析



主编 杨艳 李志敏 周淑琴



西安电子科技大学出版社
<http://www.xduph.com>

高等学校计算机类“十二五”规划教材

C++程序设计

实验指导与习题解析

主 编 杨 艳 李志敏 周淑琴
参 编 荆耀东 史 广

西安电子科技大学出版社

内 容 简 介

本书共三部分,第一部分介绍了实验软件环境 Visual C++;第二部分与本书配套教程《C++程序设计》(毕如田主编,西安电子科技大学出版社)同步设置了12个实验;第三部分与本书配套教程同步设置了对应的章节习题,并附答案。

本书可供计算机、电子信息类相关专业学生学习“C++程序设计”相关课程使用。

图书在版编目(CIP)数据

C++程序设计实验指导与习题解析/杨艳,李志敏,周淑琴主编. —西安:

西安电子科技大学出版社, 2016. 1

高等学校计算机类“十二五”规划教材

ISBN 978 - 7 - 5606 - 3540 - 8

I. ① C… II. ① 杨… ② 李… ③ 周… III. ① C语言—程序设计
—高等学校—教材 IV. ① TP312

中国版本图书馆 CIP 数据核字(2014)第 271654 号

策划编辑 胡华霖

责任编辑 阎 彬 师 彬

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 陕西华沐印刷科技有限责任公司

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 7.5

字 数 173千字

印 数 1~3000册

定 价 13.00元

ISBN 978 - 7 - 5606 - 3540 - 8/TP

XDUP 3832001 - 1

* * * 如有印装问题可调换 * * *

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

本书是《C++程序设计》配套教材,可供计算机、电子信息类相关专业学生学习“C++程序设计”相关课程使用。

本书内容分为 Visual C++ 实验环境介绍、实验指导和课后习题与答案三个部分。其中,第一部分介绍了 Visual C++ 的相关概念、项目开发过程、集成开发环境 Developer Studio 以及常用功能键及其含义。

第二部分与配套教材(《C++程序设计》(毕如田主编))同步设置了 12 个实验,每个实验的具体实验内容和学时可作适当调整,一般每个实验为 2、3 个学时。其中,前 9 个实验为必做实验,实验一指导学生掌握集成开发环境的使用,实验二指导学生掌握程序流程的控制,实验三指导学生掌握 C++ 的数据类型,实验四指导学生掌握函数的使用,实验五指导学生掌握面向对象程序设计中的类与对象,实验六指导学生掌握面向对象程序设计中的继承与派生类,实验七指导学生掌握面向对象程序设计中的多态性,实验八指导学生掌握 I/O 流类库的使用,实验九指导学生掌握模板的使用,实验十、十一和十二主要对链表、队列、栈等数据结构以及面向对象程序设计的思想进一步深入讨论,可根据不同专业及教学进度的安排选做。每一个实验都明确了实验的目的与步骤,学生需完全按照实验的步骤开展实验,并在完成思考题后提交实验报告。

第三部分与配套教材同步设置了习题,并附有答案,学生需认真完成每章习题,并参考题后所附答案进行自测。

本书由具有多年 C++ 程序设计教学经验的教师编写完成。其中,第一部分由杨艳编写,第二部分由荆耀东(实验 1~6)和史广(实验 7~12)共同编写,第三部分由周淑琴编写,全书的错误修正由李志敏完成。感谢读者和教师选用本书,欢迎对书中的不妥之处提出批评和修改建议。

编者的电子邮件地址:core_cpp@126.com。

编 者

2014 年 6 月

目 录

第一部分 Visual C++实验环境介绍	(1)
一、Visual C++简介	(1)
二、项目开发过程	(2)
三、集成开发环境 Developer Studio	(3)
四、常用功能键及其含义	(6)
第二部分 实验指导	(7)
实验一 熟悉实验环境	(8)
实验二 程序流程的控制	(16)
实验三 数据类型	(20)
实验四 函数	(24)
实验五 类和对象	(29)
实验六 继承和派生类	(33)
实验七 多态性	(37)
实验八 模板	(39)
实验九 I/O 流类库	(41)
实验十 链表与队列实验	(43)
实验十一 队列类与栈类实验	(50)
实验十二 面向对象程序设计	(55)
第三部分 《C++程序设计》课后习题与答案	(65)
第一章 C++语言概论	(66)
第二章 程序流程控制	(68)
第三章 数据类型	(73)
第四章 函数	(80)
第五章 类和对象	(83)
第六章 继承和派生类	(91)
第七章 多态性	(96)
第八章 模板	(99)
第九章 I/O 流类库	(102)
第十章 异常处理	(105)
附录 A ASCII 码表	(108)
附录 B C++程序常见错误提示中英文对照表	(110)
参考文献	(114)

第一部分 Visual C++ 实验环境介绍

一、Visual C++ 简介

Visual C++ 是 Microsoft 公司 Visual Studio 开发工具中的一个 C++ 程序开发包。Visual Studio 提供了一整套开发 Internet 和 Windows 应用程序的工具, 包括 Visual C++、Visual Basic、Visual Foxpro、Visual InterDev、Visual J++ 以及其他辅助工具, 如代码管理工具 Visual SourceSafe 和联机帮助系统 MSDN。Visual C++ 包中除包括 C++ 编译器外, 还包括所有的库、示例和为创建 Windows 应用程序所需要的文档。

从最早期的 1.0 版本发展到最新的 VS 2013 版本, Visual C++ 已经有了很大的变化, 在界面、功能和库支持方面都有较大的增强。Visual C++ 一般分为三个版本: 学习版、专业版和企业版, 不同的版本适合于不同类型的应用开发。实验中可以使用这三个版本的任意一种, 本书以 Visual C++ 6.0 学习版为实验环境。

1. Visual C++ 集成开发环境 (IDE)

IDE 是一个将程序编辑器、编译器、调试工具和其他建立应用程序的工具集成在一起的用于开发应用程序的软件系统。Visual C++ 包中的 Developer Studio 就是一个集成开发环境, 它集成了各种开发工具和 VC 编译器, 程序员可以在不离开该环境的情况下编辑、编译、调试和运行一个应用程序。IDE 中还提供大量在线帮助信息, 协助程序员做好开发工作。Developer Studio 中除了程序编辑器、资源编辑器、编译器和调试器外, 还有各种工具和向导 (如 AppWizard 和 ClassWizard) 以及 MFC 类库, 这些都可以帮助程序员快速而正确地开发出应用程序。

2. 向导 (Wizard)

向导是一个通过一步步的帮助, 引导程序员工作的工具。Developer Studio 中包含三个向导, 用来帮助程序员开发简单的 Windows 程序。

AppWizard: 用来创建一个 Windows 程序的基本框架结构。AppWizard 向导会一步步向程序员提出问题, 询问他所创建项目的特征, 然后会根据这些特征自动生成一个可以执行的程序框架, 程序员可以在这个框架下进一步填充内容。AppWizard 支持三类程序: 基于视图/文档结构的单文档应用程序、基于视图/文档结构的多文档应用程序和基于对话框的应用程序。利用 AppWizard 也可以生成最简单的控制台应用程序 (类似于 DOS 下用字符输入/输出的程序)。

ClassWizard: 用来定义 AppWizard 所创建的程序中的类。可以利用 ClassWizard 在项目中增加类, 为类增加处理消息的函数等。ClassWizard 也可以管理包含在对话框中的控件, 可以将 MFC 对象或者类的成员变量与对话框中的控件联系起来。

ActiveX Control Wizard: 用于创建一个 ActiveX 控件的基本框架结构。ActiveX 控件是用户自定义的控件, 它支持一系列定义的接口, 可以作为一个可再利用的组件。

3. MFC 库

库(library)是可以重复使用的源代码和目标代码的集合。MFC(Microsoft Fundamental Classes)是 Visual C++ 开发环境所带的类库,在该类库中提供了大量的类,可以帮助开发人员快速建立应用程序。这些类可以提供程序框架、进行文件和数据库操作、建立网络连接、进行绘图和打印等各种通用的应用程序操作。使用 MFC 库开发应用程序可以减少很多工作量。

二、项目开发过程

在一个集成的开发环境中开发项目非常容易。一个用 C++ 开发的项目通用开发过程如图 1-1 所示。

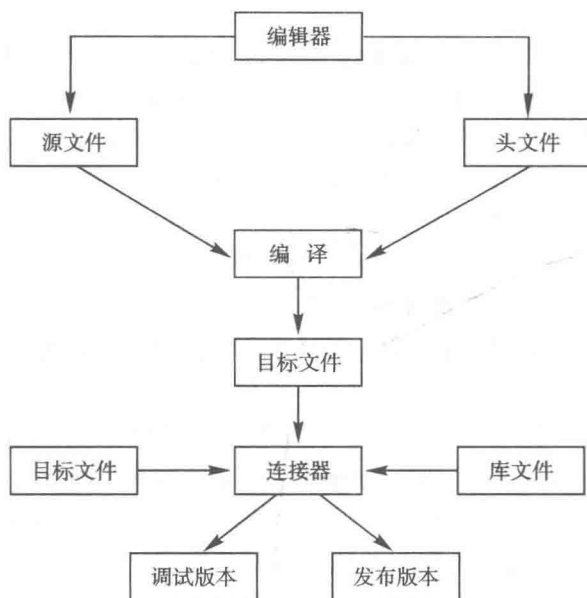


图 1-1 项目通用开发过程

建立一个项目的第一步是利用编辑器建立程序代码文件,包括头文件、代码文件和资源文件等。然后,启动编译程序。编译程序首先调用预处理程序处理程序中的预处理命令(如 `#include`, `#define` 等),经过预处理程序处理的代码将作为编译程序的输入。编译程序对用户程序进行词法和语法分析,建立目标文件,文件中包括机器代码、连接指令、外部引用以及从该源文件中产生的函数和数据名。此后,连接程序。将所有的目标代码和用到的静态连接库的代码连接起来,为所有的外部变量和函数找到其提供地点,最后产生一个可执行文件。一般由一个 Makefile 文件来协调各个部分产生可执行文件。

可执行文件分为两种版本: Debug 和 Release。Debug 版本用于程序的开发过程,该版本产生的可执行程序带有大量的调试信息,可以供调试程序使用;而 Release 版本作为最终的发行版本,没有调试信息,并且进行了优化。在上机实习过程中可以采用 Debug 版本,这样便于调试。

选择 Debug 版本还是 Release 版本的方法是:在 Developer Studio 中选择菜单“Build|Set Active Configuration”,在弹出的对话框中选择所要的类型,然后选择“OK”,关闭对话框。

Visual C++ 集成开发环境中集成了编辑器程序、编译器程序、连接器程序以及调试程序，覆盖了开发应用程序的整个过程，程序员不需要脱离这个开发环境就可以开发出完整的应用程序。

三、集成开发环境 Developer Studio

1. 进入 Developer Studio

如果用户使用的是 Visual C++ 6.0，需要单击任务栏中的“开始”后选择“程序”，找到 Microsoft Visual Studio 6.0 文件夹后，单击其中的“Microsoft Visual C++ 6.0”图标，就可以启动 Developer Studio。

2. Developer Studio 的界面

Developer Studio 用户界面是一个由窗口、工具条、菜单、工具及其他部分组成的集成界面，如图 1-2 所示。通过这个界面，用户可以在同一环境下创建、测试、调试应用程序。

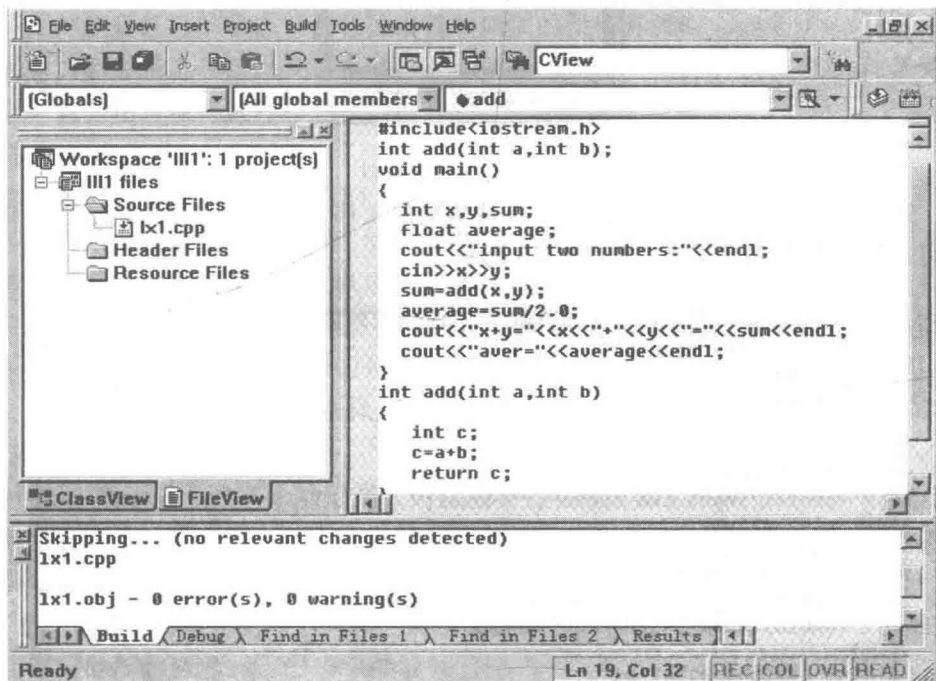


图 1-2 集成开发环境

界面主要部分介绍如下：

- 工具条和菜单，用于提供用户操作的命令接口。菜单以文字和层次化的方式提供命令接口，工具条由一系列按钮组成，这些按钮用一系列小图标来标志，可以提供快速的命令选择。菜单和工具条在开发的不同进程有不同显示内容。当第一次打开 Developer Studio 时，显示的是标准的工具条和菜单，随着开发的不同步骤，不同的工具条就会自动显示出来，菜单也会有所变化。工具条有很多种，用户可以选择显示不同的工具条。工具条可以任意移动，也可以放大缩小。工具条和菜单条功能基本相同，唯一的区别是：菜单条总占据一行，并且一般不能隐藏，如图 1-3 所示。

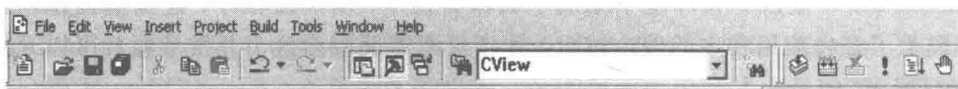


图 1-3 工具条和菜单条

● 工作区窗口，这个窗口包含关于正开发项目的有关信息。在没有开发任何项目时，该窗口显示系统的帮助目录。当打开一个项目以后，工作区窗口将会显示关于当前项目的文件信息和类的信息，如图 1-4 所示。

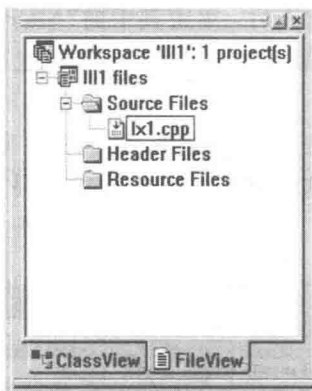


图 1-4 工作区窗口

● 文档窗口区，这个区域可以显示各种类型的文档，如源代码文件、头文件和资源文件等，可以同时打开多个文档，如图 1-5 所示。

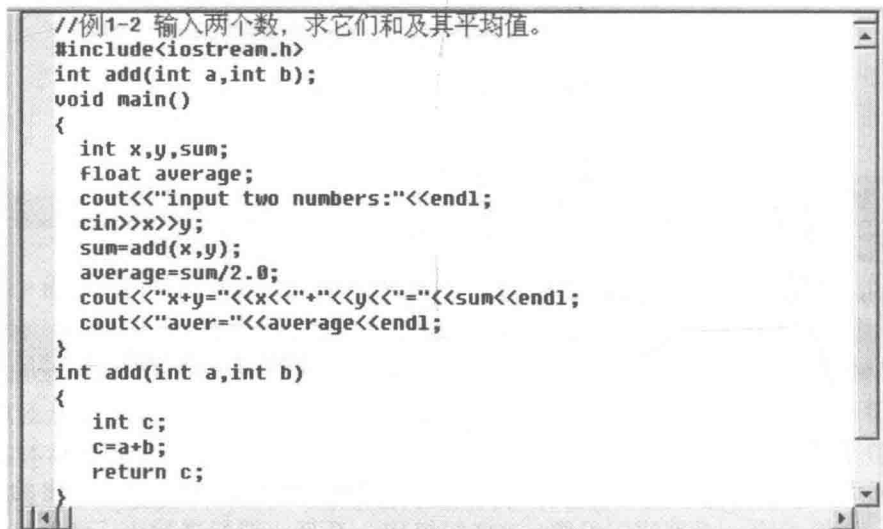


图 1-5 文档窗口区

● 输出窗口，用来显示几种信息，即可以通过选择不同的标签来显示不同的信息。这些信息包括编译连接结果信息(Build 标签)、调试信息(Debug 标签)和查找结果信息(Find in Files 标签)。其中查找结果信息有两个标签，可以显示两次在文件中查找指定内容的结果，如图 1-6 所示。

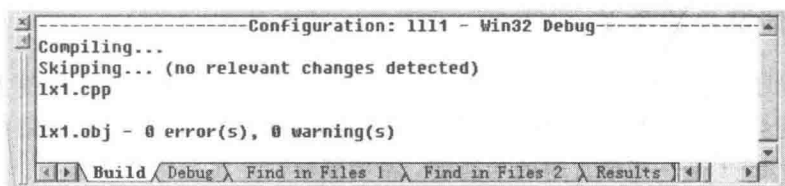


图 1-6 输出窗口

总的来说,窗口和命令接口(包括工具条和菜单)是构成界面的最主要组成部分。界面中的窗口有两种类型:文档窗口和可附着(docking)窗口。文档窗口显示在文档窗口区,用于显示和编辑文档,其大小和位置可以随其所处的 Developer Studio 窗口的改变而改变,即可以最大化和最小化。可附着窗口可以附着于应用程序窗口的边界,也可以浮在屏幕上的任何位置。可附着窗口有:工作区(workspace)窗口、输出(output)窗口和调试窗口(包括 variable、watch、local 等窗口)等。

文档窗口的位置、大小及是否可见和它所在的项目有关;可附着窗口的位置、大小及是否可见则与项目进行的状态以及各种编辑和调试的操作有关。

各种窗口和各种工具条以及菜单构成了界面的布局。一旦用户决定了一种界面布局,系统就会一直为用户保持这种布局,直到用户下一次改变该布局为止。

3. 获得帮助信息

大多数时候,用户可以通过按 F1 键得到上下文帮助。如在编辑文件时按 F1 键可以得到有关编辑的帮助,在编译连接错误信息上按 F1 键,则可以得到关于该错误的帮助信息。要想查找关于某个话题的帮助,可以选择菜单“Help|Search”,在查询对话框中进行查找。Visual C++6.0 中,可以通过选择菜单“Help|Contents”来启动 MSDN 查阅器。MSDN 查阅器是一个功能强大的程序,可以方便地浏览、查找信息,要想知道如何具体使用 MSDN 查阅器,可以在 MSDN 查阅器中选择菜单“Help”下的命令。

4. Visual C++ 的编辑器

Developer Studio 包含一个功能强大的编辑器,可以编辑 Visual C++ 源文件。这个编辑器很像字处理器,但是没有字处理器复杂的排版、文本格式等功能,它注重的是如何帮助程序员快速高效地编制程序。它具有以下特点:

- 自动语法。用高亮度和不同颜色的字来显示不同的语法成分,如注释、关键字和一般代码用不同的颜色显示。
- 自动缩进。帮助程序员排列源代码,使其可读性更强。
- 参数帮助。在编辑用到预定义的 Windows 函数时,可以自动显示函数参数。
- 集成的关键字帮助。能够使程序员快速得到任何关键字、MFC 类或 Windows 函数的帮助信息(按 F1 键即可)。
- 拖放编辑。能够用鼠标选择文本并自由拖动到任意位置。
- 自动错误定位。能自动将光标移动到有编译错误的源代码处。

在编辑器编辑程序时,源代码文件在文档显示区显示,每个文件有独立的显示窗口。如果选择用其他编辑器编辑源文件,必须将该文件以纯文本的方式保存。Visual C++ 的编译器不能处理其中有特别格式字符的文件。

四、常用功能键及其含义

为了使程序员能够方便快捷地完成程序开发，开发环境提供了大量快捷方式来简化一些常用操作的步骤。键盘操作直接、简单而且方便，因而程序员非常喜欢采用键盘命令来控制操作。表 1-1 是一些最常用的功能键，希望同学们在实验中逐步掌握。

表 1-1 常用功能键及其含义

操作类型	功能键	对应菜单	含义
文件操作	Ctrl+N	File New	创建新的文件、项目等
	Ctrl+O	File Open	打开项目、文件等
	Ctrl+S	File Save	保存当前文件
编辑操作	Ctrl+X	Edit Cut	剪切
	Ctrl+C	Edit Copy	复制
	Ctrl+V	Edit Paste	粘贴
	Ctrl+Z	Edit Undo	撤消上一个操作
	Ctrl+Y	Edit Redo	重复上一个操作
	Ctrl+A	Edit Select All	全选
	Del	Edit Del	删除光标后面的一个字符
建立程序操作	Ctrl+F7	Build Compiler current file	编译当前源文件
	Ctrl+F5	Build Run exe	运行当前项目
	F7	Build Build exe	建立可执行程序
	F5	Build Start Debugging	启动调试程序
	F5	Debug Go	继续运行
调试	F11	Debug Step into	进入函数体内部
	Shift+F11	Debug Step out	从函数体内部运行出来
	F10	Debug Step over	执行一行语句
	F9		设置/清除断点
	Ctrl+F10	Debug Run to cursor	运行到光标所在位置
	Shift+F9	Debug QuickWatch	快速查看变量或表达式的值
	Shift+F5	Debug Stop debugging	停止调试

第二部分 实验指导

【注意事项】

(1) 每次实验以前，需要详细阅读实验目的、实验要求和实验提示，以便能准确地理解实验要求，达到实验的目的。有测试数据要求的，需要给出测试结果；有要求回答问题的，需要给出回答。每次实验都有几个题目，要求为每个题目创建不同的项目，以便检查。

(2) 有的实验前后有联系，需要先完成前面的实验再进行后面的实验，请务必按照先后顺序完成实验。

(3) 每个实验完成后须提交实验报告及相应的程序。

实验一 熟悉实验环境

◇ 实验目的

- (1) 了解和使用 Visual C++ 集成开发环境。
- (2) 熟悉 Visual C++ 环境的基本命令和功能键,熟悉常用的功能菜单命令。
- (3) 学习完整的 C++ 程序开发过程。
- (4) 理解简单的 C++ 程序结构。
- (5) 熟悉最简单 C++ 程序的编写、调试和运行。
- (6) 了解用 Visual C++ 开发 Windows MFC 应用程序的过程。

◇ 实验内容

(1) 学习有关 Visual C++ 开发环境的一些知识,并尝试实现一个简单的 Visual C++ 控制台应用程序(DOS 程序)和 Windows 程序。

(2) 了解用 Visual C++ 开发 C++ 应用程序的过程,能编写和运行简单的 C++ 程序。

1. 熟悉 Visual C++ 实验环境

实验步骤

- (1) 启动 Developer Studio,看看初始化界面由哪些部分组成。
- (2) 查看各菜单项,看看都有哪些子菜单和命令。
- (3) 将鼠标放置于各工具条图标上,系统会自动显示该图标代表的命令含义,了解一下都有哪些命令。
- (4) 在任意工具条上单击鼠标右键,弹出式菜单上将显示所有可用的工具条,选择其中没有对号(✓)的项,看看有什么效果,再选择有对号的项,又有什么效果。
- (5) 将鼠标移动到左边的工作区窗口,按下鼠标左键不放,移动鼠标到屏幕中间,观察发生现象。再将它拖回到原来位置,观察发生现象。
- (6) 将鼠标移动到下边的输出窗口,按鼠标右键,弹出一个菜单,选择其中的菜单项“Hide”,结果如何?要重新显示该窗口,选择菜单“View|Output”,窗口是否又显示出来?
- (7) 学习使用帮助系统。选择菜单“Help|Contents”,启动 MSDN 联机帮助系统,学习使用该帮助系统。联机帮助系统是一个相对独立的程序,它和 Developer Studio 是两个程序,但是它的启动和停止都受 Developer Studio 影响。MSDN 联机帮助系统运行的前提条件是 Developer Studio 在运行。
- (8) 选择“File|Exit”命令退出 Developer Studio。

2. 控制台应用

用 AppWizard 建立一个控制台应用,在终端上输出“hello”。

“控制台应用程序”是一个在 DOS 窗口中运行的基于字符的程序。由于这种模式的应用程序比 Windows 程序简单,我们先选择利用 Visual C++ 来建立这样一个应用,这样使得我们可以将精力先投入到学习使用 C++ 编程语言,而不需要把过多的精力投入到学习

复杂的 Windows 编程中去。

实验步骤

1) 创建一个应用

首先创建一个项目(project), 项目将代表程序员的应用, 存放应用的所有信息, 包括源文件、资源文件和编译连接设置等。创建项目的步骤如下:

- (1) 启动 Developer Studio。
- (2) 从主菜单中选择“File|New”, 显示出“New”对话框。
- (3) 选择“Projects”标签, 并从列表中单击“Win32 Console Application”(参见图 2-1)。

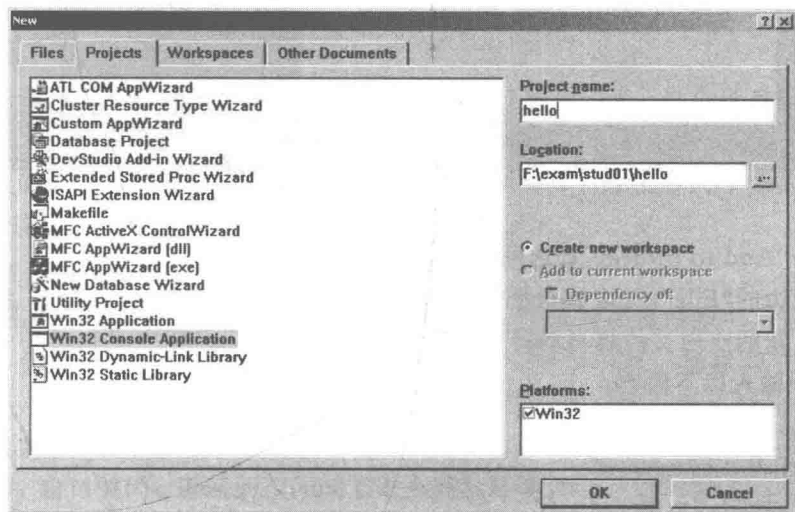


图 2-1 选择“Projects”标签

(4) 在“Location”编辑框中输入工作目录名称, 如 F: \exam\stud01(按照实验指导教师的要求)。

(5) 在对话框的右上角的“Project name”编辑框内键入项目的名字, 如“hello”, 系统将自动为项目分配一个默认的目录。

(6) 单击“OK”按钮继续。

(7) Visual C++6.0 系统将显示一个询问项目类型的程序向导, 选择“an empty project”。

(8) 单击“Finish”或“OK”按钮结束配置, 创建应用程序。

这时系统创建一个新的项目, 并且在左边的工作区窗口中出现项目的名字。工作区窗口增加了两个标签: ClassView 和 FileView。ClassView 从类的角度显示项目中建立的各个类, 双击某个类名将会在右边的文档显示区显示类的定义文件, 并把文件的当前位置定位到所选的类; FileView 显示构成项目的各个文件, 选择某一文件将会在右边的文档显示区显示文件内容。

2) 编辑一个 C++ 源程序

用下面的方法在创建项目中添加一个文件:

- (1) 在主菜单上选择“File|New”。
- (2) 在“New”对话框中选择“File”标签, 单击“C++ Source File”, 如图 2-2 所示。

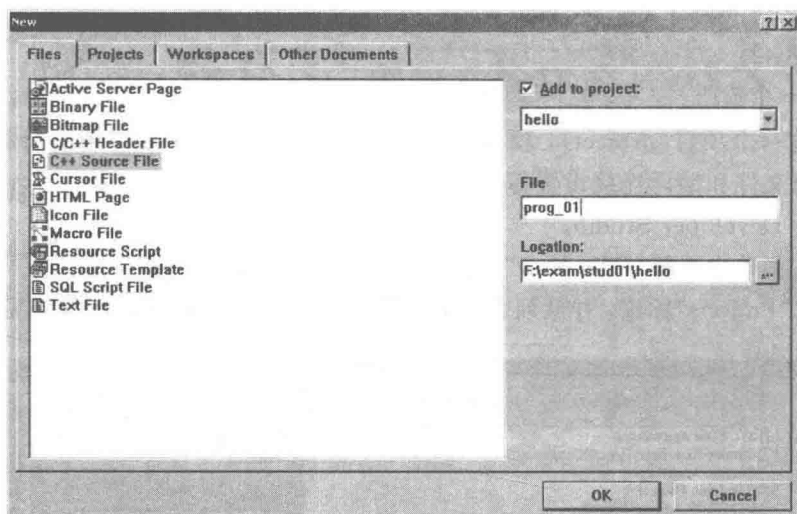


图 2-2 添加文件

(3) 选中“Add to project”复选框。

(4) 在右边的“File”name 编辑框中为文件指定一个名字，如“prog_01”，系统将自动加上后缀. cpp。新的空白文件将自动打开，显示在文档显示区。

在文件中输入以下内容：

```
// this is a c++ program
#include <iostream. h>
void main()
{
    cout << "Welcome to c++! \n" ;
    cout << "This is a c++ program. \n" ;
} //
```

虽然许多编辑命令可以通过菜单和工具栏实现，但大量的编辑命令都可以通过键盘实现。通常用键盘实现以下命令：

- 撤消前一次操作。当程序员进行了一次错误的操作时，可以通过敲击键盘上的 Ctrl+Z 完成。
- 重复前一次操作。通过敲击 Ctrl+Y 实现。
- 剪切一行。用 Ctrl+L 来删除一行并将它放到剪切板中。
- 剪切。将选中的文本删除并将它放到剪切板中，用 Ctrl+X 实现。
- 复制。将选中的文本复制到剪切板中，用 Ctrl+C 实现。
- 粘贴。将剪切板中的内容放到编辑器中文本的当前位置处(由光标指示)，用 Ctrl+V 实现。

要想了解关于键盘操作命令的完整列表，可以选择“Help”菜单下的“Keyboard Map”。程序员没有必要记住所有的命令，有些根本不常用。

3) 保存源文件

单击工具栏中的“Save”图标，或者选择“File|Save”来保存文件。

C++源文件的扩展名为.cpp。扩展名非常重要, Developer Studio 根据文件的扩展名来区分文件类型, 并且根据文件类型提供相应的编辑帮助(如正确的语法高亮显示)。

4) 编译、连接得到可执行程序

编辑结束后, 仔细检查输入的内容, 看有无错误。确认没有错误之后, 选择主菜单的“Build|Build Hello.exe”来编译项目(也可以按功能键 F7)。如果输入的内容没有错误, 那么在屏幕下方的输出窗口将会显示:

```
hello.exe - 0 error(s), 0 warning(s)
```

如果在编译时得到错误或警告, 则是源文件出现错误, 再次检查源文件, 查找错误, 然后改正它。

5) 改正源程序中的错误

编译的错误会在 Developer Studio 下方的输出窗口显示出来, 逐个查看这些错误的内容, 用鼠标双击, 光标可以自动移动到发生错误的源程序的相应地点, 仔细检查源程序, 改正发生错误的地方, 注意是否遗留了分号、引号或括号等。改正后, 再重复步骤 4) 的操作, 直到编译连接通过为止。

6) 运行程序

选择“Build|Execute hello.exe”(或者 Ctrl+F5), 在开发环境中执行程序。程序运行以后将显示一个类似于 DOS 的窗口, 在窗口中输出一行“hello”, 紧接着在下面显示出“Press any key to continue”, 这句话是系统提示程序员按任意键可退出当前运行的程序, 回到开发环境中。

按任意键, 窗口关闭, 退回到 Visual C++开发环境。

3. 编写简单的计算程序

程序功能: 输入圆的半径, 计算圆的周长和面积并输出, 测试数据见表 2-1。

表 2-1 测试数据

输 入	输 出
2	The perimeter of the circle : 12.5664 The area of the circle : 12.5664
10	The perimeter of the circle : 62.8318 The area of the circle : 314.159

实验步骤

(1) 创建一个控制台项目。选择菜单“File|New”, 在“Projects”标签下选择“Windows32 Console Application”, 输入项目名称“circle”, 然后按“OK”按钮。

(2) 在项目中增加一个文件。选择菜单“File|New”, 在“Files”标签下选择“C/C++ Source File”, 输入文件名称“circle”, 然后按“OK”按钮。

(3) 在文件中输入以下内容。

```
//该程序让用户输入圆的半径 Radius, 输出圆的周长 Perimeter 和面积 Area
#include <iostream.h>
#include <math.h>
const double PI = 3.1415926; //定义一个常量 PI
```



```
void main()
{
    int radius;
    double perimeter, area;
    cout << "Please input the radius: " ;
    cin >> radius;
    perimeter = 2 * PI * radius;           // 周长=2πR
    area = PI * pow(radius, 2);           // 面积=πR²
    cout << "The perimeter of the circle : " << perimeter << endl;
    cout << "The area of the circle : " << area << endl;
} //
```

注: $\text{pow}(\text{radius}, 2)$ 表示求 radius 的平方, $\text{pow}(x, y)$ 是系统预定义的函数, 该函数计算 x 的 y 次方。该函数的原型在文件 `math.h` 中有说明。

(4) 编译、连接并运行程序。如果程序员输入的程序有误, 则用下面的方法定位并修改错误, 直到编译连接通过。然后运行程序, 测试数据。

编译和连接错误定位包括以下方法:

- 开发环境下方的输出窗口(Output)显示编译和连接过程中出现的错误, 错误信息包括错误出现的文件名、行号和错误代码。
- 如果不懂错误消息, 将光标移动到该错误信息, 按 F1 键, 就可以显示该错误的帮助信息。
- 在 Output 窗口, 双击错误或者选择该错误再按 Enter 键, 系统自动将光标移动到发生错误的源程序行, 然后就可以改正错误。
- 按 F4 键可以选择并定位下一个错误。

思 考 题

- (1) 程序中为什么要将头文件 `math.h` 包含进来?
- (2) 建立控制台应用程序的通用步骤是什么?
- (3) 分析配套教材习题 1-4 中 i, j, k 的值, 体会前置++与后置++的区别。

4. Windows 应用程序的建立(选做)

实验步骤

1) 创建一个新项目

利用 Developer Studio 的 AppWizard 创建一个新项目, 步骤如下:

- (1) 选择菜单“File|New”, 系统将显示“New”对话框。
- (2) 选择“Projects”标签, 在显示的项目类型中选择“MFC AppWizard(exe)”。
- (3) 在右边的“Project Name”编辑框中输入项目名称, 如“helloMFC”, 然后按“OK”按钮。
- (4) MFC AppWizard 将分几步询问有关要建立的新项目配置。第一个对话框问创建哪种类型的应用(单文档、多文档还是对话框类型), 选择创建单文档应用“Single document”, 然后按“Next”按钮(初学先了解单文档部分)。
- (5) 翻过后面的五个页面(按“Next”按钮), 每个页面可以让程序员改变项目的不同选