



21世纪高校计算机规划教材

大学程序设计实训教程

夏 耘 刘丽霞 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

21 世纪高校计算机规划教材

大学程序设计实训教程

夏 耘 刘丽霞 主 编

臧劲松 黄小瑜 黄春梅 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书由全国计算机等级考试命题组专家和一线教师结合多年的教学经验及对考试范围、重点和难点的研究,为大学生精心打造而成,既重视培养学生计算思维能力和解决问题的能力,又兼顾应试指导。

本书注重基本概念的系统化,叙述简明扼要,重视开发能力的培养。书中对 C 语言和 Visual Basic 程序设计的知识点进行了有重点的讲解,结合典型案例的分析,强调学生开发能力的培养,对历年计算机等级考试的知识点和题型逐一进行了分析。本书内容精练、结构合理、重点突出。

本书共包含 6 章,其中第 1 章对有代表性的 C 语言和 Visual Basic 编程开发环境做了详细的介绍。第 2 章简明扼要地回顾了 C 语言和 Visual Basic 程序设计语言的重点语法知识。第 3 章和第 4 章分别以社团管理系统和图书管理系统为例,分析了 C 语言和 Visual Basic 面向对象程序设计开发的基本步骤和注意事项,可使学生切实了解程序设计语言如何解决问题以及解决什么样的问题,进而更加明确学习目标,增强学习兴趣。第 5 章为典型习题解析,巩固对知识点的理解。第 6 章为上海市计算机等级考试强化训练,结合计算机等级考试大纲的要求,进行模拟操作和练习。附录中提供了软件开发过程中的常用文档格式。

本书以“理论—实用—实训”为主线,注重理论联系实际,引导学生在编程中探索知识、举一反三。本书适合作为大学本科计算机程序设计基础课程的配套实训教材。

图书在版编目(CIP)数据

大学程序设计实训教程 / 夏耘, 刘丽霞主编. —北
京: 中国铁道出版社, 2010. 5
21 世纪高校计算机规划教材
ISBN 978-7-113-11214-1

I. ①大… II. ①夏…②刘… III. ①
C 语言—程序设计—高等学校—教材②
BASIC 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 055336 号

书 名: 大学程序设计实训教程
作 者: 夏 耘 刘丽霞 主编

策划编辑: 严晓舟 路 璐

责任编辑: 侯 颖

封面设计: 付 巍

版式设计: 于 洋

编辑部电话: (010) 63560056

封面制作: 白 雪

责任印制: 李 佳

出版发行: 中国铁道出版社 (北京市宣武区右安门西街 8 号 邮政编码: 100054)

印 刷: 北京新魏印刷厂

版 次: 2010 年 6 月第 1 版 2010 年 6 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 14.5 字数: 359 千

印 数: 3 600 册

书 号: ISBN 978-7-113-11214-1

定 价: 26.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签, 无标签者不得销售

凡购买铁道版图书, 如有印制质量问题, 请与本社计算机图书批销部联系调换。

前 言

大学程序设计课程借助程序设计的知识载体,传授基本问题的求解过程和基本解决思路,使学生建立算法意识,具有良好的思维习惯,从学科方法论上打基础,培养学生的计算思维能力和分析问题、解决问题的能力。程序设计是高强度的脑力劳动,不是听会的、也不是看会的,而是练会的,本书正是本着该理念编写而成的。

本书注重基本概念的系统化,叙述简明扼要,重视对学生开发能力的培养。书中对 C 语言和 Visual Basic 程序设计的知识点进行了有重点的讲解,结合对典型案例的分析,着重培养学生的开发能力。本书内容精练、结构合理、重点突出。

本书共分 6 章,其中第 1 章对有代表性的 C 语言和 Visual Basic 编程开发环境做了详细的介绍。第 2 章简明扼要地回顾了 C 语言和 Visual Basic 程序设计语言的重点语法知识。第 3 章和第 4 章分别以社团管理系统和图书管理系统为例,分析了 C 语言和 Visual Basic 面向对象程序设计开发的基本步骤和注意事项,可使学生切实了解程序设计语言如何解决问题以及能解决什么样的问题,进而使学生更加明确这门课程的学习目标,增强对本课程的学习兴趣。第 5 章为典型习题解析,帮助读者巩固对知识点的理解。第 6 章为上海市计算机等级考试强化训练,结合计算机等级考试大纲的要求,进行模拟操作和练习。此外,附录中提供了软件开发过程中的常用文档格式。

本书注重理论联系实际,由浅入深,条理清楚,引导学生在编程中探索知识、举一反三。本书适合作为大学本科计算机程序设计基础课程的配套实训教材。

本书由全国计算机等级考试命题组专家和一线教师结合多年的教学经验及对考试范围、重点和难点的研究精心编写而成,既重视培养学生的计算思维能力和解决问题的能力,又兼顾应试指导。

本书是为初学程序设计语言的高等院校学生编写而成的。根据高等院校学生的培养目标和本课程的特点,在编写过程中,始终强调实践和应用。

为了配合学习还提供了资料包,其中包含电子教案、教材案例、习题素材、考试模拟题等以及书中部分源程序文件,供读者参考。

本书由夏耘编写第 1 章和第 6 章,黄小瑜和黄春梅共同完成第 2 章和第 5 章的编写,臧劲松编写第 3 章,刘丽霞编写第 4 章。

最后,再次感谢有关专家、教师长期以来对我们的支持。

由于时间紧迫,错误和疏漏在所难免,恳请专家和广大读者批评指正。

编 者

2010 年 3 月

目 录

第 1 章 开发环境	1
1.1 Visual C++ 6.0 环境介绍	1
1.1.1 Visual C++ 6.0 概述	1
1.1.2 安装 Visual C++ 6.0 环境	1
1.1.3 Visual C++ 6.0 的组成	1
1.1.4 Windows 编程模式	4
1.1.5 Visual C++ 6.0 的引例	7
1.1.6 Visual C++ 6.0 常用控件	12
1.2 Visual Basic 2005 环境介绍	19
1.2.1 Visual Basic 2005 概述	19
1.2.2 Visual Studio 开发环境	20
1.2.3 Visual Studio 工具	22
第 2 章 程序设计基础	23
2.1 C 语言重点语法知识	23
2.1.1 基本概念	23
2.1.2 基本数据类型、基本运算和表达式	23
2.1.3 程序结构学习重点	26
2.1.4 数组学习重点	28
2.1.5 函数学习重点	31
2.1.6 指针学习重点	33
2.1.7 结构体与链表学习重点	35
2.1.8 文件学习重点	37
2.2 VB.NET 重点分析	39
2.2.1 VB.NET 程序设计语言基础	39
2.2.2 VB.NET 的流程控制	43
2.2.3 VB.NET 的数组和结构	48
2.2.4 VB.NET 的过程	53
2.2.5 VB.NET 的窗体控件和界面设计	56
2.2.6 VB.NET 面向对象语言基础	63
2.2.7 VB.NET 的文件操作	66
2.2.8 VB.NET 的图形处理	67
2.2.9 VB.NET 的数据库基础	69
2.2.10 VB.NET 的错误处理与调试	69

第3章 Visual C++典型案例分析	71
3.1 需求分析	71
3.2 系统设计分析	71
3.3 数据结构设计与实现	73
3.4 用户界面设计	74
3.5 代码实现	76
第4章 VB.NET 典型案例分析	83
4.1 系统功能与需求分析	83
4.1.1 系统功能	83
4.1.2 系统需求分析	84
4.1.3 维护管理	85
4.2 数据库结构设计 with 实现	85
4.2.1 数据实体间的关系	85
4.2.2 基本表的创建及其设计视图	86
4.3 界面设计	88
4.4 代码实现	92
4.5 实践项目	100
第5章 典型习题解析	103
5.1 C语言典型习题解析	103
5.2 VB.NET 典型习题解析	132
5.3 计算机应用基础典型习题解析	154
第6章 上海市计算机等级考试强化训练	191
6.1 上海市计算机等级考试概述	191
6.2 上海市计算机等级考试复习纲要	191
6.2.1 上海市计算机等级考试(一级)考核基本要求	191
6.2.2 上海市计算机等级考试(二级 C)考核基本要求	195
6.2.3 上海市计算机等级考试(二级 VB)考核基本要求	197
6.3 上海市计算机等级考试模拟练习题	199
6.3.1 上海市计算机等级考试(一级)模拟练习题	199
6.3.2 上海市计算机等级考试(二级 C)模拟练习题	204
6.3.3 上海市计算机等级考试(二级 VB)模拟练习题	209
6.3.4 上海市计算机等级考试(二级 VB.NET)模拟练习题	213
6.4 上海市计算机等级考试复习题答案	217
6.4.1 上海市计算机等级考试(一级)模拟练习题参考答案	217
6.4.1 上海市计算机等级考试(二级 C)模拟练习题参考答案	217
6.4.3 上海市计算机等级考试(二级 VB)模拟练习题参考答案	218
6.4.4 上海市计算机等级考试(二级 VB.NET)模拟练习题参考答案	218
附录 A	219
参考文献	225

第 1 章 | 开 发 环 境

1.1 Visual C++ 6.0 环境介绍

1.1.1 Visual C++ 6.0 概述

Visual C++是可视化编程环境,它提供了应用程序、服务、控件的开发环境。它使用了 Microsoft Windows 图形用户界面的许多先进特性和设计思想,采用了弹性可重复利用的完整的面向对象程序语言 (Object-Oriented Language)、快捷的编辑器、领先的数据库技术。具体特点如下:

- ① Visual Studio 由一套集成工具组成,用于开发 Windows 下的应用程序。
- ② 具有强大的向导工具 (MFC AppWizard、ClassWizard)。
- ③ 具有强大的实用工具程序。
- ④ 具有自动完成语句功能。
- ⑤ 具有快速宏录制功能。
- ⑥ MFC 类库支持多线程应用程序的开发。
- ⑦ 允许用户适应强大的数据库应用程序。
- ⑧ 具有强有力的 Internet 支持。
- ⑨ 支持 OLE。
- ⑩ 支持 IE5 新控件的资源编辑器。

1.1.2 安装 Visual C++ 6.0 环境

安装 Visual C++ 6.0 的步骤如下:

- ① 将 Visual C++ 6.0 安装光盘放入光驱中。
- ② 利用资源管理器打开光盘,运行 Setup.exe 程序。
- ③ 单击“确定”按钮,进入安装向导。
- ④ 单击 Next 按钮,开始安装。按屏幕提示信息进行设置,通过单击 Next 按钮,完成每一步安装。最后,单击 OK 按钮。

1.1.3 Visual C++ 6.0 的组成

Visual C++ 6.0 本身是一个庞大的系统,在使用该系统前,对它的认识是必要的。Visual C++ 6.0 由可视化的工作平台、导航、管理事件控制消息工具、组件廊、资源管理器、调试器和诊断工具、在线帮助 7 部分组成,下面将分别介绍各部分的功能。

1. 可视化的工作平台

可视化的工作平台有两个功能：

① 进入 Visual C++ 6.0 首先会建立一个工程文件，而工程文件内包含许多文件。生成工程文件前，工作平台自动生成一个工作空间（Workspace），只要不退出该工作空间，系统所生成的文件或操作的工程文件都在该工作空间中。记录工作空间所有信息的文件扩展名为.mdp，双击该文件自动进入 Visual C++ 6.0 环境并装载构成该工作空间的所有文件。

② 文本编辑器，它能根据 C/C++ 的语法自动调整程序代码格式。

2. 导航

导航组件帮助用户建立基本框架，Visual C++ 6.0 提供的导航组件有两类：AppWizard 和 ControlWizard。使用 AppWizard 能生成如下的应用程序：

- 单文档、多文档或基于对话框的应用程序。
- OLE 支持和数据库（ODBC 和 DAO）支持。
- 工具条、状态条、上下文敏感帮助和三维界面。
- 文件的内部功能，如打开、打印命令等。
- 窗口边框的风格控制。

当用户在 AppWizard 导航下输入基本信息，并单击“确定”按钮后，将自动生成 Windows 应用程序必需的开始文件（包括源文件、头文件、资源文件、工程文件等）。

ControlWizard 用来创建 OLE 控件项目的框架，它所生成的初始文件中的类、事件、属性和方法都及时记录在 ClassWizard 中，后期的可视化更改正基于此。

3. 管理事件控制消息工具

ClassWizard 用于创建 Windows 消息和命令处理函数、创建和管理类、创建类成员变量、创建 OLE Automation 的方法和属性、创建数据库类，以及其他一些工作。ClassWizard 也有助于覆盖 MFC 类中的虚函数。一般的使用方法是，先选择类，再选择需要覆盖的虚函数。该过程的其余部分与消息处理是类似的。

WizardBar 进一步简化了 ClassWizard 的日常工作，它能快速地选择一个类组件，并修改虚拟函数、Windows 消息等。

4. 组件廊

（1）ActiveX 控件

ActiveX 控件的前身就是 OLE 控件。由于 ActiveX 控件与开发平台无关，因此，在一种编程语言上开发的 ActiveX 控件无须任何修改，即可在另一种编程语言中使用，其效果如同使用 Windows 通用控件一样。例如，与 Internet Explorer 一起提供 ActiveX 控件可用于增强 Web 页功能，使其具有复杂的功能和动画。

ActiveX 控件结合了 Java 程序和 Netscape 插件的优点，还可以用许多编程语言，包括所有 Microsoft 编程和在数据库语言编写程序中使用 ActiveX 控件。

（2）ActiveX 文档

用一个 ActiveX 识别 Web 浏览器（如 Internet Explorer 浏览器）时，ActiveX 文档能够使用自己的工具栏和菜单打开程序。这意味着可以通过使用 ActiveX 识别 Web 浏览器打开非 HTML 文件，如 Microsoft Excel 或 Microsoft Word 文件。

(3) ActiveX 脚本

ActiveX 脚本支持最常用的脚本语言, 包括 Microsoft Visual Basic 脚本和 JavaScript。ActiveX 脚本的任务是让用户在能够将以 Active X 脚本开发的自定义代码升级到该脚本任务提供的更高级功能之前, 可以继续使用此类脚本。

组件库继承了 ActiveX 技术, 使代码可重用。利用它能实现创建或增加组件到组件库中, 也可以创建用户自己的组件类型, 以及将组件从一个类型转换到另一个类型。

5. 资源编辑器

使用 Visual C++ 中的资源编辑器能体验到资源编辑的功能是集成在 IDE 环境中的。若想使用它则需安装整个 Visual C++ 软件包, 这至少需要几百兆的硬盘空间。另外, 用 Visual C++ 生成的 .rc 文件中包含很多 VC 自己的头文件, 如果将它们去掉, 下次则无法再用 VC 打开文件; 如果不去掉这些多余的信息, 那么用 Rc.exe 编译的时候, 就要把所有需要的头文件复制过来。当源程序与他人交流的时候, 则要将资源脚本放到 VC 中去找到头文件。使用 Visual C++ 来编辑资源, 存盘时直接保存为 *.res 文件, 这样可以省去编译资源的步骤。到最后调试完成时, 可以保存为 .rc 文件并将文件中 VC 使用的多余内容去掉, 整理成 Rc.exe。

6. 调试器和诊断工具

Visual C++ 调试器常用的调试手段是断点。在 VC++ 中设置一个断点很简单: 装入源文件, 把光标放在想要停止的代码行上, 将鼠标指向 Insert/Remove Breakpoint 按钮, 然后单击即可。这个过程称为“设置位置断点”。当程序运行到这一行代码时就会停下来, 等待进一步操作。显然, 程序能够在用户期望的地方停下来, 用户可以对程序进行调试。常用的调试方法如下:

(1) 通过变量查看窗口调试

Visual C++ 有 3 种查看窗口: AUTOS、LOCALS、WATCH。

- AUTOS: 根据当前语句行所涉及的数据显示其相应的变量值。
- LOCALS: 根据当前程序的上下文, 显示最近的局部变量值。
- WATCH: 显示用户所指定的变量值。

在 AUTOS 和 LOCALS 窗口中显示的值是自动格式的, 只能在十进制和十六进制之间切换, 但在 WATCH 窗口中则有很多参数。例如, 显示整型变量 i 的值, 可以写成:

```
i,u           //将 i 显示为无符号的十进制整数
i,x           //将 i 显示为十六进制整数
i,c           //将 i 显示为单字符
```

这种格式码很多, 具体可以看 MSDN。此外, 在 WATCH 中还能够显示 CPU 寄存器, 例如:

```
@EAX,x       //显示 EAX 寄存器的内容
```

也可以在 WATCH 窗口中显示一个表达式, 例如:

```
i+j
Object.m_strName
```

当然, 也可以把 WATCH 作为一个计算器使用, 例如:

```
1000/365
```

甚至还能在 WATCH 中调用一个函数。

(2) 查看堆栈

可以在 CALL STACK 窗口中看到程序中函数是如何调用的, 例如:

```
demo.exe!main() Line 112
demo.exe!mainCRTStartup() Line259+0x12
kernel32.dll!77e5eb69()
```

这说明在 demo.exe 模块中的 mainCRTStartup() 被 kernel32.dll 模块中的 77e5eb69() 位置的函数调用了, 而它又调用了 demo.exe 模块中的 main() 函数。如果函数有参数, 那么这些参数也会显示出来。双击显示的某一行, 即可调出相应的源代码。

(3) 其他

能够让 Visual C++ 显示出多线程的运行信息、内存信息、模块信息、反汇编信息以及寄存器信息等。

在断点所在行右击, 在弹出的快捷菜单中选择 Breakpoint Properties 命令, 这里会出现一个 Condition 和 Hit Count 按钮。在 Hit Count 中输入一个数字, 表示在停止运行前通过这个断点的次数, 或者使用其他判断条件。在 Condition 中输入一个条件表达式, 表示如果满足这个条件则在这个断点停下。此处只能使用 C 类型比较运算符, 不能调用任何函数, 不能包含任何宏。在 Condition 中还可以选择当这个表达式的值改变后在断点停下。通过使用高级断点能够大大增强调试器的功能, 但是它还有其他的断点类型, 通过进一步使用断点语法, 能够设置更加详细、具体的断点。

调试可以分成以下几个步骤:

(1) 复制错误

所谓复制错误, 就是把错误再现出来, 便于进一步对其进行分析。这是整个调试过程中非常重要的一个步骤。有时候这一步并不是十分容易, 甚至有时候不能复制错误, 这就要求程序本身对错误有一定的处理, 最简单的就是把相关数据导出, 如在 MFC 中支持的 DUMP 功能等。

(2) 分解错误

一旦错误再次出现, 就需要确定错误的发生地点。在这个阶段需要对错误的原因进行猜想, 同时依据各种变量、对象的状态分析程序当前是否正常。通过逐步排除正确的程序段, 把错误限定在一个较小的范围内。随着调试过程的不断进行, 猜想会越来越准确, 当积累了一定的经验后, 在某个时间时就能够“领悟”到错误的原因。

(3) 解决错误

虽然找到了错误的原因, 可是解决错误有时是很困难的。如发现错误是源于对系统的错误设计, 为了解决这个错误需要对程序进行很大的改动。当然, 也有相当一部分错误是很容易解决的。

(4) 校验错误

一旦解决了错误后, 一定要对改正的结果进行校验。在很多时候, 解决了一个错误会产生一些副作用, 为此必须对更正进行校验。

7. 在线帮助

Books Online 是 Visual C++ 6.0 自带的、十分全面的在线帮助系统, 在主窗口区域单击 Info view 选项卡就会出现树状组织的整个内容。每个文档都是用 HTML 编写的。

1.1.4 Windows 编程模式

Windows 程序中必须要有 WinMain 函数, 因为该函数最重要的任务是创建该应用程序的主窗口。Windows 程序与基于 MS DOS 程序的最大差别就在于: MS DOS 程序是通过调用操作系统的功能来获得用户输入的, 而 Windows 程序是通过操作系统发送的消息来处理用户输入的。Windows 消息都是经过严格定义的, 并且适用于所有的程序。

Windows 提供了通用的图形设备接口 (GUI)，通过调用 (GDI) 函数和硬件打交道，不必理会设备环境，Windows 会自动将设备环境结构映射到相应的物理设备。

Windows 程序设计中所需要的数据是存储在资源文件中的，这样连接器就可以把编译好的二进制代码和二进制资源文件结合起来生成可执行程序。资源文件可以包括位图、图标、菜单定义、对话框设计，甚至可以包含用户自己定义的格式。

Windows 程序允许动态的链接目标模块，并且多个应用程序可以共享同一个动态链接库。VC++ 的源程序浏览器能够使人们从类或函数的角度来了解或编辑程序，而不是直接从文件入手。在看别人的源代码时如果能熟练地使用源代码浏览器将会事半功倍。源程序浏览器主要的查看状态有以下几种：

- **Definitions and References:** 选择任何函数、变量、类型、宏定义，可以看到其在项目中的定义，以及在何处用到它。
- **Call Graph/Caller Graph:** 对于所选择的函数，给出它的调用与被调用函数的图示。
- **Derived Class Graph/Base Class Graph:** 给出类层次关系的图形表示，可以看到所选择的类的派生类和基类以及成员。
- **File Outline:** 对于所选的文件，列出文件中的类、函数和数据成员，同时还显示它们定义的位置和使用位置。

SDK 即 software develop kit (软件开发工具包)，它包含了进行 Windows 软件开发的文档和 API 函数的输入库、头文件 (因为 API 在动态链接库中，这些动态链接库是系统的组成部分，因此不用再提供，而输入库和头文件则是必需的，这样才能在程序中使用 API 函数)。早期 SDK 是一个单独发放的包，现在 Visual C++ 和其他一些开发环境中已经包含了它。如果已经安装了 VC++，那么就可以开始编写 Windows 程序了。随着 Windows 系统的发展，SDK 的内容越来越多。

一个基本的 Windows SDK 程序有几十行，尽管也不算长，但比 C 版的 hello world 长得多。而且，里面充斥着奇怪的变量类型和常量定义，首先会有一个 `#include`，然后是一个函数声明 `"LRESULT CALLBACK WndProc( HWND, UINT, WPARAM, LPARAM )"`，函数名前面有两个修饰符。

Windows 的数据类型尽管看上去很陌生，但它们是由 C 的基本数据类型定义的。例如，UINT 就是 `unsigned int`，PSTR 就是 `pointer to string` 的意思。Windows 还有很多系统定义的结构体，如 `WNDCLASS`、`MSG` 等，这些东西见得多了就自然明白了。Windows 还有一个重要的概念——句柄。通过句柄就可以操作 Windows 对象。HWND、HINSTANCE、HDC 等都是句柄。

Windows 程序的结构。一般有一个 `WinMain()` 函数作为程序的入口点，在 `WinMain()` 里面定义窗口类，进行消息循环。消息循环就是普通的 `while` 循环，在其中接收消息、分发消息。然后是窗口函数 `WndProc()`，名称可以由用户自己定。在其中用一个大的 `switch` 结构检索消息，在每个 `case` 下面写处理消息的代码。最简单的 Windows SDK 程序只要写这两个函数即可。当程序较长时，就要把特定的消息处理代码写成函数，以便在处理消息时调用，甚至可以使用 C++ 来写程序。等熟悉这种结构以后，就可以任意发挥了。学 SDK 很重要的是不要期望在开始时就弄清楚每行代码。

MFC 借助 C++ 的优势为 Windows 开辟了一片新天地，同时也借助 `ApplicationWizard` 使开发者脱离了那些每次都必写的基本代码，借助 `ClassWizard` 和消息映射使开发者摆脱了定义消息处理时那种混乱和冗长的代码段。更令人兴奋的是利用 C++ 的封装功能可使开发者摆脱 Windows 中各

种句柄的困扰,只需要面对 C++ 中的对象,这样一来就使得开发更接近开发语言而远离系统。(但笔者个人认为了解系统原理对开发很有帮助。)

MFC 中常用类、宏、函数如下:

(1) 常用类

CRect: 用来表示矩形的类,拥有四个成员变量: top、left、bottom、right。分别表示左上角和右下角的坐标。可以通过以下的方法构造:

<code>CRect(int l,int t,int r,int b);</code>	指明四个坐标
<code>CRect(const RECT& srcRect);</code>	由 RECT 结构构造
<code>CRect(LPCRECT lpSrcRect);</code>	由 RECT 结构构造
<code>CRect(POINT point,SIZE size);</code>	由左上角坐标和尺寸构造
<code>CRect(POINT topLeft,POINT bottomRight);</code>	由两点坐标构造

成员函数:

<code>int Width() const;</code>	得到宽度
<code>int Height() const;</code>	得到高度
<code>CSize Size() const;</code>	得到尺寸
<code>CPoint& TopLeft();</code>	得到左上角坐标
<code>CPoint& BottomRight();</code>	得到右下角坐标
<code>CPoint CenterPoint() const;</code>	得到中心坐标

此外矩形可以和点 (CPoint) 相加进行位移,和另一个矩形相加得到“并”操作后的矩形。

CPoint: 用来表示一个点的坐标,有两个成员变量 x 和 y。可以和另一个点相加。

CString: 用来表示可变长度的字符串。使用 CString 可不指明内存大小, CString 会根据需要自行分配。

<code>GetLength</code>	得到字符串长度
<code>GetAt</code>	得到指定位置处的字符
<code>operator +</code>	相当于 strcat
<code>void Format(LPCTSTR lpszFormat,...);</code>	相当于 sprintf
<code>Find</code>	查找指定字符、字符串
<code>Compare</code>	比较
<code>CompareNoCase</code>	不区分大小写比较
<code>MakeUpper</code>	改为小写
<code>MakeLower</code>	改为大写

CStringArray: 用来表示可变长度的字符串数组。数组中每一个元素为 CString 对象的实例。

<code>Add</code>	增加 CString
<code>RemoveAt</code>	删除指定位置的 CString 对象
<code>RemoveAll</code>	删除数组中所有 CString 对象
<code>GetAt</code>	得到指定位置的 CString 对象
<code>SetAt</code>	修改指定位置的 CString 对象
<code>InsertAt</code>	在某一位置插入 CString 对象

(2) 常用宏

RGB
TRACE
ASSERT
VERIFY

(3) 常用函数

```
CWndApp* AfxGetApp();  
HINSTANCE AfxGetInstanceHandle();  
HINSTANCE AfxGetResourceHandle();
```

“int AfxMessageBox(LPCTSTR lpszText,UINT nType=MB_OK,UINT nIDHelp=0);”用于弹出一个消息框。

1.1.5 Visual C++ 6.0 的引例

【任务】

设计一个欢迎进入系统的界面，其效果如图 1-1-1 所示。

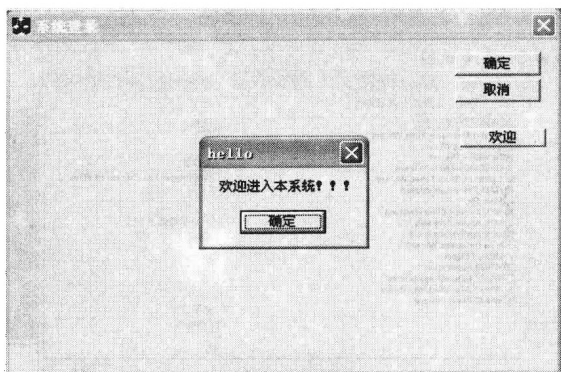


图 1-1-1 引例执行界面

其中，“欢迎”按钮是引例通过代码控制的，即单击“欢迎”按钮，立即显示 hello 对话框，在此对话框中显示文字“欢迎进入本系统!!!”。单击该对话框中的“确定”按钮，对话框立即关闭，单击应用程序左上角图标，其状态如图 1-1-2 所示，在此状态下，选择“关于 hello”选项则出现图 1-1-3 所示的界面。单击“关于 hello”对话框的“确定”按钮，退出应用程序。

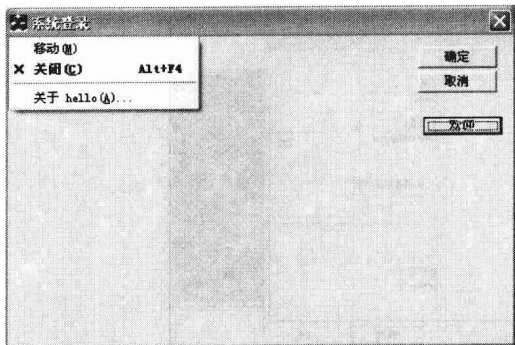


图 1-1-2 单击应用程序左上角图标时的界面

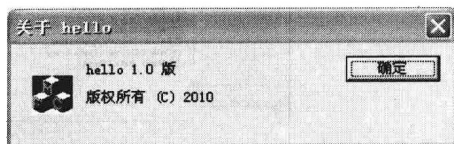


图 1-1-3 hello 版权界面

【生成引例工程文件的步骤】

① 进入 Visual C++ 6.0 系统，选择菜单“文件”→“新建”命令，如图 1-1-4 所示。

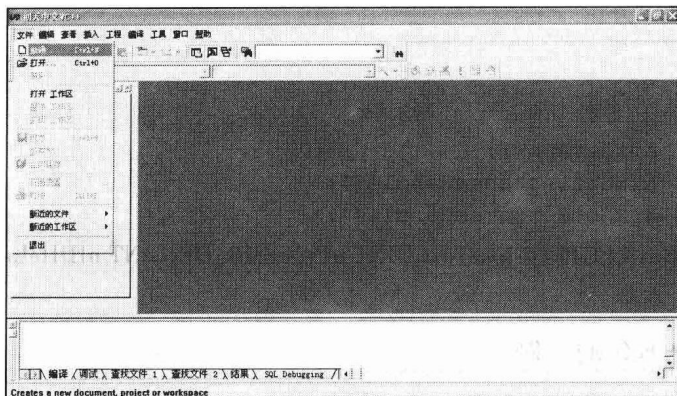


图 1-1-4 选择“文件”→“新建”命令

② 在弹出的“新建”对话框（见图 1-1-5）中选择“工程”选项卡。

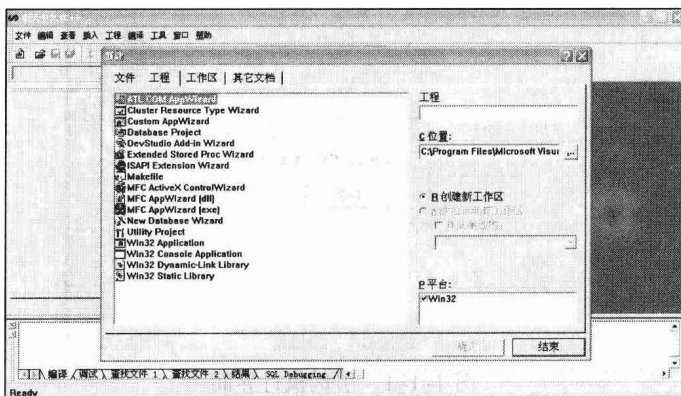


图 1-1-5 “新建”对话框界面

③ 按图 1-1-6 所示输入相关值，如选择“MFC AppWizard[exe]”选项，在“工程”文本框中输入工程名称“hello”，在位置下拉框中选择工程在磁盘中的位置，选择“创建新工作区”单选按钮，选择“平台”复选框，并选择其下的“Win32”复选框表示此程序在 32 位操作系统下运行。

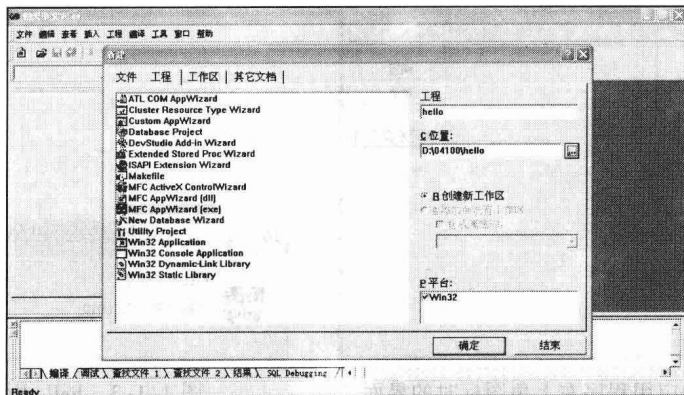


图 1-1-6 选择工程的相关参数界面

④ 单击“确定”按钮，进入自动创建框架的第1步，如图1-1-7所示。在应用程序类型中选择“基本对话框”单选按钮（因为引例是基于对话框的应用程序），单击“下一个”按钮。

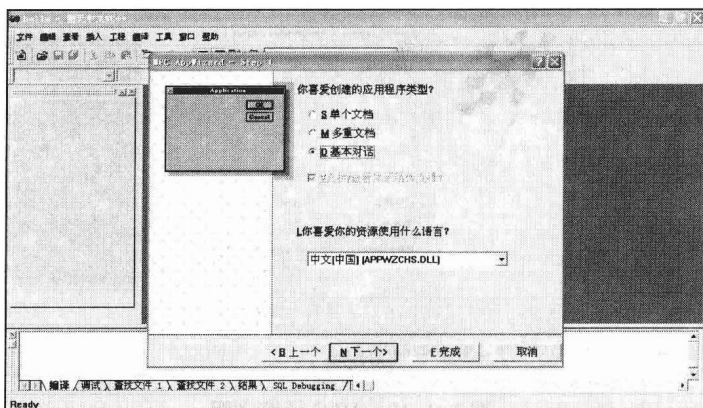


图 1-1-7 MFC 导航之一

⑤ 进入自动创建框架的第2步，如图1-1-8所示。根据自己的想法输入标题对话，单击“下一个”按钮。

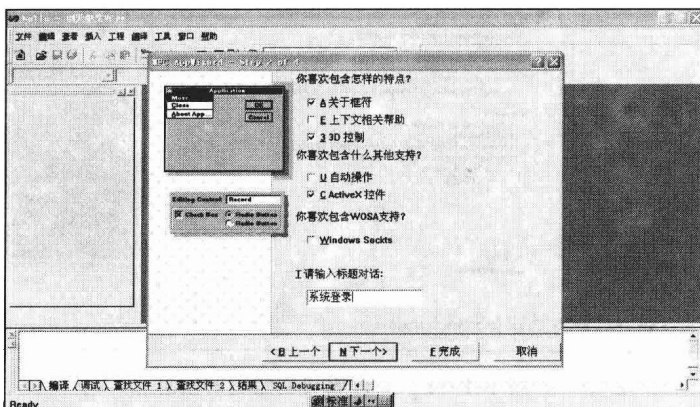


图 1-1-8 MFC 导航之二

⑥ 进入自动创建框架的第3步，如图1-1-9所示，单击“下一个”按钮。

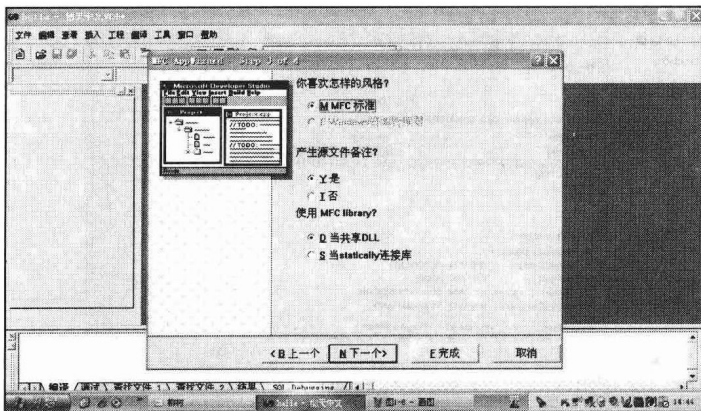


图 1-1-9 MFC 导航之三

⑦ 进入自动创建框架的第4步,如图1-1-10所示,单击“完成”按钮。

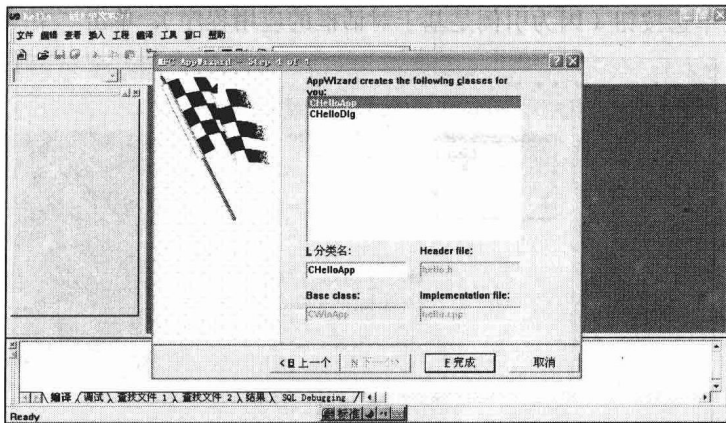


图 1-1-10 MFC 导航之四

⑧ 工程建立完成,显示图1-1-11所示的对话框,单击“确定”按钮。

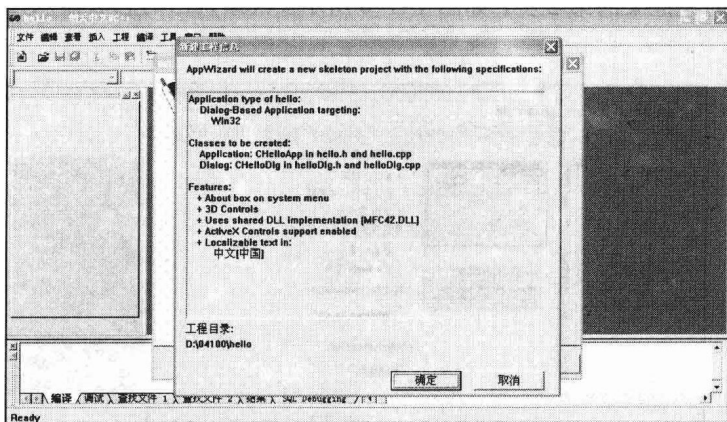


图 1-1-11 工程成功创建后的信息

⑨ 进入设计界面,插入一个控件——按钮,选中该按钮,右击并选择“建立类向导”命令,显示图1-1-12所示的对话框,单击“Add Function”按钮。

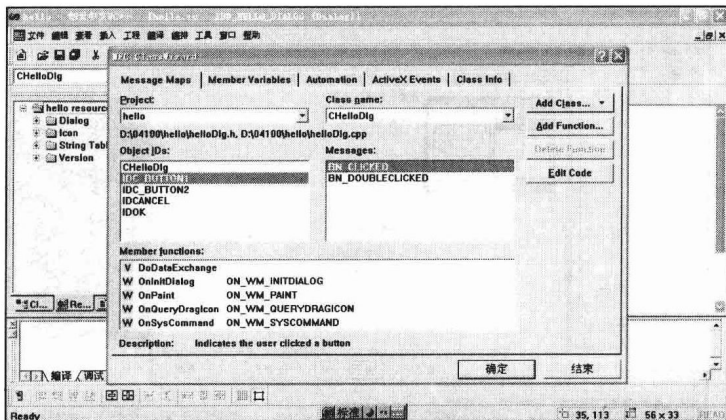


图 1-1-12 设计界面

⑩ 单击“Add Function”按钮后的界面如图 1-1-13 所示,单击“Edit Code”按钮。

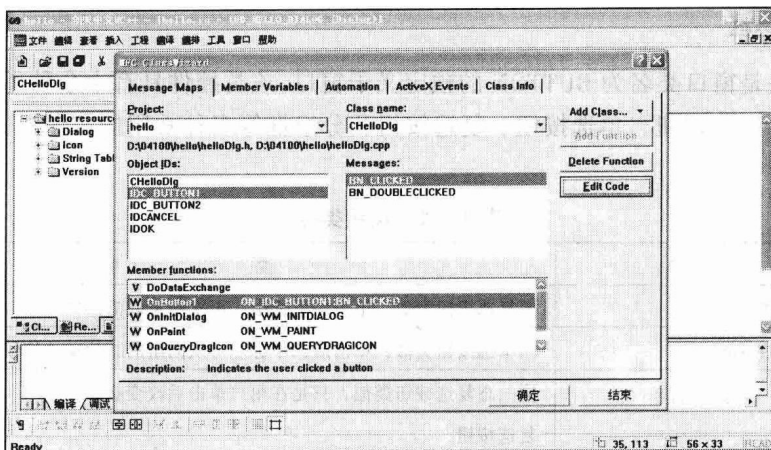


图 1-1-13 进入代码设计前界面

⑪ 按图 1-1-14 所示输入代码。

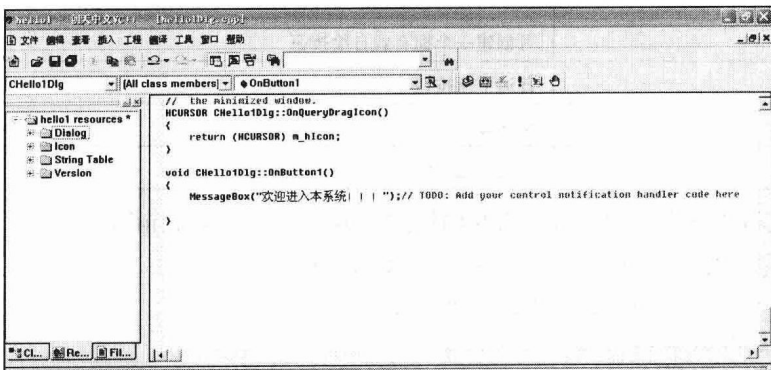


图 1-1-14 代码设计界面

⑫ 选择“文件”→“保存”命令。然后,按图 1-1-15 所示,选择“编译”→“构建”命令,编译成功后,单击“!”图标,执行程序。

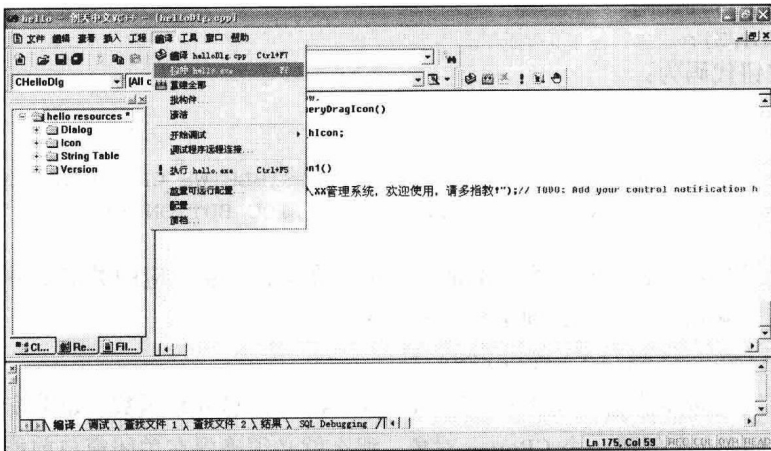


图 1-1-15 编译界面