

21世纪高等学校规划教材 | 计算机应用



Visual Basic程序设计 + Access 2007数据库 应用系统开发案例教程

曹风华 张利军 主编
徐 军 李翠梅 副主编



清华大学出版社

- ❖ 教学目标明确，注重理论与实践的结合
- ❖ 教学方法灵活，培养学生自主学习的能力
- ❖ 教学内容先进，强调计算机在各专业中的应用
- ❖ 教学模式完善，提供配套的教学资源解决方案
- ❖ 可提供教学资料：<http://www.tup.tsinghua.edu.cn>

清华大学出版社数字出版网站

WQBook  
www.wqbook.com

ISBN 978-7-302-27765-1



9 787302 277651 >

定价：26.00元

Visual Basic程序设计 + Access 2007数据库 应用系统开发案例教程

曹风华 张利军 主编
徐 军 李翠梅 副主编

内 容 简 介

本书以程序设计与数据库应用为主要内容,以案例驱动的方式着重介绍 Visual Basic 与 Access 的基本知识与基本应用,难易适度、深入浅出,便于学生学习与掌握。本书以数据库应用开发为主线,介绍 Visual Basic 编程、数据库基础知识、Access 应用基本知识以及使用 Visual Basic 开发 Access 应用,每章配有较为典型的习题以帮助学生对各知识点的学习、复习。

本书可作为财经类高等院校计算机基础教育程序设计与数据库应用课程的教材,也可作为培训教材和广大计算机爱好者的自学参考用书。

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 Basic 程序设计+Access 2007 数据库应用系统开发案例教程 / 曹风华, 张利军主编. —北京: 清华大学出版社, 2012.2

(21 世纪高等学校规划教材·计算机应用)

ISBN 978-7-302-27765-1

I. ①V… II. ①曹… ②张… III. ①BASIC 语言-程序设计-高等学校-教材 ②关系数据库-数据库管理系统, Access 2007-高等学校-教材 IV. ①TP312 ②TP311.138

中国版本图书馆 CIP 数据核字(2011)第 280361 号

责任编辑: 闫红梅 薛 阳

封面设计: 傅瑞学

责任校对: 时翠兰

责任印制: 何 芊

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

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

印 装 者: 北京兆成印刷有限公司

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 17

字 数: 413 千字

版 次: 2012 年 2 月第 1 版

印 次: 2012 年 2 月第 1 次印刷

印 数: 1~4000

定 价: 26.00 元

编 委 会

编委会主任：刘文清

副主任：杜金柱 赵俊岚 乔晓华

委员：（按姓氏拼音排序）

陈连锁	杜金柱	韩 勇	刘文清	马鹏烜
乔晓华	舒力迪	唐建平	王 彪	徐 军
张凯文	张学众	赵俊岚		



前言

本书是财经类院校计算机基础教学改革教材之一。根据教育部高等教育司组织制定的《中国高等院校计算机基础教育课程体系》(简称 CFC)最新教学改革的要求,结合当前财经类专业计算机基础教学“面向应用,加强基础,普及技术,注重融合,因材施教”的教育理念,我们特别在教学体系的设计上做了大胆改革,即由原来的“计算机文化基础”、“Visual FoxPro 数据库及其应用”两门课程,新建成相对独立但又自成体系的 6 大部分,包括计算机应用基础、程序设计与数据库应用、多媒体应用技术、网页制作、计算机系统维护、Excel 在经济管理中的应用等,并组织相关骨干教师本着“任务驱动、案例贯穿”的原则,编写了这套计算机基础教学系列教材。本套教材力求突出案例驱动的教学思想,一方面,充分尊重和符合学生的认知规律;另一方面,可以和各专业后续课程进行有效对接,使计算机及网络技术能更好地运用于经济管理领域,为学生在本专业领域中实现二次开发奠定较好的基础。

本书围绕着以 Visual Basic 开发数据库应用程序这一主线,着重介绍 Visual Basic 基本编程知识,控件、数据库基本概念、Access 基本知识与应用等。本教材难度适中,便于学习掌握。通过本教材的学习,学生可以利用 Visual Basic 开发一些小型的数据库应用程序。

本书共 11 章,第 1~6 章主要介绍了 Visual Basic 编程基本知识,包括 Visual Basic 简介、Visual Basic 编程基础、Visual Basic 窗体与常用控件、Visual Basic 界面设计;第 7 章介绍了数据库基本知识;第 8~10 章介绍了 Access 基本知识,包括 Access 创建数据库与数据表、Access 查询应用、Access 的窗体设计和报表设计;第 11 章介绍了利用 Visual Basic 开发数据库应用相关知识。本教材结构合理,前后联系紧密,适合学生学习。

本书由曹风华、张利军任主编,徐军、李翠梅任副主编。徐军编写了第 2 章和第 4 章,李翠梅编写了第 1 章和第 3 章,常桂英编写了第 5 章和第 6 章,曹风华编写了第 8 章、第 9 章,张凯文编写了第 10 章,张利军编写了第 7 章和第 11 章。全书由曹风华统稿。

在本教材的编写过程中,内蒙古财经学院赵俊岚教授、乔晓华教授给予了大力支持和帮助。

由于学识、水平有限,书中的疏漏、不当之处请各位读者指正。

编者

2011 年 10 月

第 1 章 Visual Basic 概览	1
1.1 Visual Basic 概述.....	1
1.1.1 Visual Basic 简介.....	1
1.1.2 Visual Basic 的安装和启动.....	2
1.1.3 Visual Basic 的特点.....	3
1.2 Visual Basic 系统集成开发环境界面.....	4
1.3 VB 常用术语.....	8
1.3.1 窗体、控件和对象.....	8
1.3.2 属性、事件和方法.....	9
1.3.3 工程、模块和程序.....	10
1.4 设计 Visual Basic 应用程序的步骤.....	10
1.5 窗体.....	13
1.5.1 窗体的基本属性、事件、方法.....	13
1.5.2 多窗体处理.....	16
1.5.3 MDI 窗体.....	19
本章小结.....	20
习题 1.....	20
第 2 章 Visual Basic 程序设计基础	22
2.1 基本数据类型.....	22
2.2 常量与变量.....	24
2.2.1 常量.....	24
2.2.2 变量.....	25
2.3 运算符和表达式.....	28
2.4 常用内部函数.....	30
2.4.1 数学函数.....	30
2.4.2 字符函数.....	32
2.4.3 日期时间函数.....	34

第 1 章

Visual Basic 概览

Visual Basic (以下简称 VB) 是美国微软公司推出的在 Windows 操作平台上广泛使用的一种可视化程序设计语言, 使用 VB 可以方便快捷地开发 Windows 应用程序。在深入学习 VB 编程之前, 本章先介绍 VB 的特点、集成开发环境及面向对象的基本概念。

1.1 Visual Basic 概述

1.1.1 Visual Basic 简介

知识要点分析

Visual 的意思是“可视的”, 即直观的编程方法。在 Visual Basic 中引入了控件的概念, 如按钮、标签、文本框和复选框等。Visual Basic 把这些控件模块化, 并且每个控件都有若干属性用来控制其外观和行为。这样, 不需要编写大量代码去描述界面元素的外观和位置, 只要简单地把控件加到窗体上即可。

Basic 是指 BASIC (Beginners All-Purpose Symbolic Instruction Code) 语言, 是一种在计算机技术发展史上应用得最为广泛的语言之一。

Visual Basic 在原有 BASIC 语言的基础上进一步发展, 既继承了 BASIC 语言编程的简便性, 又具有 Windows 的图形窗口工作环境; 既是一种可供非计算机专业的设计人员学习和掌握 Windows 编程的最简单易学的程序设计语言, 同时又是一种可供专业程序开发人员开发 Windows 应用程序的设计语言。如果学过 BASIC 语言, 那么现在学习 Visual Basic 语言就变得更加容易了。

Visual Basic (VB) 是 Microsoft 公司在 1991 年推出的基于 Windows 环境的软件开发产品。从 1.0 版到现在的 6.0 版, Visual Basic 在广泛的使用中不断发展和完善。目前, Visual Basic 使用较多的版本是 Visual Basic 6.0, 且有 Visual Basic 6.0 中文版。

Visual Basic 6.0 有 3 种不同的发行版本, 以满足不同开发目的需要。

1. Visual Basic 学习版 (Learning Edition)

这个版本包括所有的内部控件, 以及网络、选项卡和数据绑定控件。使用学习版能方便地开发 Windows 95/98 和 Windows NT 4.0 下的应用程序。

2. Visual Basic 专业版 (Professional Edition)

专业版包括学习版的全部功能, 以及附加的 ActiveX 控件、IIS 应用程序设计器、集成

的可视化数据工具和数据环境、ActiveX 数据对象和动态 HTML 页面设计器等。专业版为专业编程人员提供了一套功能齐全的开发工具。

3. Visual Basic 企业版 (Enterprise Edition)

企业版包括专业版的全部功能，并带有 Back Office 工具，如 SQL Sever、Microsoft Transaction Server、Internet Information Server 和 Visual SourceSoft 等，使用企业版能够开发出功能强大的应用程序。

本书内容的讲解以 VB 6.0 中文企业版为准，系统全面地介绍 VB 6.0 版的数据类型、常用标准函数、语句、函数和过程、文件、标准控件、数据库管理等内容。

1.1.2 Visual Basic 的安装和启动

知识要点分析

1. VB 的安装

将 VB 6.0 的安装光盘放入光驱，执行安装光盘上的 Setup.exe 程序，启动 VB 的安装向导，按照安装向导的指示逐步完成 Visual Basic 6.0 的安装。

在安装的过程中要求用户输入产品的 ID 号 and 对应的安装方式，安装程序为用户提供了两种选择：即选择“典型安装”，则只安装最典型的组件，安装过程无须用户干预；如选择“自定义安装”，则会打开“自定义安装”对话框，用户可以根据需要有选择地安装需要的组件。单击“继续”按钮后，安装程序将复制文件到计算机硬盘中，复制结束后重新启动计算机完成 VB 6.0 的安装。

计算机重新启动后，安装程序将自动打开“安装 MSDN”对话框，询问用户是否需要安装联机帮助文档 MSDN。MSDN LIBRARY 是开发人员的重要参考资料，包含了容量约 1GB 的编程技术信息，包括示例代码、文档、技术文章、Microsoft 开发人员知识库及开发程序时所需的其他资料。它是 Microsoft Visual Studio 6.0 套件之一，由两张光盘组成。注意：VB 6.0 的联机帮助文档只有在安装了 MSDN 后才可使用。如要安装 MSDN 需要选择“安装 MSDN”复选框，在光驱中放入 MSDN 安装盘后，单击“下一步”按钮，按屏幕提示操作即可完成安装；否则在取消选中“安装 MSDN”复选框后直接单击“下一步”按钮，MSDN 留待以后安装。

2. VB 的启动和退出

VB 安装完成后，在开始菜单的程序组中将多出一个“Microsoft Visual Basic 6.0 中文版”菜单选项，单击其级联菜单中的“Microsoft Visual Basic 6.0 中文版”即可启动 VB。

VB 启动后，首先显示如图 1-1 所示的“新建工程”对话框。

在“新建工程”对话框中，选择“新建”选项卡可以建立新的工程或应用程序，选择“现存”选项卡可以打开原有的工程，选择“最新”选项卡可以打开最近建立或使用过的工程。在“新建”选项卡中选中“标准 EXE”（缺省选项）后，单击“打开”按钮，即可进入如图 1-2 所示的“VB 集成开发环境”的主界面，进行应用程序的开发。



图 1-1 “新建工程”对话框

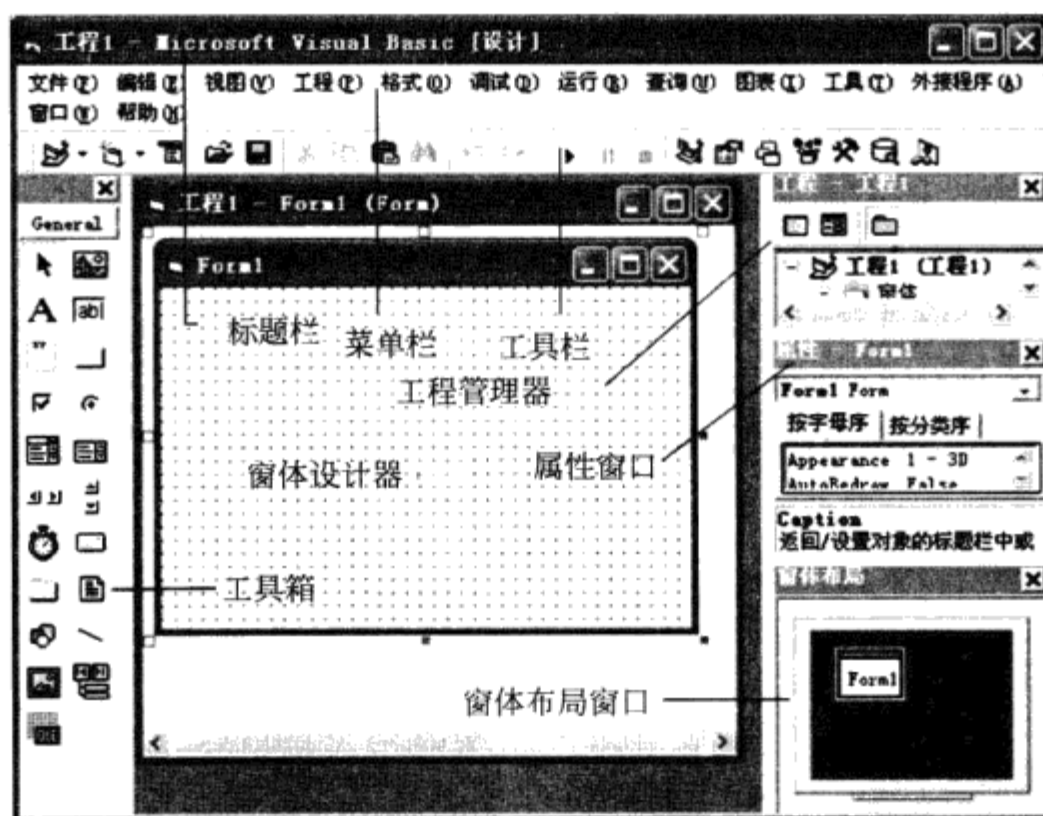


图 1-2 VB 的集成开发环境

在工程的编辑修改结束后，如果要结束 VB 集成开发环境的运行，则单击图 1-2 中 VB 窗口的“关闭”按钮或选择“文件”菜单中的“退出”命令，VB 会自动判断用户是否修改了工程的内容，并询问用户是否进行保存文件或直接退出。

1.1.3 Visual Basic 的特点

📌 知识要点分析

VB 是在原有的 BASIC 语言基础上发展而来的。它具有 BASIC 语言简单易用的优势，

同时增加了面向对象和可视化程序设计语言的特点。

1) 可视化的设计平台

VB 提供可视化的设计平台, 把 Windows 界面设计的复杂性“封装”起来, 开发人员不必再为界面的设计而编写大量程序代码, 只需用系统提供的工具在屏幕上“画出”命令按钮、文本框等各种对象。VB 系统将自动产生界面设计代码, 编程人员只需编写实现程序功能的那部分程序代码, 从而大大提高了程序设计的效率。

2) 面向对象的设计方法

VB 应用面向对象的程序设计方法 (Object oriented program, OOP), 把程序和数据“封装”起来成为一个对象, 每个对象都是可视的, 并为每个对象赋予应有的属性。

3) 事件驱动的编程机制

VB 通过事件来执行对象的操作, 通常由用户操作引发某个事件来驱动完成某种功能。例如, 命令按钮是一个对象, 当用户单击该按钮时, 将产生 (或称“触发”) 一个“单击” (Click) 事件, 而在发生该事件时, 系统将自动执行一段相应程序 (或称为“事件过程”), 用以实现指定的操作和达到运算、处理的目的。

在 VB 中, 编程人员只需针对这些事件编写相应的处理代码 (即事件过程), 这样的代码一般较短, 所有程序既易于编写又易于维护。

4) 结构化的设计程序

VB 是在结构化的 BASIC 语言基础上发展起来的, 加上面向对象的设计方法, 因此是更出色的结构化程序设计语言。

5) 友好的 VB 集成开发环境

VB 提供了易学易用的应用程序集成开发环境。在该集成开发环境中, 编程人员可以设计用户界面、编写代码和调试程序, 直至把应用程序编译成可执行文件, 直接在 Windows 环境下运行。

6) 强大的功能

VB 可以对多种数据库系统进行数据访问。VB 支持对象的链接与嵌入 (Object Linking and Embedding, OLE)、动态数据交换 (Dynamic Data Exchange, DDE)、动态链接库 (Dynamic Link Library, DLL) 及 Active 等技术, 它能够充分利用 Windows 资源, 开发出集文字、声音、图像、Web 等对象于一体的应用程序。

1.2 Visual Basic 系统集成开发环境界面

知识要点分析

集成开发环境是指在一个公共环境里融会了众多不同的功能, 覆盖了开发应用程序设计、编辑、编译和调试等各个方面。VB 的集成开发环境由标题栏、菜单栏、工具栏、工具箱、属性窗口、工程资源管理器、窗体设计器、窗体布局窗口等组成。

1. 标题栏和菜单栏

标题栏中显示有当前应用程序相应的工程名称和当前所处的工作模式以及最小化、最

大化（或还原）、关闭等按钮。

VB 6.0 集成开发环境有三种工作模式：设计模式、运行模式和中断模式，这三种模式分别表示当前应用程序的三种工作状态。如标题栏中显示信息为

工程 1—Microsoft Visual Basic [设计]

方括号中的“设计”表明当前的工作状态是“设计阶段”。随着工作状态不同，方括号内的信息也随之改变，可能会是“运行”或“Break”，分别代表“运行阶段”或“中断阶段”。

菜单栏中显示有“文件”、“编辑”、“视图”、“工程”等多个菜单，每一个菜单中又包含了大量的菜单命令，辅助用户进行应用程序的开发、编译和调试。单击菜单栏中的菜单名，即可打开下拉菜单，进行菜单命令的选择。

2. 工具栏和数字显示区

在菜单栏下面是工具栏，工具栏上提供了许多常用命令的快速访问按钮，单击某个按钮即可完成对应的操作。默认情况下，VB 集成开发环境中显示的工具栏是标准工具栏。用户可通过视图菜单中的工具栏命令调整各种工具栏的显示和隐藏，也可通过鼠标拖动将工具栏拖动到 VB 窗口的任意部分。

工具栏右侧是一个数字显示区，左数字区用来显示当前对象的坐标位置（默认窗体工作区的左上角为坐标原点，水平向右为横坐标轴正向，垂直向下为纵坐标轴正向），右数字区用来显示当前对象的大小，即对象的高度和宽度。

3. 工具箱

在新建或打开“标准 EXE”工程时，VB 将打开标准工具箱，在标准工具箱中提供了一个指针和 20 个标准控件（也称为内部控件）。另外，VB 还提供了大量的 ActiveX 控件，这些控件可以通过特定的方法添加到工具箱中。

向工具箱中添加控件的步骤为：

(1) 在工具箱任意空白处单击鼠标右键，在弹出的快捷菜单中选择“部件”命令，将会弹出如图 1-3 所示的“部件”对话框。

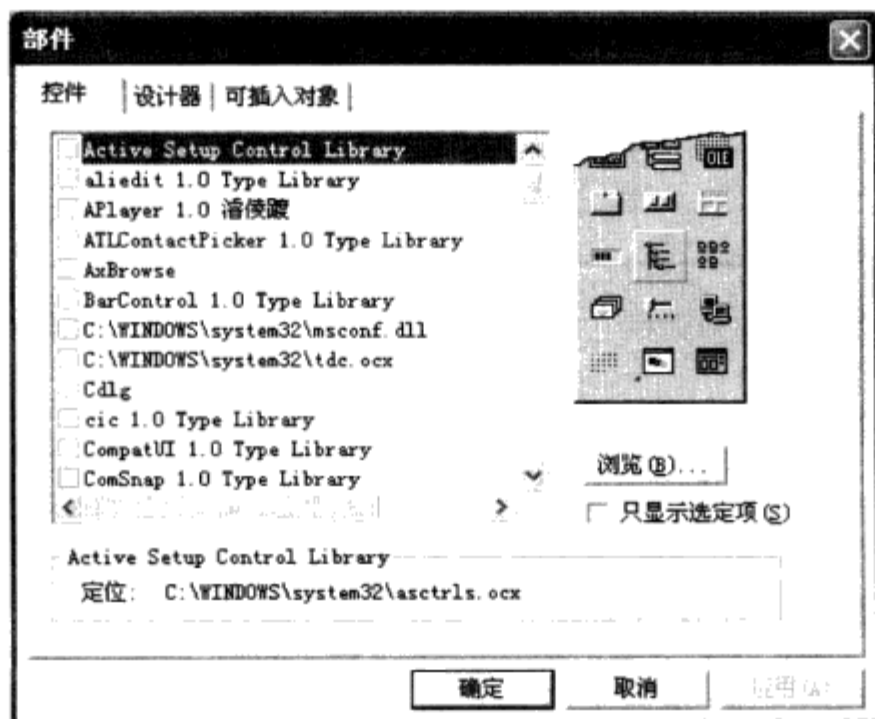


图 1-3 “部件”对话框

(2) 在打开的“部件”对话框中，将自己需要的控件加上选中标记，然后单击“确定”按钮后退出，所选的控件即可添加到工具箱中。

要删除工具箱中的 ActiveX 控件，只要按照上述方法去掉选中标记即可。

注意：内部控件无法从工具箱中删除。

在 VB 中还可以定义选项卡来对工具箱上的控件进行重新的组织安排，以形成一个新的工具箱，方法为：

(1) 在工具箱的空白处单击鼠标右键，在弹出的快捷菜单中选择“添加选项卡”命令。

(2) 在打开如图 1-4 所示的“新选项卡名称”对话框中输入选项卡的名称，如 abc，然后单击“确定”按钮后退出。

(3) 使用鼠标左键将所需的控件拖动到创建的选项卡 abc 上，释放鼠标左键，即可完成新工具箱的创建。以后，通过单击选项卡名称就可以在不同的工具箱之间进行切换。

4. 窗体设计器窗口

窗体设计器窗口主要用来设计应用程序的用户界面，如设计窗体的外观，在窗体上添加控件、图形，移动控件，改变控件大小等。一个应用程序可以拥有多个窗体，每个窗体必须有一个唯一的标识名称，VB 在默认情况下分别以 Form1, Form2, …命名窗体。

窗体工作区中布满了对齐用的小点，如果想清除这些小点或改变点与点之间的距离（即改变定位网格线的距离），可选“工具”菜单中的“选项”命令，然后在“选项”对话框的“通用”选项卡中进行调整。

说明：在“选项”对话框中还可以设置 VB 开发环境中的各种属性及各种窗口的格式，如所用的字体、字号、颜色等。

5. 工程资源管理器窗口（简称工程窗口）

工程资源管理器窗口如图 1-5 所示，它以树状层次结构方式列出了当前工程（或工程组）中的所有文件，并对工程进行管理。在窗口中，双击某个文件图标，即可打开相应的文件。

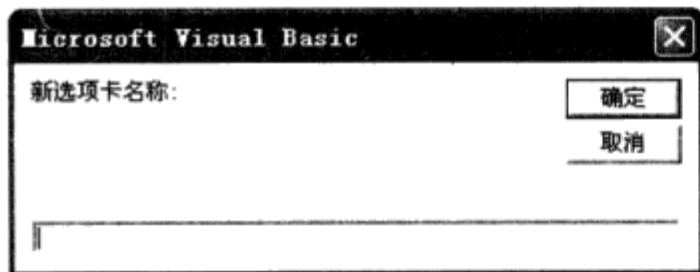


图 1-4 新选项卡名称对话框



图 1-5 工程资源管理器窗口

在工程资源管理器窗口的标题栏下方有“查看代码”、“查看对象”和“切换文件夹”3 个工具按钮。单击“查看代码”按钮可打开代码窗口，以便编辑和查看程序代码；单击“查看对象”按钮可打开窗体设计器窗口，以便查看和编辑正在设计的窗体；单击“切换文件夹”按钮可显示或隐藏包含对象文件夹中的个别项目列表。

6. 属性窗口

属性窗口通常位于工程资源管理器窗口的下方。单击工具栏上的“属性窗口”按钮，或按 F4 键，或选择“视图”菜单中的“属性窗口”命令，均可打开如图 1-6 所示的属性窗

口。在程序设计阶段，可通过属性窗口修改程序运行时各对象属性的初始值，调整对象的外观和相关数据。

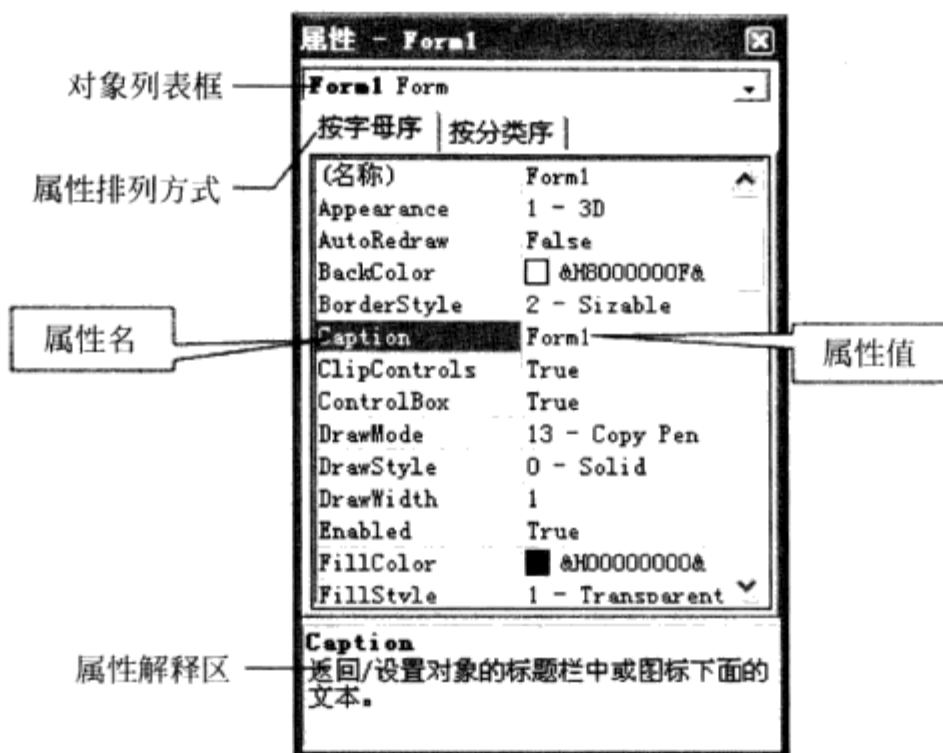


图 1-6 属性窗口

属性窗口由以下 4 部分构成。

(1) 对象列表框：用于标识当前选定对象的名称以及所属的类。单击其右端的下拉箭头，可列出当前窗体以及所包含的全部对象的名称列表，可从中选择要更改其属性的对象。

(2) 属性排列方式：可通过“按字母序”选项卡和“按分类序”选项卡分别显示当前对象的各种属性。

(3) 属性列表：属性列表的第一列为当前选定对象的所有属性的名称，第二列为对应各属性的属性值。通过在任一属性的属性值上单击，即可重新设置该属性的属性值。值得注意的是，不同的属性，其属性值的设置方法也可能各不相同。有的属性值可直接输入，有的则必须借助于右侧显示的省略号按钮或下拉箭头式按钮才能输入。出现省略号按钮的，单击该按钮，将会弹出新的对话框，在该对话框中完成对应属性的设置；出现下拉箭头式按钮的，单击该按钮，在属性值列表中选择相应属性值。

(4) 属性解释区：显示所选属性的简单说明。

使用属性窗口修改控件属性的一般方法为：在窗体设计器中，单击选中想要修改属性的控件，然后使用属性窗口中的属性列表对相应的属性进行重新设置。

7. 代码编辑器窗口（简称代码窗口）

代码编辑器窗口用于编写和修改各种通用过程代码和事件过程代码。双击窗体的任何地方，或选择“视图”菜单中的“代码窗口”命令，或单击工程窗口的“查看代码”按钮均可打开如图 1-7 所示的代码编辑器窗口。

代码编辑器窗口中由对象下拉列表框、过程下拉列表框和代码区构成。对象下拉列表框列出了当前窗体以及所包含的所有对象名。在过程列表框中列出了前面所选对象的所有的事件名。代码区用来编辑和修改代码。它具有以下三种特性：自动列出成员特性、自动

显示快速信息和自动语法检查。

所谓自动列出成员特性，是指在代码窗口中，当要输入控件的属性和方法时，在控件名后输入小数点，VB 就会自动显示一个包含了该控件的所有属性和方法的下拉列表，并且如果输入属性名或方法的前几个字母，系统会自动检索，缩小查找的范围，并显示出需要的属性。编程人员可以从列表中双击选择所需的内容，从而快速地完成内容的录入。

所谓自动显示快速信息，是指在代码窗口中，当输入合法的 VB 语句或函数名后，VB 会在当前行的下方自动显示出该语句或函数的语法，辅助用户完成语句或函数的输入，避免不必要的语法错误。

自动语法检查是指在用户输入某行代码回车后，系统就会进行语句语法检查。如果其中包含错误，系统会显示警告提示框，同时将对应的语句变为红色。

代码窗口有两种查看视图，使用代码区左下角的“过程查看”按钮和“全模块查看”按钮可在这两种视图之间切换。单击“过程查看”按钮一次只能查看一个过程，单击“全模块查看”按钮可查看程序中的所有过程。

8. 窗体布局窗口

窗体布局窗口如图 1-8 所示。窗体布局窗口用来设置应用程序中各窗体的位置。使用鼠标拖动或右击窗体布局窗口中的小窗体图标，可方便地调整程序运行时窗体显示的位置。

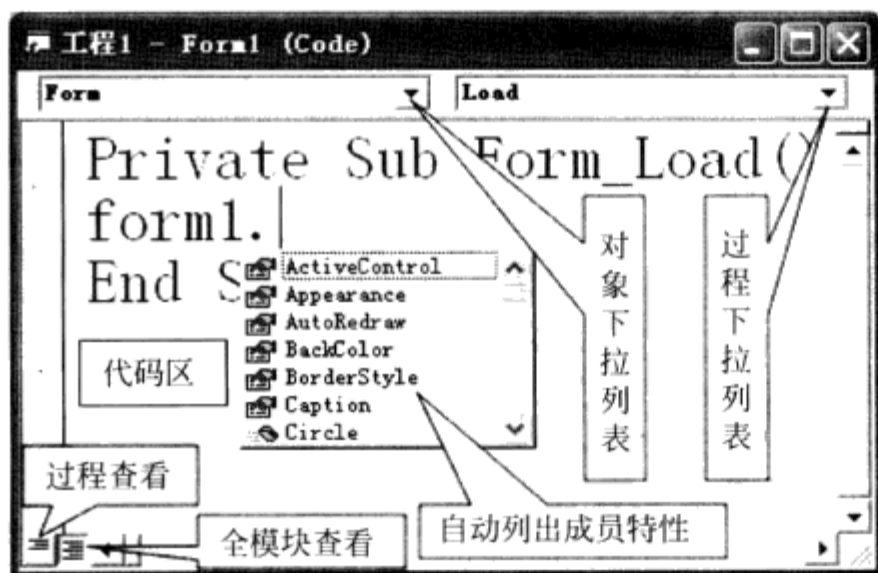


图 1-7 代码编辑器窗口



图 1-8 窗体布局窗口

1.3 VB 常用术语

知识要点分析

1.3.1 窗体、控件和对象

在 VB 中，窗体是为用户设计应用程序界面而提供的窗口。它是多数 VB 应用程序设计界面的基础。它相当于一块画布，应用程序界面会全部在此画出，而无须编写任何有关界面的程序代码。设计中使用工具箱在窗体上画出的各图形对象，如标签、命令按钮、文

本框、列表框等统称为控件 (Control)，而窗体和控件均称为对象。

在 Visual Basic 6.0 中，用户无须用大量对象命令来定义对象的代码、属性和数据块，就可生成用户自定义的 VB 对象。因此，对象不仅包括窗体和控件，而且 VB 中许多可访问的实物也称为对象。

1.3.2 属性、事件和方法

在可视化编程中，VB 的窗体与控件都是程序被操作的对象。这些对象都有自己的事件、属性和方法。

1. 事件 (Event)

所谓事件是由 VB 预先定义的、能够被对象识别的动作。在 VB 中对象与应用程序的往来是通过事件来进行的。对象的事件是固定的，不能定义新的事件，所以，VB 为每个对象提供了丰富的事件，这些事件足以满足 Windows 中大多数操作的需要。

对每个对象，可能会有多个事件产生，当一个对象识别到某一事件发生时，就去响应事件的程序代码段，该代码段称为事件过程。

事件过程是用来完成事件发生后所要做的工作。如应用程序中有一命令按钮，该命令按钮用于完成结束应用程序的任务，则编程者只需在命令按钮的单击事件中写入结束语句即可。当应用程序运行时，单击命令按钮，该按钮就接收到 Click 单击事件，此事件响应事件过程的结束语句，程序运行结束。

不同的对象能够识别的事件是不同的，如窗体能识别单击和双击事件，而计时器只能接收 Timer 事件。

一般地，一个对象能识别多个事件，但在设计应用程序时不一定对每个事件编写事件过程。在程序设计中，编程者根据程序的实际需要，对必须响应的事件编写相应的事件过程，而不必理会的事件则不需要编写事件过程。

事件过程的一般格式如下。

```
Private Sub 对象名_事件名称 ()  
:  
程序代码  
:  
End Sub
```

2. 属性 (Properties)

所谓属性指对象的名字、大小、位置和颜色等特性。在 VB 中，属性用来描述对象的状态。每个对象都由若干属性来描述，不同的对象具有不同的属性，如命令按钮有 Caption 属性而无 Text 属性，文本框无 Caption 属性而有 Text 属性。在设计应用程序时，通过改变对象的属性值来改变对象的外观和行为。

属性值的设置或修改有两种方法：一种是通过属性窗口设置，另一种是通过编程的方法在程序运行时来改变对象的属性。这里介绍第二种方法。

在程序代码中设置属性的格式如下。

```
[对象名.]属性名=属性值
```

若对象名缺省则隐含指当前窗体。例如，将文本框 Text1 的 Text 属性设置为“内蒙古财经学院”的命令是

```
Text1.Text = "内蒙古财经学院"
```

3. 方法 (Method)

方法是指 VB 提供的用来完成特定操作的特殊子程序。与事件过程不同，方法不能响应某一事件，只是完成与对象相关联的特殊操作。如显示、隐藏及移动对象等操作在 VB 中分别对应 Show、Hide 及 Move 等可执行的子程序，即方法。用户只需要调用它们来实现这些功能。

方法不是一个独立的实体，它的功能要有对象来体现，不同的对象具有不同的内部方法，所以调用方法应指明是针对哪个对象的。调用的一般格式为：

```
[对象名.]方法 [参数]
```

若对象名缺省则隐含指当前窗体。例如，清除窗体 Form1 的内容的方法是：

```
Form1.Cls
```

1.3.3 工程、模块和程序

工程文件（简称工程）是用来管理构成应用程序对象的所有文件和对象的清单。工程文件的扩展名是 vbp。一个工程一般包括窗体模块和标准模块等类型文件，其中，窗体模块文件的扩展名是 frm，标准模块文件的扩展名是 bas。

在早期的 VB 版本中，一个完整的应用程序只包含一个工程，但在 VB 5.0 以上版本中，一个完整的应用程序可由若干工程形成一个工程组构成。最简单的 VB 程序只有一个工程，且工程中只包含一个窗体。

所谓模块是由若干子程序（即过程）组成的，可以独立编译的程序代码。在 VB 中，程序是模块结构的。

窗体模块文件描述了窗体的各种信息。其中包括窗体和窗体内各控件的大小和位置等界面信息和在窗体内编写的过程、函数等程序代码以及在窗体内定义的变量和注释等。工程文件中的每个窗体都对应一个窗体模块文件。

标准模块文件是存放与特定窗体或控件无关的变量、通用过程和函数等程序代码。

前面介绍了一些 VB 程序设计中常用的术语。如果读者一时还无法掌握，可通过对后面章节的学习，加深对这些术语的理解。

1.4 设计 Visual Basic 应用程序的步骤

应用实例

VB 可视化编程的一般步骤为：首先设计界面，包括建立窗体和利用控件在窗体上创

建各种对象；然后再设置窗体和控件的属性；最后在代码编辑器中编写代码。下面通过实例来说明使用 VB 6.0 开发应用程序应有的步骤。

【实例 1-1】 设计一个程序，运行时界面如图 1-9、图 1-10 所示，单击“英文”按钮，显示“英文”界面（见图 1-9），单击“中文”按钮，显示“中文”界面（见图 1-10）。



图 1-9 “英文”按钮界面

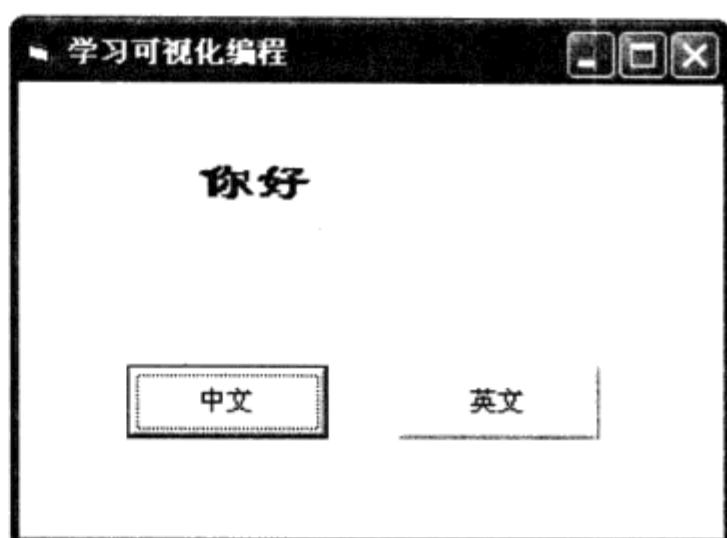


图 1-10 “中文”按钮界面

该应用程序设计步骤如下。

1. 新建工程

在 VB 中开发的每个应用程序都被称之为工程，新建一个工程的方法有 2 种：

(1) 启动 VB 后，系统将显示“新建工程”对话框。在对话框中选择“标准 EXE”选项，单击“打开”按钮，即可新建一个工程。

(2) 使用“文件”菜单中的“新建工程”命令，在“新建工程”对话框中选择“标准 EXE”，单击“打开”按钮，也可新建一个工程。

在新建工程的同时，VB 会自动地创建一个窗体，默认窗体名为 Form1。根据工程的设计需要，还可以使用“工程”菜单中的“添加窗体”命令继续向工程中添加窗体，添加的窗体名依次为 Form2, Form3 等。

本例中，采用第 1 种方法新建一个工程。

2. 设计应用程序界面

通过绘制控件的方法，可以在窗体上添加各种各样的控件。

本例中，在窗体上添加一个标签控件和两个命令按钮，窗体布局如图 1-11 所示。

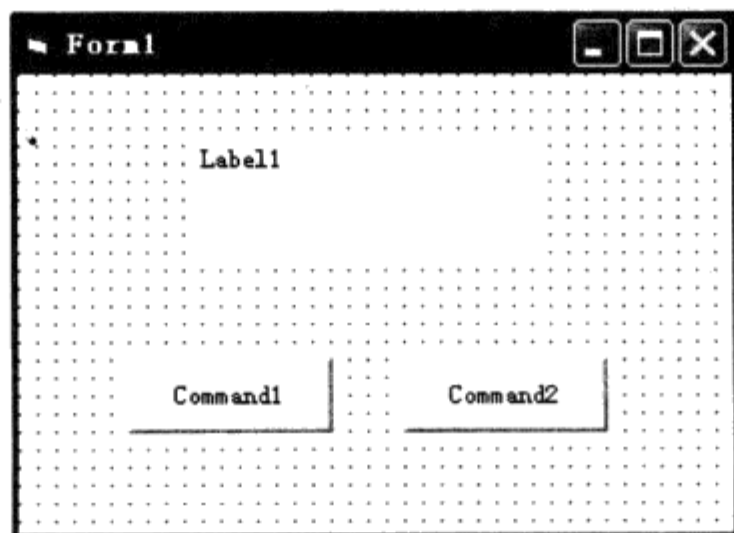


图 1-11 窗体布局界面

3. 设置对象的属性值

窗体和控件的属性很多。在进行应用程序的设计时，应根据实际需要有选择地进行其中某些属性的重新设置。设置属性时，要确保对应的控件被选中。

单击窗体的任意空白区域选中窗体，在属性窗口中设置窗体的 Caption 属性为“学习可视化编程”。

单击选中标签 Label1，设置其 Caption 属性为空（无内容）；通过 Font 属性对话框设置字体为“隶书”2号加粗。

单击选中第一个命令按钮 Command1，设置其 Caption 属性为“中文”。

单击选中第二个命令按钮 Command2，设置其 Caption 属性为“英文”。

4. 编写程序代码

打开代码窗口，在对象下拉列表框中分别选择 Command1，Command2，在事件下拉列表框中选择 Click 事件，为两个命令按钮编写单击事件过程。代码如下。

```
Private Sub Command1_Click()
    Label1.Caption = "你好"      ' 设置在标签中显示你好字样
End Sub

Private Sub Command2_Click()
    Label1.Caption = "Hello"    ' 设置在标签中显示 Hello 字样
End Sub
```

5. 运行、调试和修改工程

单击工具栏上的“启动”按钮或按 F5 键，即可运行工程。单击“英文”按钮，显示图 1-9；单击“中文”按钮，显示图 1-10；单击工具栏上的“结束”按钮结束工程的运行。

如果在程序的运行过程中发生错误，可使用“调试”菜单中的各种命令进行调试。在找到错误的原因后，即可进行修改。继续测试，直到程序运行结果正确无误为止。

6. 保存工程和窗体

使用“文件”菜单中的“保存工程”命令或“工程另存为”命令，可以进行工程的保存。

如果是从未保存过的新建工程，系统将会打开“文件另存为”对话框，在“文件另存为”对话框中，首先进行窗体的保存。窗体文件的扩展名为.frm。当所有的窗体文件保存完毕以后，系统将会弹出“工程另存为”对话框，提示用户进行工程的保存。工程文件的扩展名为.vbp。

在进行文件的保存时，还存在一个设置保存路径的问题。由于一个工程中可包含多种文件，所以在保存工程时应将所有类型的文件存放在同一个文件夹中，便于修改和管理程序文件。

本例中只有一个窗体 Form1，因此，只需保存一个窗体文件（如 sl1-1.frm）和一个工程文件（如 sl1-1.vbp），将它们都放置在同一个文件夹下（如 d:\第一章实例）。

7. 工程的编译

使用“文件”菜单中的“生成工程.EXE”命令即可对工程进行编译。编译的结果是产生一个扩展名为.EXE 的可执行文件。该文件在 Windows 环境下可直接运行，无须依赖于

VB 环境的存在。

8. 工程的关闭

使用“文件”菜单中的“移除工程”命令可以关闭当前工程。如果当前工程或当前工程中的某些文件在编辑完成后未进行保存，系统将会提示用户保存对应的文件。建议用户在新建工程之前最好关闭上一个建立的工程，否则会形成工程组。

如果在关闭工程的同时希望退出 VB 的运行，可使用“文件”菜单中的“退出”命令或单击 VB 窗口上的“关闭”按钮。

1.5 窗体

窗体 (Form) 是设计 VB 应用程序的基本平台。窗体本身又是一个对象，它有自己的属性、事件和方法，以便控制窗体的外观和行为。窗体又是其他对象的载体或容器，几乎所有的控件都设计在窗体上。

通常应用程序是从窗体开始执行的。每一个窗体对应于程序的一个窗口。对于一个简单程序，一个窗体已经足够了，但对于一个复杂的程序，也许需要几个、十几个甚至几十个窗体。

1.5.1 窗体的基本属性、事件、方法

知识要点分析

1. 窗体的常用属性

窗体属性决定着窗体的外观和行为。新建工程时，VB 系统会自动建立一个空白窗体，并为该窗体设置默认属性。属性的设置可在属性窗口中完成，有的也可在代码中设置。以下介绍窗体的一些常用的属性。

(1) Name (名称) 属性：指定窗体的名称。在工程中首次创建窗体时默认为 Form1，添加第二个窗体时，其默认名称为 Form2，以此类推。用户可在属性窗口的“名称”栏中设置窗体名，但在应用程序运行时，它是只读的，即不能在应用程序中修改。

(2) Caption (标题) 属性：用于确定窗体的标题。窗体使用的默认标题为 Form1，Form2，…。

(3) BackColor (背景颜色) 和 ForeColor (前景颜色) 属性：这两个属性用于设置窗体的背景色和前景色。

(4) BorderStyle (边框类型) 属性：用于设置窗体的边框风格。共有 6 种属性，如 0——None (无边框)，1——Fixed Single (窗体大小不变且具有单线边框) 等。该属性只能在设计阶段设置。

(5) ControlBox (控制框) 属性：设置窗体是否具有控制菜单。默认值为 True。

(6) MaxButton (最大化按钮) / MinButton (最小化按钮) 属性：设置窗体是否具有最大化按钮和最小化按钮。

(7) **WindowState** (窗口状态) 属性: 设置窗体装载时的状态。有 3 种属性值: 0 (默认) —— 正常状态; 1 —— 最小化状态; 2 —— 最大化状态。

(8) **Picture** (图形) 属性: 设置在窗体中显示的图形。在属性窗口中单击该属性行右端的三点式按钮, 打开一个“加载图片”对话框, 可以从中选择一个合适的图形文件。也可以在应用程序中使用以下语句格式来设置。

```
[对象.]Picture=LoadPicture("文件名")
```

其中, **LoadPicture** 是一个装载图片函数。

(9) **Font** (字体) 属性: 设置在窗体中显示的正文文字的字体。在属性窗口中单击该属性行右端的三点式按钮, 打开一个“字体”对话框, 从中选择相应的选项。

在代码窗口可以使用以下格式引用该对象的属性。

Font Name: 字体名称。

Font Size: 字体大小 (字号)。

Font Bold: 是否粗体, 设置值为 **True** 或 **False**。

Font Italic: 是否斜体, 设置值为 **True** 或 **False**。

Font Strikefhru: 是否删除线, 设置值为 **True** 或 **False**。

Font Underline: 是否有加划线, 设置值为 **True** 或 **False**。

Font transparent: 确定显示的信息是否与背景重叠, 设置值为 **True** (默认值) 时, 表示保留背景, 使前景的文本或图形与背景重叠显示; 当设置属性值为 **False** 时, 背景将被前景的文本或图形覆盖。

(10) **Visible** (可见性) 属性: 设置窗体是否可见。

(11) **AutoRedraw** (自动重画) 属性: 设置在运行时调整窗体大小或被另一对象遮住后, 重新显示时窗体是否重画。

(12) **Icon** (图标) 属性: 指定在窗体最小化时显示的图形。

(13) **Left** (左边位置) / **Top** (顶边位置) 属性: 设置窗体运行时距离屏幕左上角的位置 (窗体布局窗口可见)。

(14) **Width** (宽) / **Height** (高) 属性: 设置窗体的宽度和高度。

2. 窗体的事件

窗体作为对象, 能够对事件做出响应。窗体事件过程的一般格式为:

```
Private Sub Form_事件名([参数表])
:
End Sub
```

注意, 在事件过程名中的窗体只能使用 **Form** (如 **Form_Load**), 但在过程内容对窗体进行引用时必须用到窗体名字 (如 **Form1.Caption**)。

与窗体有关的常用事件有如下几种。

(1) **Load** (装载) 事件: 装载窗体时触发 **Load** 事件, 如果这个事件过程 **Form_Load** 存在, 便立即执行它。

启动应用程序时, 自动装载和显示“启动窗体” (单个窗体通常就是启动窗体), 在此

期间会先后触发 Load, Activate 等事件。在启动窗体显示之前触发 Load 事件。对于未被装载的窗体, 如果使用 Load 语句调用该窗体, 或者在其他窗体中引用该窗体的控件, 都会触发 Load 事件。

通常, 窗体的 Load 事件过程是应用程序中第一个被执行的过程, 常用来进行初始化处理。

(2) Unload (卸载)事件: 卸载窗体时触发 Unload 事件。单击窗体上的“关闭”按钮或使用 Unload 语句时可触发该事件。

(3) Activate(活动)/DeActivate(非活动)事件: 当一个窗体变为活动窗体时触发 Activate 事件, 当本窗体不再是活动窗体时触发 DeActivate 事件。

(4) Click (单击) 和 DblClick (双击) 事件: 鼠标单击和双击窗体时触发。

(5) Keypress (按键) 事件: 当按下键盘某个键时, 将触发 Keypress 事件。其事件的过程格式为

```
Private Sub 对象_Keypress(KeyASCII As Integer)
:
End Sub
```

其中参数 KeyASCII 返回所按键的 ASCII 码。例如, 按下 A 键, KeyASCII 的值为 65; 按下 a 键, KeyASCII 的值为 97。

3. 窗体的方法

在事件过程中, 可能使用到的窗体的方法有:

(1) Show 方法: 用于显示窗体, 可将后台的窗体调到前台 (此时将触发 Activate 事件)。如果调用 Show 方法时指定的窗体未被装载, 则 VB 自动装载该窗体 (此时将触发 Activate 事件和 Load 事件)。语法为:

```
窗体对象名.Show
```

(2) Hide 方法: 用于隐藏窗体。

(3) Print 方法: 用于在窗体中显示文本。

(4) Cls 方法: 用于清除运行时在窗体上显示的文本或图形。但不能清除在设计阶段设置的文本和图形。

(5) Move 方法: 用于移动并改变窗体或控件的位置和大小。其格式为:

```
[对象.] Move Left[,Top[,Width[,Height]]]
```

其中, Left 和 Top 参数表示将要移动对象的目标位置的 x, y 坐标; Width 和 Height 参数表示移动到目标位置后, 对象的宽度和高度, 以此改变对象的大小。

(6) Refresh 方法: 用于刷新窗体。

应用实例

【实例 1-2】 设置窗体的初始属性为: 背景为红色; 窗体图标为任意一个 VB 的 .ico 图标文件; 窗体的最大化最小化按钮不可用; 窗体的标题为“欢迎进入 VB 6.0 世界”; 设置窗体上显示的文本字体为“宋体”3 号加粗; 设置窗体启动时距离显示器左边距和上边距

分别为 10 和 20。在窗体上添加名称为 Command1 和 Command2 的命令按钮，更改标题分别为“显示”和“清除”。编写适当的代码，程序运行时单击“显示”按钮，在窗体显示“对 VB 的初步认识”，单击“清除”按钮，清除窗体上的文本。程序的运行界面如图 1-12 所示。

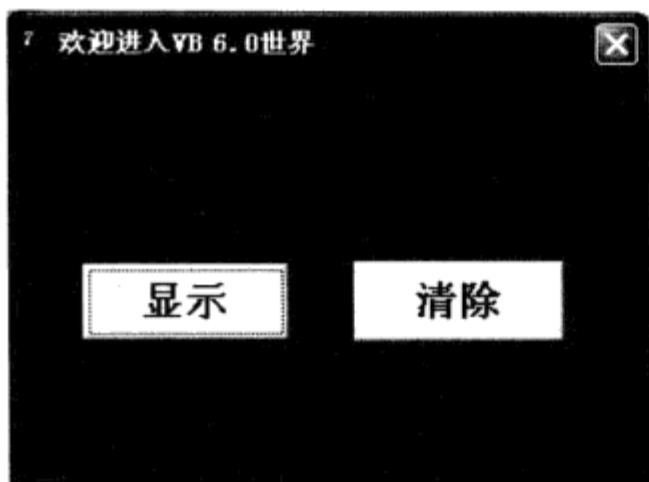


图 1-12 实例 1-2 程序运行界面

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加两个命令按钮。

(2) 设置控件属性。

单击窗体的任意空白区域选中窗体，在属性窗口中设置窗体的 BackColor 属性为红色；通过 Icon 属性对话框在“C:\Program Files\Microsoft Visual Studio\Common\Graphics”目录下寻找需要的图标文件来设置窗体的图标；将 MaxButton 和

MinButton 属性值均设为 false；将 Caption 属性设置为“欢迎进入 VB 6.0 世界”；通过 Font 属性对话框设置字体为“宋体”3 号加粗；将 Left 和 Top 属性值设为 10 和 20。

单击选中第一个命令按钮 Command1，设置其 Caption 属性为“显示”；单击选中第二个命令按钮 Command2，设置其 Caption 属性为“清除”。

(3) 编写程序代码。

打开代码窗口，在对象下拉列表框中分别选择 Command1, Command2，在事件下拉列表框中选择 Click 事件，为两个命令按钮编写单击事件过程。代码如下。

```
Private Sub Command1_Click()
    Print "对 VB 的初步认识"
End Sub
Private Sub Command2_Click()
    Cls
End Sub
```

1.5.2 多窗体处理

📖 知识要点分析

在前面的讨论中只涉及一个窗体，而在实际中特别是较为复杂的应用程序中，单一窗体往往不能满足应用需要，通常需要用到多个窗体 (MultiForm)。在多窗体程序中，每个窗体可以有自已的界面和程序代码，完成不同的操作。

1. 添加窗体

在多窗体程序中，要建立的界面由多个窗体组成。要在当前工程中添加一个新的窗体，可以通过“工程”菜单中的“添加窗体”命令来实现。每执行一次该命令，则建立一个新窗体。这些窗体的默认名称为 Form1, Form2, …

2. 删除窗体

要删除一个窗体，可按以下步骤进行。

在工程资源管理器窗口选定要删除的窗体。选择“工程”菜单中的“移除”命令。

3. 保存窗体

在工程资源管理器窗口中选定要保存的窗体，再选择“文件”菜单中的“保存”或“另存为”命令，即可保存当前窗体文件。

注意：工程中的每一个窗体都需要分别保存。

4. 设置启动窗体

启动窗体是应用程序运行时第一个出现的窗体。在单一窗体程序中，运行程序时会从这个窗体开始执行。对于多窗体程序，默认情况下，应用程序会把设计阶段建立的第一个窗体作为启动，其他窗体必须通过 Show 等方法才能看到。

如果要设置其他窗体为启动窗体，可以采用以下操作。

- (1) 从“工程”菜单中选择“工程属性”命令，打开“工程属性”对话框。
- (2) 选择“通用”选项卡，在“启动对象”列表框中选取要作为启动窗体的窗体。
- (3) 单击“确定”按钮。

5. 窗体处理

前面介绍的窗体的属性和方法，同样适用于多窗体程序设计。

在多窗体程序中，需要在多个窗体间切换，即需要打开、关闭、隐藏或显示指定的窗体，这些都可以通过相应的语句及方法来实现。除了前面介绍的方法外，还有一些常用的语句如下。

1) 装载语句 Load

装载语句的语法为：

Load 对象名

功能为：将窗体或控件载入内存，但不显示。装载窗体时，先完成窗体的初始化，即将窗体属性设置为初始值（在属性窗口设置或 Initialize 事件中提前设置），然后再进行装载。使用 Load 语句将触发窗体的 Load 事件。

执行 Load 语句后，可以引用窗体中的控件及各种属性。

例如：

```
Load Form1          '加载窗体 Form1
```

2) 卸载语句 Unload

卸载语句完成的是装载语句的逆过程，语法为：

Unload 对象名

功能为：从内存中卸载指定的窗体或控件。

例如：

```
Unload Form2       '卸载窗体 Form2  
Unload Me          '卸载当前窗体
```

其中，Me 是系统关键字，用来代表当前窗体。

如果卸载的窗体是程序中唯一的窗体，将终止程序的执行。使用 Unload 语句将触发对象的 Unload 事件。

应用实例

【实例 1-3】 设计一个程序，该程序包含两个窗体，第一个窗体为启动窗体，两个窗体的界面如图 1-13、图 1-14 所示，当单击第一个窗体上的按钮“显示”时，显示第二个窗体，隐藏第一个窗体，当单击第二个窗体上的按钮“返回”时，隐藏第二个窗体，重新显示第一个窗体。



图 1-13 实例 1-3 启动窗体



图 1-14 实例 1-3 第二个窗体

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加命令按钮 Command1，使用“工程”菜单中的“添加窗体”命令添加一个新的窗体，窗体名自动为 Form2，在 Form2 中添加一个命令按钮 Command1。

(2) 设置控件属性。

单击选中 Form1 窗体，通过属性窗口将 Form1 和 Command1 的 Caption 的属性分别设置为“启动窗体”和“显示”。

单击选中 Form2 窗体，通过属性窗口将 Form2 和 Command1 的 Caption 的属性分别设置为“第二个窗体”和“返回”。

(3) 编写程序代码。

单击选中 Form1，打开代码窗口，编写代码：

```
Private Sub Command1_Click()
    Form1.Hide
    Form2.Show
End Sub
```

单击选中 Form2，打开代码窗口，编写代码：

```
Private Sub Command1_Click()
    Form2.Hide
    Form1.Show
End Sub
```

(4) 保存工程。

执行“文件”菜单的“保存工程”命令，对两个窗体模块和工程分别进行保存。

1.5.3 MDI 窗体

知识要点分析

在 Windows 中的应用程序具有多种多样的窗口界面，总的看来有三种形式：单文档界面（Single Document Interface, SDI）、多文档界面（Much Document Interface, MDI）和资源管理器界面。VB 支持这三种分割的文档界面。

单文档界面（SDI）是指在应用程序中，每次只能打开一个文档，如果想要打开新的文档，则必须关闭当前文档。这种风格的窗体不能包含子窗体。例如记事本和写字板。

多文档界面（MDI）是指在应用程序中可同时打开多个文档，每个文档都显示在自己的窗口中，包含文档的子窗体被放置在父窗体中，父窗体为应用程序中所有的子窗体提供工作空间。子窗体为文档窗体，实际上是 MDIChild 属性设置为 True 的普通窗体。一个应用程序可包含多个相似或不相似的 MDI 子窗体。在运行时，子窗体显示在 MDI 父窗体的工作区当中。当子窗体最小化时，其图标显示在 MDI 窗体的工作区中，而不是任务栏中。

创建 MDI 窗体的方法为：使用“工程”菜单中的“添加 MDI 窗体”子菜单，将会弹出“添加 MDI 窗体”对话框，选择“新建”选项卡中的“MDI 窗体”选项，单击“打开”按钮，即可完成 MDI 窗体的添加。此后，程序员可根据需要使用属性窗口对 MDI 的外观和其他属性进行设置。

值得注意的是，在任何时候，一个应用程序只能有一个 MDI 窗体，它与普通窗体不同，除非控件具有 Align 属性（如 PictureBox）或具有不可见界面（如 Timer），否则不能将控件直接放置在 MDI 窗体上。

当加载子窗体时，其父窗体（MDI 窗体）自动加载，而加载 MDI 窗体时，子窗体则不会自动加载，子窗体的加载只能以命令和命令的快捷方式来加载。不能把 MDI 子窗体或 MDI 窗体显示为模式窗体。

应用实例

【实例 1-4】 设计一个程序，该程序包含两个窗体 Form1 和 MDIform1，将 Form1 设置为 MDI 窗体的子窗体，如图 1-15 所示。编写适当的代码，在程序运行时，当单击 Form1 中的按钮“改变 MDI 窗体标题”时，MDIform1 的标题改变为“MDI 窗体”，如图 1-16 所示。

设计步骤如下。

（1）设计程序界面。

新建一个工程，在 Form1 窗体中添加命令按钮 Command1；使用“工程”菜单中的“添加 MDI 窗体”子菜单，在弹出的“添加 MDI 窗体”对话框中选择“新建”选项卡中的“MDI 窗体”选项，单击“打开”按钮，添加 MDI 窗体 MDIform1。

（2）设置控件属性。

通过属性窗口将 Form1 中 Command1 的 Caption 的属性设置为“改变 MDI 窗体标题”，将 MDIChild 属性设置为 True。

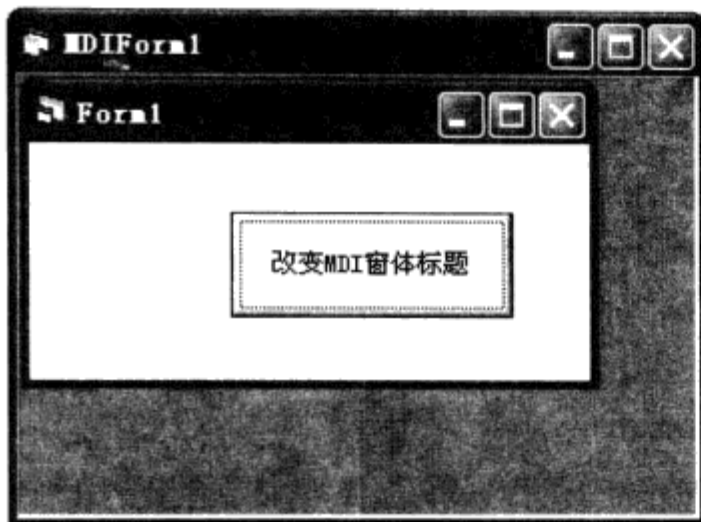


图 1-15 实例 1-4 运行界面 1

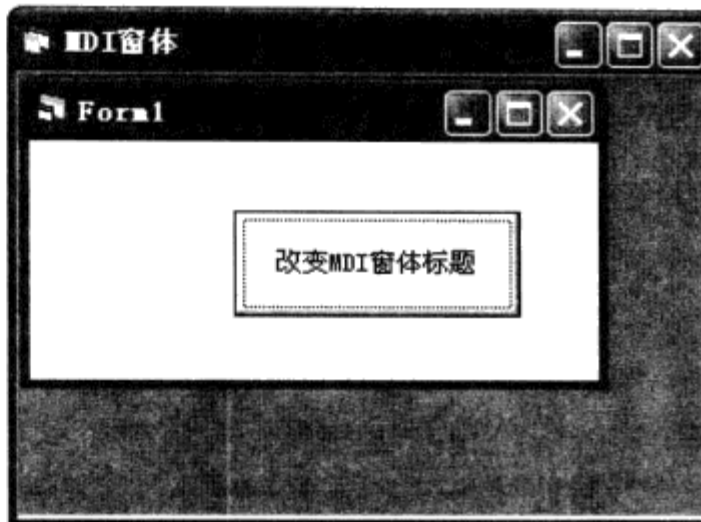


图 1-16 实例 1-4 运行界面 2

(3) 编写程序代码。

单击选中 Form1，打开代码窗口，编写代码：

```
Private Sub Command1_Click()
MDIForm1.Caption = "MDI 窗体"
End Sub
```

(4) 保存工程。

执行“文件”菜单的“保存工程”命令，对窗体 Form1、MDI 窗体 MDIform1 和工程分别进行保存。

本章小结

本章对 VB 6.0 的安装、启动、退出、系统集成开发环境界面及 VB 常用术语做了简要的介绍，并通过简单的例子描述了 VB 应用程序的设计步骤；对设计 VB 应用程序的基本平台窗体（Form）做了详细的介绍，为后续内容的学习打下了基础。

习 题 1

一、判断题

1. VB 用于开发 Windows 环境下的应用程序。()
2. 一个窗体上至少必须有一个控件。()
3. 对于一个窗体，其窗体名和窗体文件名必须相同。()
4. 对象是窗体和控件的总称。()
5. VB 中默认的对象是当前窗体。()
6. 事件是对象能够识别的动作。()
7. 同一个事件的名称在不同的程序中可以不同。()
8. 事件是编程人员定义的。()

9. 对象的事件是不固定的。()
10. 一个应用程序可以有多个窗体。()
11. 一个应用程序可以有多个 MDI 窗体。()
12. 当加载子窗体时, 其父窗体 (MDI 窗体) 自动加载, 而加载 MDI 窗体时, 子窗体则不会自动加载。()
13. 对象的属性设置只能在设计阶段完成。()
14. 保存一个工程至少应保存两个文件。()

二、单选题

1. VB 程序的运行机制是 ()。
A. 可视化 B. 面向对象 C. 面向图形 D. 事件驱动
2. VB 可视化编程有三个基本步骤, 依次是 ()。
A. 设计界面, 建立窗体, 建立对象 B. 创建工程, 设计界面, 保存工程
C. 创建工程, 设计对象, 编写代码 D. 设计界面, 设置属性, 编写代码
3. 窗体设计器用来设计 ()。
A. 应用程序的代码段 B. 应用程序的界面
C. 对象的属性 D. 对象的事件
4. 在 VB 中, 编写程序代码应在 () 中进行。
A. 对象窗口 B. 属性窗口 C. 代码窗口 D. 窗体布局窗口
5. 假设窗体上已有一个控件是活动的, 为了在属性窗口中设置窗体的属性, 预先要执行的操作是 ()。
A. 单击窗体上没有控件的地方 B. 单击任一控件
C. 双击任一控件 D. 双击窗体上没有控件的地方
6. 编制一个简单的 VB 程序, 该程序只有一个窗体, 则该程序有 () 个文件需要保存。
A. 1 B. 2 C. 3 D. 4
7. 通过 () 可以在界面设计阶段直观地调整窗体在屏幕上的位置。
A. 代码窗口 B. 窗体布局窗口
C. 窗体设计器窗口 D. 属性窗口
8. 在设计应用程序时, 通过 () 窗口可以查看到应用程序工程中的所有组成部分。
A. 代码窗口 B. 窗体布局窗口
C. 窗体设计器窗口 D. 工程资源管理器窗口

第2章

Visual Basic 程序设计

Visual Basic 程序是由界面和程序代码两部分构成的，界面是在窗口的，而程序大部分的功能还是要通过编写代码来实现。本章先从 Visual Basic 语言的基础谈起，介绍数据类型、常量、变量、表达式、内部函数、程序后续的学习打好基础。

2.1 基本数据类型

☞ 知识要点分析

在 Visual Basic 中，数据类型控制了数据的内部存储方式，规定数据类型、占用内存空间的大小及能够参与的运算。Visual Basic 中使用数据类型可以有效地使用内存空间，提高程序运行的效率。每个变量、常量或表达式都有一定的数据类型。Visual Basic 提供了丰富的数据类型，可以用来存储各种不同的数据类型。Visual Basic 提供了丰富的数据类型，可以用来存储各种不同的数据类型。Visual Basic 提供了丰富的数据类型，可以用来存储各种不同的数据类型。

Visual Basic 数据类型可分为基本数据类型和用户自定义类型两大类。用户自定义类型的基础，用户自定义类型是由基本数据类型组合而成的数据类型。

1. 数值型数据

数值型数据包括整型、实型和字节型 3 类。

1) 整型数据

整型数是指不带小数和指数符号的数，根据其取值范围的不同，分为短整型 (Integer) 和长整型 (Long)。

2) 实型数据

实型用来存放小数数据，实型数按其取值范围和精确度的不同，分为单精度实型 (Single)、双精度实型 (Double) 和货币型 (Currency)。

3) 字节型数据

字节型 (Byte) 用来存储二进制数据，用 1 个字节存储，不能表示负数，可以用 Byte 型表示。

2. 字符型数据

字符型 (String) 用来存放字符串, 字符型可分为定长字符型和可变长度字符型两种, 变长字符型长度不固定, 它的长度随着所赋值的变化而变化。

3. 逻辑型数据

逻辑型 (Boolean) 也叫布尔型, 只能存储 True 或 False, 适合数据只有两种状态的情况。当把逻辑数据转换成数值型时, True 转换成-1, False 转换成 0。反之当把数值型数据转换成逻辑型数据时, 0 会转换成 False, 其他非 0 的数据则转换成 True。

4. 日期型数据

日期型 (Date) 用于存储日期和时间。日期型数据需用“#”号括起来, 日期型变量如果没有赋值, 系统默认日期为 1899 年 12 月 30 日, 时间为 0:00:00; 如果给日期型变量赋值为数值, 则该数值表示的是天数, 负数表示公元 1899 年 12 月 30 日之前的日期和时间。

5. 变体型数据

变体型 (Variant) 可以存储系统定义的所有数据类型, 它是一种可变的数据类型, 占用的内存比其他类型多, 因此一般不建议使用变体型, 而当用户事先无法预料结果的类型, 可以考虑使用变体型。当定义变量时没有说明数据类型, 则默认变体型。变体型还包括 3 种值: Empty, Null 和 Error。

6. 对象型数据

对象型 (Object) 存储任何类型的对象, 可以引用应用程序中的对象或者其他应用程序中的对象。

表 2-1 列出了各种数据类型的存储空间大小、取值的范围和类型标识符。

表 2-1 Visual Basic 的基本数据类型

数据类型	类型标识	类型符	占字节数	取值范围
字符型	Byte	无	1	0~255
整型	Integer	%	2	-32 768~32 767
长整型	Long	&	4	-2 147 483 648~2 147 483 647
单精度型	Single	!	4	负数: -3.4E38~-1.4E-45 正数: 1.4E-45~3.4E38
双精度型	Double	#	8	负数: -1.79D308~-4.94D-324 正数: 4.94D-324~-1.79D308
货币型	Currency	@	8	-922 337 203 685 477.580 8~ 922 337 203 685 477.580 7
逻辑型	Boolean	无	2	True 或 False
日期型	Date	无	8	100 年 1 月 1 日—9999 年 12 月 31 日
变长字符串	String	\$	字符串长度	0~大约 20 亿个字符
定长字符串	String*size	无	size	0~65 535 个字节 (64KB)
对象型	Object	无	4	任何对象
变体型 (数值)	Variant	无	16	最大可达 Double 型的范围
变体型 (字符)	Variant	无	字符串长度	与变长字符串有相同的范围

2.2 常量与变量

常量是程序执行过程中始终保持不变的量，包括直接常量、符号常量和系统常量。变量是指在程序运行过程中其值可以改变的量。每一个变量都有一个唯一的名字以区别其他变量，变量与常量在内存中均要占据一定的存储空间。

2.2.1 常量

☞ 知识要点分析

1. 直接常量

直接常量是指各种类型的常数值，包括数值常量、字符常量、日期型常量和逻辑型常量。

1) 数值常量

数值常量由正负号、数字和小数点组成，分为整数和小数。小数又分定点数和浮点数，定点数是固定的小数点位置决定了固定位数的整数和小数部分的正数或负数，表示数值的范围比较小；浮点数是用科学记数法表示的十进制数，分单精度浮点数和双精度浮点数，用字母 E 或 D 区分，例如 3.48E6 表示单精度、2.35D-10 表示双精度。

2) 字符常量

字符常量是用半角双引号（" "）括起来的字符序列。例如"abc"、"中国"等都是合法的字符常量，双引号中不包含任何字符，也不包含空格，则表示一个空字符串。字符常量一般赋值给字符型变量。

3) 日期型常量

日期型常量在格式上要求用“#”将日期时间值括起来。例如：#08/28/98#，#1998-08-28 10:30:12 AM #，#18:35:20 PM#等都是合法的日期型常量。

4) 逻辑型常量

逻辑型常量只有 True（真）和 False（假）两个值。

2. 符号常量

符号常量指的是在程序中用符号表示的常量，又可分为系统常量和用户定义的符号常量两种。

1) 用户定义符号常量

当程序中用到的某个数据很长很难记忆或者多次用到，则可以定义一个容易书写的符号来代替它，这个符号就叫符号常量。例如圆周率如果在程序中多次使用，可以使用符号常量来代替它，若要将圆周率的精度提高，则只需修改符号常量的值，而不需要一条一条语句的去查找修改，非常方便。

符号常量需要先声明后使用，声明符号常量的语法为：

```
Const <符号常量名> [As 数据类型] = 表达式
```

符号常量的命名遵循标识符的命名规则，若省略[As 数据类型]，符号常量的类型由表

达式的数据类型决定。表达式可以由数值、字符、运算符等组成，也可以使用之前已经定义过的符号常量。例如：

```
Const PI=3.1415926
Const PI As Single=3.1415926
```

使用已经定义过的 PI 来定义新的符号常量：

```
Const P=2*PI
```

符号常量形式上和变量相似，但本质上仍然是常量，因此只能引用不能被赋值。

2) 系统常量

系统常量是 Visual Basic 在程序运行过程中，由系统自动定义的系统内部的常量，与应用程序的对象、方法和属性一起使用，一般以标志符 VB 打头。

2.2.2 变量

☞ 知识要点分析

1. 变量的命名规则

Visual Basic 通过变量名来引用其存储的数据，变量名属于 Visual Basic 的标识符。标识符是编程时为变量、符号常量、用户定义数据类型、过程、函数等定义的名字。标识符的命名应遵循以下规则。

- 字母或汉字开头，由字母、汉字、数字或下划线组成。
- 最长不能超过 255 个字符。
- 不区分大小写，符号常量一般用大写。

不允许使用下列字符。

- 运算符、标点符号和空格。
- 不能使用 Visual Basic 中的关键字(在 Visual Basic 中有特殊意义的单词、字母组合)。
- 除了最后一个字符外不能包含类型说明符。
- 撇号(')或 Rem 为程序注释的引导不能使用。

2. 变量类型的声明

变量声明就是定义变量的名称和变量的数据类型，有什么样的数据类型就有什么样的变量类型。在 Visual Basic 中有显式声明和隐式声明 2 种声明变量的方法。

1) 显式声明

显式声明变量可以使用 Dim, Static, Public, Private 4 个语句，声明变量的作用范围不同，Dim 语句最为常用。语法如下：

```
Public | Dim | Static | Private 变量名[As 数据类型][, 变量名[As 数据类型...]]
```

说明：

- 方括号表示其内容可以省略。若省略了数据类型，则默认该变量为 Variant 型，变体变量占用内存空间多，有时候它的值可能无效，所以最好给变量定义确定的数据

类型。

- 在用 Dim 语句声明变量后，系统会自动按照变量的数据类型给变量赋值，数值型初值为 0，字符型初值为空串，逻辑型初值为 False。
- 字符型变量有定长和变长之分，在声明定长字符型变量时，用“String*长度”来表示。
- 同时声明多个变量，各变量名以逗号进行分隔。

例如：

```
Dim a As Integer,b As Single      '定义 a 为整型变量, b 为单精度型变量
Dim a,b                          '定义 a 和 b 为变体变量
Dim s As String *12              '定义 s 为 12 位(定长)字符变量
Dim c As string                  '定义 c 为可变长度字符变量
Dim x,y,z As Integer            '定义 x,y 为变体型变量, z 为整型变量
```

可以使用类型声明符来代替“As 数据类型”。变量名和说明符之间不能有空格，例如：

```
Dim x%,y&,z!
```

等价于

```
Dim x As Integer,y As Long,z As Single
```

2) 隐式声明

在 Visual Basic 中可以未经声明，而直接使用变量，这种声明变量的方法叫隐式声明。隐式声明变量有两种方法。

方法一：直接使用。

```
Private Sub Command1_Click()
    '单击命令按钮在窗体上显示"欢迎使用 VB"
    a = "欢迎使用 VB" '隐式声明变量 a
    Print a
End Sub
```

方法二：用类型声明符隐式声明变量。

```
Private Sub Command1_Click()
    '单击命令按钮在窗体上显示数值 1000
    b% = 1000 '用类型声明符隐式声明变量 b
    Print b
End Sub
```

方法一中 Visual Basic 系统会根据赋予变量的值来决定变量的类型。方法二先用类型声明符将变量 b 隐式声明为整型，然后进行赋值，因此限制了给变量 b 赋值的常量必须是整型或者是能自动转换为整型的值。

隐式声明变量的好处是使用起来非常的方便快捷。但隐式声明变量会增加程序调试难度，破坏程序的可读性，一旦将变量名拼错可能会导致难以查找的错误，因此使用变量应先声明后使用；也可以在代码编辑器窗口，从对象下拉列表中选择“通用”，在过程下拉列

表中选择“声明”，在通用声明段使用 Option Explicit 语句来强制对所有使用的变量进行显式声明（界面操作：单击菜单“工具/选项”，选择“编辑器”选项卡，勾选“要求变量声明”）。从而养成一种良好的编程习惯。

3. 变量的赋值

在程序的执行过程中，声明变量后，系统一般根据数据类型赋予变量一个初值。可以为变量重新赋予具体的数据，或者读出其代表的数值，即变量的赋值与引用。赋值语句是程序设计中最基本、最常用的语句。

语法格式：

<变量名> = <表达式>

功能：

计算表达式的值，并将计算结果赋予赋值号左边的变量。

说明：

(1) 表达式可以是单个常量或变量，也可以是由常量、变量、函数、运算符及括号等组成的字符串。

(2) “=”号右侧表达式中的变量必须是赋过值的，否则变量的初值自动取零（变长字符串变量取空字符）。例如：

```
a=1  
c=a+b+5      '变量 b 未赋值，为 0
```

执行后，c 值为 6。

(3) 赋值号(=)不同于数学中的等号，如： $x=x+1$ 在数学中不成立，而在 Visual Basic 中，表示将变量 x 的值+1 再重新赋予变量 x。数学中 $a=b+c$ 等价于 $b+c=a$ ，但 Visual Basic 中， $b+c=a$ 是非法的赋值语句。

(4) 在使用赋值语句赋值时，原则上要求赋值运算符两边数据类型相同，而实际应用中，如果表达式的数据类型系统能够自动转换为变量的数据类型，也能成功赋值，这个值是类型转换后该有的值。但不是所有数据类型之间都可以强制转换，一旦赋值运算符两边数据类型不同且不能转换，则会出现类型不匹配的错误。例如：

```
Private Sub Command1_Click()  
    Dim a As Integer, b As Integer, c As String, d As Date  
    a = 1.5          '转换 1.5 为整型数 2 (四舍五入)，再赋值给 a  
    b = "ABCD"      '出错，类型不符  
    c = 123         '转换为字符串"123"，再赋值给 c  
    d = 2.5         '2.5 转换为日期型数据，整数是日期，小数为时间  
End Sub
```

(5) 数值型变量在赋值的时候如果超出了该类型规定的范围，会提示溢出错误；定长字符型变量赋值时，如果超出了定长，超出的部分截掉显示，不足定长则用空格补足。例如：

```
Dim x As Byte, y As String * 3, z As String * 3
```

```

x = 258           '溢出错误
y = "abcde"     '结果只有 abc 赋值给变量 y
z = "a"         '将"a"赋值给变量 z，可以用 Len 函数来测字符串的长度为 3

```

2.3 运算符和表达式

Visual Basic 提供了丰富的运算符，通过运算符实现简单运算，这些运算符包括算术运算符、字符连接运算符、关系运算符和逻辑运算符。表达式由运算符、常量、变量和函数等组成。

☞ 知识要点分析

多数表达式是由各种类型的运算符和操作数综合而成。在计算表达式的过程中，各种运算必须按一定的顺序依次进行，这种顺序就是运算符的优先级。按照优先级从高到底进行计算，才能够得出正确的表达式结果。表 2-2 显示了 Visual Basic 中的 4 类运算符及相应的运算优先级。

表 2-2 Visual Basic 的运算符分类

运算符种类	优先级	运算符（按优先顺序从高到底）
圆括号	1	()
算术运算符	2	^幂运算，-求负，*乘、/除，\整除，Mod 取余，+加、-减
字符串运算符	3	+、&连接
关系运算符（级别相同）	4	=等于、>大于、>=大于等于、<小于、<=小于等于、<>不等于、like 字符串匹配、is 对象引用比较
逻辑运算符	5	Not（取反），And（与），Or（或）

1. 算术运算符和算术表达式

算术运算符用来进行算术运算。用算术运算符将常量、变量等连接起来组成的式子叫算术表达式。Visual Basic 规定了算术运算符的优先级，计算表达式时，按优先级的高低顺序执行。算术运算符包括+，-，*，/，\，^和 Mod，有几点说明和举例如下。

- -（减号）也可以作为求负运算符，这时-号为单目运算符（只有一个运算对象）。
- \（整除）运算符求一个数除以另一个数的整数商，为双目运算，比如：18\4 的结果为 4。
- ^（乘方）用于求一个数的幂（指数）运算，如：3^4 的结果为 81，也是双目运算。
- Mod（取模）计算一个数除以另一个数后的余数，15 Mod 4 的结果为 3，注意 Mod 运算符左右要有一个空格隔开。

2. 字符连接运算符和字符表达式

字符连接运算符是合并字符串的运算符，有+和&两种。+作为字符串连接运算符时，操作数必须是字符串，否则可能无法确定是进行字符串的连接还是做加法运算，还可能出现类型不匹配无法连接的错误；&的作用也是连接，而且运算符和运算数之间要加空格，

它可以将非字符串型的数据转换成字符串后实现连接。因此在进行字符连接运算时尽量使用 &。

以下是一些连接表达式的例子。

"Visual" + "Basic" 结果为: "VisualBasic"。

10 & 10 结果为: 1010。

10+10 结果为: 20。

"10" + "10" 结果为: 1010。

3. 比较运算符和比较表达式

比较运算符（关系运算符）用来进行比较运算，用比较运算符连接起来的式子叫比较（关系）表达式。比较表达式的结果只能是 True 或 False，当比较表达式表示的比较关系成立时，结果为 True；否则为 False。Visual Basic 中的比较运算符有 <, <=, >, >=, =, <>, Is 和 Like。在使用比较运算符时，需要注意以下几点。

- 比较运算符“=”与赋值运算符“=”不同。赋值运算符的优先级低于比较运算符，例如：语句 b=10=15，先进行 10=15 的比较运算，再将结果为 False 赋值给变量 b。
- 比较运算的对象可以是字符、数值、日期、逻辑型等数据类型，当运算符的两个运算数是相同数据类型时进行比较。数值按大小比较；日期按先后比较，日期早的小于日期晚的；逻辑型 False(0)大于 True(-1)；字符串按 ASCII 码序比较，汉字大于西文；比较时，若第一个字符相等，再比较第二个字符，直到比较出大小或比较完为止。
- Like 比较两个字符串的模式是否匹配。涉及一些通配符的使用，?号表示单个字符、*号表示多个字符、#号表示单个数字、[字符列表]：表示列表中的单个字符、[! 字符列表]：表示列表外的单个字符。例如：

"ab" Like "a?" 判断字符串匹配，结果为 True；

"abcde" Like "a*" 判断字符串匹配，结果为 True；

133 Like "13[6-9]" 判断字符串匹配，结果为 False；

133 Like "13[!6-9]" 判断字符串匹配，结果为 True。

- Is 比较两个对象是否一致，如 Object1 Is Object2。

4. 逻辑运算符和逻辑表达式

逻辑运算符（布尔运算符）进行逻辑运算，用逻辑运算符连接组成的式子叫逻辑表达式。逻辑表达式的结果是 True 或 False。逻辑运算符有 And（与）、Or（或）和 Not（非）等，其中 Not（非）为单目运算。表 2-3 列出了 x 和 y 进行各种逻辑运算的结果。

表 2-3 逻辑运算

x	y	Not x	x And y	x Or y
True	True	False	True	True
True	False	False	False	True
False	True	True	False	True
False	False	True	False	False

在程序中通常用逻辑运算符连接多个关系表达式来表示复合条件。例如：条件为“成绩在 85 分以上的女生或者成绩在 90 分以上的男生”时，可以用以下的逻辑表达式：

成绩 > 85 And 性别 = "女" or 成绩 > 90 and 性别 = "男"

2.4 常用内部函数

Visual Basic 的函数可分为内部函数和用户自定义函数两大类。内部函数也称公共函数，是 Visual Basic 事先编制好的相应的程序，编程时可以直接使用。内部函数包括数学函数、字符函数、日期时间函数、类型转换函数等。每个内部函数都有它特定的功能，每个函数的使用可能都需要考虑函数的三要素：函数名、函数参数和函数的返回值。

函数的使用语法为：

函数名(参数 1, 参数 2, ...)

函数名是系统规定的函数的名称；函数参数的多少根据不同的函数而不同，参数用圆括号括起来，如果有多个参数，则用半角逗号隔开；每个函数都有一个确切的结果，这个结果就是该函数的返回值，在使用函数的过程中需注意函数返回值的数据类型。调用函数一般的方法是将函数的结果直接用 Print 语句输出，或者用函数构成表达式，将表达式的结果赋值给某个变量。在“代码编辑器”窗口输入函数名和“(”后，系统会列出该函数的参数类型和个数等信息，方便用户参考。

2.4.1 数学函数

知识要点分析

数学函数是 Visual Basic 系统提供的进行算术运算的函数，函数参数假设用数值 x 来表示，表 2-4 列出了常用数学函数的功能、用法和函数返回值的类型，函数的运算结果保留一位小数。

表 2-4 数学函数及用法

函数	功能	返回值类型	示例	结果
Abs(x)	取绝对值	同 x	Abs(-8.6)	8.6
Sin(x)	正弦函数	Double	Sin(60*3.14/180)	0.9
Cos(x)	余弦函数	Double	Cos(60*3.14/180)	0.5
Tan(x)	正切函数	Double	Tan(60*3.14/180)	1.7
Exp(x)	指数函数 e^x	Double	Exp(2)	7.4
Log(x)	e 为底的对数函数	Double	Log(30)	3.4
Rnd(x)	随机数函数	Single	rnd	[0,1]之间的数
Sgn(x)	符号函数 x>0、x=0、x<0 时分别 返回 1, 0, -1	Integer	Sgn(-10)	-1
Sqr(x)	平方根函数	Double	Sqr(20)	4.5
Int(x)	取小于等于 x 的最大整数	Double	Int(-5.5)	-6
Fix(x)	取整函数	Double	Fix(-5.5)	-5
Round(x)	四舍五入函数	Double	Round(-5.5)	-6

对部分数学函数说明如下。

(1) 数学三角函数如 Sin 函数、Cos 函数、Tan 函数等，参数都以弧度为单位，可以是任何数值表达式，如果是角度数值，先要转为弧度，即角度数乘以 $\pi / 180$ 得到弧度数值。

(2) Int(x)和 Fix(x)函数都是取整，但有所不同。Fix(x)函数只是去掉小数部分，返回其整数部分。而 Int(x)函数是返回小于等于 x 的最大整数，但不同于四舍五入，例如：

Fix(8.6)和 Int(8.6)结果都为 8；

Fix(-8.6)结果为-8，Int(-8.6)结果都为-9。

当函数参数 x 是正数时，这两个函数的结果相同，而当 x 是负数时，结果不同。

(3) Rnd(x)函数是一个产生随机数的函数，返回一个随机的 (0, 1)之间的 Single 型小数，参数 x 是随机数种子，若省略 x 则默认 $x > 0$ 。当 $x > 0$ ，重复执行 Rnd(x)产生随机数序列的下一个数；当 $x < 0$ ，重复执行 Rnd(x)产生相同的一个随机数；当 $x = 0$ ，重复执行函数，会出现一个相同的随机数。

当 $x = 0$ 时重复调用随机数函数也只能产生同一个随机数，当 $x > 0$ 时，重复调用函数能够产生随机数序列，但这是一个相同的序列，每次启动程序调用函数产生的随机数序列都是相同的。而如果将 Randomize 语句放到随机数函数的前面，可以避免这两种情况的发生。Randomize 语句的语法如下。

```
Randomize [(x)]
```

x 是随机数发生器的种子数（整型），可以省略。

在程序设计的过程中，经常需要产生某个范围的随机数或某个范围的随机数整数，可以参考下面的规则。

若要产生(x,y)范围内的随机数：

```
Rnd * (y - x) + 1
```

若要产生(x,y)范围的随机整数：

```
Int ((y - x + 1) * Rnd) + x
```

应用实例

【实例 2-1】 在窗体上添加 3 个标签 (Label1, Label2, Label3), 3 个文本框 (Text1, Text2, Text3), 一个命令按钮 Command1。界面如图 2-1 所示。编写适当的代码，程序运行时在 Text1 中输入一个整数，单击 Command1 在 Text2 中计算出该数的绝对值，在 Text3 中计算出求绝对值后该数的平方根。

主要操作步骤：

(1) 设计界面。

新建工程和窗体，在窗体上添加相应的控件。设置控件的属性如表 2-5 所示。

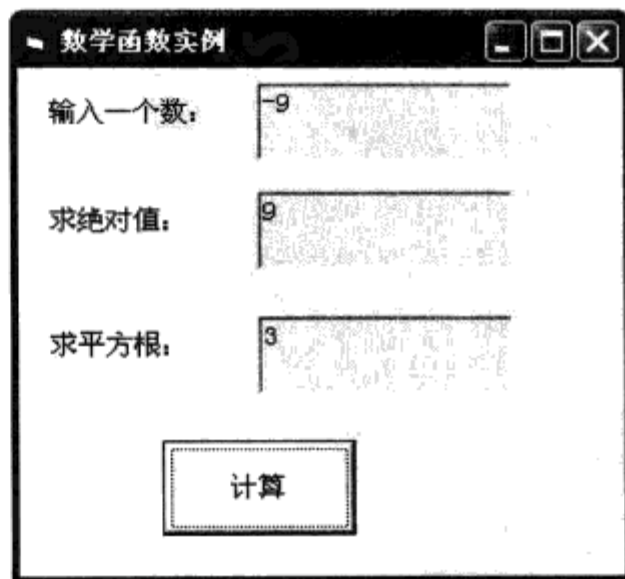


图 2-1 数学函数实例程序运行界面

表 2-5 控件属性设置

对象	控件名	属性名	属性值
Form	Form1	Caption	数学函数实例
Label	Label1	Caption	输入一个数
	Label2	Caption	求绝对值
	Label3	Caption	求平方根
Text	Text1	Text	空
	Text2	Text	空
	Text3	Text	空
Command	Command1	Caption	计算

(2) 代码编写。

```
Private Sub Command1_Click()
'单击命令按钮完成计算并输出
    Dim x As Integer, y As Integer, z As Single
    x = Text1.Text
    y = Abs(x)
    z = Sqr(Sgn(x) * x)
    Text2 = y
    Text3 = z
End Sub
```

注意:文本框的 Text 属性为字符型,而 x,y,z 三个变量的数据类型为数值型中的 Integer 和 Single,类型不匹配。鉴于该例中系统能够自动进行这两种类型之间的转换,所以本例忽略了这个问题。

(3) 保存工程和窗体。

2.4.2 字符函数

知识要点分析

字符函数用于对字符串进行处理,在 Visual Basic 中,采用了新的字符处理方式,将英文字符和中文字符统一编排,每个字符都用两个字节表示,每个英文字符和汉字的字符长度都是 1,这种处理方式称为“Unicode 方式”。用字母 s 表示一个字符串, n 表示数值,表 2-6 列出了常用字符函数的名称、功能、返回值类型和用法。

表 2-6 常用字符函数及用法

函数格式	功能	返回值的类型
InStr([n],s1, s2[,m])	在 s1 中从 n 开始查找 s2, 返回 s2 在 s1 中开始的位置	Integer variant
Left(s,n)	从 s 左边截取 n 个字符	String
Right(s,n)	从 s 右边截取 n 个字符	String

续表

函数格式	功能	返回值的类型
Mid(s,n1[,n2])	从 n1 位置开始截取 n2 长度的字符串	String
Len(s varname)	返回字符串的长度或变量的字符数	Integer Variant
LenB(s varname)	返回字符串的长度或变量的字节数	Integer Variant
Ltrim(s)	去掉字符串左边的空格	String
Rtrim(s)	去掉字符串右边的空格	String
Trim(s)	去掉字符串两边的空格	String
Space(n)	产生由 n 个空格组成的字符串	String
String(n,s n1)	返回由 n 个 s 字符串首字符组成的字符串或 n1 对应的 ASCII 字符组成的字符串	String
StrReverse(s)	返回字符串的反串	String
Replace(s,s1,s2,[n1][,n2][,m])	在 s 中从 n1 开始由 s2 替代 s1 共替代 n2 次, m 表示是否区分大小写	String
Ucase(s)	将 s 的小写字母换成大写字母	String
Lcase(s)	将 s 的大写字母换成小写字母	String

比较复杂的字符函数说明:

1. InStr([n],s1, s2[m])

确定 s2 子串在 s1 主串中的起始字符位置。

说明: 若参数 n 缺省, 该函数返回 s2 在 s1 中的首次出现的起始字符位置。否则, 该函数返回 s2 在 s1 中从第 n 位起出现的起始字符位置。例如:

```
Dim x
x = InStr(2, "This is a book", "is", 0)
Print x      'x 的值是 3
```

2. string(n,s|n1)

功能: 返回由 n 个 s 字符串首字符组成的字符串或是由 n 个 n1 数字对应的 ASCII 字符组成的字符串。例如:

```
Dim s, s1, s2 As String
s1 = String(10, 65)
s2 = String(10, "abc")
Print s1      's1 的值是 10 个 A"AAAAAAAAAAAA"
Print s2      's2 的值是 10 个 a"aaaaaaaaaa"
```

3. Replace(s,s1,s2[,n1][,n2][,m])

说明: 如果省略 n1 和 n2, 在字符串 s 中, 从第一个字符开始, 由字符串 s2 替代所有的 s1; 若有 n1, 则从 n1 开始替代, 若有 n2, 则从 n1 开始连续替代 n2 次; m 表示是否区分大小写, m=0 区分, m=1 不区分, 省略 m 为区分大小写。例如:

```
Dim s As String, s1 As String, s2 As String
s = "These Is books And those Is pens."
```

```

s1 = Replace(s, "is", "are", 1, 2, 1)
s2 = Replace(s, "is", "are", 1, 2, 0)
Print s1 's1 的值是"These are books And those are pens."
Print s2 's1 的值是"These Is books And those Is pens."

```

应用实例

【实例 2-2】 在窗体上添加 9 个标签 (Label1, Label2, …), 9 个文本框 (Text1, Text2, …),

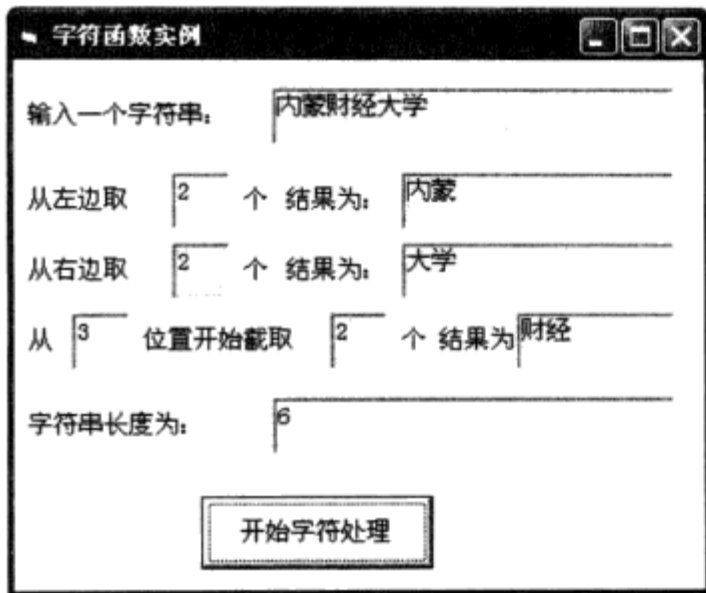


图 2-2 字符函数实例程序运行界面

一个命令按钮 Command1。界面如图 2-2 所示。编写适当的代码, 程序运行时在 Text1 中输入一个字符串, 在 Text2、Text4、Text6、Text7 中分别输入数字 2、2、3、2, 单击 Command1 在 Text3、Text5、Text8 和 Text9 中得到结果。

主要操作步骤:

(1) 界面设计。

新建工程和窗体, 添加控件, 设置属性。窗体的 Caption 属性设置为“字符函数实例”; 标签的 Caption 属性和文本框的 Text 属性设置为如图 2-2 所示的内容; 命令按钮的 Caption 属性设置为“开始字符处理”。

(2) 代码编写。

```

Private Sub Command1_Click()
    Dim s As String, n1 As Integer, n2 As Integer, n3 As Integer, n4 As Integer
    s = Text1
    n1 = Val(Text2) '类型转换函数, 将文本框中的字符串转换为数值
    n2 = Val(Text4)
    n3 = Val(Text6)
    n4 = Val(Text7)
    Text3 = Left(s, n1)
    Text5 = Right(s, n2)
    Text8 = Mid(s, n3, n4)
    Text9 = Len(s)
End Sub

```

注意: 本例中用到了一个 Val(s)函数, 这是一个类型转换函数, 将文本框中的字符串转换为数值。

(3) 保存工程和窗体。

2.4.3 日期时间函数

知识要点分析

日期时间函数和日期或者时间相关, 常用的日期时间函数的功能、用法、返回值类型

等如表 2-7 所示，参数 d 表示日期，t 表示时间，n 为一个数值。

表 2-7 常用日期函数

函数名	功能	返回类型	示例	结果
Date	返回系统日期	Date	Date	系统日期
Now	返回系统日期和时间	Date	Now	系统日期时间
Time	返回系统时间	Date	Time	系统时间
Year(d)	返回年份	Integer	Year(#2010/3/15#)	2010
Month(d)	返回月	Integer	month(#2010/3/15#)	3
Day(d)	返回日	Integer	Day(#2010/3/15#)	15
MonthName(n)	返回月份名	String	MonthName(3)	3 月
WeekDay(d)	返回日期号星期日为 1， 以此类推	Integer	WeekDay(#2011/07/21#)	5
WeekDayName(n)	返回星期名	String	WeekDayName(5)	星期四
Hour(t)	返回小时	Integer	Hour("9:30:35 AM")	9
Minute(t)	返回分钟	Integer	Minute (#9:30:35 AM#)	30
Second(t)	返回秒	Integer	Second (#9:30:35 AM#)	35

说明：系统日期与时间是用户计算机中内部时钟的日期与时间，该日期与时间不一定与实际日期与时间一致。系统日期与时间可以由用户根据需要而设置。

应用实例

【实例 2-3】 在窗体上添加 7 个标签 (Label1, Label2, …)，5 个文本框 (Text1, Text2, …)，一个命令按钮 Command1。界面如图 2-3 所示。编写适当的代码，程序运行时单击命令按钮“显示日期时间”，则在相应的文本框中显示日期和时间信息。

主要操作步骤：

(1) 界面设计。

新建工程和窗体，添加控件，设置属性。窗体的 Caption 属性设置为“日期时间函数实例”；标签的 Caption 属性和文本框的 Text 属性设置为如图 2-3 所示的内容；命令按钮的 Caption 属性设置为“显示日期时间”。

(2) 代码编写。

```
Private Sub Command1_Click()
    Text1 = Year(Date)
    Text2 = Month(Date)
    Text3 = Day(Date)
    Text4 = Time
    Text5 = WeekdayName(Weekday(Date))
End Sub
```

(3) 保存工程和窗体。

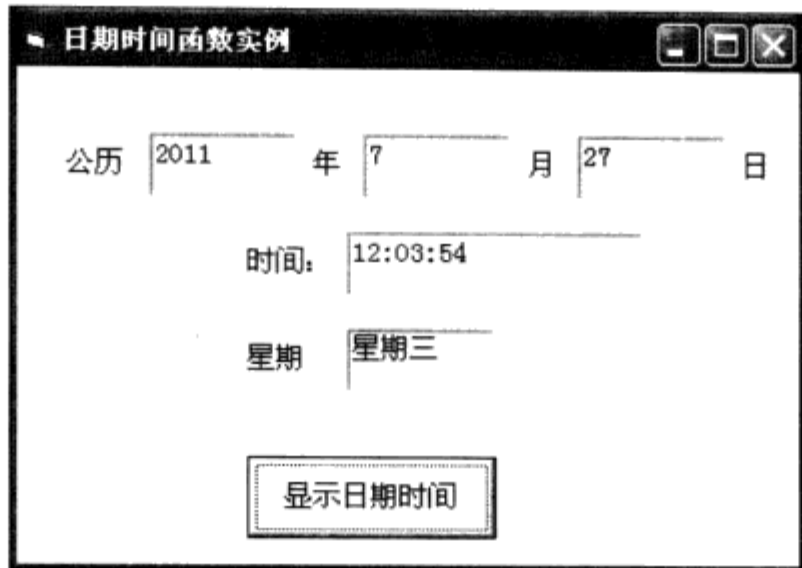


图 2-3 日期函数实例程序运行界面

2.4.4 类型转换函数和判断函数

☎ 知识要点分析

1. 类型转换函数

类型转换函数的功能是强制进行数据类型的转换，这里只介绍主要的 4 个类型转换函数，函数的功能、用法和返回值类型如表 2-8 所示，其中参数 s 表示字符串，n 表示数值。

表 2-8 类型转换函数

函数名	功能	返回类型	示例	结果
Asc(s)	将首字符转换为 ASCII 值	Integer	Asc("abc")	97
Chr(n)	将 ASCII 码值转换成字符	String	Chr(68)	"D"
Str(n)	将数值转换成字符串	String	Str(23.87)	"23.87"
Val(s)	将数字字符串转换成数值	Double	Val("15.3")	15.3

1) Asc(s)

返回字符串 s 首字符对应的 ASCII 码数值（十进制），在一般情况下，返回 0~255 之间的整数，s 不能是空字符串。

2) Chr(n)

将数值（十进制）转换成相应的 ASCII 码字符，n 正常范围为 0~255，Chr()与 Asc()为互逆函数。

3) Str(n)

将数值转换为数字字符串，转换后的字符串左边增加一个符号位。

4) Val(s)

将字符串中第一个数字字符到第一个非数字字符之间的所有数字字符转换为数值。小数点、0~9、正负号等被认为是数字字符。若第一个字符就是非数字字符，其转换结果为 0。

例如：

```
Dim n1 As Single, n2 As Single, n3 As Single
```

```
n1 = Val("38a.76b")
n2 = Val("abc")
n3 = Val("56.8")
Print n1, n2, n3    'n1 的结果是 38, n2 的结果是 0, n3 的结果为 56.8
```

应用实例

【实例 2-4】 考虑数据类型的匹配，重新编写实例 2-1 程序的代码。代码编写如下。

```
Private Sub Command1_Click()
'单击命令按钮完成计算并输出
    Dim x As Integer, y As Integer, z As Single
    x = val(Text1.Text)    '文本框中是数值字符串，转为数值后赋值给 x
    y = Abs(x)
    z = Sqr(Sgn(x) * x)
    Text2 = str(y)        '变量 y 的值是 Integer 数值型，转换为字符后通过文本框输出
    Text3 = str(z)
End Sub
```

2. 类型判断函数

Visual Basic 还提供了一些用来判断数据类型的函数，这里仅介绍 Typename 函数。该函数可以返回变量、一些常量、函数及表达式的类型。

函数应用举例：

```
Private Sub Command1_Click()
Dim b As Date
Print TypeName(1000)    '常量类型，结果为 Integer
Print TypeName("fale") '常量类型，结果为 String
Print TypeName(Rnd)    '判断函数的类型，结果为 single
Print TypeName(b)      '判断变量的类型，结果为 Date
Print TypeName(Sqr(9) + 6 > 10) '判断表达式的类型，结果为 Boolean
End Sub
```

2.5 程序代码编写规则

知识要点分析

1. 语句及语法

- Visual Basic 编写程序时，语句是按行书写的，不能随意回车，每行最多允许 255 个字符。
- 如果一行语句过长，想分成多行书写，需要加续行符（一个空格和一个下划线“_”组成）后换到下一行。

- 如果多条语句比较短（比如若干条给变量赋值的语句），也可以合并到一行上，语句与语句之间用半角的“:”号分隔。

2. 代码书写规则

- 代码书写不区分大小写字母，通常在 Visual Basic 中命令词的首字母大写。
- 各关键字之间，关键字和变量名、常量名、过程名之间一定要用空格分隔。
- 最好使用缩进格式，以使程序结构更具可读性。
- 严格按照 Visual Basic 规定的格式和符号编写程序。

3. 结构化程序的基本结构

Visual Basic 是结构化的程序设计语言，有 3 种结构：顺序结构、选择（分支）结构和循环结构。结构化的特点是程序结构清晰，可读性强，容易调试。

1) 顺序结构

程序严格按照语句书写的先后顺序执行，这样的程序结构叫顺序结构。顺序结构是程序结构中最常见、最简单的一种程序结构，一般由赋值语句、输入语句和输出语句等组成。

2) 选择结构

程序设计中常会遇到根据不同的情况采用不同的处理方法的情况。需要根据判断的结果来决定执行哪部分程序。比如成绩处理中，根据不同的分数成绩来判断等级，用选择结构就比较好处理。选择结构有 If...Else 和 Select...Case 两种语句。

3) 循环结构

在编程的过程中经常会遇到某一段代码需要反复执行多次，循环结构的程序设计就可以很容易地解决问题。循环结构允许当某个条件为真时进入循环，反复执行一些语句，而直到条件为假才跳出循环。Visual Basic 提供了多种形式的循环语句，包括 For...Next 循环、Do...Loop 循环等。

在编程过程中，经常是 3 种程序控制结构综合使用的，这样才能游刃有余地控制程序的执行顺序，充分发挥程序的功能。

本章小结

本章是 Visual Basic 程序设计语言的基础，主要介绍了基本数据类型、常量、变量、运算符、表达式和常用内部函数，并且简单介绍了程序代码的编写规则和程序结构等，以便为后面章节的深入学习打好基础。

习 题 2

一、判断题

1. Visual Basic 数据类型可分为基本数据类型和用户自定义类型两大类。（ ）
2. 表达式 $25 \bmod 6$ 的值是 4。（ ）
3. 函数 Asc("b") 的值是 95。（ ）

4. 单精度型和双精度型是完全相同的两个数据类型。()
5. 字节型 (Byte) 用来存储二进制数据, 用 1 个字节存储。()
6. 当定义变量时没有说明数据类型, 则默认变体型。()
7. 字符常量是用半角单引号 (") 括起来的字符序列。()
8. 符号常量实质上就是变量。()
9. 字符型 (String) 变量有定长和变长之分, 在声明定长字符型变量时, 用 "String*长度" 来表示。()
10. 一条 Dim 语句只能声明一个变量。()
11. 赋值号 (=) 和数学中的等号形式上相同, 实质不同。()
12. 在使用赋值语句赋值时, 原则上要求赋值运算符两边数据类型相同。()
13. Visual Basic 有 3 种程序控制结构: 顺序结构、选择 (分支) 结构和循环结构。()
14. Left("呼和浩特", 2) 的结果是 "呼"。()
15. 设 a 是字符型变量, a="243516", 则表达式 Val(Right(a, 2) + Mid(a, 3, 2)) 的结果为 1643。()

二、单选题

1. 下列表示正确的关系表达式是 ()。

A. $100 \leq x \leq 200$	B. $x \geq 100 \text{ or } x \geq 200$
C. $x \geq 100 \text{ and } x \leq 200$	D. 以上都不对
2. 执行语句 S=Len(Mid("内蒙古财经大学", 4, 2)) 后, S 的值是 ()。

A. 5	B. 4	C. 3	D. 2
------	------	------	------
3. 表达式 String(6, "abc") 输出的值为 ()。

A. aaaaaa	B. bbbbbb
C. cccccc	D. abcabcabcabcabcabc
4. 表达式 UCase(Mid("Visual Basic 程序设计", 1, 6)) 的值是 ()。

A. visual	B. VISUAL	C. basic	D. BASIC
-----------	-----------	----------	----------
5. 下列不属于数值型数据类型的是 ()。

A. Integer	B. Long	C. Single	D. Date
------------	---------	-----------	---------
6. LenB("VB 程序设计") 的返回值为 ()。

A. 6	B. 10	C. 12	D. 14
------	-------	-------	-------
7. 要生成 50~60 之间的随机整数, 应采用的表达式为 ()。

A. Int(Rnd(60)+50)	B. Int(Rnd*60)
C. Int(Rnd*61)	D. Int(Rnd*11)+50
8. 语句 Print Sgn(-5) + Abs(-3 ^ 2) + Int(-3 ^ 2) 的输出结果是 ()。

A. 18	B. 23	C. 1	D. -1
-------	-------	------	-------
9. 可以同时删除字符串前导和尾部空格的函数是 ()。

A. Ltrim	B. Trim	C. Rtrim	D. Mid
----------	---------	----------	--------
10. 下列语句成功执行后, 变量 a 的数据类型为 ()。

```
dim a as string
```

```
a=100  
print a
```

11. 成功执行如下命令后, 变量 a 的值为 ()。
- A. 字符型 B. 数值型 C. 变体型 D. 以上都不对

```
Private Sub Command1_Click()  
    Dim a As String * 3  
    a = "abcdef"  
    Print a  
End Sub
```

12. 表达式#2011-07-12#-#2011-06-12#的结果为 ()。
- A. 1 B. 30 C. 无结果 D. 以上都不对
13. 表达式"student" Like "stu*"的执行结果为 ()。
- A. True B. False C. 空 D. 以上都不对
14. 表达式"student" & 100 的结果为 ()。
- A. "student" B. 100 C. 出错提示 D. "student100"
15. 表达式 TypeName(Day(Date))用来判断 Day(Date)的数据类型, 结果为 ()。
- A. Variant B. Byte C. Integer D. Date

第3章

顺序结构程序设计

一个完整的 VB 应用程序，一般都包含三部分内容，即输入数据、计算处理、输出结果。

程序严格按照语句书写的先后顺序执行，这样的程序结构叫顺序结构。顺序结构是程序结构中最常见、最简单的一种程序结构，一般由赋值语句、输入数据语句和输出数据语句等组成。VB 的输入输出有着十分丰富的内容和形式，它提供了多种手段，并可通过有关控件实现输入输出操作。

3.1 赋值语句

知识要点分析

在程序的执行过程中，说明变量后，可以为变量赋予具体的数据，或者读出其代表的数值，即变量的赋值与引用。赋值语句是程序设计中最基本、最常用的语句。

1. 格式

<变量名> = <表达式>

2. 功能

计算表达式的值，并将计算结果赋予赋值号左边的变量。

3. 说明

(1) 表达式中的变量必须是赋过值的，否则变量的初值自动取零（变常字符串变量取空字符）。例如：

```
a=1  
c=a+b+5          '变量 b 未赋值。为 0
```

执行后，c 值为 6。

(2) 赋值号 (=) 不同于数学中的等号，如： $x=x+1$ 在数学中不成立，而在 Visual Basic 中，表示将变量 x 的值+1 再重新赋予变量 x。数学中 $a=b+c$ 等价于 $b+c=a$ ，但 Visual Basic 中， $b+c=a$ 是非法的赋值语句。

(3) 表达式可以是单个常量或变量，也可以是由常量、变量、函数、运算符及括号等组成的字符串。

(4) 在使用赋值语句赋值时，原则上要求赋值运算符两边数据类型相同，对于某些情

况，如将数值赋给字符串变量、将数字形式的字符串赋给数值变量等，VB 也能自动进行数据类型的转换。若赋值运算符两边数据类型不同且不能转换，会出现类型不符错误。例如：

```
Dim a As Integer, b As Integer, c As String
a=1.5           '转换 1.5 为整型数 2 (四舍五入), 再赋值给 a
b="ABCD"       '出错, 类型不符
c=123          '转换为字符串"123", 再赋值给 c
```

3.2 注释、暂停与结束语句

☞ 知识要点分析

3.2.1 注释语句

1. 格式

'<注释内容 >

或

Rem <注释内容>

2. 功能

为程序的相关内容加注释与说明，以便于程序的理解。例如

```
Rem 交换变量 a 与 b 的值
c = a '借助于第三个变量 c
a = b
b = c
```

3. 说明

- (1) 注释语句是非执行语句，对程序的执行过程不产生任何影响，也不被编译与解释。
- (2) 用 ' 加的注释内容，习惯于将注释内容置于可执行语句的后面，而使用 Rem 加的注释内容单独占一行。

3.2.2 暂停语句

1. 格式

Stop

2. 功能

在程序中设置“断点”，暂停程序的执行，同时打开“立即”窗口，用户可在此对程序进行检测和调试。

3.2.3 结束语句

1. 格式

End

2. 功能

结束当前程序的执行，关闭所有（运行）文件，重置所有的变量。

3. 说明

End 语句除用于结束程序的执行外，在不同的环境下还有其他一些用途，主要包括：

End Sub	'结束一个通用过程
End Function	'结束一个函数过程
End If	'结束一个条件分支语句块
End Select	'结束一个多情况语句

3.3 使用 Print 方法输出数据

知识要点分析

1. 格式

[对象名称.]Print [表达式列表][,|;]

2. 功能

按指定的输出格式在窗体、图片框或打印机上输出各个表达式的值。

3. 说明

(1) 对象名称为可选参数。可以是窗体名称、图片框名称或打印机名称。如果是向程序所在的当前窗体输出信息，则可省去该窗体对象的名称。

(2) 表达式列表，可以是一个或多个任何类型的表达式。若是数值型表达式，则先计算表达式的值，然后，输出其值。若是字符表达式，则原样输出。若省略表达式列表，则输出一个空行。

(3) 若输出的是多个表达式，各表达式以“,”分隔，按标准格式输出（即：分区输出格式）。各表达式以“;”分隔，则按紧凑格式输出（即：后项输出紧接前一项输出）。

应用实例

【实例 3-1】 设计一个交换变量 x、y 的值的程序，当单击窗体时，显示变量 x 和 y 交换前后的值。程序运行结果如图 3-1 所示。

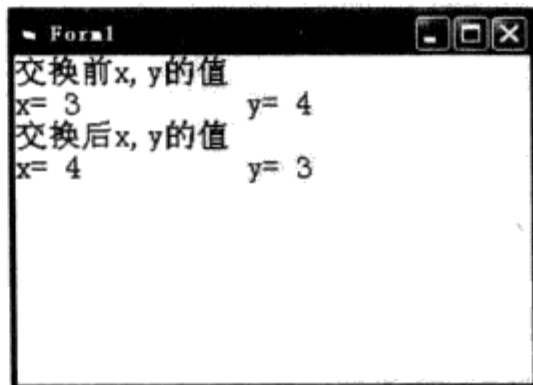


图 3-1 实例 3-1 运行结果界面

新建一个工程，编写如下所示的程序代码。

```
Private Sub Form_Click()
Dim x As Integer, y As Integer '声明变量类型
x = 3: y = 4 '给变量 x,y 赋值
Rem 以下程序段用于显示交换前变量的值
Print "交换前 x,y 的值"
Print "x="; x, "y="; y
Rem 以下程序段用于交换 x,y 的值
z = x '借助于第三个变量 z
x = y
y = z
Rem 以下程序段用于显示交换后变量的值
Print "交换后 x,y 的值"
Print "x="; x, "y="; y
End Sub
```

3.4 基本控件

前面在第 1 章介绍了窗体，窗体为应用程序提供了一个窗口，但是仅有窗体是不够的，还需要在其中放置各种控件才能实现用户与应用程序之间的信息交互。

本节介绍控件的公共属性和三种基本控件——命令按钮、标签和文本框。其他常用控件将在以后章节中陆续介绍。

3.4.1 控件的公共属性

知识要点分析

控件有很多公共的属性，下面介绍多数控件所共有的属性。以后介绍具体控件时，将不再重复这些属性。

1. Name 属性

用于定义控件对象的名称，从而可以唯一地确定对象。在定义对象属性、使用对象方法、定义对象事件过程时使用，该属性为所有控件的公共属性。

每当新建一个控件时，VB 会给该控件指定一个默认名，如 Command1, Command2, …; Text1, Text2, …; Label1, Label2…等。用户可在属性窗口的“名称”栏中设置控件的名称，控件的 Name（名称）属性必须以字母开头，其后可以是字母、数字和下划线，名称长度不能超过 40 个字符。但在程序运行时，Name 属性是只读的，即不能在应用程序中修改。

2. Caption 属性

用于确定控件的标题，即显示在控件上的文本。Caption 属性是说明性的文字，可以是任意字符串。

当创建一个控件时，其默认标题与默认的 Name 属性值相同，可以通过属性窗口改变其值，也可通过程序代码改变其值，例如：

```
Command1.Caption="确定"
```

执行该语句将使命令按钮 Command1 的标题更改为“确定”。

可以在 Caption 属性中为控件指定一个访问键。访问键是指控件标题后带下划线的字母。在按住 Alt 键的同时按下访问键，相当于单击此控件。定义访问键的方法是：在输入标题的同时在访问键字母前加“&”符号即可，如“确定(&E)”。此方法为通用方法。

3. Enable 属性

决定控件是否对用户产生的事件做出响应。如果将控件的 Enable 属性值设置为 True (默认)，则控件有效，允许控件对事件做出响应；当设置 Enable 属性值设置为 False 时，则控件变成浅灰色，不允许使用。

4. Visible 属性

决定控件是否可见，默认值为 True。当设置属性值为 False 时，控件不可见。

5. Height, Width, Top 和 Left 属性

Height 和 Width 属性确定控件的高度和宽度，Top 和 Left 属性确定控件在窗体中的位置。Top 表示控件到窗体顶部的距离，Left 表示控件到窗体左边框的距离。

6. BackColor 和 ForeColor 属性

这两个属性用于设置控件的背景色和前景色。

7. Font 对象属性

与窗体中类似。

3.4.2 命令按钮

知识要点分析

命令按钮 (CommandButton) 作为应用程序中使用最多的控件对象之一，常常用来接收用户的操作信息，激发相应的事件过程。

通常可以用双击工具箱中的按钮图标或直接在窗体画出的方法建立按钮。当按钮建立后，程序员可根据应用程序的需要进行命令按钮的属性设置，并编制相应的事件代码。

1. 常用属性

(1) Default 属性和 Cancel 属性：设置该按钮是否为窗体的“默认”按钮和是否为窗体的缺省“取消”按钮，所谓“默认”按钮即只要用户按下 Enter 键相当于单击该按钮，自动执行该命令按钮的 Click 事件过程；“取消”按钮是指只要用户按下 Esc 键就等价于单击该按钮，自动执行该命令按钮的 Click 事件过程。

以上属性其值为 True 时，表示指定；否则为不指定。

(2) Style 属性：设置命令按钮的外观：值为 0 (默认) 时以标准的 Windows 按钮方式显示；值为 1 时以图形按钮方式显示，此时可用 Picture, DownPicture 和 DisabledPicture 属性来分别指定按钮在正常、被按下和不可用三种状态的图片。

(3) Value 属性: 该属性只能在程序运行期间使用。设置为 True 表示该命令按钮被按下。

2. 常用事件

命令按钮最基本的事件是 Click (单击) 事件, 但不支持 DblClick (双击) 事件。以下操作均可触发 Click 事件。

- (1) 单击此按钮。
- (2) 当焦点在按钮上时, 按空格键或 Enter 键。
- (3) 对于默认按钮按 Enter 键, 对于缺省取消按钮按 Esc 键。
- (4) 使用 Alt+访问键。
- (5) 在代码中将按钮的 Value 属性设置为 True。

3. 常用方法

命令按钮常用方法是 SetFocus 方法。

应用实例

【实例 3-2】 在窗体上添加如图 3-2 所示的两个命令按钮, 指定 Command1 和 Command2 的访问键分别为 T 和 B。编写适当的代码, 程序运行时单击 Command1 将窗体标题更改为“内蒙古财经学院欢迎你”, 单击 Command2 将窗体背景改为红色。

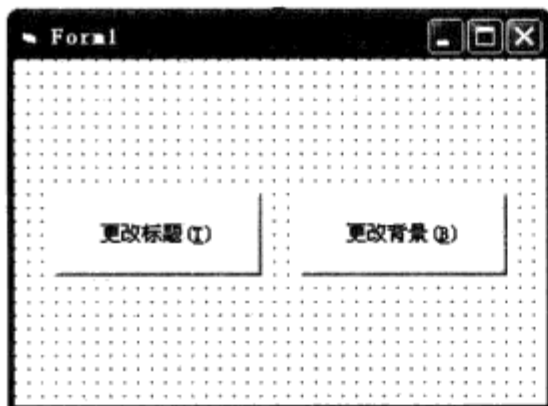


图 3-2 实例 3-2 设计界面

设计步骤如下。

- (1) 设计程序界面。

新建一个工程, 在 Form1 窗体中添加两个命令按钮。

- (2) 设置控件属性。

将命令按钮 Command1 的 Caption 属性设置为“更改标题(T)”, 将命令按钮 Command2 的 Caption 属性设置为“更改背景(B)”。

- (3) 编写程序代码。

编写的两个事件过程代码如下。

```
Private Sub Command1_Click()
    Form1.Caption = "内蒙古财经学院欢迎你"
End Sub

Private Sub Command2_Click()
    Form1.BackColor = RGB(255, 0, 0)
End Sub
```

3.4.3 标签

知识要点分析

标签 (Label) 作为输出控件, 一般用于显示只读性质的文本, 如处理结果、提示帮助等信息。但也可以用于标识窗体或其他本身不具备 Caption 属性的控件 (如文本框 TextBox、

滚动条 ScrollBar), 以说明它们的名称或功能。其默认名称为 Label1, Label2, ...

1. 常用属性

在标签的所有属性中, 最重要的就是 Caption 属性。它规定了在标签中显示的文本信息的内容, 可在程序设计时通过属性窗口设置或在运行时用代码赋值。常见的标签属性如下。

(1) Alignment 属性: 设置标签内文本针对于标签的对齐方式, 共有 3 个可选项: 0 (左对齐, 默认值), 1 (右对齐) 和 2 (居中)。

(2) AutoSize 属性: 设置标签是否自动改变大小以显示其全部内容, 值为 True 自动改变标签大小以显示全部内容; 值为 False (默认值) 保持标签大小不变, 超出标签区域的内容不显示。

(3) BackStyle 属性: 设置标签的背景模式, 共有 2 个选项, 是否透明, 0 表示透明, 1 (默认值) 表示不透明。

(4) BorderStyle 属性: 设置标签有无边框: 0 (默认值) 表示无边框, 1 表示固定单边框。

(5) WordWrap 属性: 设置一个 AutoSize 属性为 True 的 Label 控件, 是否要进行水平或垂直展开以适合其 Caption 属性中指定的文本的要求。值为 True 时, 文本卷绕, Label 控件水平大小不变, 垂直展开或缩短, 以使其与文本和字体大小相适应; 值为 False (默认值), 文本不卷绕, Label 水平地展开或缩短以使其与文本的长度相适。

2. 常用事件和方法

标签可触发 Click (单击) 和 DblClick (双击) 等事件。标签支持 Move 方法, 用于实现控件的移动。

应用实例

【实例 3-3】 在窗体上添加如图 3-3 所示的名称为 Label1 的标签和两个命令按钮, Label1 的标题设置字体为“宋体”, 字号为“二号”, 设置为自动大小。编写适当的代码, 程序运行时单击 Command1 将标签标题字体更改为“黑体”, 单击 Command2 将标签标题字号更改为 20。

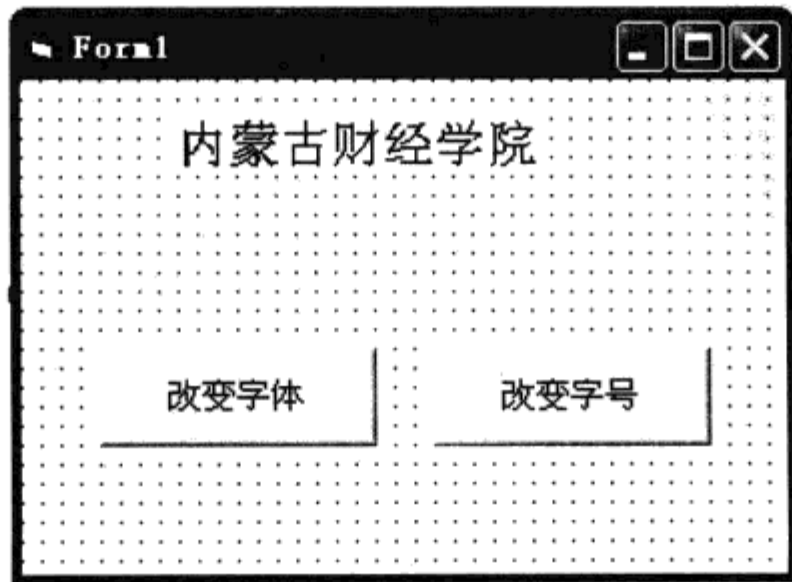


图 3-3 实例 3-3 设计界面

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加名称为 Label1 的标签和两个命令按钮。

(2) 设置控件属性。

将标签 Label1 的 Caption 属性设置为“内蒙古财经学院”，AutoSize 属性设置为 True，通过 Font 属性对话框设置字体为“宋体”，字号为“二号”。将命令按钮 Command1 的 Caption 属性设置为“改变字体”，将命令按钮 Command2 的 Caption 属性设置为“改变字号”。

(3) 编写程序代码。

编写的两个事件过程代码如下。

```
Private Sub Command1_Click()
    Label1.FontName = "黑体"
End Sub
Private Sub Command2_Click()
    Label1.FontSize = 20
End Sub
```

3.4.4 文本框

知识要点分析

文本框 (TextBox) 是最重要的输入输出控件之一。通过文本框，用户可以方便快捷地输入和编辑文本。

1. 常用属性

文本框具有一般控件的常用属性，但文本框没有 Caption 属性。下面介绍它的一些特殊属性。

(1) MultiLine 属性：设置文本框是否可接收和显示多行文本：值为 True，允许多行文本，当到达文本框右边界时文本自动换行；值为 False (默认值)，将忽略回车符，数据在一行内显示，多余的内容不显示。

(2) MaxLengh 属性：设置显示在文本框中的最多字符数，超出部分不接收并发出嘟嘟声。对于单行显示的文本框，指定最大长度为 2KB；对于多行显示的文本框，指定最大长度为 32KB。

(3) PasswordChar 属性：设置在文本框中显示字符的替代符，如一串“*”号。一般用于口令输入。若 MultiLine 属性已设为 True，则设置 PasswordChar 属性将毫无效果。

(4) Scrollbars 属性：设置文本框内是否有滚动条显示及滚动条的样式：0 (默认值) 没有滚动条，1 显示水平滚动条，2 显示垂直滚动条，3 同时显示水平和垂直滚动条。

注意：如果要显示滚动条，必须将 MultiLine 属性设置为 True。

(5) Text 属性：设置或返回文本框中的内容。默认值为 Text1, Text2, ...

(6) Locked 属性：设置文本框是否可以编辑修改，值为 True 时为只读。

以下属性只能在代码中使用。

(7) SelStart 属性：设置或返回文本框中插入点的位置，0 表示最左边的位置；如果属性值大于或等于文本的字符数，那么插入点将被放在最后一个字符之后。

(8) SelLength 属性: 将其设为大于 0 的数值, 系统会自动选中从插入点开始的 SelLength 个字符。

(9) SelText 属性:

① 返回文本框中所选的文本, 如果没有字符被选中, 则为空串。

② 设置其值, 则新的值将自动替换选中的文本, 若无选中文本, 则新的值在当前插入点位置插入。

2. 常用事件和方法

文本框支持 Click, DblClick 等鼠标事件, 同时支持 Change, GotFocus, LostFocus 等事件。文本框常用的方法有 Refresh 方法和 SetFocus 方法。

应用实例

【实例 3-4】 在窗体上添加如图 3-4 所示的两个文本框 Text1、Text2 和一个命令按钮, Text1 初始内容如图 3-4 所示。编写适当的代码, 在程序运行时单击“返回选中文本”按钮在 Text2 中返回 Text1 中选中文本。

设计步骤如下。

(1) 设计程序界面。

新建一个工程, 在 Form1 窗体中添加两个文本框 Text1、Text2 和一个命令按钮。

(2) 设置控件属性。

设置两个文本框的 MultiLine 属性值为 True, 设置 Text1 的 Scrollbars 属性值为 2, Text2 的 Scrollbars 属性值为 1, 在 Text1 的 Text 属性中输入如图所示的内容, 将 Text1 的 Text 属性清空; 将命令按钮 Command1 的 Caption 属性设置为“返回选中文本”。

(3) 编写程序代码。

编写的两个事件过程代码如下。

```
Private Sub Command1_Click()  
    Text2.Text = SelText  
    Text1.SetFocus  
End Sub
```

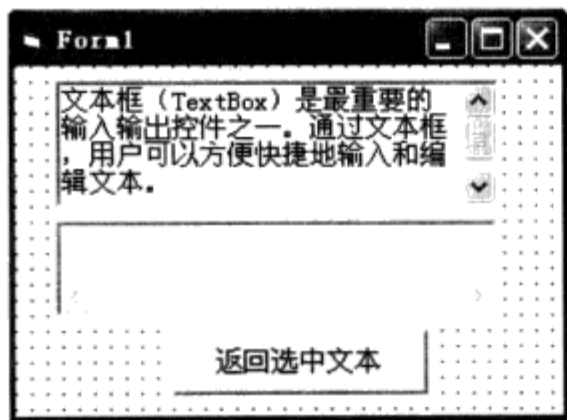


图 3-4 实例 3-4 设计界面

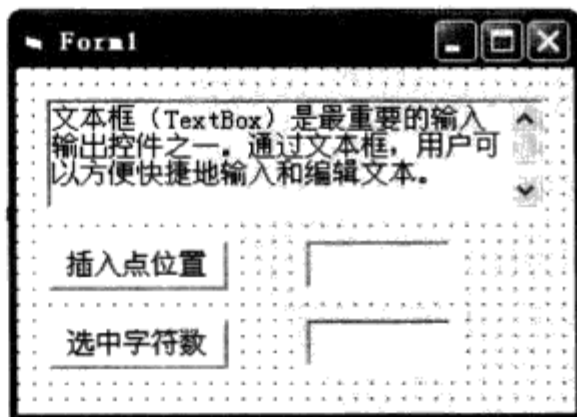


图 3-5 实例 3-5 设计界面

【实例 3-5】 在窗体上添加如图 3-5 所示的 3 个文本框 Text1, Text2, Text3 和 2 个命令按钮, Text1 初始内容如图 3-5 所示。编写适当的代码, 在程序运行时单击“插入点位置”按钮在 Text2 中返回 Text1 的光标位置, 单击“选中字符数”按钮时 Text3 中返回 Text1 中选中的文本个数。

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加 3 个文本框 Text1, Text2, Text3 和 2 个命令按钮。

(2) 设置控件属性。

设置文本框 Text1 的 MultiLine 属性值为 True, Scrollbars 属性值为 2, 在 Text1 的 Text 属性中输入如图所示的内容，将 Text2, Text3 的 Text 属性清空；将命令按钮 Command1, Command2 的 Caption 属性分别设置为“插入点位置”和“选中字符数”。

(3) 编写程序代码。

编写的两个事件过程代码如下。

```
Private Sub Command1_Click()
    Text2.Text = Text1.SelStart
    Text1.SetFocus
End Sub
Private Sub Command2_Click()
    Text3.Text = Text1.SelLength
    Text1.SetFocus
End Sub
```

【实例 3-6】 在窗体上添加两个如图 3-6 所示的文本框 Text1, Text2, 初始内容为空，编写适当的代码，使程序运行时 Text2 中同步显示 Text1 的内容。

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加 2 个文本框 Text1, Text2。

(2) 设置控件属性。

设置文本框 Text1, Text2 的 MultiLine 属性值为 True, Scrollbars 属性值为 2, 将 Text1, Text2 的 Text 属性清空。

(3) 编写程序代码。

编写的事件过程代码如下。

```
Private Sub Text1_Change()
    Text2.Text = Text1.Text
End Sub
```

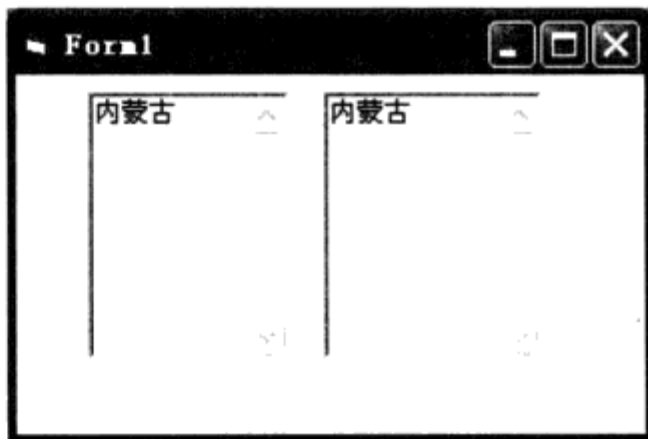


图 3-6 实例 3-6 设计界面

3.5 列表框与组合框

3.5.1 列表框

☞ 知识要点分析

列表框 (ListBox) 作为输入控件，可为用户提供选项的列表，并支持用户从列表中选

选择一个或多个选项完成输入。列表框中的选项可呈一列显示，也可以呈多列显示。由于列表框大小有限，当选项数目超过列表框可容纳的选项数目时，列表框会自动出现滚动条，供用户上下滚动选择。

在列表框内的项目称为表项，表项的加入是按一定的顺序进行的，这个顺序号称为索引。

1. 常用属性

(1) Columns 属性：确定列表框是水平滚动还是垂直滚动，以及显示列表中表项的方式。默认值为 0，表示每一个表项占一行，列表框按单列垂直滚动方式显示。当属性值为 n (大于 1) 时，多个表项占一行，列表框按多列水平滚动方式显示。在列表框容纳不下项目时会自动出现滚动条。

(2) List 属性：该属性是一个字符型数组（数组的概念将在第 5 章介绍），用于存放列表框的表项。List 数组的下标（可以理解为索引，通过它可以指定数组中的某一个元素）从 0 开始。例如，在图 3-7 中，List1.List(0) 的值为“教授”，List1.List(1) 的值为“副教授”，以此类推。

可以通过 List 属性向列表框添加表项，其操作是：在属性窗口中单击 List 属性，再单击其右端的下拉箭头，用户可以在该下拉方框中输入列表框中的表项。每输入一项按 Ctrl+Enter 键换行，全部输入完后按 Enter 键，所输入的表项即出现在列表框中。

(3) MultiSelect 属性：设置是否允许同时选择多个列表项，0 为不允许（默认值）；1 为允许，使用鼠标单击或按下空格键的方法可在列表中选中多个选项或取消对某个选项的选择；2 也为允许，可使用 Ctrl 和 Shift 键配合鼠标操作选中多个选项。

(4) Sorted 属性：设置列表框中各表项在运行时是否按字母顺序排列。值为 True 表示按字母顺序排序，但要注意：此后使用带 Index 参数的 AddItem 方法向列表中的某一确定位置添加元素可能违反排序顺序，后面项目的添加可能不会正确地排序。值为 False 表示不按字母顺序排序（默认）。

(5) Style 属性：确定控件的样式。默认值为 0，表示标准形式；设置该属性值为 1 时，表示复选框形式，即在每个表项前增加一个复选框以表示该表项是否被选中。

以下属性只能在代码中使用，部分属性为只读。

(6) ListCount 属性：该属性为只读属性，返回列表框中表项的数目。ListCount-1 表示列表中的最后一项的序号。

(7) ListIndex 属性：该属性为只读属性，返回当前选定表项的索引值。如选中第一个表项值为 0，选中第二个表项值为 1，以此类推，如选中最后一个表项值为 ListCount-1，没有表项被选中值为-1。

(8) Selected 属性：该属性是一个逻辑型数组，表示列表框中某一项是否被选中。例如 List1.Selected(3)=True 表示选中列表框中的第 4 条选项；若为 False，表示未选中。List1.Selected(List1.ListIndex)=False 表示撤销对当前选中项的选中。

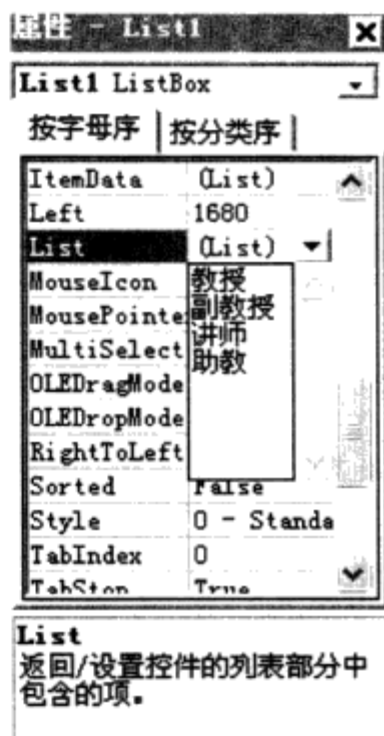


图 3-7 List 属性

(9) Text 属性: 该属性为只读属性, 返回当前选定表项的文本内容。

2. 事件

列表框常见的事件主要有单击事件 (Click) 和双击事件 (DblClick)。

3. 方法

列表框中的表项可以通过 List 属性设置, 也可在程序中用 AddItem 方法来添加, 用 RemoveItem 或 Clear 方法进行删除。

(1) AddItem 方法: 使用 AddItem 方法可向列表框中添加新的选项, 语法格式为:

```
[对象名].AddItem 列表项[,索引]
```

其中, “列表项” 用于指定要插入的列表选项内容, String 型; 索引用于指定插入选项的具体位置, Integer 数据。当索引默认时, 代表在列表框的最后插入新的表项 (追加)。有索引时, 在索引指定的位置进行插入。

例如, 要在省份列表框 List1 的第 28 个位置后插入 “海南省”, 可以采用

```
List1.AddItem "海南省", 28
```

(2) RemoveItem 方法: RemoveItem 方法用于清除列表框中指定的选项, 语法格式为:

```
[对象名].RemoveItem 索引
```

其中, “索引” 用于指定要删除的列表项所处的具体位置, Integer 数据。

(3) Clear 方法: Clear 方法用于清除列表框中的全部选项, 语法格式为:

```
[对象名].Clear
```

4. 列表框表项的输出

输出列表框中表项, 有 3 种常用方法:

(1) 用鼠标单击列表框内某一表项, 则该表项值存放在 Text 属性中。例如:

```
X=list1.text '把选定的表项值存放在 x 变量中
```

(2) 指定索引号来获取表项的值, 例如:

```
List1.ListIndex=3
```

```
X=List1.Text
```

(3) 从 List 数组中读取表项的值, 例如:

```
X=List1.List(3)
```

应用实例

【实例 3-7】 设计一个程序, 通过一个列表框的内容对另一个列表框的表项进行添加、删除及表项数目的统计。程序设计界面如图 3-8 所示, 运行界面如图 3-9 所示。

设计步骤如下。

(1) 设计程序界面。

新建一个工程, 在 Form1 窗体中添加 2 个标签, 2 个列表框 List1, List2, 3 个命令按钮 Command1, Command2, Command3, 1 个文本框 Text1。



图 3-8 实例 3-7 设计界面

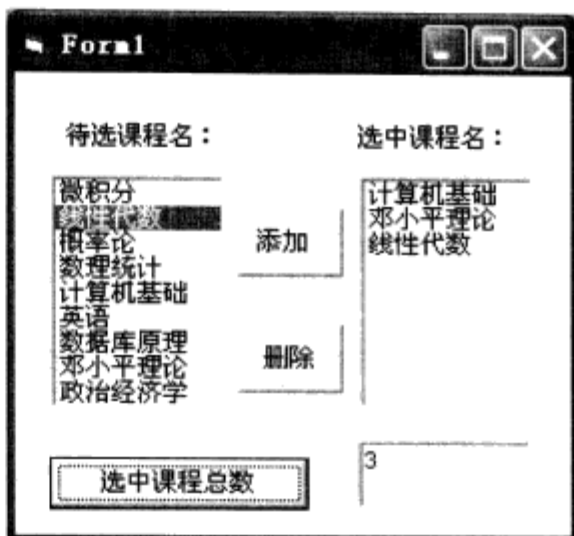


图 3-9 实例 3-7 运行界面

(2) 设置控件属性。

设置 2 个标签的标题分别为“待选课程名:”、“选中课程名:”，列表框 List1 的 List 属性值如图 3-8 所示，3 个命令按钮 Command1, Command2, Command3 的标题分别为“添加”、“删除”、“选中课程总数”，将 Text1 的 Text 属性清空。

(3) 编写程序代码。

编写的 3 个事件过程代码如下。

```
Private Sub Command1_Click()
List2.AddItem List1.Text
End Sub
Private Sub Command2_Click()
List2.RemoveItem List2.ListIndex
End Sub
Private Sub Command3_Click()
Text1.Text = List2.ListCount
End Sub
```

3.5.2 组合框

☎ 知识要点分析

组合框的功能和使用方法与列表框基本相同。不同的是：通过有效的设置，组合框允许用户输入未出现在指定列表中的选项内容，并且可将选项折叠起来，以节省控件所占用的空间。一般而言，它是一个文本框与一个下拉列表框的组合，ListIndex=-1 代表从文本框中输入的项目。

组合框具有列表框和文本框的大部分属性和方法，还有一些自己的属性。

(1) Style 属性：该属性取值为 0、1 或 2，分别决定了组合框的 3 种不同类型，即下拉组合框（默认）、简单组合框和下拉列表框如图 3-10 所示。

① 下拉组合框：执行时，用户可以直接在文本框内输入内容，也可单击其下拉箭头，再从打开的列表框中选择，选定内容会显示在文本框上。

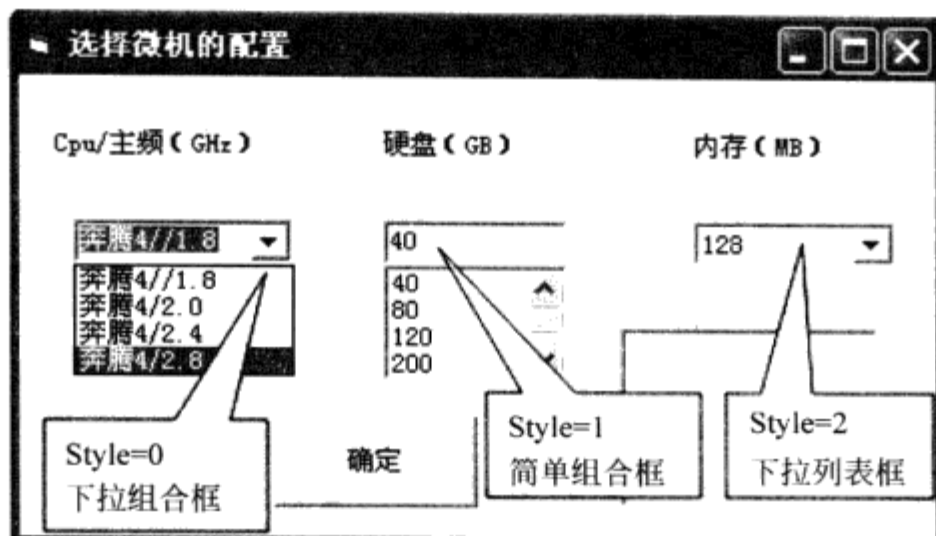


图 3-10 Style 属性值不同时的组合框

② 简单组合框：它列出所有的项目供用户选择，没有下拉箭头，列表框不能收起。这种组合框也允许用户直接在文本框内输入内容。

③ 下拉列表框：不允许用户输入内容，只能从下拉列表框中选择。

(2) Text 属性：该属性是用户所选定项目的文本或直接从文本框输入的文本。

同列表框一样，在应用程序的运行过程中，组合框的选项内容也不是一成不变的，同样可以通过 AddItem, RemoveItem 和 Clear 方法进行选项内容的添加或删除。

由于组合框的样式可能为下拉式组合框，因此除了单击事件 (Click) 以外，组合框还有可能要响应按键弹起事件 (KeyUp)。

应用实例

【实例 3-8】 在窗体上设置 3 个组合框，用于选择微机部分配置，包括 CPU/ 主频、硬盘及内存。运行时，用户在选择各项内容之后，单击“确定”按钮，在文本框内显示所选择的微机配置。程序运行界面如图 3-11 所示。

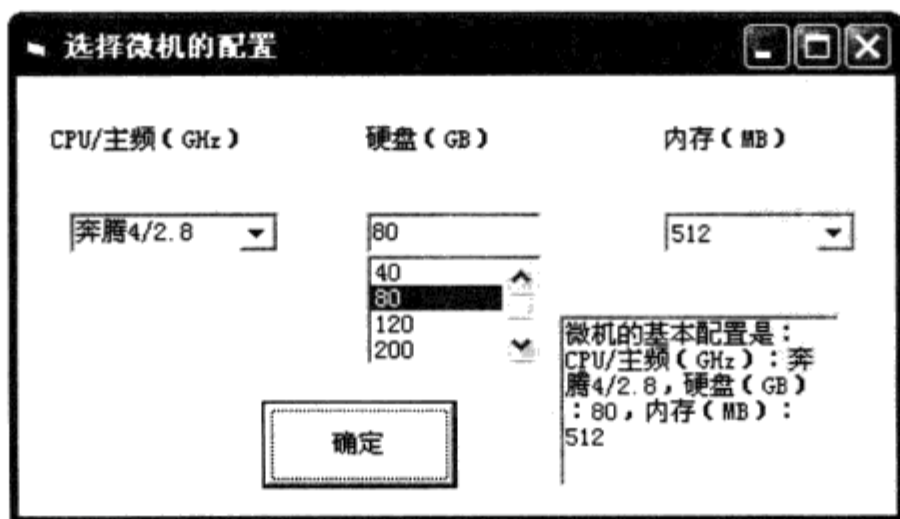


图 3-11 实例 3-8 程序运行界面

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加 3 个标签，3 个组合框，1 个命令按钮，1 个文本框 Text1。

(2) 设置控件属性。

设置窗体的标题为“选择微机的配置”，3个标签的标题分别为“CPU/主频 (GHz)”、“硬盘 (GB)”和“内存 (MB)”，3个组合框的 List 属性值如图 3-10 所示，命令按钮的标题为“确定”，将 Text1 的 Text 属性清空，MultiLine 属性为 True。

(3) 编写程序代码。

在代码窗口编写如下所示的事件代码。

```
Private Sub Form_Load()  
    Combo1.Text = Combo1.List(0)  
    Combo2.Text = Combo2.List(0)  
    Combo3.Text = Combo3.List(0)  
End Sub  
  
Private Sub Command1_Click()  
    Text1.Text = "微机的基本配置是: " & "CPU/主频 (GHz): " & Combo1.Text _  
        & ", 硬盘 (GB): " & Combo2.Text & ", 内存 (MB): " & Combo3.Text  
End Sub
```

本章小结

本章介绍了赋值语句、注释、暂停与结束语句、Print 方法等 VB 程序设计的基本语句及控件的公共属性；并通过命令按钮、标签、文本框、列表框与组合框等基本控件的使用描述了顺序结构程序设计的方法。

习题 3

一、判断题

1. 一个程序代码行只能写入一条语句。()
2. 字符型数据可以用英文的双撇号或单撇号括起来。()
3. 当某一控件的 Enable 属性值设置为 False 时，则该控件在程序运行时不起作用。()
4. 标签和文本框都能显示文字，且在运行时都可由用户编辑这些文字。()
5. 命令按钮只能响应单击事件，不能响应双击事件。()
6. 所有窗体和控件都有 Name 属性，其值不能为空。()
7. Name 属性值可以以字母或数字开头。()
8. Name 属性值可以在属性窗口修改，也可以在通过代码来修改。()
9. 窗体和控件的所有属性都可以在属性窗口中设置。()
10. 窗体和控件的所有属性都可以在程序中通过语句设定。()
11. 控件的某些属性只能在程序运行时设定。()

12. 文本框没有 Caption 属性。()
13. 在 3 种组合框中, 只有下拉列表框仅供选择其中表项数据, 但不允许添加数据。()
14. 对象的属性设置是预先设置好的。()

二、单选题

1. 语句 S=S+1 的正确含义是 ()。
- A. 变量 S 的值与 S+1 的值相等 B. 将变量 S 的值存到 S+1 中去
C. 将变量 S 的值加 1 后赋给变量 S D. 变量 S 的值为 1
2. 语句 print "sqr(16)";sqr(16)的输出结果是 ()。
- A. sqr(16)=sqr(16) B. sqr(16)=4
C. "4"="4" D. 4=sqr(16)
3. 在文本框中输入一个字符时, 能同时触发的事件是 ()。
- A. KeyPress 和 Change B. Click 和 Change
C. DblClick 和 KeyPress D. DblClick 和 Change
4. 在窗体上已建立了两个文本框 Text1 和 Text2, 并编写如下 3 个事件过程。

```
Private Sub Form_Click()
    Text2.Text = Text1.Text
    Text1.Text = Text2.Text + "p"
End Sub
Private Sub Form_Load()
    Text1.Text = "m"
    Text2.Text = "n"
End Sub
Private Sub Text1_Change()
    Text2.Text = Text2.Text + Text1.Text
End Sub
```

运行程序后单击窗体, 则在文本框 Text2 中显示的内容是 ()。

- A. MP B. MMP C. MMPP D. NMMP
5. 读取列表框中的 3 个表项值, 把值赋给变量 X, 不可以常用 ()。
- A. X=List1.List(2) B. X=List1.text(2)
C. List1.Selected(2)=true D. List1.ListIndex=2
X=List1.Text X=List1.Text
6. 下列控件没有 Caption 属性的对象是 ()。
- A. 标签 B. 命令按钮 C. 窗体 D. 文本框
7. 列表框中被选择的选项条目序号是由 () 属性决定的。
- A. List B. Index C. ListIndex D. TabIndex
8. 清除列表框的所有选项条目时所用的方法是 ()。
- A. Remove B. Clear C. ReFresh D. Move
9. 在标签控件中, 更改文字对齐方式的属性是 ()。

- A. Justify B. Font C. Alignment D. AutoSize
10. 要在命令按钮上显示图形, 应首先设置的属性是 ()。
- A. Style B. Enable C. Caption D. Icon

三、编程及上机调试

在窗体上设置如图 3-12 所示的一个命令按钮、一个文本框和一个标签, 3 个控件的 Visible 属性值均为 False, 命令按钮的标题是“显示”, 标签能自动调整大小。编写适当的代码使程序运行后, 单击窗体时显示出命令按钮, 再单击命令按钮时则显示标签和文本框, 并以“宋体”10号在标签上显示“已下达显示命令”及以“隶书”20号在文本框中显示“内蒙古财经学院欢迎你”。运行后界面如图 3-13 所示。

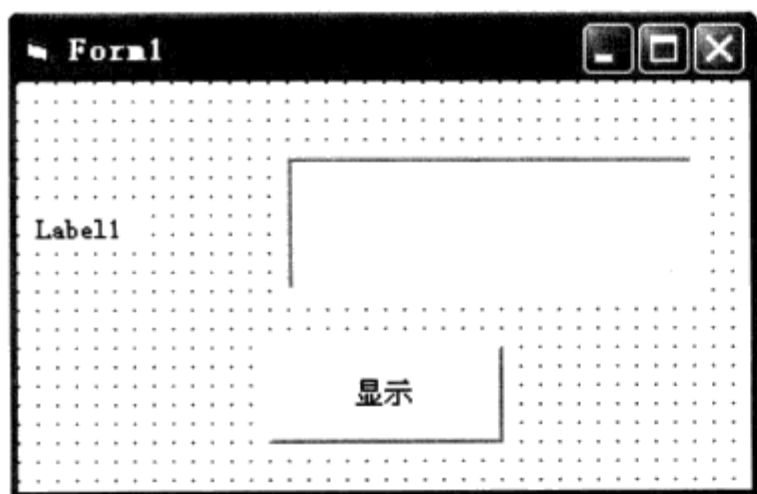


图 3-12 习题三设计界面

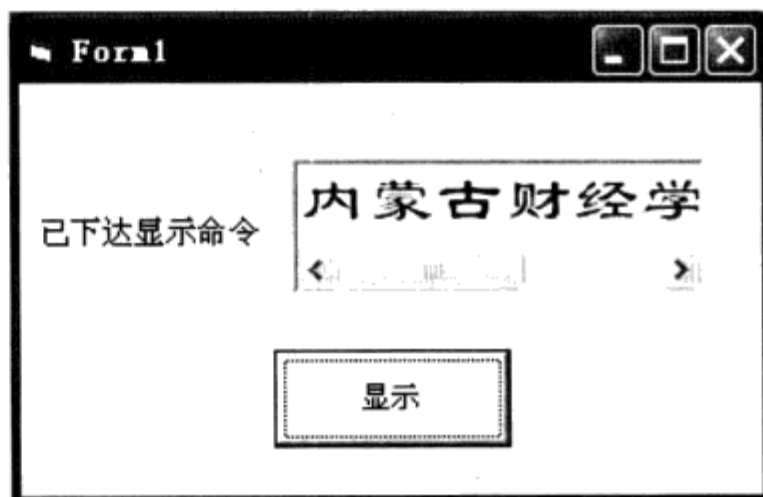


图 3-13 习题三运行界面

第4章

选择结构程序设计

用顺序结构编写的程序比较简单，只能实现一些简单的处理，在实际应用中，有许多问题需要判断某些条件，根据判断的结果来控制程序的流程。使用选择结构的程序，可以实现这样的处理。

VB 中实现选择结构的语句主要有 If 和 Select...Case 语句。

4.1 条件语句

4.1.1 单行条件语句

☎ 知识要点分析

1. 格式

```
If <条件> Then <语句序列 1> [ Else <语句序列 2> ]
```

2. 功能

计算条件的值，若为真，则执行语句序列 1；否则，执行语句序列 2。

3. 说明

条件是逻辑表达式。单行条件语句要求在一行内输入完，不能超过 255 个字符。语句序列 1 或语句序列 2，可以是任何语句，可以是个单语句，也可以是用“:”分隔的多个语句组成的语句序列。

✍ 应用实例

【实例 4-1】 设计一个程序，实现从第一个文本框输入一个成绩，单击“判断”按钮，通过第二个文本框输出判断结果。判断条件是当成绩 ≥ 60 为合格，否则为不合格。程序运行界面如图 4-1 所示。

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加两个文本框 Text1、Text2，添加两个标签 Label1、

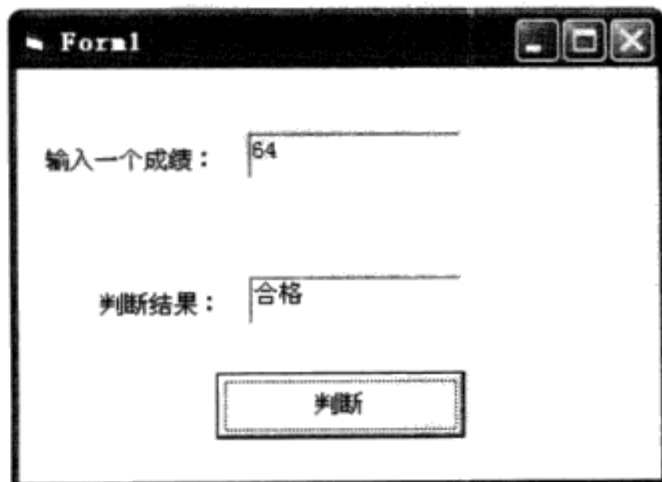


图 4-1 实例 4-1 运行界面

Label2, 一个命令按钮 Command1。

(2) 设置控件属性。

将两个文本框 Text1, Text2 的 text 属性值清空, 将两个标签 Label1, Label2 的 Caption 属性设置分别设置为“输入一个成绩”和“判断结果”, 将命令按钮 Command1 的 Caption 属性设置为“判断”, 如图 4-1 所示。

(3) 编写程序代码。

在代码窗口编写如下代码。

```
Private Sub Command1_Click()  
    Dim x As Single  
    x = Text1.Text  
    If x >= 60 Then Text2.Text = "合格" Else Text2.Text = "不合格"  
End Sub
```

4.1.2 多行条件语句

知识要点分析

1. 格式

```
If <条件> Then  
    <语句序列 1>  
[ Else  
    <语句序列 2> ]  
End If
```

2. 功能

计算条件的值, 若其值为真, 则执行语句序列 1。然后, 跳过语句序列 2, 执行 End If 的后续语句。否则, 执行语句序列 2, 然后, 执行 End If 的后续语句。

3. 说明

多行条件语句又称为块结构条件语句。If 语句为分支的开始语句, 是条件分支点。End If 为分支的结束语句, 是分支的汇合点。语句序列 1 或语句序列 2 中可以包含任何语句。

应用实例

【实例 4-2】 使用多行条件语句实现实例 4-1。

在代码窗口编写如下代码。

```
Private Sub Command1_Click()  
    Dim x As Single  
    x = Text1.Text  
    If x >= 60 Then  
        Text2.Text = "合格"  
    Else  
        Text2.Text = "不合格"
```

```
End If
End Sub
```

【实例 4-3】 设计一个程序，实现从 3 个文本框输入 3 个数，单击“判断”按钮，在第 4 个文本框中显示其中最大的数。运行界面如图 4-2 所示。

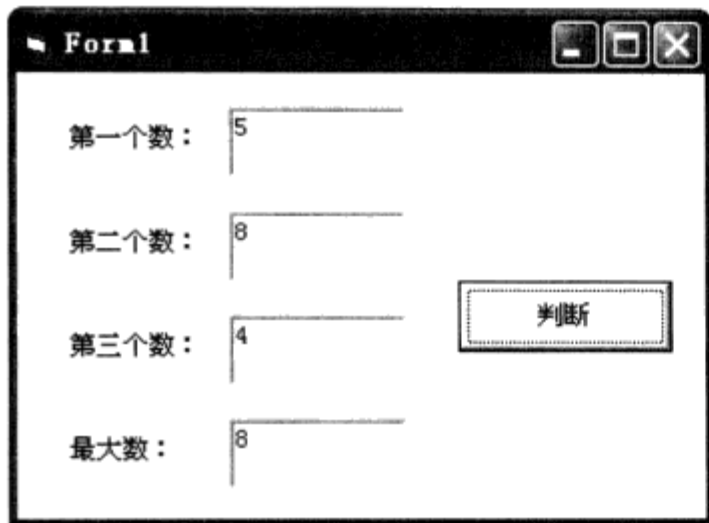


图 4-2 实例 4-3 运行界面

“第二个数”、“第三个数”和“最大数”，将命令按钮 Command1 的 Caption 属性设置为“判断”，如图 4-2 所示。

(3) 编写程序代码。

```
Private Sub Command1_Click()
    Dim a As Integer, b As Integer
    Dim c As Integer, d As Integer
    a = Val(Text1.Text)
    b = Val(Text2.Text)
    c = Val(Text3.Text)
    If a > b Then
        d = a      'd 用来存放较大值
    Else
        d = b
    End If
    If c > d Then d = c
    Text4.Text = d
End Sub
```

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加 4 个文本框 Text1, Text2, Text3, Text4，添加 4 个标签 Label1、Label2、Label3、Label4，添加一个命令按钮 command1。

(2) 设置控件属性。

将 4 个文本框的 Text 属性值清空，将 4 个标签的 Caption 属性分别设置为“第一个数”、

4.1.3 条件语句的嵌套

☎ 知识要点分析

1. 一般格式

由于条件语句中的每个语句序列中均可以包含任意的 Visual Basic 语句，所以在条件语句中可以再包含条件语句，这种一个条件语句中再包含另一个条件语句的语句结构，即称之为条件语句的嵌套。条件语句的嵌套结构的执行过程与简单条件类似，条件语句的嵌套一般没有具体的嵌套层数的限制，在实际应用中，只要内存能够满足要求，即可不断嵌

套下去。其语法格式为：

```
If <条件 1> Then
    If <条件 2> Then
        :
        End If
    Else
        :
    End If
```

条件语句不但可以嵌套，也可以嵌入到其他程序结构中，例如，可以将条件语句嵌入到循环结构中，也可以将循环结构嵌入到条件结构中。

2. ElseIf 格式

如果出现多层 IF 语句嵌套，将使程序冗长，不便阅读，为此 VB 提供了带有 ElseIf 的语句结构。其语法格式为：

```
If <条件 1 >Then
    <语句序列 1>
ElseIf <条件 2> Then
    <语句序列 2>
ElseIf <条件 3 >Then
    <语句序列 3>
:
[Else
    <语句序列 n+1>]
EndIf
```

这是多分支的条件语句。该语句执行时先测试“条件 1”，如果为假，依次测试“条件 2”，以此类推，直到找到为真的条件。一旦找到一个为真的条件，VB 会执行相应的语句序列，然后执行 End If 语句后面的代码。如果所有的条件都是假，VB 便执行 Else 后面的“语句序列 N+1”，然后执行 End If 语句后面的代码。

应用实例

【实例 4-4】 设计程序，根据不同的时间段在窗体上用“隶书”加粗小 1 显示问候语，如 0 时—12 时，显示“早上好”；12 时—18 时，显示“下午好”；18 时—24 时，显示“晚上好”。

本例要求分别用两种不同的条件语句的嵌套格式编写程序代码，采用 Print 方法直接在窗体上输出结果，运行结果界面如图 4-3 所示。

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加两个

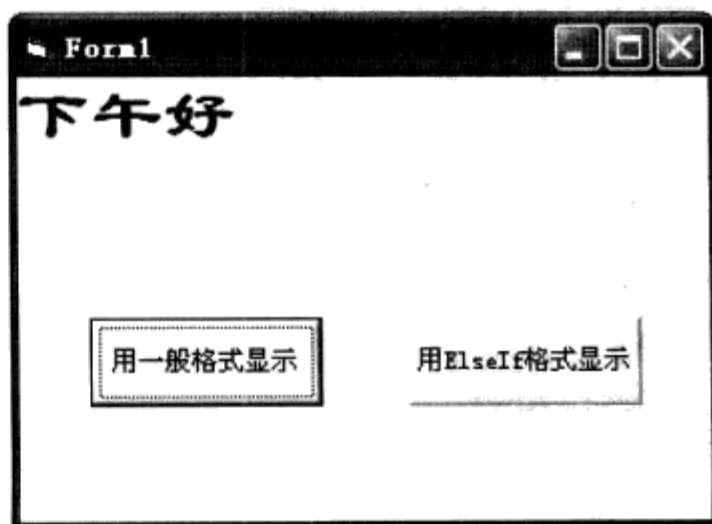


图 4-3 实例 4-4 运行界面

命令按钮 Command1, Command2。

(2) 设置控件属性。

通过窗体的 Font 属性对话框将窗体的显示字体设置为“隶书”加粗小 1, 将命令按钮 Command1, Command2 的 Caption 属性分别设置为“用一般格式显示”和“用 ElseIf 格式显示”, 如图 4-3 所示。

(3) 编写程序代码。

在代码窗口分别编写命令按钮 Command1, Command2 的单击事件代码如下。

```
Private Sub Command1_Click()  
Dim h As Integer  
h = Hour(Time) '取系统的时间  
If h < 12 Then  
    Print "早上好"  
Else  
    If h < 18 Then  
        Print "下午好"  
    Else  
        Print "晚上好"  
    End If  
End If  
End Sub
```

```
Private Sub Command2_Click()  
Dim h As Integer  
h = Hour(Time) '取系统的时间  
If h < 12 Then  
    Print "早上好"  
ElseIf h < 18 Then  
    Print "下午好"  
Else  
    Print "晚上好"  
End If  
End Sub
```

4.2 多分支语句

知识要点分析

虽然使用条件语句的嵌套可以实现多分支选择, 但结构不够简明。使用多分支语句 Select...Case 也可以实现多分支选择, 它比起上述条件语句嵌套更有效, 更易读, 并且易于跟踪调试。多分支语句也称为多情况语句。

1. 格式

```
Select Case <测试表达式>
```

```

Case <表达式列表 1>
    <语句序列 1>
Case <表达式列表 2>
    <语句序列 2>
:
Case <表达式列表 n>
    <语句序列 n>
[Case Else
    <语句序列 n+1>]
End Select

```

2. 功能

计算“测试表达式”的值，并比较该测试值与哪个 Case 子句的“表达式”匹配，若找到了，则执行该情况对应的语句序列，然后跳到 End Select 出口语句。若所有情况均不匹配，则执行 Case Else 对应的语句序列。

3. 说明

多情况选择语句以 Select Case 开始，End Select 结束。“测试表达式”可以是数值表达式或字符表达式，通常是变量。所谓匹配是指“测试表达式”的值包含在域值中。“表达式列表”称为域值，可以是下列形式之一。

- (1) <表达式 1>[,<表达式 2>]…，如：Case 2, 4, 6, 8。
- (2) <表达式 1> To <表达式 2>，如：Case 1 To 10。
- (3) Is <关系表达式>。使用的运算符包括：<, <=, >, >=, <>, =，如：Case Is >=10。
- (4) 在一个 Case 语句中，三种形式可以混用，如：Case Is <=10,12,14,16,18,100 To 200。

应用实例

【实例 4-5】 使用多分支语句实现实例 4-4。

程序运行界面如图 4-4 所示。

设计步骤如下。

- (1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加一个命令按钮 Command1。

- (2) 设置控件属性。

通过窗体的 Font 属性对话框将窗体的显示字体设置为“隶书”加粗小 1，将命令按钮 Command1 的 Caption 属性设置为“显示问候”，如图 4-5 所示。

- (3) 编写程序代码。

在代码窗口编写命令按钮 Command1 的单击事件代码如下。

```

Private Sub Command1_Click()
Dim h As Integer

```



图 4-4 实例 4-5 运行界面

```

h = Hour(Time) '取系统的时间
Select Case h
    Case Is < 12
        Print "早上好"
    Case Is < 18
        Print "下午好"
    Case Else
        Print "晚上好"
End Select
End Sub

```

【实例 4-6】 输入学生成绩（百分制），判断该成绩的等级。

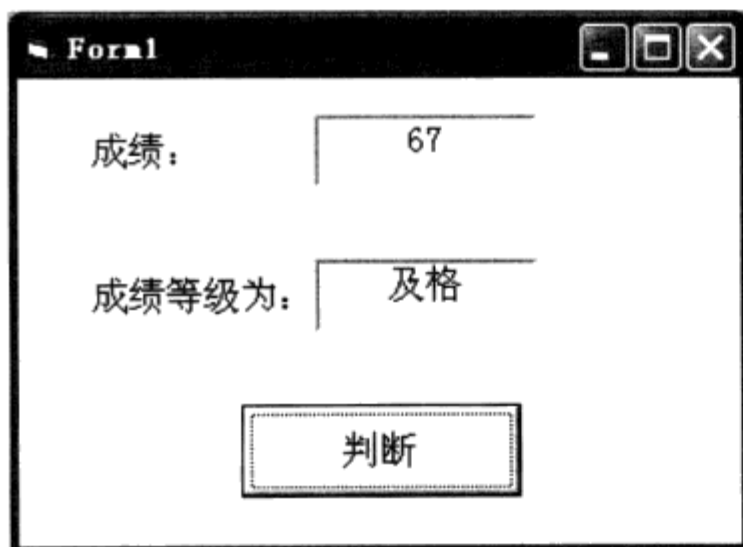


图 4-5 实例 4-6 运行界面

在代码窗口编写如下代码。

```

Private Sub Command1_Click()
Dim cj As Integer, dj As String
cj = Val(Text1.Text)
Select Case cj
    Case 0 To 59
        dj = "不及格"
    Case 60 To 75
        dj = "及格"
    Case 75 To 90
        dj = "良"
    Case 90 To 100
        dj = "优"
    Case Else
        dj = "成绩输入出错"
End Select
Text2.Text = dj
End Sub

```

判断条件为：成绩<60 为不及格，成绩在 60~75 为及格，成绩在 75~90 为良，成绩在 90~100 为优。程序运行界面如图 4-5 所示。

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加两个标签、两个文本框和一个命令按钮。

(2) 设置控件属性。

通过窗体的 Font 属性对话框将各控件的字号设置为小四，其余属性设置如图 4-5 所示。

(3) 编写程序代码。

4.3 使用对话框

对话框是程序与用户进行交互的重要途径。对话框既可以用来显示信息，也可以用于输入信息。在 VB 中提供了两种内容的对话框：输入对话框和消息对话框。这两种对话框可以通过系统提供的函数（InputBox 和 MsgBox）来创建，而不必为对话框另建窗体。

4.3.1 输入对话框

☎ 知识要点分析

1. InputBox 函数的语法格式

```
<变量>=InputBox("提示" [, "标题"] [, "默认值"] [, xpos] [, ypos])
```

其中：

(1) “提示”指定在对话框中显示的文本。要使“提示”文本换行显示，可在换行处插入回车符(Chr(13))、换行符(Chr(10))（或系统符号常量 VbCrLf）或回车换行符(Chr(13)+Chr(10))，使显示的文本换行。

(2) “标题”指定对话框的标题。

(3) “默认值”用于指定输入框中显示的默认文本。

(4) xpos 和 ypos 分别指定对话框的左边和上边与屏幕左边和上边的距离。

2. 功能

该函数的作用是打开一个对话框，等待用户输入，当用户输入内容后单击“确定”按钮或 Enter 键时，函数返回输入的值，按“取消”按钮，返回的将是一个空字符串。其值的类型为字符串。

例如下列语句：

```
FileName = InputBox("请输入文件名（不超过8个字符）", "文件名", "vbfile")
```

将产生一个如图 4-6 所示的输入对话框，当用户在对话框中输入文本后，单击“确定”按钮，输入的文本将返回给变量 FileName。若用户单击“取消”按钮，返回的将是一个空字符串。

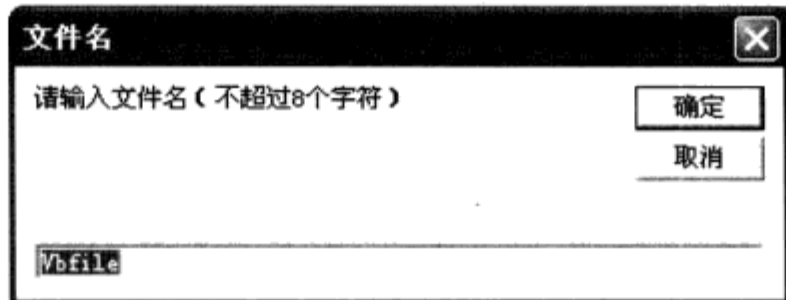


图 4-6 输入对话框

如果把上述语句改为：

```
FileName = InputBox("请输入文件名" + Chr(13) + "(不超过8个字符)", "文件名", "vbfile")
```

则“提示”信息分两行显示，如图 4-7 所示。

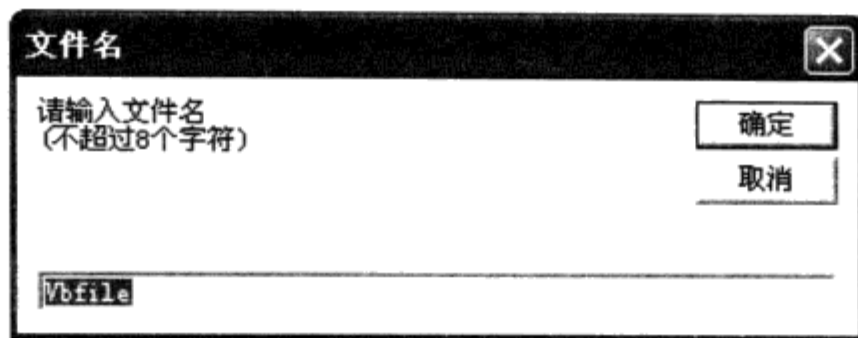


图 4-7 “提示”信息分行显示的输入对话框

应用实例

【实例 4-7】 通过输入对话框输入学生成绩，实现实例 4-6，运行界面如图 4-8 及图 4-9 所示。

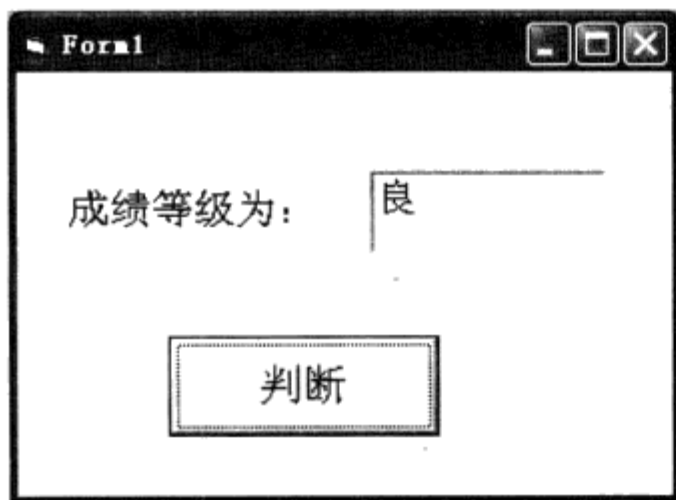


图 4-8 实例 4-7 运行界面

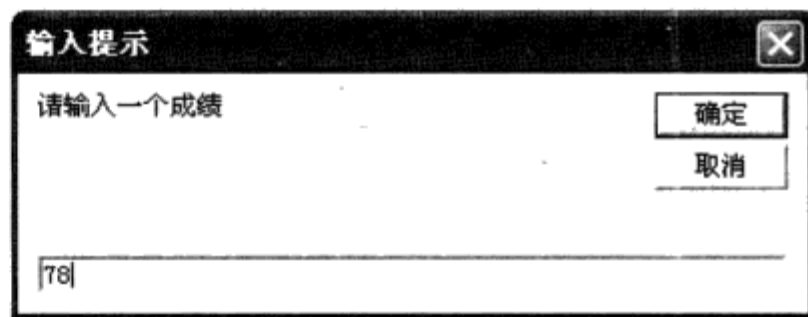


图 4-9 实例 4-7 输入成绩对话框

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加一个标签、一个文本框和一个命令按钮。

(2) 设置控件属性。

通过窗体的 Font 属性对话框将各控件的字号设置为小三，其余属性设置如图 4-8 所示。

(3) 编写程序代码。

在代码窗口编写如下代码。

```
Private Sub Command1_Click()
    Dim cj As Integer, dj As String
    cj = InputBox("请输入一个成绩", "输入提示")
    Select Case cj
        Case 0 To 59
            dj = "不及格"
        Case 60 To 75
            dj = "及格"
        Case 75 To 90
```

```

    dj = "良"
Case 90 To 100
    dj = "优"
Case Else
    dj = "成绩输入出错"
End Select
Text1.Text = dj
End Sub

```

4.3.2 消息对话框

📞 知识要点分析

1. MsgBox 函数的语法格式

<变量>=MsgBox("提示" [,对话框类型[, "对话框标题"]])

2. 功能

使用 MsgBox 函数可以产生一个对话框来显示消息（图 4-10）。当用户单击某个按钮后，将返回一个数值以标明用户单击了哪个按钮。

其中：

(1) “提示”指定在对话框中显示的文本。在“提示”文本中使用回车符(Chr(13))、换行符(Chr(10))或回车换行符(Chr(13)+Chr(10))可使显示的文本换行。

(2) “对话框标题”指定对话框的标题。

(3) “对话框类型”指定对话框中出现的按钮和图标，一般有 3 个参数，其取值和含义如表 4-1~表 4-3 所示。

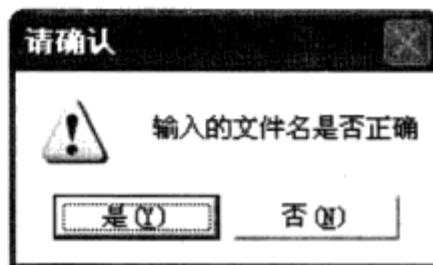


图 4-10 消息对话框

表 4-1 参数 1——出现按钮

符号常量	值	显示的按钮
VbOKOnly	0	“确定”按钮
VbOKCancel	1	“确定”和“取消”按钮
VbAbortRetryIgnore	2	“终止”、“重试”和“忽略”按钮
VbYesNoCancel	3	“是”、“否”和“取消”按钮
VbYesNo	4	“是”和“否”按钮
VbRetryCancel	5	“重试”和“取消”按钮

表 4-2 参数 2——图标类型

符号常量	值	显示的图标
VbCritical	16	显示停止图标
VbQuestion	32	显示问号(?)图标
VbExclamation	48	显示感叹号(!)图标
VbInformation	64	显示消息图标

表 4-3 参数 3——默认按钮

符号常量	值	默认的活动按钮
VbDefaultButton1	0	第一个按钮
VbDefaultButton2	256	第二个按钮
VbDefaultButton3	512	第三个按钮

这 3 种参数值决定了对话框的模式。可以把这些参数值（每组值只取一个）相加以生成一个组合的按钮参数值。例如：

```
S = MsgBox("输入的文件名是否正确", 4 + 48, "请确认")
```

显示的对话框如图 4-10 所示。

(4) MsgBox 返回值指明了用户在对话框中选择了哪一个按钮，如表 4-4 所示。

表 4-4 MsgBox 函数返回值

符号常量	返回值	对应按钮
VbOk	1	确定
VbCancel	2	取消
VbAbort	3	终止
VbRetry	4	重试
VbIgnore	5	忽略
VbYes	6	是
VbNo	7	否

(5) 选项中的值可以是数值，也可以是符号常量，例如：

```
X = VbYesNoCancel + VbQuestion + VbDefaultButton1
Y = MsgBox("输入的文件名是否正确", x, "请确认")
```

(6) 如果省略了某一项，必须加入相应的逗号分隔符，例如：

```
Y = MsgBox("输入的文件名是否正确", , "请确认")
```

(7) 若不需要返回值，则可以使用 MsgBox 的语句格式：

```
MsgBox "提示" [,对话框类型[, "对话框标题"]]
```

应用实例

【实例 4-8】 设计一个密码输入的简单检验程序。密码假定为“123456”，密码输入时在屏幕上不显示输入的字符，而以“*”代替。程序的运行界面如图 4-11 所示。当输入密码正确时，弹出如图 4-12 所示的消息框；当输入的密码不对时，弹出如图 4-13 所示的消息框，当用户单击“重试”按钮时，通过 MsgBox 函数返回值的条件判断将焦点定位在原文本框，等待用户重新输入密码；当用户单击“取消”按钮时，通过 MsgBox 函数返回值的条件判断弹出如图 4-14 所示的消息框，结束程序。

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加一个标签、一个文本框和一个命令按钮。

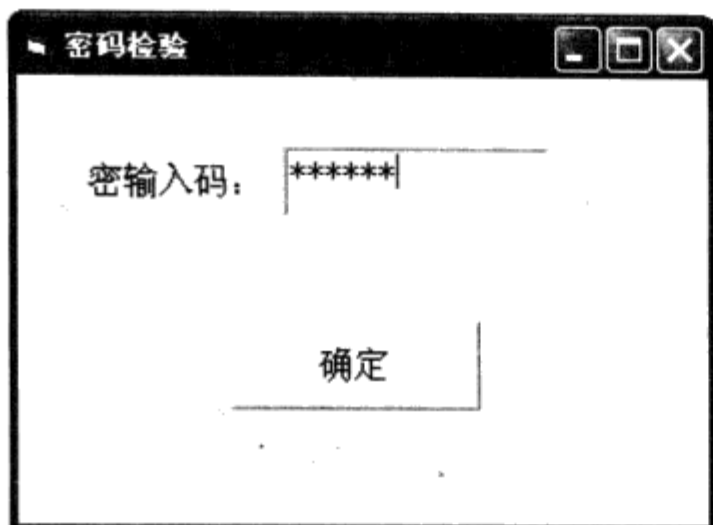


图 4-11 实例 4-8 运行界面

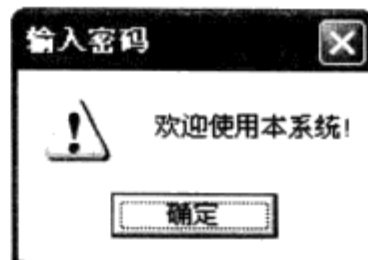


图 4-12 密码输入正确时的消息框

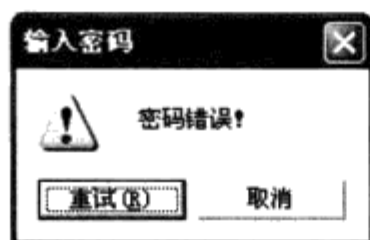


图 4-13 密码输入错误时的消息框

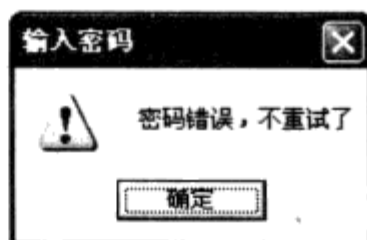


图 4-14 不重试密码时的消息框

(2) 设置控件属性。

通过窗体的 Caption 将窗体的标题设置为“密码检验”，通过文本框的 Password Char 属性将运行时密码输入显示“*”、MaxLengh 属性将文本框最多字符数设置为 6，其余属性设置如图 4-11 所示。

(3) 编写程序代码。

在代码窗口编写如下代码。

```
Private Sub Command1_Click()
Dim x As Integer
If Text1.Text = "123456" Then
    MsgBox "欢迎使用本系统!", 48, "输入密码" '在消息框上显示"!"图标
Else
    x = MsgBox("密码错误!", 5 + 48, "输入密码") '在消息框上显示"重试"和
                                                ' "取消"按钮以及"!"图标
                                                ' 4 表示单击了"重试"按钮
    If x = 4 Then
        Text1.Text = ""
        Text1.SetFocus '焦点定位在原输入的文本框中
    Else
        MsgBox "密码错误, 不重试了", 48, "输入密码" '在消息框上显示"!"图标
    End
End If
End Sub
```

4.4 选择性控件

4.4.1 单选按钮

☞ 知识要点分析

单选按钮 (OptionButton) 作为常用的输入控件之一, 通常以成组的形式出现。在程序运行时, 通过单选按钮组可为用户的操作提供多种可能。在任何时候, 一个单选按钮组必须选中且仅能选中一个选项, 即当组中某个按钮被选择时, 这个组内其他按钮的选择会由系统控制自动放弃。因此, 单选按钮可以用于在多种选项中由用户选择其中一项的情况。

1. 常用属性

(1) Caption 属性: 设置或返回单选按钮的标题。

(2) Alignment 属性: 设置标题是显示在按钮左侧 (值为 0, 默认值) 还是右侧 (值为 1)。

(3) Value 属性: 设置或返回该选项是否被选中。选中时 Value 值为 True, 否则为 False。

(4) Style 属性: 设置选项按钮的外观, 默认值为 0, 表示标准方式; 其值为 1 时, 表示以图形方式显示单选按钮。

2. 事件

单选按钮最重要的事件就是单击 (Click) 事件。当运行时单击单选按钮, 或在代码中改变单选按钮的 Value 属性值 (从 False 改变为 True), 将触发 Click 事件。在应用程序中可创建一个事件过程, 检测控件对象 Value 属性值, 再根据检测结果执行相应的处理。

✍ 应用实例

【实例 4-9】 设计一个程序, 通过单选按钮更改标签字体。程序运行界面如图 4-15 所示。

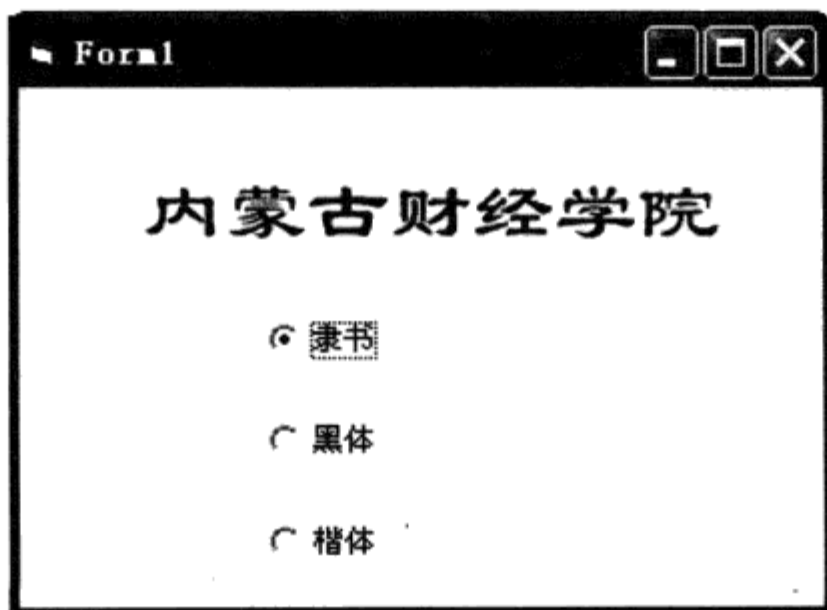


图 4-15 实例 4-9 运行界面

“内蒙古财经学院”，通过设置 `AutoSize` 的属性值为 `True` 使标签能随内容自动伸缩，通过 `Font` 属性将字号设置为小 1 号；其余属性设置如图 4-16 所示。

(3) 编写程序代码。

在代码窗口编写如下代码。

```
Private Sub Check1_Click()
If Check1.Value = 1 Then           '判断复选框 1 是否选中
    Label1.FontName = "楷体_GB2312" '选中时设置为"楷体_GB2312"字体
Else
    Label1.FontName = "宋体"       '不选中时设置为"宋体"字体
End If
End Sub

Private Sub Check2_Click()
If Check2.Value = 1 Then           '判断复选框 2 是否选中
    Label1.FontItalic = True       '选中时设置为"斜体"字形
Else
    Label1.FontItalic = False
End If
End Sub

Private Sub Check3_Click()
If Check3.Value = 1 Then           '判断复选框 3 是否选中
    Label1.FontSize = 25           '选中时设置为"25"号字
Else
    Label1.FontSize = 9            '不选中时设置为"9"号字
End If
End Sub

Private Sub Check4_Click()
If Check4.Value = 1 Then           '判断复选框 4 是否选中
    Label1.ForeColor = RGB(255, 0, 0) '选中时设置为"红色"
Else
    Label1.ForeColor = RGB(0, 0, 0)  '不选中时设置为"黑色"
End If
End Sub
```

4.5 框架控件

知识要点分析

框架 (Frame) 是常用的控件容器之一。在窗体中的控件可以借助于框架分成多个可标识的控件组。

1. 常用属性

框架的常用属性有 `Name` 属性和 `Caption` 属性。

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加一个标签、三个单选按钮。

(2) 设置控件属性。

通过标签的 Caption 将标题设置为“内蒙古财经学院”，通过设置 AutoSize 的属性值为 True 使标签能随内容自动伸缩，通过 Font 属性将字号设置为小 1 号；其余属性设置如图 4-15 所示。

(3) 编写程序代码。

在代码窗口编写如下代码。

```
Private Sub Option1_Click()  
Label1.FontName = "隶书"  
End Sub  
Private Sub Option2_Click()  
Label1.FontName = "黑体"  
End Sub  
Private Sub Option3_Click()  
Label1.FontName = "楷体_GB2312"  
End Sub
```

4.4.2 复选框

☎ 知识要点分析

复选框 (CheckBox) 作为输入控件，可为用户提供选项并显示该选项是否被选中，通常以控件数组的形式出现。与单选按钮不同的是，位于同一个复选框组中的这些组项，彼此之间互不矛盾。在运行时，用户可根据自己的需要，从提供的多个选项选择一个或多个。

复选框的属性和事件与单选按钮完全相同的，在此不在赘述。只有 Value 属性有三种取值：0——未选中（默认值），1——选中，2——不可用。

✍ 应用实例

【实例 4-10】 设计一个程序，用复选框来控制标签中文字的字体、字形、字号及颜色。程序运行界面如图 4-16 所示。

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加一个标签、4 个复选框。

(2) 设置控件属性。

通过标签的 Caption 属性将标题设置为

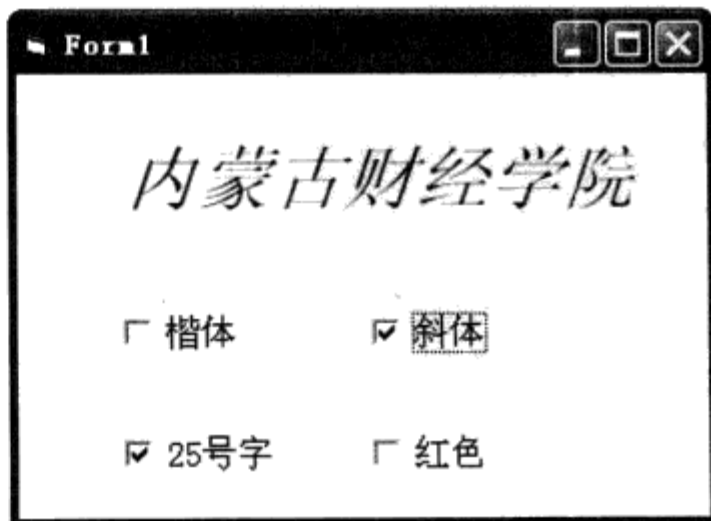


图 4-16 实例 4-10 运行界面

2. 事件

框架可以响应 Click 和 DblClick 事件。在应用程序中一般不需要编写有关框架的事件过程。

应用实例

【实例 4-11】 设计一个程序，通过框架分组控制标签中文本的字体、字号、字形。程序运行界面如图 4-17 所示。

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加一个标签、3 个框架和一个命令按钮，3 个框架中分别添加 2 个单选按钮和 2 个复选框。

(2) 设置控件属性。

通过标签的 Caption 将标题设置为“内蒙古财经学院”，通过设置 AutoSize 的属性值为 True 使标签能随内容自动伸缩，其余属性设置如图 4-17 所示。

(3) 编写程序代码。

在代码窗口编写如下代码。

```
Private Sub Option1_Click()
    Label1.FontName = "隶书"
End Sub
Private Sub Option2_Click()
    Label1.FontName = "黑体"
End Sub
Private Sub Option3_Click()
    Label1.FontSize = 20
End Sub
Private Sub Option4_Click()
    Label1.FontSize = 25
End Sub
Private Sub Check1_Click()
    If Check1.Value = 1 Then
        Label1.FontItalic = True
    Else
        Label1.FontItalic = False
    End If
End Sub
Private Sub Check2_Click()
    If Check2.Value = 1 Then
        Label1.FontBold = True
    Else
```

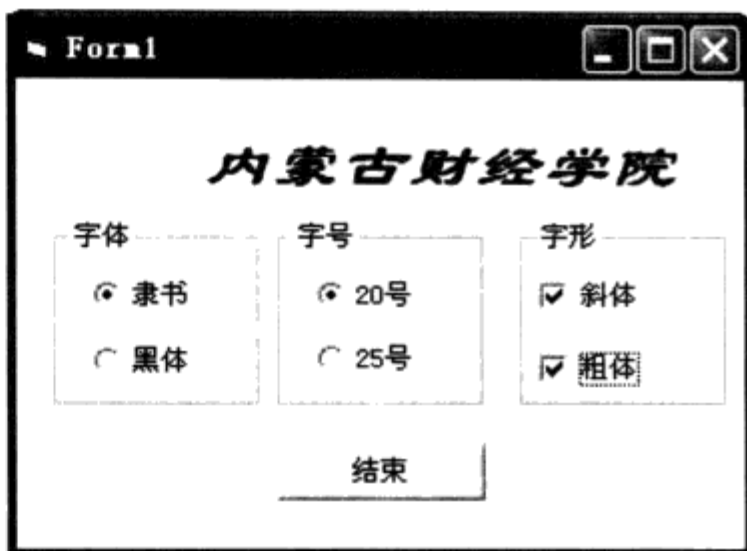


图 4-17 实例 4-11 运行界面

```
'判断复选框 1 是否选中
'选中时设置为"斜体"字形
'判断复选框 2 是否选中
'选中时设置为"粗体"字形
```

```

    Label1.FontBold = False
End If
Private Sub Command1_Click()
End
End Sub

```

4.6 计时器控件

☞ 知识要点分析

计时器 (Timer) 也是工具箱中的一个标准控件, 它每隔一定的时间产生一次 Timer 事件 (或称报时), 可根据这个特性来定时控制某些操作或进行计时。

计时器控件在设计时显示为一个小时钟图标, 在运行时不显示在屏幕上, 通常另设标签或文本框来显示时间。计时器的默认名称为 Timer1, Timer2, …。

1. 常用属性

(1) Enabled 属性: 确定计时器是否可用。默认值为 True; 当设置为 False 时, 表示不可用, 此时计时器不产生任何事件。

(2) Interval 属性: 设置两个 Timer 事件之间的时间间隔, 其值以毫秒 (1ms=1/1000s) 为单位, 取值范围为 0~65 535。例如, 如果希望每隔半秒钟产生一个 Timer 事件, 那么 Interval 属性值应该设置为 500, 这样, 每隔 500ms 就会触发一次 Timer 事件, 从而执行相应的 Timer 事件过程。若 Interval 属性值设置为 0 (默认), 表示计时器不可用。

2. 事件

计时器只响应一个 Timer 事件。也就是说, 计时器控件对象在间隔了一个 Interval 设定时间后, 触发一次 Timer 事件。

✍ 应用实例

【实例 4-12】 建立一个电子时钟。程序窗体设计界面如图 4-18 所示, 程序运行界面如图 4-19 所示。

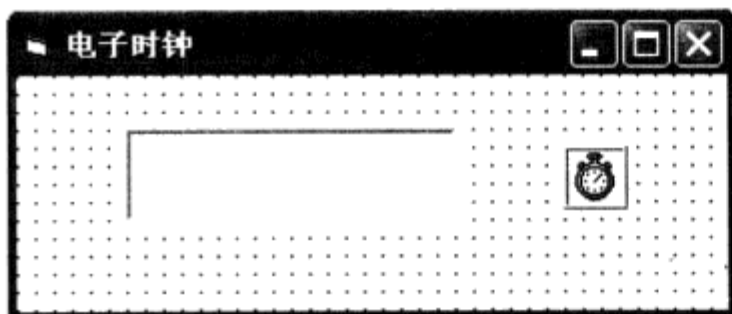


图 4-18 实例 4-12 设计界面

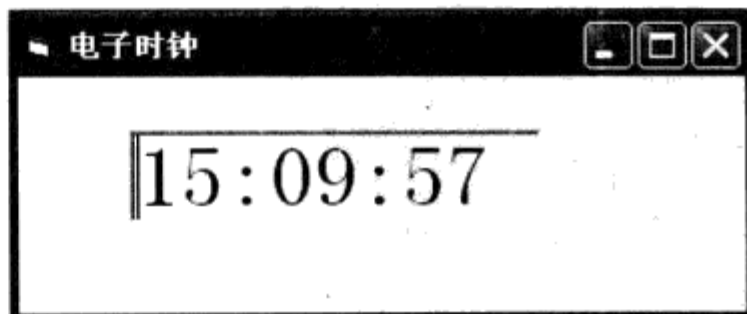


图 4-19 实例 4-12 运行界面

设计步骤如下。

(1) 设计程序界面。

新建一个工程, 在 Form1 窗体中添加一个文本框和一个计时器控件。

(2) 设置控件属性。

把窗体的 Caption 属性设置为“电子时钟”；将计时器的 Interval 属性值设定为 1000；将文本框的 Text 属性设置为空，字体“大小”设定为 28，如图 4-18 所示。

(3) 编写程序代码。

在代码窗口编写如下代码。

```
Private Sub Timer1_Timer() '每隔 1 秒触发 1 次 Timer 事件，并执行本事件过程
    Text1.Text = Time      'Time 是 VB 系统时间函数
End Sub
```

程序运行后，即可见到如图 4-19 所示的电子时钟。

【实例 4-13】 通过计时器实现文字字号的逐步放大。程序的窗体设计界面如图 4-20 所示，运行界面如图 4-21 所示。

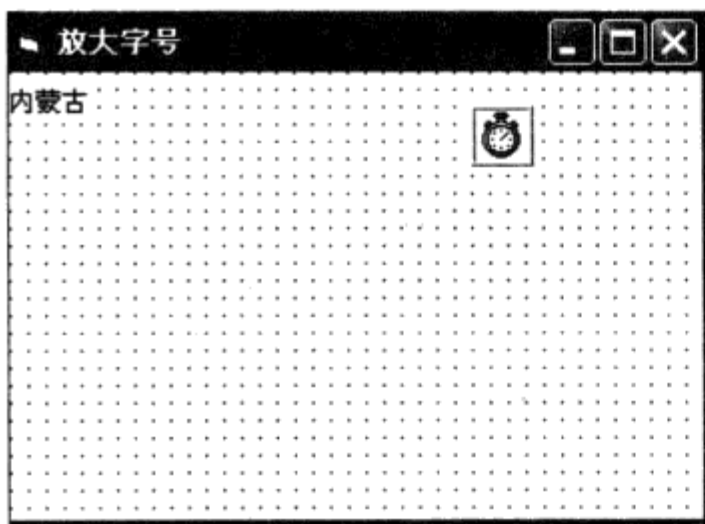


图 4-20 实例 4-13 的设计界面

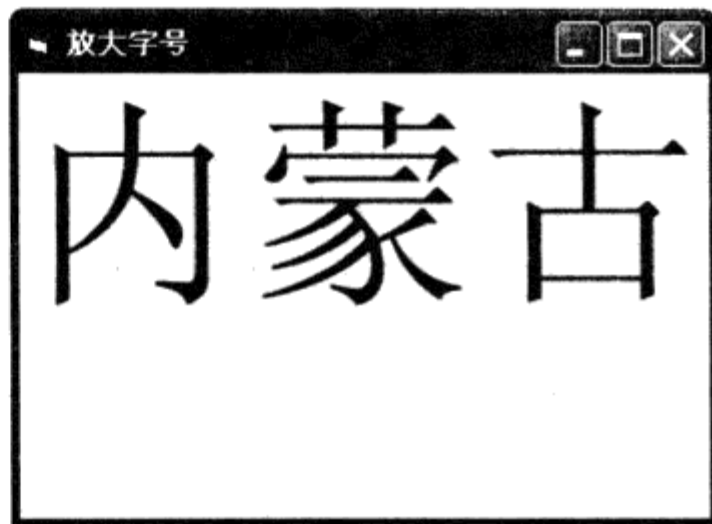


图 4-21 实例 4-13 的运行界面

设计步骤如下。

(1) 设计程序界面。

新建一个工程，在 Form1 窗体中添加一个标签和一个计时器控件。

(2) 设置控件属性。

把窗体的 Caption 属性设置为“放大字号”；将计时器的 Interval 属性值设定为 500；将标签的 Caption 属性设置“内蒙古”，将 AutoSize 属性设置为 True，如图 4-20 所示。

(3) 编写程序代码。

在代码窗口编写如下代码。

```
Private Sub Timer1_Timer()
    If Label1.FontSize < 70 Then
        Label1.FontSize = Label1.FontSize * 1.2 '每次放大 1.2 倍
    Else
        Label1.FontSize = 8 '恢复为 8 号字
    End If
End Sub
```

在程序运行时，每隔半秒触发一次 Timer 事件，每次放大字号 1.2 倍，如果字号大于或等于 70，就把字号恢复为 8，运行界面如图 4-21 所示。

本章小结

本章通过条件语句、多分支语句及输入对话框、消息对话框等对话框的使用和单选按钮、复选框等选择性控件介绍了选择结构的程序设计方法，并介绍了框架控件、计时器控件的使用。

习 题 4

一、判断题

1. 单选按钮组中所有按钮都采用相同的名称 (Name)。 ()
2. 单选按钮的 Value 属性能确定该按钮是否被选中。 ()
3. 一个窗体上的所有单选按钮一次只能有一个被选中。 ()
4. 在运行期间用鼠标单击单选按钮时，按钮的 Value 属性值变为 True。 ()
5. 在代码中采用语句 Option1.Value=True，把单选按钮 Option1 的 Value 属性值变为 True，将会触发 Click 事件。 ()
6. 计时器控件能有规律地以一定时间间隔触发 Timer 事件，并执行该事件过程的程序代码。 ()
7. 可以设置计时器的 Visible 属性使其在运行时可见。 ()
8. 可以在窗体上设置计时器的大小。 ()
9. 计时器可以识别 Click 事件。 ()
10. 如果计时器的 Interval 属性值为 0，这计时器无效。 ()
11. 窗体上的复选框可以都不选中。 ()
12. 放置在不同框架控件中的单选按钮可以同时被选中。 ()

二、单选题

1. 关于语句 "If x=0 Then s=0"，下列说法正确的是 ()。

A. x 必须是逻辑型变量	B. S 不能是逻辑型变量
C. x=0 是关系表达式，s=0 是赋值语句	D. x=0 是赋值语句，s=0 是关系表达式
2. 下列程序段的执行结果是 ()。

```
a = "abcde": b = "cdefg"
c = Right(a, 3): d = Mid(b, 2, 3)
If c < d Then y = c + d Else y = d + c
Print y
```

- | | | | |
|-----------|-----------|-----------|-----------|
| A. abcdef | B. cdebcd | C. cdeefg | D. cdedef |
|-----------|-----------|-----------|-----------|
3. 执行下列程序段后，变量 x 的值是 ()。

```
x = -3
```

```
If Abs(x) <= 2 Then x = x - 1 Else x = x + 8
Select Case x
    Case Is < 5
        x = x + 1
    Case 5, 7, 9
        x = x + 2
    Case 10 To 15
        x = x + 3
    Case Else
        x = x + 4
End Select
Print x + 1
```

A. 8 B. 7 C. 15 D. 6

4. 如果要使计时器每分钟触发一个 Timer 事件, 则 Interval 属性应该设置为 ()。

A. 10 B. 100 C. 1000 D. 10 000

5. 一个单选按钮的标题为“Open”, 如要为其设置访问键 P, 可将其 Caption 属性值设置为 ()。

A. P B. Open (&P) C. Open+P D. Open

三、编程及上机调试

设计程序, 利用 3 个复选框分别代表红、绿、蓝 3 种颜色值, 当选中复选框时表示颜色值为 255, 不选中时表示颜色值为 0, 把通过 RGB 函数调配的颜色作为标签的背景色, 程序运行界面如图 4-22 所示。



图 4-22 习题三运行界面

第5章

循环结构程序设计

循环结构就是用来重复执行某个程序段的结构。在编写程序的过程中，常常会遇到某一段代码需要反复执行若干次以完成某种功能，或按指定条件，重复执行一组语句的操作。例如：计算 $1+2+3+4+\dots+N$ 的值，使用顺序结构写程序代码如下。

```
Private Sub Command1_Click()  
    Dim n As integer,s As long  
    n=Val (InputBox ("请输入 n"))  
    s=s+1:s=s+2  
    :  
    S=s+n  
    Print "s="; s  
End Sub
```

采用这种顺序结构，计算前 n 项和，需要 n 个赋值语句，其实这样做完全没有必要，因为循环结构就可以用来解决这个问题。

5.1 循环语句

Visual Basic 提供了几种不同的循环结构语句，这里只重点介绍 For...Next 语句和 Do...Loop 语句。

5.1.1 For...Next 循环语句

☞ 知识要点分析

For...Next 循环语句是计数型循环语句（又称其为步长型循环语句），它使用一个计数器变量（循环变量），每重复一次循环后，循环变量的值就会自动增加或减少。适用于循环次数预知的情况。

1. 格式

```
For<循环变量>=<初值>To<终值>[Step<步长>]  
    <循环体语句序列>  
Next<循环变量>
```

2. 功能

根据循环变量值的变化，重复执行循环体语句序列，直到循环变量的值超出了初值和终值确定的范围。

3. 说明

(1) For 语句称为循环起始语句，Next 语句称为循环终端语句。

(2) For...Next 循环必须要有一个循环变量，每循环一次，该变量的值就会增加或减少步长确定的值。循环变量必须为数值型变量。

(3) Step 步长可以省略，如果省略系统默认步长为 1。步长为数值型表达式，其值可正可负，若步长为正，则初值应小于等于终值，否则不执行循环体语句；若为负，则初值应大于等于终值，否则不执行循环体的语句。若为 0，则形成无限循环（又称“死循环”），此时按下 Ctrl+Break 组合键可以结束程序的运行。

(4) 在正常循环的情况下，循环的次数由表达式 $\text{Int}((\text{终值}-\text{初值}) / \text{步长}+1)$ 确定。

4. 执行步骤

首先把“初值”赋给循环变量，如果步长为正，则测试循环变量的当前值是否大于终值，如果大于则结束循环，执行 Next 的后续语句；否则执行循环体语句序列，遇到 Next 语句，将循环变量的当前值加步长，再赋给循环变量，转向 For 语句继续上一过程。如果步长为负，则测试循环变量的当前值是否小于终值，如果小于则结束循环，执行 Next 的后续语句；否则执行循环体语句序列，遇到 Next 语句，将循环变量的当前值减去步长，再赋给循环变量，转向 For 语句继续上一过程。

应用实例

【实例 5-1】 在窗体上添加 2 个标签 (Label1 和 Label2)，2 个文本框 (Text1 和 Text2)，2 个命令按钮 Command1 和 Command2。界面如图 5-1 所示。编写适当的代码，程序运行时在 Text1 中输入一个整数，单击 Command1（计算）在 Text2 中计算出从 1 累加到该数的和。单击 Command2（退出）结束程序的运行。

主要操作步骤：

(1) 设计界面。

新建工程和窗体，在窗体上添加相应的控件。设置控件的属性：窗体的 Caption 属性值为“For...Next 循环实例”；Label1 和 Label2 的 Caption 属性分别为“请输入一个数(n):”和“For 循环计算 1-n 的和为:”；Text1 和 Text2 的 Text 属性值为空；Command1 和 Command2 的 Caption 属性值为“计算”和“退出”。

(2) 代码如下（步长省略，默认为 1）。

```
Private Sub Command1_Click()
    Dim n As Integer, s As Integer
    n = 0: s = 0
    n = Val(Text1)
```

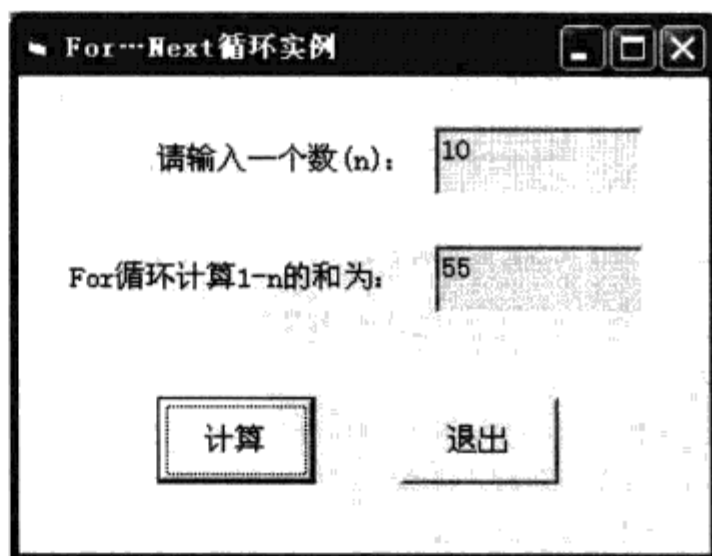


图 5-1 For 循环实例求和界面

```

    For i = 1 To n
        s = s + i
    Next i
    Text2 = s
End Sub

Private Sub Command2_Click()
    End
End Sub

```

上面 Command1 的单击事件中，如果步长为-1，代码可以这样写：

```

Private Sub Command1_Click()
    Dim n As Integer, s As Integer
    n = 0: s = 0
    n = Val(Text1)
    For i = n To 1 step -1
        s = s + i
    Next i
    Text2 = s
End Sub

```

【实例 5-2】 在窗体上添加 1 个标签 Label1，1 个文本框 Text1，2 个命令按钮 Command1 和 Command2。界面如图 5-2 所示。编写适当的代码，程序运行时单击 Command1（计算）弹出输入对话框，输入一个整数 n（本例输入 4），确定后在 Text1 文本框中计算出 n 的阶乘。单击 Command2（退出）结束程序的运行。



图 5-2 For 循环实例求阶乘界面

分析：该例中的窗体和窗体上添加控件的属性设置如图 5-2 所示，这里不再赘述。在写代码的时候需要注意，存放阶乘结果的变量 s，初值应该为 1，而不能为 0，这是和实例 5-1 中求和相区别的一点。

编写代码如下。

```

Private Sub Command1_Click()
    Dim n As Integer, s As Integer
    s = 1
    n = Val(InputBox("请输入一个整数"))
    For i = 1 To n
        s = s * i
    Next i
    Text1 = s
End Sub

Private Sub Command2_Click()
    End
End Sub

```

5.1.2 Do...Loop 循环语句

知识要点分析

Do...Loop 循环结构是适用于事先不知道循环次数的一种循环结构，其有两种不同的循环控制格式：前测型和后测型。

1. 前测型 Do While|Until...Loop 循环控制语句

(1) 格式：

```
Do While|Until 条件表达式  
    <循环体语句序列>  
Loop
```

(2) 执行过程与功能说明。

- Do While...Loop 首先判断条件表达式的值（因此叫前测型），如果条件为真（True），执行循环体语句序列，遇到 Loop 则返回到 Do While 继续判断条件，如此反复。当条件为假（False）时跳出循环，执行 Loop 后面的语句。如果条件一开始就为假则循环一次也不被执行；而如果条件一直为真就会形成无限循环（死循环）。
- Do Until...Loop 首先也是判断条件表达式的真假，只是 Until 和 While 判断条件正好相反，Until 条件为假时执行循环体语句序列，为真时结束循环。

2. 后测型 Do...Loop While|Until 循环控制语句

(1) 格式：

```
Do  
    <循环体语句序列>  
Loop While|Until <条件表达式>
```

(2) 执行过程与功能说明。

先无条件执行循环体语句序列（因此叫后测型），直到 Loop 时才判断条件表达式的值，如果是 Loop While，则条件为真，返回到 Do 继续循环，否则结束循环。如果是 Loop Until，则条件表达式的值为假，程序返回 Do 语句继续循环，否则跳出循环。跳出循环后执行循环体下面的后续语句。

应用实例

【实例 5-3】 分别用前测型的 While 和 Until 条件实现实例 5-1。

分析：Do...Loop 循环结构和 For...Next 循环不同，Do...Loop 循环中没有一个像 For...Next 中的循环自增或自减的变量，因此设计 Do...Loop 循环要考虑人为地去设计这样一个能最终导致循环跳出的变量。另外一个需要注意的就是，While 和 Until 是两个完全相反的条件。

Command1 的鼠标单击事件过程的代码设计如下。

```
Private Sub Command1_Click()  
    Dim n as integer, k As Integer, s As Long
```

```

n = Val(Text1)
s = 0: k = 1
Do While k <= n    '如果是 Until 条件, 则为 Do Until k>n 语句
    s = s + k
    k = k + 1      'k 就是一个人为设计的自增变量, 当 k 增加到大于 n 时便跳出循环
Loop
Text2 = s
End Sub

```

思考:

该程序中用 Do Until k>n 替换 Do While k <= n 语句对程序的结果没有影响。如果是用 Do Until k<=n 语句替换结果有什么不同? 程序的执行过程发生了什么变化?

s=s+k 和 k=k+1 两条语句的顺序对程序的执行结果有没有影响? 为什么?

【实例 5-4】 分别用后测型的 While 和 Until 条件实现实例 5-2。

分析: 后测型 Do...Loop 循环和前测型功能基本相同, 不同之处在于如果条件在一开始就不满足, 前测型循环一次也不能执行, 而后测型无论条件的值为真还是假, 都要无条件执行一次循环。

Command1 命令按钮的鼠标单击事件代码如下。

```

Private Sub Command1_Click()
    Dim n As Integer, k As Integer, s As Long
    n = Val(InputBox("请输入一个整数 n"))
    s = 1: k = 1
    Do
        s = s * k
        k = k + 1
    Loop While k <= n    '采用 Loop Until k>n 语句, 其程序的功能与原程序相同
    Text1 = s
End Sub

```

5.1.3 循环出口语句

知识要点分析

在 For...Next 循环的循环体语句序列中有一个[Exit For]选项, 在 Do...Loop 循环的循环体语句序列中有 Exit Do 可选项, 执行循环的过程中如果遇到这两条语句则跳出循环, 执行 Next 后面或 Loop 后面的语句。这两个可选项一般都是用在条件判断 If...Then 中或者 Select Case 语句中。

应用实例

【实例 5-5】 新建工程和窗体, 在窗体上添加两个标签 (Label1 和 Label2)、两个文本框 (Text1 和 Text2) 和一个命令按钮 Command1, 属性设置如图 5-3 所示。程序运行时, 在 Text1 文本框中输入一个大于 1 的正整数, 单击“开始判断”命令按钮, 在 Text2 中输

出判断的结果。

分析：首先要明白什么是质数，这是算法的本质。质数又称素数，是指一个大于1的正整数，除了1和它本身外，不能被其他自然数整除。比1大但不是质数的正整数称为合数。2是最小的质数，1和0既不是质数也不是合数。

然后设计算法判断，判断一个数 m 是不是质数，就要用 m 分别除以 2, 3, 4, ..., $m-1$ ，若所有的数都不能被 m 整除，则说明 m 是质数。但是考虑到如果一个数是合数，那么一定可以由两个自然数相乘得到，其中一个大于或等于它的平方根，一个小于或等于它的平方根。并且成对出现。因此判断一个数是否是质数，则用这个数分别去除以 2 到其平方根就可以了。

Command1 的单击事件过程如下。

```
Private Sub Command1_Click()
    Dim i As Integer, k As Long, flag As Boolean
    k = Val(Text1)
    flag = True
    For i = 2 To Int(Sqr(k))
        If k Mod i = 0 Then
            flag = False
            Exit For '只要能被整除即可判断不是素数，则跳出循环
        End If
    Next i
    If flag = True Then
        Text2 = "是质数"
    Else
        Text2 = "不是质数"
    End If
End Sub
```

【实例 5-6】 用 Do...Loop 循环实现实例 5-5。
程序代码如下。

```
Private Sub Command1_Click()
    Dim i As Integer, k As Long, flag As Boolean
    k = Val(Text1)
    flag = True
    i = 2
    Do While i <= Int(Sqr(k))
        If k Mod i = 0 Then
            flag = False
            Exit Do '只要能被整除即可判断不是素数，则跳出循环
        End If
        i = i + 1
    Loop
```

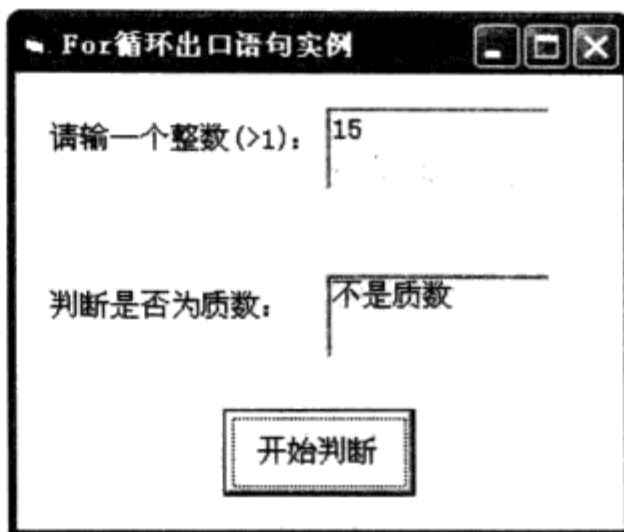


图 5-3 For 循环出口语句实例程序界面

```

        End If
        i = i + 1
    Loop
    If flag = True Then
        Text2 = "是质数"
    Else
        Text2 = "不是质数"
    End If
End Sub

```

5.2 循环嵌套

☞ 知识要点分析

在编程的过程中，经常会用到顺序结构中有循环，循环结构中又嵌套其他循环结构，或嵌套分支结构的情况。各种程序控制结构的综合使用才能使编程更加顺畅、结构更清晰、功能更强大。如果一个循环结构的内部又包含一个完整的循环结构，则称其为循环嵌套。循环可以嵌套多层，这种循环的多层嵌套又叫多重循环。常见的是二重循环或三重循环。各种结构嵌套时需要注意以下问题。

- (1) 内循环和外循环的循环变量不能同名，否则会提示“编译错误”。
- (2) 外循环必须包含完整的内循环，或者完整的分支嵌入，不能交叉。

✎ 应用实例

【实例 5-7】 计算并输出 100~200 之间的所有质数，窗体上有一个标签 Label1，一个可以显示多行文本的文本框，有垂直滚动条，一个命令按钮 Command1。控件属性设置后的界面如图 5-4 所示。

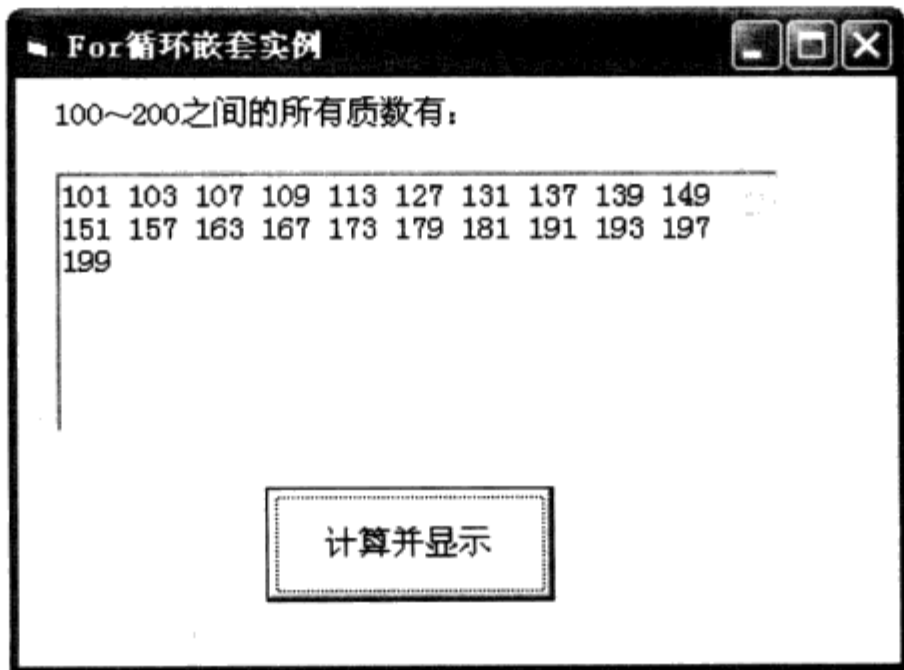


图 5-4 For 循环嵌套实例程序界面

程序代码如下。

```
Private Sub Command1_Click()  
Dim i As Integer, j As Integer, flag As Boolean  
For i = 100 To 200  
    flag = True  
    For j = 2 To Int(Sqr(i))  
        If i Mod j = 0 Then  
            flag = False  
            Exit For  
        End If  
    Next j  
    If flag = True Then Text1 = Text1 & i & " "  
Next i  
End Sub
```

【实例 5-8】 深入了解 For…Next 循环中循环变量、初值和终值的变化。新建一个工程和窗体，在窗体的鼠标单击事件中编写如下代码。

```
Private Sub form_click()  
Dim a, b, c, x, y, z As Integer  
For a = 1 To 7  
    For b = 1 To 8  
        For c = 1 To 9  
            x = x + 1  
        Next c  
        y = y + 1  
    Next b  
    z = z + 1  
Next a  
Print "a="; a, "b="; b, "c="; c  
Print "x="; x, "y="; y, "z="; z  
End Sub
```

程序的运行结果是：

```
a= 8      b= 9      c=10  
x= 504    y=56      z= 7
```

【实例 5-9】 计算并显示九九乘法表。

创建工程和窗体，在窗体的合适位置放一个可显示多行的文本框。一个命令按钮 Command1，设置属性，界面如图 5-5 所示。程序运行时单击命令按钮，在文本框上方位置显示“-----九九乘法表-----”，在文本框中显示九九乘法表。

程序代码如下。

```
Private Sub Command1_Click()  
Dim i As Integer, j As Integer
```

```

Print
Print Tab(23); "-----九九乘法表-----" 'Tab 定位输出位置为第 23 列
For i = 1 To 9
    For j = 1 To i
        Text1 = Text1 & i & "*" & j & "=" & j * j & " "
    Next j
    Text1 = Text1 & vbCrLf '文本框中换行
Next i
End Sub

```

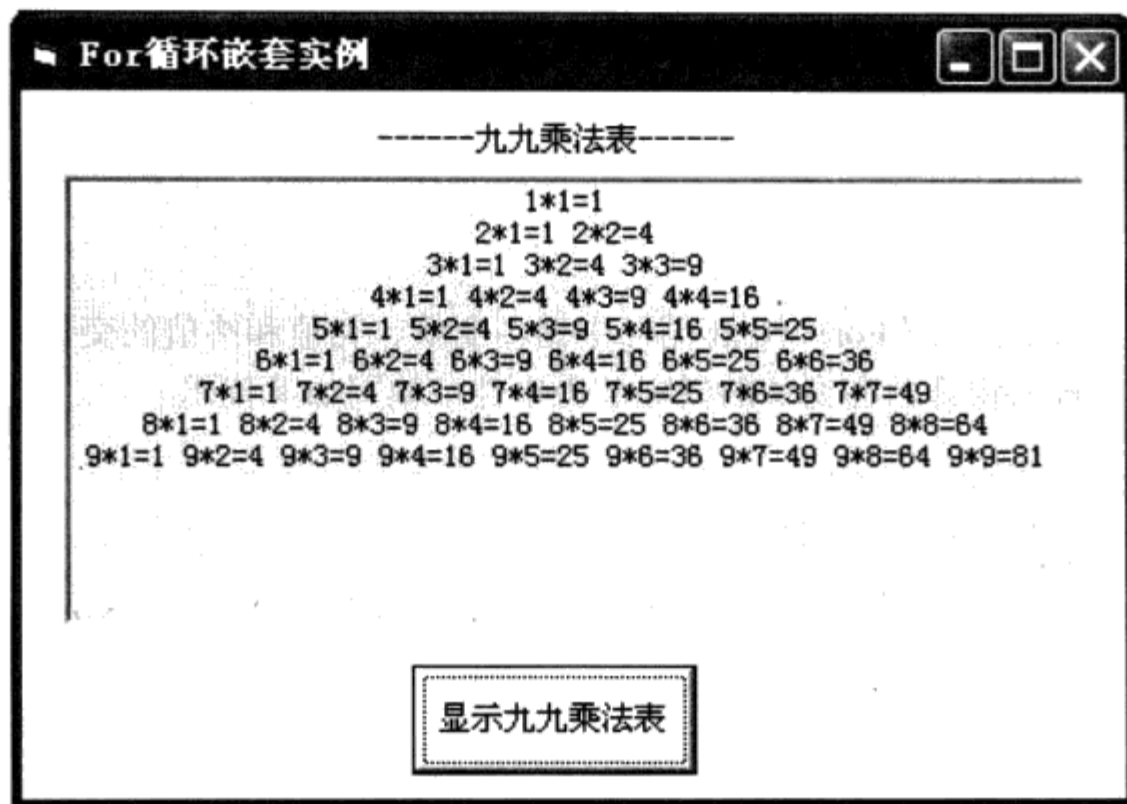


图 5-5 程序运行界面

【实例 5-10】 深入了解 Do 循环嵌套程序执行过程。

新建工程和窗体，在窗体上添加一个命令按钮 Command1，该命令按钮的鼠标单击事件过程代码如下。

```

Private Sub Command1_Click()
    Dim a As Integer, b As Integer
    Do While a <= 1 '外循环
        Print "&";
        b = 0 '重置变量 b 的值
        Do While b <= 1 '内循环
            Print "***";
            b = b + 1
        Loop
        a = a + 1
    Loop
    Print
    Print "a="; a; "b="; b
End Sub

```

程序分析：程序的执行结果为分两行显示，第一行显示“&*****&*****”，第二行显示变量 a 和 b 的值都是 2。外层的 Do 循环执行了 2 次，内层的 Do 循环执行了 4 次。

【实例 5-11】 新建工程和窗体，在窗体上添加一个命令按钮，在命令按钮的鼠标单击事件中书写如下代码，读程序，分析程序运行的结果。

```
Private Sub Command1_Click()  
Dim i As Integer, flag As Boolean  
flag = False  
i = 1  
Do Until flag = True      '外循环，直到 flag 的值为真时结束外循环  
    Do Until i > 100      '内循环  
        i = i + 1  
        If i = 20 Then  
            flag = True  
            Exit Do      '退出内循环  
        End If  
    Loop  
Loop  
Print i, flag  
End Sub
```

这是一个 Do...Loop 循环的二重嵌套，内循环中又嵌套了分支。外循环的结束条件是 flag 为真，内循环的结束条件是 i>100，所以正常应该循环 100 次，但是因为内嵌了分支，当 i=20 时，将 True 赋值给 flag 变量，然后跳出内循环，返回到外循环的 Do 语句判断，Until 条件为真跳出外循环。所以程序运行的结果输出 i 的值为 20，flag 的值为 True。

5.3 数组

5.3.1 数组的概念和声明

数组是一类特殊的变量，用来处理大量和比较复杂的数据。而且数组变量能够反映出的一组变量之间的关系，比如处理一个班级学生的学号、姓名、成绩等，数组变量分别是学号、姓名和成绩，而每个学生具体的学号、姓名和成绩则分别构成一组数组元素变量。

知识要点分析

1. 数组的概念

数组是一组同类变量的集合，如果用 a 表示一个数组变量，则 a(0)，a(1)，a(2)，…分别用来表示数组的不同元素，这些具有同一变量名、不同下标的下标变量集合就称为数组。

正确理解和使用数组要注意以下几个问题。

(1) 数组名是数组变量的名称，它的命名和普通变量相同，符合 Visual Basic 的标识符规则。但需要注意，数组名不是一个单个变量，它代表一组变量。

(2) 数组元素就是数组名代表的那一组变量，一般具有相同的名称和数据类型，用下标进行标识。数组的下标表示数组元素在数组中的位置，下标必须放在数组名之后的圆括号中。

(3) 下标的个数决定数组的维数，有一个下标为一维数组，有两个或多个下标，称其为二维数组或多维数组。当有多个下标时，下标之间用半角逗号隔开。

(4) 数组也遵循变量的先声明后使用的原则，声明数组就是要说明数组名、数组元素的数据类型、数组的维数以及每一维的上下界。上界指可使用下标的最大值，下界指的是可使用下标的最小值。

2. 数组的声明

语法格式：

```
Dim 数组名(下标[,...]) As 类型名称
```

功能：声明数组名，数组的维数、数组中可使用的数组元素的个数、数据类型。

说明：

- 下标的格式为：[下界 To]上界，如果省略了下界，系统默认 0；或者在代码编辑窗口，单击对象下拉列表框选择“通用”，在声明段输入“Option Base 0|1”语句来确定数组下标默认的下界为 0 或为 1，这样声明后，本模块中用到的所有数组省略下界都取该默认下界。
- 上下界必须是数值表达式，范围为 Long 数据类型内的整数值，若上下界为小数时，系统自动取整。
- 数组中的所有元素具有相同的数据类型，但当数组被声明为变体型 Variant 时，数组中的元素可以有不同的数据类型。

下面是一些声明数组的例子。

Dim a(3) As Integer	'一维数组 a, 从 a(0) 到 a(3) 有 4 个整型元素
Dim b(1 To 3) As Single	'一维数组 b, 从 b(1) 到 b(3) 有 3 个单精度型元素
Dim c(2, 3) As String	'二维数组 c, 从 c(0,0) 到 c(2,3) 有 12 个字符型元素
Dim d(2, -1 To 2) As Long	'二维数组 d, 从 d(0,-1) 到 d(2,2) 有 12 个长整型元素
Dim e(2 To 4)	'一维数组 e, 变体数据类型, 从 e(2) 到 e(4) 有 3 个元素
Dim f(2.5) As Integer	'一维数组 f, 从 f(0) 到 f(2) 有 3 个整型元素

5.3.2 数组的基本操作

数组的基本操作实质上是对数组元素进行的操作，包括对数组的赋值、引用、运算、输出和清除等。

知识要点分析

1. 数组元素的赋值和输出

给数组元素赋值有 3 种方法：用赋值语句分别为每个数组元素赋值、Array 函数赋值和用循环赋值。数组元素值的输出可以直接用 Print 语句单个输出，更方便快捷的方法是用 For 循环输出。

1) 用赋值语句为单个元素赋值

如果数组中元素的个数较少, 可以用赋值语句像普通变量一样分别赋值。输出值也像普通变量一样用 `Print` 语句输出。

例如: 在窗体上添加一个命令按钮, 单击命令按钮时执行程序。

```
Private Sub Command1_Click()  
    Dim x(3) As Integer  
    x(0) = 10: x(1) = 20: x(2) = 30: x(3) = 40  
    Print x(0); x(1); x(2); x(3)  
End Sub
```

这种方法为数组元素赋值的前提是数组元素的个数很少, 否则将非常烦琐, 而且很容易出错。

2) 用 `Array` 函数为一维数组元素赋值

用 `Array` 函数只能为一维数组的所有元素赋值。其格式为:

数组变量名 = `Array`(数组元素值)

说明:

- 数组变量必须声明为 `Variant` 型, 而且不指定下标。
- `Array` 函数的参数是数组元素的值, 每个值之间用半角逗号隔开。
- 赋值后数组下标的下界取默认值 0 或者 `Option Base` 语句指定的默认值, 上界由赋值数据的个数来决定。
- 用 `LBound` 和 `UBound` 函数可以测试数组变量下标的上下界。

`LBound` 和 `UBound` 函数的语法为:

`LBound|UBound`(数组名[, 维数])

返回某一维下标的下界或上界, 如果省略维数, 默认第一维。

例如:

```
Private Sub Command1_Click()  
    Dim x  
    x = Array(10, 20, 30, 40, 50, 60, 70, 80)  
    Print LBound(x), UBound(x)  
End Sub
```

程序运行后的结果为 0 和 7, 表示数组 `x` 下标的下界为 0, 上界为 7, 共 8 个元素。

3) 用 `For` 循环为数组元素赋值并输出

当数组元素比较多, 赋值有规律的数据、随机数或者键盘输入的数据时, 这种方法是首选。由于 `For` 循环的循环变量具有按照步长自增或自减的特点, 因此用它来动态地表示数组元素的下标, 可以非常方便快捷地为数组元素赋值和输出。

例如: 给一维数组 `x(10)` 赋值并输出。

```
Private Sub Command1_Click()  
    Dim x(10) As Integer, i As Integer  
    For i = 0 To 9
```

```

    x(i) = i      '把循环变量的值赋给数组元素
Next i

For i = 0 To 9
    Print x(i)   '输出数组元素的值
Next i
End Sub

```

程序运行的结果，x 数组元素的值为 0, 1, 2, 3, 4, 5, 6, 7, 8, 9，正好是循环变量的值。

给二维数组元素赋值用二重 For 循环嵌套，例如：

```

Private Sub Command1_Click()
Dim x(3, 3) As Single
For i = 0 To 3
    For j = 0 To 3
        x(i, j) = Int((Rnd * 11) + 10) '为 x 数组赋 10~20 之间的随机整数
    Next j
Next i

For i = 0 To 3
    For j = 0 To 3
        Print x(i, j)      '输出
    Next j
Next i
End Sub

```

2. 数组的清除

数组声明后，在其生存周期内，将长期占用相应的存储空间，直到程序运行结束，因此已经没用的数组如果不及时删除会造成存储空间的浪费。可使用 Erase 语句释放不需要的动态数组占用的存储空间。

语法格式：

```
Erase <数组名 1, 数组名 2, ...>
```

功能：对静态数组重新设置初始值，清除数组元素的值，释放动态数组的存储空间使其成为一个空数组。用 Dim 语句定义的数组就是动态数组，这里只讨论动态数组。

例如：

```
Erase a,b,c
```

如果 a, b, c 是用 Dim 定义的动态数组，这条语句的功能就是释放 a, b, c 三个数组的内存空间。

应用实例

【实例 5-12】 新建工程和窗体，在窗体上添加一个标签 Label1、一个文本框 Text1 和

一个命令按钮 Command1，属性设置和程序运行界面如图 5-6 所示。

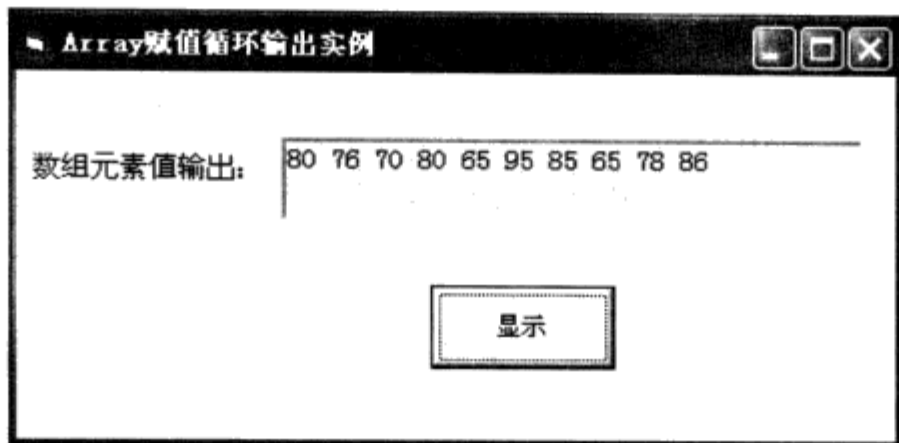


图 5-6 Array 函数为数组元素赋值并输出运行界面

程序代码如下。

```
Option Base 1
Private Sub Command1_Click()
    Dim x
    x = Array(80, 76, 70, 80, 65, 95, 85, 65, 78, 86)
    For i = 1 To UBound(x)
        Text1 = Text1 & x(i) & " "
    Next i
End Sub
```

程序分析：程序中用 Array 函数给一维数组 x 赋值，所以声明 x 时必须指定为变体型或省略数据类型，并且不能给出下标。输出时用 For 循环，通过 Option Base 1 语句，可以判断循环变量的初值也就是数组下标的下界为 1，而上界用 UBound 函数得到。

【实例 5-13】 新建工程和窗体，添加两个标签 Label1 和 Label2，两个文本框 Text1 和 Text2，一个命令按钮 Command1，属性设置和程序运行界面如图 5-7 所示。

声明定义一个 Integer 型数组，用 For 循环输出数组的初值到 Text1 文本框中；然后用循环重新为数组元素赋值，用到 InputBox 函数，随意接收键盘输入的一些整数值，并在 Text2 文本框中输出。

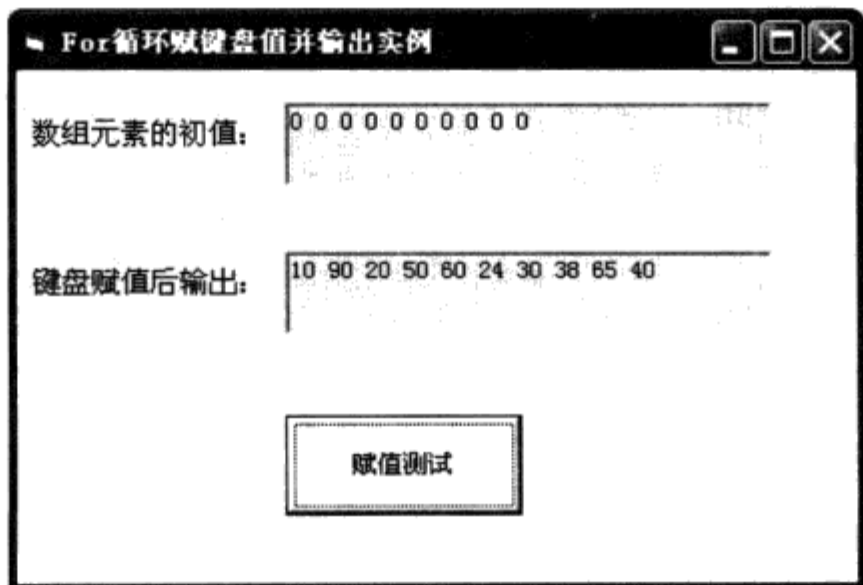


图 5-7 循环为数组赋值并输出运行界面

程序代码如下。

```
Option Base 1
Private Sub Command1_Click()
Dim x(10) As Integer
For i = 0 To 10      '声明数组后输出初值
    Text1 = Text1 & x(i) & " "
Next i
For i = 1 To 10     '为数组元素重新赋值
    x(i) = Val(InputBox(""))
Next i
For i = 1 To 10     '输出数组元素的值
    Text2 = Text2 & x(i) & " "
Next i
End Sub
```

5.3.3 数组的应用

在实际编程的过程中，数组的应用非常广泛，特别是一维和二维数组。熟练使用数组，能够达到事半功倍的效果。

应用实例

【实例 5-14】 新建工程和窗体，在窗体上添加三个标签（Label1，Label2 和 Label3）、三个文本框（Text1，Text2 和 Text3），两个命令按钮（Command1 和 Command2）。程序运行后单击 Command1，弹出 Inputbox 对话框并向其中输入 10 个数给数组元素赋值，并将赋的值在文本框 Text1 中输出，计算出最大值和最小值在 Text2 和 Text3 文本框输出。单击 Command2 退出程序的运行，属性设置和程序运行界面如图 5-8 所示。

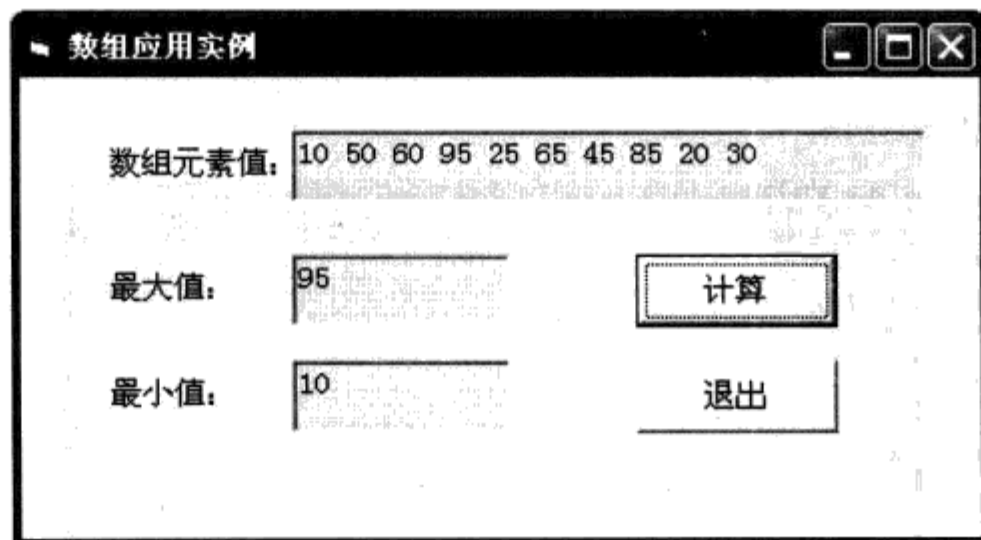


图 5-8 数组求最大最小值程序运行界面

程序思路分析：首先假设最大值 max 和最小值 min 都是数组的第一个元素，然后用最大值和最小值依次和数组的其他 9 个元素进行比较，如果比最大值大，那么把较大者重新赋值给 max；如果比最小值小，则重新把较小者赋值给 min。比较过程结束后，max 和 min

变量中存放的就是最大值和最小值了。

主要程序代码如下。

```
Option Base 1
Private Sub Command1_Click()
Dim x(10) As Integer, max As Integer, min As Integer
For i = 1 To 10
    x(i) = Val(InputBox("为数组元素赋值"))
    Text1 = Text1 & x(i) & " "
Next i
max = x(1): min = x(1)
For i =2 To 10
    If max < x(i) Then max = x(i)
    If min > x(i) Then min = x(i)
Next i
Text2 = max
Text3 = min
End Sub
```

【实例 5-15】 新建工程和窗体，在窗体上添加两个标签（Label1, Label2）、两个文本框（Text1, Text2），两个命令按钮（Command1 和 Command2）。程序运行后单击 Command1，产生 10 个 1~100 范围内的随机整数给数组元素赋值，并将赋的值在文本框 Text1 中输出，将这 10 个数据降序排列后在文本框 Text2 中输出。单击 Command2 退出程序的运行，属性设置和程序运行界面如图 5-9 所示。

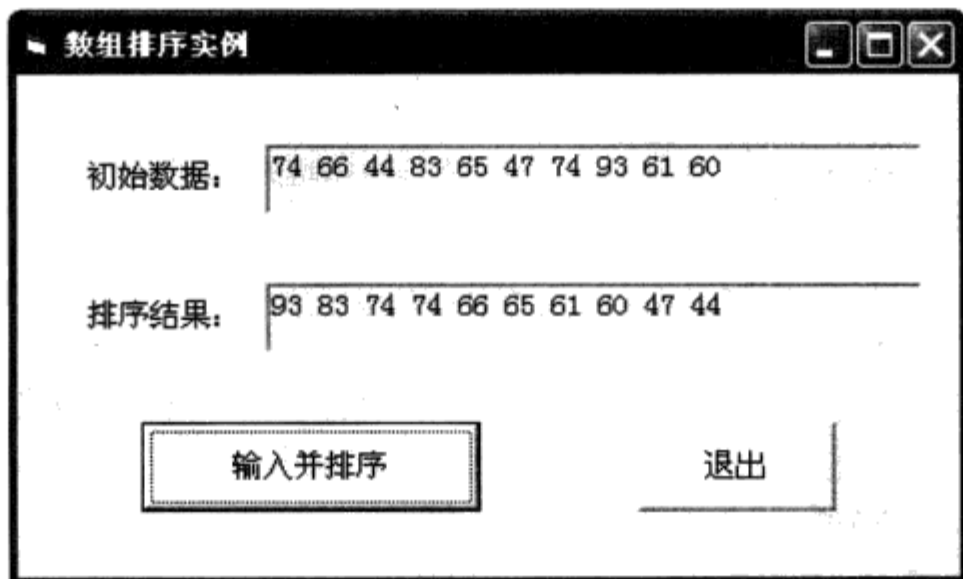


图 5-9 用数组实现排序程序运行界面

分析：简单排序有两种方法：一是直接排序法，二是冒泡排序法。

所谓直接排序法，是将数组中的 $x(1)$ 依次与 $x(2)$, $x(3)$, $x(4)$, ..., $x(10)$ 比较，如果 $x(1)$ 小于比较的数组元素，则将它们交换值，使得 $x(1)$ 中存放较大者；再将 $x(2)$ 依次与 $x(3)$ ~ $x(10)$ 比较，根据比较的结果进行必要的交换，使得 $x(2)$ 存放次大者；以此类推，直到将前 9 个数分别比较并交换后，整个数组就完成了降序排列。

冒泡排序法：对数组中两两相邻的元素比较大小，将值较大的元素放在前面（降序），值较小的元素放在后面。一趟比较完成后，一个最小的数成为数组中的最后一个元素，其他数像气泡一样上浮一个位置，重复这个过程直到没有数据需要交换为止。

两种方法的共同点是随着比较过程的进行，比较次数都是递减的。

主要程序代码如下（直接排序法降序排列）。

```
Private Sub Command1_Click()
Dim x(1 To 10) As Integer
For i = 1 To 10
    Randomize
    x(i) = Int(Rnd * 99) + 1           '产生 1~100 范围的随机数赋值给数组
    Text1 = Text1 & x(i) & " "
Next i
For i = 1 To 9                       '直接排序法降序排序
    For j = i + 1 To 10
        If x(i) < x(j) Then
            m = x(i): x(i) = x(j): x(j) = m   '中间变量实现交换
        End If
    Next j
Next i
For i = 1 To 10                       '输出
    Text2 = Text2 & x(i) & " "
Next i
End Sub
```

如果排序的过程采用冒泡法，可采用下面的代码来替换排序过程。

```
For i = 1 To 9                         '10 个数需要比较 9 趟
    For j = 1 To 10 - i                 '每趟的比较过程
        If x(j) < x(j + 1) Then
            m = x(j): x(j) = x(j + 1): x(j + 1) = m
        End If
    Next j
Next i
```

【实例 5-16】 用 Array 函数给数组赋值，输入 20 个学生的成绩，统计不同等级的学生人数（90 分以上为优秀，80 分以上为良好，60~70 为及格，60 分以下为不及格）。

新建工程和窗体，在窗体上添加一个框架（Frame1）、4 个文本框（Text1，Text2，Text3 和 Text4），一个命令按钮 Command1。各控件的属性设置和程序运行界面如图 5-10 所示。

程序代码如下。

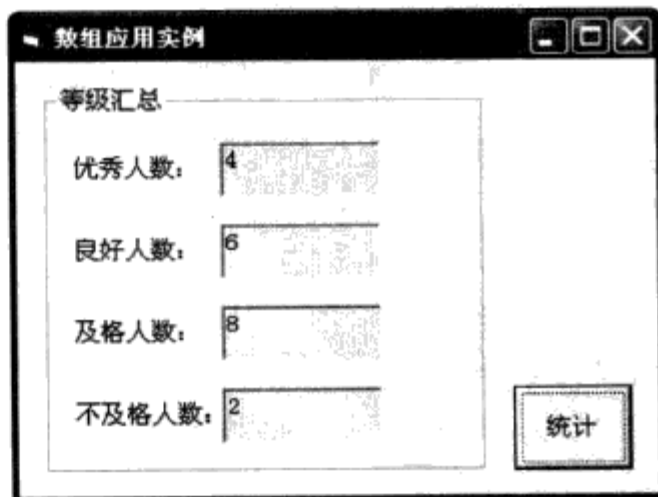


图 5-10 统计成绩程序运行界面

```

Option Base 1
Private Sub Command1_Click()
Dim n
Dim i As Integer
Dim n1, n2, n3, n4
n = Array(98, 85, 65, 88, 90, 75, 80, 82, 70, 68, 73, 60, 92, 77, 86, 55,
79, 87, 50, 95)
For i = 1 To UBound(n)
    Select Case n(i)
        Case Is >= 90
            n1 = n1 + 1
        Case Is >= 80
            n2 = n2 + 1
        Case Is >= 60
            n3 = n3 + 1
        Case Else
            n4 = n4 + 1
    End Select
Next i
Text1 = n1
Text2 = n2
Text3 = n3
Text4 = n4
End Sub

```

【实例 5-17】 二维数组应用。随机生成一个 5 行 5 列的二维矩阵，用一个二维数组来存放，每个数组元素的值为一位的正整数。

新建工程和窗体，添加两个标签（Label1 和 Label2），一个图片框 Picture1，一个文本框 Text1 和一个命令按钮 Command1，程序运行后单击 Command1，二维矩阵显示在图片框中，计算出二维矩阵主对角线上的数据之和，输出到 Text1 文本框中，如图 5-11 所示。

分析：5 行 5 列的矩阵，用二维数组来处理非常方便，矩阵中的每个数据对应二维数组的一个元素。二维数组元素的赋值需要 For...Next 循环嵌套来完成。每个数组元素的值可以通过随机数函数来生成。

程序代码如下。

```

Private Sub Command1_Click()
Dim x(1 To 5, 1 To 5) As Integer
Dim i As Integer, j As Integer, s As Integer
For i = 1 To 5
    '二重循环嵌套为数组元素赋值

```

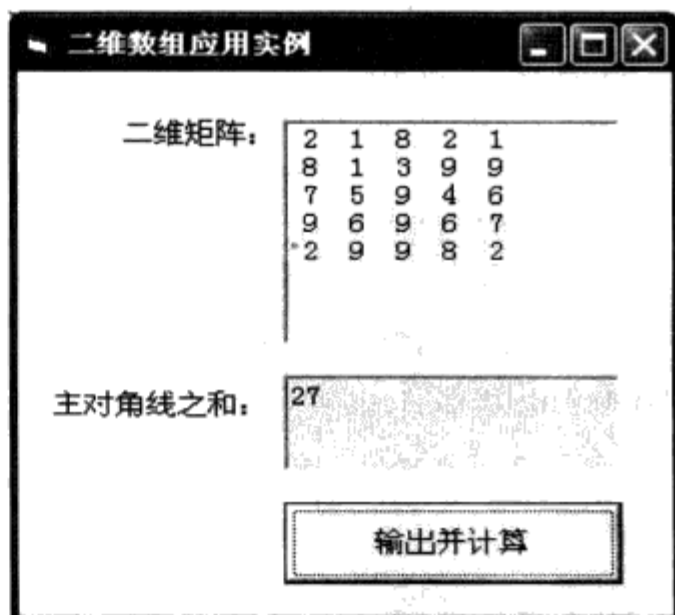


图 5-11 二维数组应用实例程序运行界面

```

For j = 1 To 5
    Randomize
    x(i, j) = Int(Rnd * 9) + 1
    Picture1.Print x(i, j); '在图片框中显示二维矩阵
Next j
Picture1.Print           '图片框中一行结束后换行
Next i

For i = 1 To 5           '二重循环嵌套读取每个数组元素
    For j = 1 To 5
        If i + j = 6 Then '判断是不是主对角线上的数
            s = s + x(i, j)
        End If
    Next j
Next i
Text1 = s
End Sub

```

实例中用到了图片框 `Picture`，有关图片框和图像框的内容在下一节有详细介绍。

5.4 图片框和图像框

图片框 (`PictureBox`) 和图像框 (`Image`) 是用来显示图形的基本控件，图像框显示静态图像，图片框除了可以显示图像外，还能作为控件容器，用 `Print` 显示文本、简单动画等，还可以在图片框中用 `Line` 等方法绘制图形。因此图片框占用内存较大，如果只是用来显示静态图片，一般图像框就可以了。

☞ 知识要点分析

1. 图片框和图像框的主要属性

1) `Picture` 属性

图片框和图像框都有 `Picture` 属性，用来设置图片框或图像框中要显示的图形文件的文件名，可以在属性窗口通过“浏览”按钮找到文件来设置。也可以在程序代码中调用 `Loadpicture` 函数来设置，该函数的参数给出图形文件所在的路径和文件名。例如：

```

Picture1.Picture = LoadPicture("C:\hua.jpg")
Image1.picture= LoadPicture("C:\hua.jpg")

```

如果要删除图片框或者图像框中的图像，可以在代码中使用下面的语句：

```

Picture1.Picture = LoadPicture("")
Image1.Picture = LoadPicture("")

```

2) 图片框的 `AutoSize` 属性

图片框的 `AutoSize` (自动大小) 属性有两个取值：`True` 和 `False` (默认)，用来设置图

片框如何与图像适应。当值为 `False` 时，图片框保持大小不变，图像如果超出了图片框的大小，则超出的部分不能显示；值为 `True` 时图片框会自动调整其大小以适应图像的正常显示。

3) 图像框的 `Stretch` 属性

图像框没有 `AutoSize` 属性，但是有 `Stretch` 属性，该属性用来设置图像框如何与图像相适应。`Stretch` 属性的取值也是 `True` 或 `False`（默认），其值为 `True` 时，图像将根据图像框的大小来调整自身的大小，可能会导致图像变形；值为 `False` 时，图像框根据图像的大小来调整自身的大小，以适应图像的正常显示。

2. 图片框的主要方法

图片框 `PictureBox` 可以作为控件容器，也可以用 `Print` 方法在图片框中输出文字。例如：

```
Picture1.Print "欢迎使用"
```

清除图片框中的文字可以用方法 `Cls`，用法如下：

```
Picture1.Cls
```

应用实例

【实例 5-18】 新建工程和窗体，添加 2 个框架（`Frame1` 和 `Frame2`），2 个图像框（`Image1` 和 `Image2`），2 个图片框（`Picture1` 和 `Picture2`），8 个标签（`Label1`, `Label2`, ...），通过设置图像框和图片框的 `Stretch` 属性和 `Picture` 属性来更好地理解这两项属性的内涵。各类控件的属性设置和程序运行的界面如图 5-12 所示。



图 5-12 图片框图像框实例程序运行界面

【实例 5-19】 新建工程和窗体，添加一个框架（`Frame1`，`Caption` 属性值为“各评委打分”），在框架中放置 6 个图像框（`Image1`, `Image2`, ..., `Image6`），设置图像框的 `Stretch` 属性值为 `False`，设置 `Picture` 属性值显示一些小图像，7 个标签说明信息，一个命令按钮

Command1。程序运行时单击命令按钮 Command1，弹出 InputBox 对话框输入 6 个评委的打分，去掉一个最高分和一个最低分，计算出选手的最后得分显示在文本框 Text1 中。各控件的属性设置及程序运行界面如图 5-13 所示。



图 5-13 图像框实例程序运行界面

程序代码如下。

```
Private Sub Command1_Click()
    Dim n(1 To 6) As Single, max As Single, min As Single, x As Single, s As Single
    Dim i As Integer
    For i = 1 To 6 '输入评委打分成绩赋值给数组元素
        n(i) = Val(InputBox("请输入评委打分"))
    Next i
    max = n(1): min = n(1)
    For i = 1 To 6
        s = s + n(i)
        If max < n(i) Then max = n(i)
        If min > n(i) Then min = n(i)
    Next i
    Text1 = (s - max - min) / 4
End Sub
```

【实例 5-20】 新建工程和窗体，添加一个标签（Label1，Caption 属性值为“显示图片列表”），添加 2 个图片框（Picture1 和 Picture2），AutoSize 属性设置为默认的 False；2 个命令按钮 Command1（Caption 属性值为显示）和 Command2（Caption 属性值为清除）。程序运行时单击命令按钮 Command1，弹出信息框提示，在 Picture1 中显示第几幅图片的信息，在 Picture2 分别显示 5 幅图片；单击 Command2 清除两个图片框中的内容。各控件的属性设置及程序运行界面如图 5-14 所示，界面显示的是最后一幅图片。



图 5-14 图片框应用实例程序界面

程序代码如下。

```
Private Sub Command1_Click()  
Dim i As Integer  
For i = 1 To 5  
    MsgBox "显示第" & i & "幅图片"  
    Picture1.Print "显示第" & i & "幅图片"  
    Picture2.Picture = LoadPicture(App.Path & "\" & i & ".jpg")  
Next i  
End Sub  
  
Private Sub Command2_Click()  
Picture1.Cls  
Picture2.Picture = LoadPicture("")  
End Sub
```

程序中 `App.Path` 获取的是当前工程和窗体保存的路径。`MsgBox` 的作用是让每幅图片的显示能够停顿一下，否则显示的过程就无法看到了，而只能看到最后一幅图片。

5.5 菜单设计

菜单是图形用户界面的重要组成部分，为用户灵活操作应用程序提供了便捷的手段，更多的用户习惯于用菜单进行操作。菜单可分为下拉式菜单和弹出式菜单（快捷菜单）。在 Visual Basic 中，可以方便地创建这两类菜单。


5.5.1 下拉式菜单

一般在默认的情况下，在窗体的标题栏下面有一个菜单栏，菜单栏中包含一个或多个菜单标题，每个菜单标题也被称为顶级菜单。单击菜单标题，则出现一个由菜单项组成的下拉列表，所以叫做下拉式菜单。这些菜单项可以是命令、分隔条和子菜单标题。菜单项应按其功能的不同分组存放。同种功能的菜单项应处于同一个分组，同时菜单项的标题也应符合菜单的基本约定。如：对于那些在使用后会弹出对话框的菜单命令，应在其标题后加“...”号；而对于带子菜单的菜单项，应在其菜单标题后加上“▶”标志等。

☞ 知识要点分析

在 Visual Basic 中，可以把菜单也理解为控件，菜单控件作为一个对象，与其他对象一样具有自己的属性和事件，但这类控件在标准的工具箱中没有。系统提供了专门制作菜单的工具——菜单编辑器，菜单的设计在菜单编辑器中完成。

1. 菜单编辑器

单击窗口中“工具”菜单下的“菜单编辑器”命令；或单击工具栏上的菜单编辑器按钮；或按下快捷键 Ctrl+E 可打开菜单编辑器窗口。

菜单编辑器由属性区、编辑区和菜单控件列表框组成，如图 5-15 所示。

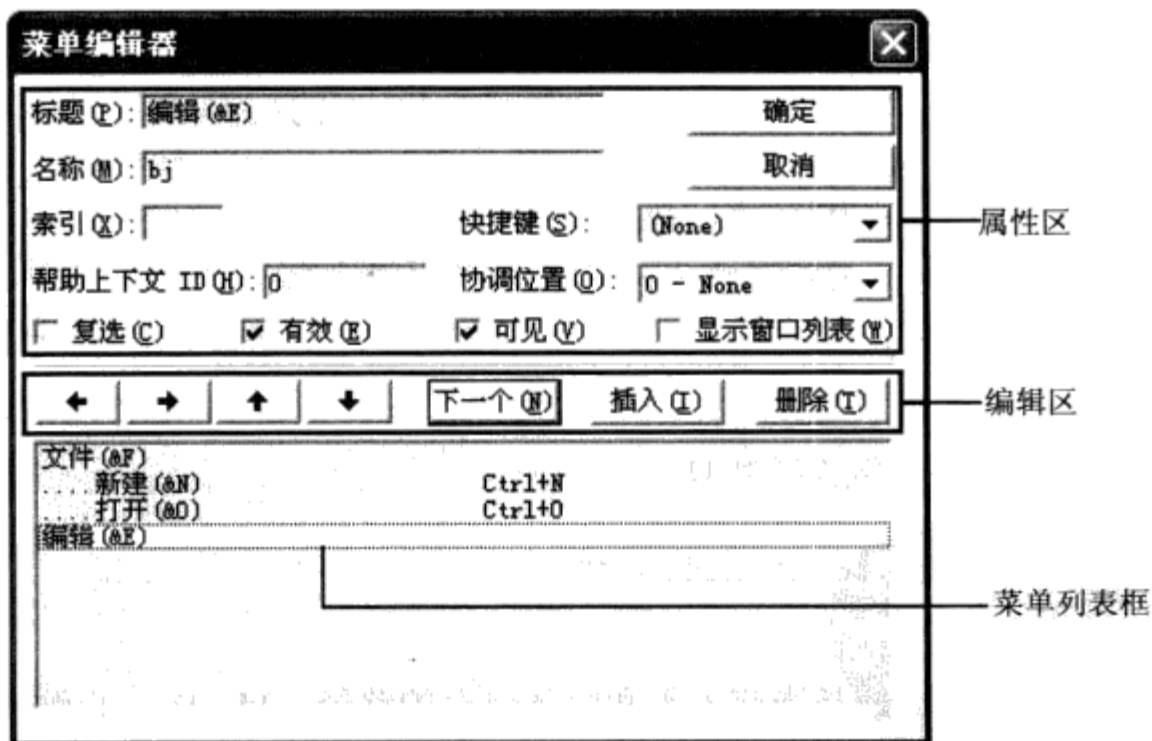


图 5-15 “菜单编辑器”窗口

属性区设置菜单控件的属性。

编辑区有 7 个按钮，对菜单进行编辑，具体包括：

- “↑”或“↓”：将选定菜单项的位置上移或下移。
- “←”或“→”：设置菜单缩进，提升或降低菜单的级别。
- “下一个(N)”：在选定菜单项下面添加一个新的菜单项，若选定菜单项下面有菜单，则下移到该菜单。

- “插入(I)”: 在选定菜单项的上方插入一个菜单项。
- “删除(T)”: 删除选定的菜单项。

菜单列表框以缩进的形式显示当前窗体的所有菜单项，包括菜单的标题、访问键和快捷键等，缩进量越大代表菜单项的级别越低。

2. 菜单的主要属性

菜单控件的属性在属性区设置。菜单控件的主要属性：

1) 标题 (Caption)

用于设置菜单标题，即在菜单栏上显示的文本。

- 当设置值为连字符“-”时，可在菜单中插入分组线。
- 当需要设置菜单的访问键时，可以在标题文本的后面用“(&+访问字符)”的格式设置。程序运行时可以用 Alt+访问字符来操作菜单。
- 当设置的菜单在程序运行后单击要打开一个对话框，则在标题文本的后面加上“…”。

2) 名称 (Name)

设置菜单或菜单项在代码中引用时所使用的名称，菜单项的名称应该是唯一的。一般菜单命名前面加“mnu”便于标识。

3) 索引 (Index)

用于建立菜单控件数组，通过索引值来标识数组中的不同元素（一个元素就是一个菜单项）。可通过代码实现菜单和菜单命令的动态添加和删除。

4) 快捷键 (Shortcut)

为设计的菜单选择一个快捷键。程序运行时，按下快捷键就相当于用鼠标单击了这个菜单项。顶级菜单不能设置快捷键。

5) 复选 (Checked)

该属性值设置为 True (勾选)，程序运行后在菜单项前出现“√”标记，一般用于创建开关式的菜单命令，指出开关状态。

6) 有效 (Enabled)

设置值为 True，程序运行时菜单清晰显示，表示可用；值为 False 时，菜单呈灰暗显示，表示不可用。

7) 可见 (Visible)

设置值为 True，程序运行后该菜单可见，否则菜单隐藏。隐藏的菜单是不能执行的。

3. 下拉式菜单的创建

下拉式菜单创建的一般步骤：

- 选定窗体，选择一种方法打开“菜单编辑器”窗口。
- 在菜单编辑器窗口中设计菜单的外观和属性，在菜单列表框中可浏览，然后单击“确定”按钮关闭菜单编辑器。
- 菜单的代码设计，菜单只有一个 Click 事件。
- 调试程序，保存工程和窗体。

应用实例

【实例 5-21】 新建工程和窗体，添加 1 个标签 (Caption 属性值为“改变图片框的背

景色：”），1 个图片框（Picture1），设计两个顶级菜单“颜色”和“退出”，颜色菜单下有三个菜单项：红色、绿色和蓝色。所有菜单都有访问键。所有控件的属性设置及程序运行界面如图 5-16 所示。

分析：所有的菜单都有访问键，因此在设置菜单标题属性时，需要在标题文本后面用“（&访问字符）”来设置；菜单的名称属性都以 mnu 打头，后面加英文表示。

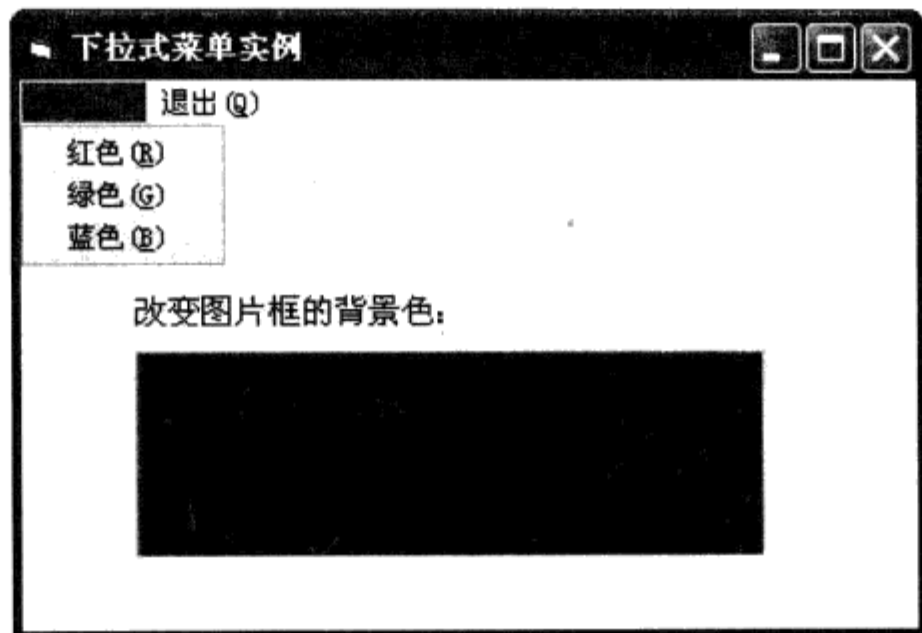


图 5-16 下拉式菜单实例运行界面

菜单代码如下。

```
Private Sub mnublue_Click()
Picture1.BackColor = vbBlue
End Sub

Private Sub mnugreen_Click()
Picture1.BackColor = vbGreen
End Sub

Private Sub mnured_Click()
Picture1.BackColor = vbRed
End Sub

Private Sub mnuquit_Click()
End
End Sub
```

【实例 5-22】 在程序运行过程中动态修改属性。程序运行界面如图 5-17 所示。

设计界面：新建工程和窗体，在窗体上添加 1 个标签（Label1，Caption 属性值为“改变文本框中文本的字体和字型”）和 1 个文本框（Text1，初始 Text 属性值为空）；选定窗体，按下 Ctrl+E 组合键打开菜单编辑器设计菜单，菜单列表和菜单的访问键、快捷键如图 5-18 所示。

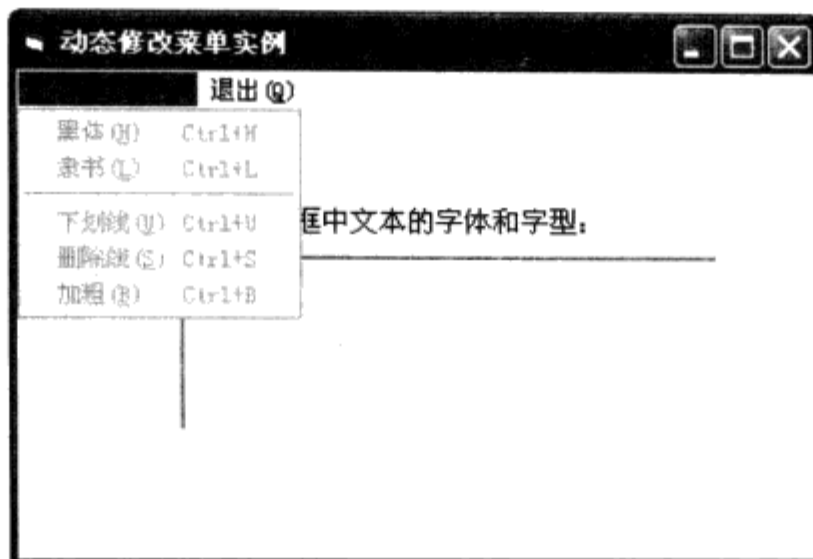


图 5-17 动态修改菜单实例运行界面

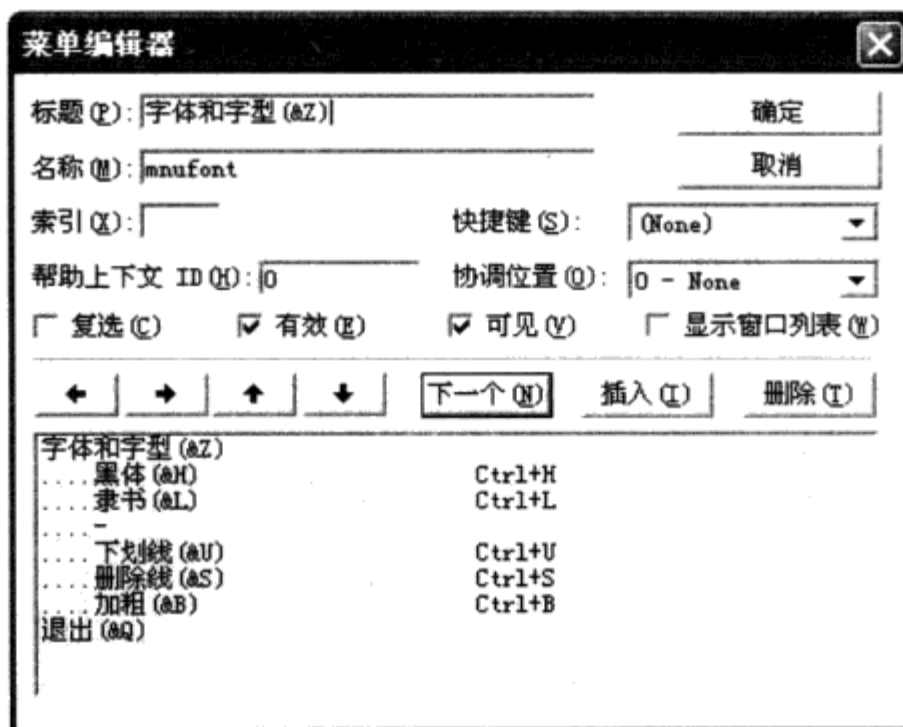


图 5-18 实例 5-22 菜单设计

编写代码如下。

```
Private Sub mnufont_Click()
    If Text1.Text = "" Then
        mnuheit.Enabled = False      ' 黑体菜单不可用
        mnulish.Enabled = False      ' 隶书菜单不可用
        mnushancx.Enabled = False    ' 删除线菜单不可用
        mnuxiahx.Enabled = False     ' 下划线菜单不可用
        mnujiacu.Enabled = False     ' 加粗菜单不可用
    Else
        mnuheit.Enabled = True
        mnulish.Enabled = True
        mnushancx.Enabled = True
        mnuxiahx.Enabled = True
        mnujiacu.Enabled = True
    End If
End Sub
```

```
Private Sub mnuheit_Click()  
    mnulish.Checked = False  
    mnuheit.Checked = True  
    Text1.FontName = "黑体"  
    mnulish.Visible = True  
End Sub  
  
Private Sub mnulish_Click()  
    mnuheit.Checked = False  
    mnulish.Checked = True  
    Text1.FontName = "隶书"  
End Sub  
  
Private Sub mnushancx_Click()  
    Text1.FontStrikethru = True  
    mnushancx.Visible = False  
End Sub  
  
Private Sub mnuxiahx_Click()  
    Text1.FontUnderline = True  
    mnuxiahx.Visible = False  
End Sub  
  
Private Sub mnujiacu_Click()  
    Text1.FontBold = True  
    mnujiacu.Visible = False  
End Sub  
  
Private Sub mnutuic_Click()  
    End  
End Sub  
  
Private Sub Text1_Change()  
    If Text1 = "" Then  
        mnushancx.Visible = True  
        mnuxiahx.Visible = True  
        mnujiacu.Visible = True  
    End If  
End Sub
```

程序结果分析：程序运行后，当文本框中为空时，“字体和字型”菜单的所有菜单项都灰色显示不可用；当文本框输入内容后菜单都变为可用。当设置文本框中的文本为某个字体后，该字体菜单加复选标记，同时另外的字体菜单取消复选。当设置文本框中的文本为某个字型后，则该字型菜单隐藏。当文本框清空时，字型菜单又都显示出来。单击“退出”菜单结束程序的运行。

【实例 5-23】 在程序运行过程中动态添加或删除菜单项。程序运行界面如图 5-19 所示。

分析：程序运行的过程中动态地添加和删除菜单项，只有菜单控件数组才能实现。菜单控件数组是指共享相同名称和事件过程的菜单项目的集合，其所有元素都应处于同一个

缩进级别且应连续排列。每个菜单控件数组元素都由唯一的索引值 (index) 来标识, 该值通过菜单编辑器的“索引”文本框中来指定, 默认从 0 开始, 连续排列。

设计界面: 新建工程和窗体, 在窗体上添加 1 个标签 (Label1, Caption 属性值为“请设置字号:”), 1 个文本框 (Text1, 初始 Text 属性值为空)。选定窗体, 打开“菜单编辑器”窗口, 设计菜单, 其中 18, 22, 30 为菜单控件数组的三个元素, 它们具有相同的名称 mnusize 和不同的 Index 索引值 0, 1 和 2。

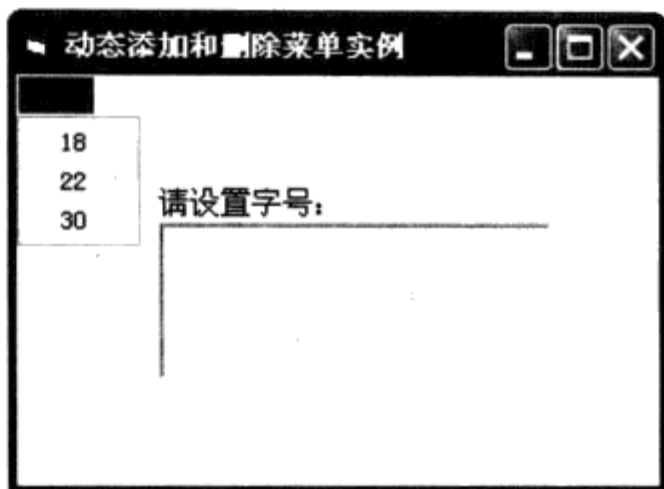


图 5-19 添加和删除菜单实例运行界面

程序代码:

```
Dim flag As Boolean '通用段声明变量, 声明后 flag 的初始值为 False
```

```
Private Sub mnusize_Click(Index As Integer)
```

```
    Select Case Index
```

```
        Case 0
```

```
            Text1.FontSize = 18
```

```
            If flag = True Then
```

```
                Unload mnusize(3)
```

```
                flag = False
```

```
            End If
```

```
        Case 1
```

```
            Text1.FontSize = 22
```

```
        Case 2
```

```
            Text1.FontSize = 30
```

```
            If flag = False Then
```

```
                Load mnusize(3)
```

```
                mnusize(3).Caption = "36"
```

```
                flag = True
```

```
            End If
```

```
        Case 3
```

```
            Text1.FontSize = 36
```

```
    End Select
```

```
End Sub
```

程序运行后, 在文本框中输入内容, 单击菜单项 30, 设置字号为 30, 如果没有菜单项 36, 则添加该菜单; 单击菜单项 18, 设置文本框内容的字号为 18, 如果已经添加了菜单项 36, 则删除该菜单。

5.5.2 弹出式菜单

弹出式菜单是单击鼠标右键时弹出的菜单, 由于弹出式菜单中显示的菜单项取决于按

下鼠标右键时指针所处的位置，因而弹出式菜单也被称为上下文菜单或快捷菜单。弹出式菜单能够以更灵活的方式为用户提供更便利的操作。

知识要点分析

创建弹出式菜单也在菜单编辑器中完成，它和下拉式菜单的区别在于需要设置顶级菜单不可见。

在 Visual Basic 中，创建弹出式菜单的步骤为：

- (1) 使用菜单编辑器创建菜单，使顶级菜单项不可见，即不勾选“可见”框。
- (2) 编写与相应弹出式菜单相关联的 MouseUp 事件过程。用到 PopupMenu 方法。

MouseUp 事件过程的结构如下。

```
Private Sub Form_MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
...
End Sub
```

选择了这个事件后，系统会自动定义一些变量，其中 Button 为整型数据，用以返回在实际操作中用户按下了鼠标的哪一个键，其值可以为 1（左键）、2（右键）、4（中键）；Shift 为整型数据，用以返回在实际操作中用户在按下鼠标按键的同时是否结合了键盘按键，其值可以为 1（在按下 Shift 键的同时按下鼠标按键）、2（在按下 Alt 键的同时按下鼠标按键）、4（在按下 Ctrl 键的同时按下鼠标按键）；X 和 Y 参数为单精度型数据，用以返回按键的位置（横、纵坐标）。

PopupMenu 方法：

```
[对象名.]PopupMenu <菜单名>[, Flag 参数[, 横坐标 X[, 纵坐标 Y]]]
```

其中，X（Y）参数用来指定弹出式菜单的横（纵）坐标，缺省时默认鼠标坐标。

Flag 参数包括位置参数和行为参数两部分，位置参数的值为 0（默认），4，8 分别表示弹出式菜单的左上角位于 X、上边中点位于 X 和右上角位于 X；行为参数取值 0（默认）或 2，表示弹出式菜单只响应鼠标右击或响应单击和右击。

在使用弹出式菜单时需要注意，很多 Windows 的控件自身就带有弹出式菜单（如文本框），不要试图再给这样的控件添加弹出式菜单，这样会产生两个弹出式菜单，而原始菜单具有更高的优先权，容易造成混乱。

应用实例

【实例 5-24】 用弹出式菜单来代替实例 5-21 中的下拉式菜单。

设计界面：只需要在原实例界面的基础上，修改菜单，选定窗体，打开菜单编辑器，将顶级菜单“颜色”（名称属性值为 mnucolor）设置为不可见。

代码编写：

```
Private Sub Form_MouseUp(Button As Integer, Shift As Integer, X As Single,
Y As Single)
If Button = 2 Then
```

```
    PopupMenu mnucolor
End If
End Sub

Private Sub mnublue_Click()
    Picture1.BackColor = vbBlue
End Sub

Private Sub mnugreen_Click()
    Picture1.BackColor = vbGreen
End Sub

Private Sub mnured_Click()
    Picture1.BackColor = vbRed
End Sub

Private Sub mnuquit_Click()
    End
End Sub
```

本章小结

循环结构是用来重复执行某个程序段的结构。本章重点介绍了 For...Next 和 Do...Loop 两类循环语句。For...Next 循环用于计数型的能够预知循环次数的循环；Do...Loop 循环结构适用于事先不知道循环次数的循环设计需要。介绍了循环退出语句 Exit For 和 Exit Do。在编程的过程中，经常会用到顺序结构中有循环，循环结构中又嵌套其他循环结构，或嵌套分支结构的情况。各种程序控制结构的综合使用才能使编程更加顺畅、结构更清晰、功能更强大。因此在本章中提供了大量嵌套结构的程序应用实例，并且用到了一类特殊的用来处理大量和比较复杂数据的数组变量。最后结合程序结构，介绍了基本控件图片框和图像框的一些属性和方法，以及下拉式菜单和弹出式菜单的制作。

习 题 5

一、判断题

1. For...Next 循环适合不知道循环次数的循环设计需要。（ ）
2. For...Next 循环必须要有一个循环变量，每循环一次，该变量的值都需要人为设置语句改变，否则就会进入到死循环。（ ）
3. Ctrl+Break 组合键可以结束“死循环”程序的运行。（ ）
4. For...Next 循环，步长不可以为负值。（ ）

5. Exit Do 语句一般都是用在条件判断 If...Then 中或者 Select...Case 语句中结束 Do...Loop 循环。()
6. Do...While...Loop 循环控制为前测型,如果条件为真,执行循环体语句序列。()
7. Do...Loop 循环的前测型和后测型没有区别。()
8. Do...Loop While<条件>,若条件为假,则 Do 后面的循环体语句一次也不执行。()
9. 循环结构中可以嵌套其他循环结构,这种循环的多层嵌套又叫多重循环。()
10. 内循环和外循环的循环变量可以同名。()
11. 数组是一类特殊的变量,用来处理大量的和比较复杂的数据。()
12. 具有同一变量名、不同下标的下标变量集合称为数组,数组的下标必须放到方括号[]中。()
13. 下标的个数决定数组的维数,有一个下标为一维数组。()
14. 当声明数组时省略数据类型,则系统默认 Integer 型。()
15. 数组中的所有元素具有相同的数据类型,但当数组被声明为变体型 Variant 时,数组中的元素可以有不同的数据类型。()
16. Dim a(1 To 9) As Single 表示声明了一个包含 9 个元素的单精度型一维数组。()
17. 给数组元素赋值有 3 中方法,其中用 For...Next 循环赋值是常用的一种方法。()
18. 用 Array 函数为一维数组元素赋值的前提是该数组被规范的定义了名称、上下标和数据类型。()
19. 图片框和图像框都有 Picture 属性。()
20. 图片框和图像框相同,可以混用。()
21. Image1.Picture = LoadPicture("")表示删除图像框 Image1 中的图形。()
22. 图片框和图像框都用 Stretch 属性来设置框如何与图像大小相适应。()
23. 在 Visual Basic 中,下拉式和弹出式菜单都在菜单编辑器中设计。()
24. 选定窗体,按下快捷键 Ctrl+E 可以打开菜单编辑器。()
25. 菜单控件数组可以实现在程序运行的过程中动态地添加和删除菜单。()

二、单选题

1. 下列程序运行后,其循环体被执行的次数是()。

```
For i = 1 to 100 step 0
    Print i
Next i
```

- A. 1 B. 100 C. 无限循环 D. 以上都不对

2. 下列程序段,有关循环描述,错误的是()。

```
For j = m to n step k
    Print j
Next j
```

- A. 若 $m > n$,则必须要求 $k > 0$,如果此时 $k < 0$,则循环体被执行的次数为 0
 B. 若 $m < n$,则必须要求 $k > 0$,如果此时 $k < 0$,则循环体被执行的次数为 0

- C. 若 $m > n$, 且 $k < 0$ 或若 $m < n$, 且 $k < 0$, 则循环体被执行的次数为 $\text{Int}(1+(n-m)/k)$
D. 循环结束后, 循环变量 j 的值一定是 n
3. 在窗体上添加一个命令按钮 Command1, 命令按钮的单击事件如下。

```
Dim i as integer
For i=2 to 10 step 2
    i=i+2
    Print i
Next i
```

程序运行后, 单击命令按钮, 结果为 ()。

- A. 5 8 12 B. 2 4 8 C. 4 5 8 D. 4 8 12
4. 在窗体上添加一个命令按钮 Command1, 命令按钮的单击事件如下。

```
Private Sub Command1_Click()
Dim i As Integer, s As Integer
For i = 30 To 1 Step -1
    If i Mod 3 = 0 And i Mod 5 = 0 Then
        s = s + 1
    End If
Next i
Print s
End Sub
```

程序运行后, 单击命令按钮, 结果为 ()。

- A. 30 B. 15 C. 2 D. 1
5. 在窗体的单击事件中编写程序代码, 程序运行后单击窗体, 结果为 ()。

```
Private Sub Command1_Click()
Dim i As Integer, s As Integer
Do While i <= 20
    s = s + i
    If s >= 20 Then Exit Do
    i = i + 1
Loop
Print s
End Sub
```

- A. 22 B. 21 C. 20 D. 19

6. 在窗体上添加一个命令按钮 Command1 和一个文本框 Text1, 编写如下事件过程, 程序运行后单击命令按钮, 在文本框中显示的值为 ()。

```
Private Sub Command1_Click()
Dim i As Integer, s As Integer
Do
    s = s + 3
    i = i + 1
```

```

Loop Until i = 5
Text1 = s
End Sub

```

- A. 14 B. 15 C. 16 D. 17

7. 在窗体上添加命令按钮 Command1, 编写如下代码, 程序运行后单击 Command1, 输出的结果为 ()。

```

Private Sub Command1_Click()
Dim i As Integer, j As Integer, s As Integer
For i = 1 To 3
    For j = 1 To i
        s = s + 1
    Next j
Next i
Print s
End Sub

```

- A. 6 B. 7 C. 8 D. 9

8. 关于数组的描述, 错误的是 ()。

- A. 下标的个数决定数组的维数
 B. 数组遵循变量的先声明后使用的原则
 C. 可以用 LBound 和 UBound 函数来测试数组下标的下界和上界
 D. 一维数组可以用 For 循环赋值, 但二维数组不可以

9. 在程序运行过程中, 要退出 For 循环, 可以使用的语句是 ()。

- A. Exit Do B. Exit Sub C. Exit D. Exit For

10. 在窗体上添加一个命令按钮 Command1 和一个文本框 Text1, 编写程序代码如下。

```

Private Sub Command1_Click()
Dim x As Integer, y As Integer, z As Integer
x = 10: y = 20: z = 30
For i = 1 To 2
    x = y: y = z: z = x
Next i
Text1 = x + y + z
End Sub

```

程序运行后, 单击命令按钮, 在文本框中显示的结果是 ()。

- A. 80 B. 70 C. 60 D. 50

11. 在窗体上添加一个命令按钮 Command1 和一个文本框 Text1, 编写程序代码如下。

```

Private Sub Command1_Click()
Dim t As String
x = "a": y = "b": z = "c"
For i = 1 To 2
    t = x: x = y: y = z: z = t

```

```
Next i
Text1 = x + y + z
End Sub
```

程序运行后，单击命令按钮，在文本框中显示的结果是（ ）。

A. abc B. acb C. cab D. bca

12. 以下循环体语句执行了（ ）次。

```
Dim i As Integer
i = 10
Do While i >= 1
    i = i - 2
Loop
```

A. 10 B. 5 C. 3 D. 一次都不执行

13. 以下程序执行后，输出的结果为（ ）。

```
Option Base 1
Private Sub Command1_Click()
Dim a(10), b(3) As Integer
k = 5
For i = 1 To 10
    a(i) = i
Next i
For i = 1 To 2
    b(i) = a(i * i)
Next i
For i = 1 To 2
    k = k + b(i) * 2
Next i
Print k
End Sub
```

A. 15 B. 20 C. 25 D. 30

14. 补充下面程序，使语句 $x=x+1$ ，被执行 15 次（ ）。

```
For i = 1 To 3
    For j = 2 To _____ Step 2
        x=x+1
    Next j
Next i
```

A. 14 B. 12 C. 10 D. 8

15. 补充下面程序，循环结束后 t 的值为 7，则条件 t 小于等于（ ）。

```
t = 1
Do
    t = t + 1
```

Loop While t <= _____

- A. 8 B. 7 C. 6 D. 5

16. 补充下面程序, 循环结束后输出最大值 ()。

```
Option Base 1
Private Sub Command1_Click()
Dim a, Max, t
a = Array(18, 90, 65, 30, 55, 20, 60, 40)
_____
For i = 1 To _____
    If a(i) > Max Then Max = a(i): t = i
Next i
Print "最大值是: "; max
End Sub
```

- A. Max = a(0) UBound(a) B. Max = a(1) LBound(a)
C. Max = a(0) LBound(a) D. Max = a(1) UBound(a)

17. 用语句 Dim a(-1 To 2, 1.5 To 3)定义的数组有 () 个元素。

- A. 8 B. 9 C. 10 D. 不能定义

18. 补充完善下面程序, 使其能够输出 0~100 之内能被 3 整除且个位数是 6 的所有整数 ()。

```
Private Sub Command1_Click()
Dim m As Integer, n As Integer
For m = 0 To _____
    n = m * 10 + 6
    If _____ Then Print n
Next m
End Sub
```

- A. 10 n\3 = int(n\3) B. 10 n Mod 3 = 0
C. 100 n\3 = int(n\3) D. 100 n Mod 3 = 0

19. 程序运行后输出的结果为 ()。

```
Private Sub Command1_Click()
Dim m As Integer, n As Integer
m = 1: n = 1
Do While m <= 2
    Do While n <= 2
        m = m + n
        n = n + 1
    Loop
    m = m + 1
Loop
Print m, n
End Sub
```

A. 6 2 B. 6 3 C. 5 2 D. 5 3

20. Do 循环的结构如下, 请补充完整 ()。

```
Do While|Until 条件
  循环体语句序列
```

A. End Do B. Loop C. Exit Do D. Next

三、程序题

1. 编写程序, 用随机函数产生 100 个 1~100 之间的整数, 在文本框中输出大于等于键盘输入的某个数 (1~100) 的数及个数。分别用 For...Next 循环和 Do...Loop 循环两种方法实现。
2. 在文本框中输出如图 5-20 所示的等腰三角形。
3. 从键盘输入 5 个正整数, 请在文本框中输出最大值和最小值。
4. 将 0~100 范围中能同时被 4 和 6 整除的数显示在文本框中。
5. 编写程序, 在图片框中随机生成一个 6 行 6 列的二维矩阵, 计算每行的平均值在文本框中输出。程序运行界面如图 5-21 所示。



图 5-20 星号三角形程序运行界面

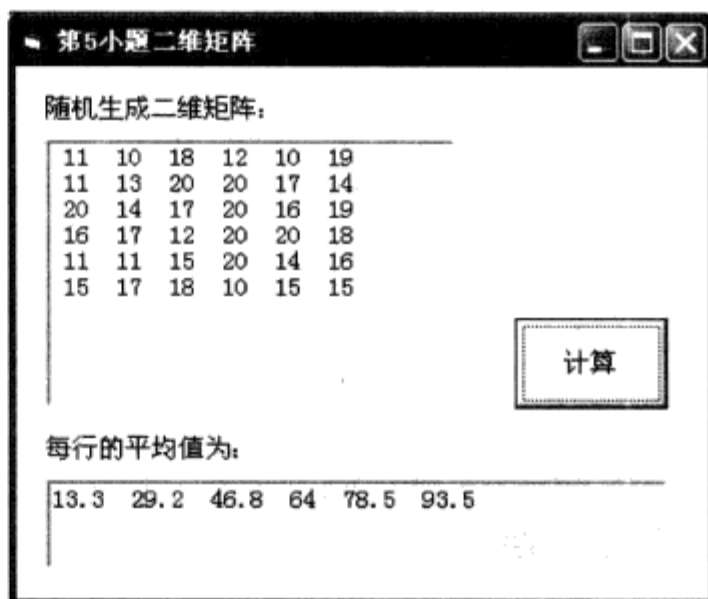


图 5-21 随机生成 6 行 6 列矩阵求每行平均值程序界面

6. 制作下拉式和弹出式菜单, 界面如图 5-22。程序运行后, 单击“基本信息管理”下面的菜单项, 能够打开对应的窗体; 单击“退出”能够结束程序的运行; 右击窗体能够弹出图 5-23 所示的快捷菜单, 单击颜色菜单项可以设置窗体的背景为相应的颜色。

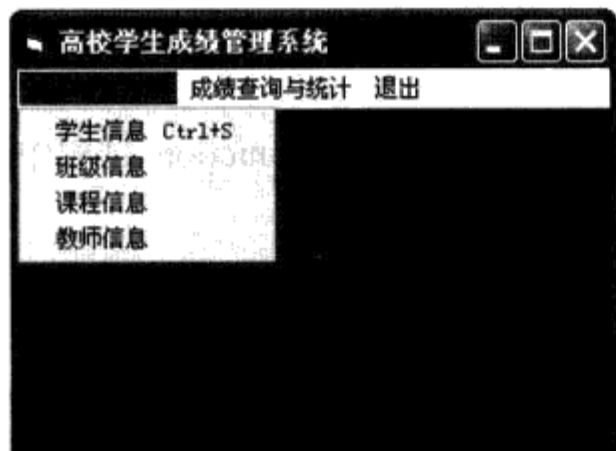


图 5-22 下拉式菜单程序界面



图 5-23 弹出式菜单程序界面

第 6 章

Visual Basic 过程与文件

6.1 Visual Basic 过程使用

在 Visual Basic 中过程分为两大类：Sub 过程和 Function 过程。把 Sub...End Sub 定义的过程称为子程序，把由 Function...End Function 定义的过程称为自定义函数。

6.1.1 Sub 过程的建立与调用

知识要点分析

1. 建立格式

```
|Private||Public||Static|Sub 过程名(形参列表)  
...[Exit Sub]  
End Sub
```

2. 要点说明

(1) Sub 过程以 Sub 开头，End Sub 结束，之间是语句块，称为“过程体”或“子程序体”。格式中各参量的含义如下。

- **Private:** 使用时表示只有本模块中的其他过程可以调用该 Sub 过程。
- **Public:** 使用时表示所有模块的所有其他过程者可调用该 Sub 过程。
- **Static:** 指定过程中的局部变量在内存中的默认储存方式。如果使用了 Static，则在每次调用过程时，局部变量的值保持不变；如果省略了 Static，则在每次调用过程时，局部变量被初始化为 0 或空字符串。
- **过程名:** 长度不超过 255 个字符的变量名。在 Sub 过程和 Function 过程中变量名不能相同。
- **形参列表:** 表示调用时传递给 Sub 过程的参数变量列表，多个列表之间用逗号隔开。
- **Exit Sub:** 在过程体中用于退出 Sub 过程。
- **Sub 过程**可定义在窗体模块或标准模块中。

(2) 每个 Sub 过程必须有一个 End Sub 子句标志 Sub 过程的结束。当程序执行到 End Sub 时，将退出该过程，并立即返回到调用语句下面的语句。

(3) Sub 过程不能嵌套定义，可以嵌套调用。不能用 GoTo 语句进入或转出一个 Sub 过程，只能通过调用执行 Sub 过程。

3. 程序调用

Sub 过程可以通过下面 2 种方式调用。

(1) 使用 Call 语句调用。

```
Call <过程名>(实参列表)
```

注意：参数列表中的括号不能省略。

(2) 过程名作为一个语句。

```
<过程名> 实参列表
```

注意：参数列表中不能加括号。

应用实例

【实例 6-1】 通过 Sub 过程调用，输入圆的半径，求圆的面积和周长，效果如图 6-1 所示。

代码如下。

```
Private Sub Command1_Click() '圆面积
Dim r As Double
r = InputBox("请输入半径", "提示")
Call mj(r) '调用时括号不能省略
End Sub
```

```
Private Sub Command2_Click() '圆周长
Dim r As Double
r = InputBox("请输入半径", "提示")
zc r '调用时括号省略
End Sub
```

模块程序如下。

```
Public Sub zc(K As Double) '周长
T = 2 * 3.14159 * K
Form1.Text1.Text = T
End Sub
```

```
Public Sub mj(K As Double) '面积
s = 3.14159 * K ^ 2
Form1.Text1.Text = s
End Sub
```

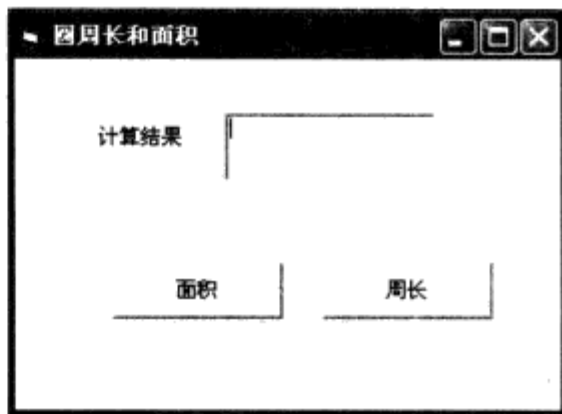


图 6-1 求圆面积和周长窗体

6.1.2 Function 过程

知识要点分析

1. 建立格式

```
[Private] | [Public] | [Static] | Function 函数名(形参列表)
```

```
...[Exit Function]
End Function
```

2. 要点说明

(1) Function 过程以 Function 开头, End Function 结束, 之间是语句块, 即函数体。格式中各参量的含义如下。

- Private、Public、Static、形参列表等项含义与 Sub 过程相同。
- Exit Function: 从 Function 过程中退出。

(2) 调用 Function 过程要返回一个值, 格式为“过程名=表达式”, 因此可以像内部函数一样在程序中使用。

3. 程序调用

直接在表达式中调用。

```
变量名=函数名(实参列表)
Print 函数名(实参列表)
```

应用实例

【实例 6-2】 通过 Function 过程调用, 输入圆的半径求圆的面积和周长, 效果如图 6-2 所示。

代码如下。

```
Private Sub Command1_Click() '圆面积
Dim r As Double
r = InputBox("请输入半径", "提示")
S = mj(r) '调用时括号不能省略
Text1.Text = S
'也可通过下面的形式调用
Text1.Text = mj(r)
End Sub
```

```
Private Sub Command2_Click() '圆周长
Dim r As Double
r = InputBox("请输入半径", "提示")
Print zc(r) '调用时括号不能省略
End Sub
```

模块程序如下。

```
Public Function zc(K As Double) '周长
T = 2 * 3.14159 * K
zc = T '函数返回值
End Function

Public Function mj(K As Double) '面积
S = 3.14159 * K ^ 2
```

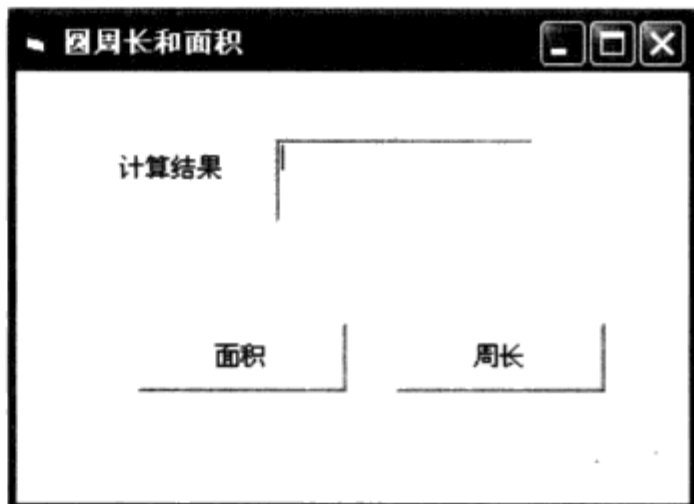


图 6-2 求圆面积和周长窗体

```

mj = S '函数返回值
End Function

```

6.1.3 参数传递

📌 知识要点分析

在调用一个过程时，可以把实际参数传送给过程中的形式参数。在参数传递时，实参和形参可以同名，也可以不同名，但在传递过程中，参数的类型、个数、顺序要保证一一对应，否则程序会出错。

在 Visual Basic 中，形参与实参的传递方式有两种，即按值传递和按地址传递。

1. 按值传递

```
Sub|Function<过程名>(ByVal<参数 1>,ByVal<参数 2>...)
```

定义过程时用 **ByVal** 关键字指出参数是按值来传递的，即形参值在 Sub 过程或 Function 过程中的改变不会影响到主程序中实参的值。

2. 按地址传递

```
Sub|Function<过程名>(<参数 1>,<参数 2>...)
```

按地址传递是在按值传递的基础上省略了 **ByVal**，是指将实参的地址传给形参，这样实参和形参共用相同的地址，即共享同一段内存，在被调过程中改变形参的值，则相应实参的值也被改变。

🔗 应用实例

【实例 6-3】 通过按值传递和按地址传递调用过程，效果如图 6-3 所示。
程序代码如下。

```

Private Sub Command1_Click() '按值传递
Dim A As Integer
A = 100
B = 200
Call abcl(A, B)
'传递后 A 和 B 的值不会改变
Text1.Text = A
Text2.Text = B
End Sub

Private Sub Command2_Click() '按地址传递
Dim A As Integer, B As Integer
A = 100

```

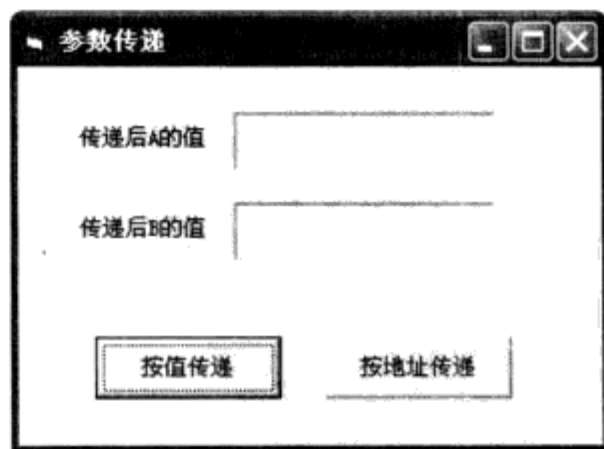


图 6-3 参数传递窗体

```

B = 200
abc2 A, B
'传递后 A 和 B 的值会改变
Text1.Text = A
Text2.Text = B
End Sub

```

模块程序如下。

```

Public Sub abc1(ByVal x As Integer, ByVal y)
x = 800
y = 900
End Sub

```

```

Public Sub abc2(x As Integer, y As Integer)
x = 800
y = 900
End Sub

```

思考：abc1 与 abc2 的参数传递有什么区别？

6.1.4 综合应用

应用实例

【实例 6-4】 在窗体中定义过程，求 $1!+2!+3!+\dots+10!$ 的和。

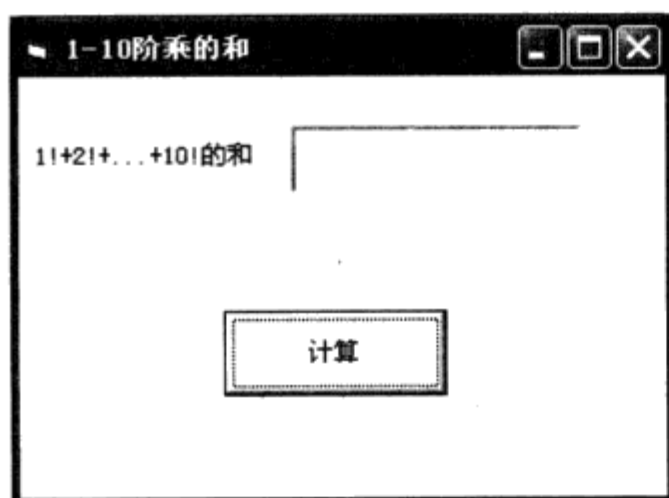


图 6-4 求阶乘窗体

```

sum = sum + jc(i)
Next
Text1.Text = sum
End Sub
Function jc(n As Integer)

```

分析：

- (1) 该过程完成的是 1~10 中每个数的阶乘，因此 1~10 是传递给过程的实参。
- (2) 可以通过循环体实现多次参数传递。
- (3) 该过程接收到参数后，要完成阶乘的计算，同时必须有返回值。

效果如图 6-4 所示。

程序代码如下。

```

Private Sub Command1_Click()
Dim i As Integer, sum As Long
For i = 1 To 10

```

```
t = 1
For i = 1 To n
    t = t * i
Next
jc = t
End Function
```

【实例 6-5】 在窗体中定义过程，将 10~20 的素数在文本框中显示出来。

分析：

(1) 该过程完成的是 10~20 之间参数的传递。

(2) 可以通过循环体实现多次参数传递。

(3) 该过程接收到参数后，要完成素数的判断，同时返回 True 或 False。

效果如图 6-5 所示。

代码如下。

```
Private Sub Command1_Click()
    Dim i As Integer
    For i = 10 To 20
        If ss(i) = True Then
            Text1.Text = Text1.Text & i & " "
        End If
    Next
End Sub
```

```
Public Function ss(n As Integer)
    Dim bj As Boolean
    bj = True
    For i = 2 To n - 1
        If n Mod i = 0 Then
            bj = False
            Exit For
        End If
    Next
    ss = bj
End Function
```

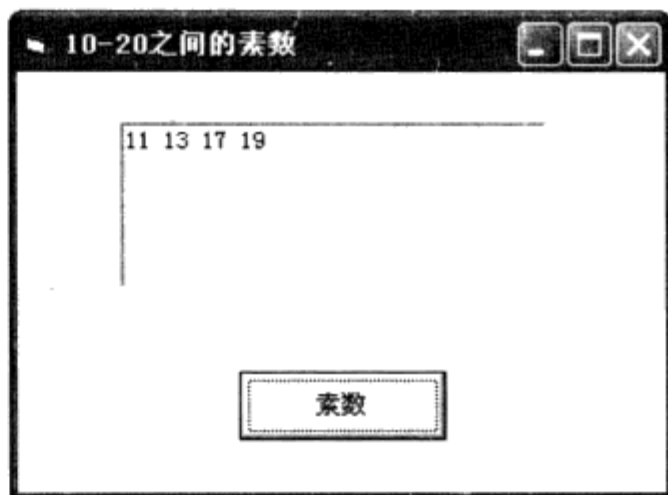


图 6-5 素数统计窗体

6.2 文件的访问模式

Visual Basic 的输入输出既可以在标准的输入输出设备上进行，也可以通过外部存储器进行，而计算机处理的数据一般都以文件的形式存放在外部存储上。存放文件的方式也称

文件结构，Visual Basic 文件由记录组成，记录由字段组成，字段由字符组成。根据计算机访问文件的方式可将文件分成顺序文件，随机文件。

6.2.1 顺序文件

☞ 知识要点分析

顺序文件即文本文件，其中的记录按顺序一个个排列，读取时必须按顺序逐个进行。顺序文件无法灵活地存取和增减数据，适合不经常修改的数据。优点是占空间少，容易使用。

1. 打开文件

Visual Basic 用 Open 语句打开或建立文件，其格式为：

```
Open"文件名"for <打开方式> As #文件号
```

打开方式又包括：

(1) Output: 把数据写入文件，文件不存在创建新文件，文件已存在覆盖原文件。

(2) Append: 追加数据到文件末尾，文件不存在创建新的文件。

(3) Input: 从文件中读入数据，文件如不存在，则会出错。

文件号为 1~512。

2. 写文件

1) Print#文件号，输入项列表

例如：

```
Print #1,a,b,c
```

把变量 a, b, c 的值写入文件号为 1 的文件中。

2) Write#文件号，输入项列表

例如：

```
Write #1,a,b,c
```

将把变量 a, b, c 的值写入文件号为 1 的文件中。

3. 读文件

1) Line input#文件号，变量名

按行读入文件。

2) Input#文件号，变量列表

按数据读入文件。

4. 关闭文件

例如 Close #1 是关闭 1 号文件。

✎ 应用实例

【实例 6-6】 编写程序将 1~100 写入 fl.txt 文件中，并比较 Print 与 Write 的区别，效果如图 6-6 所示。

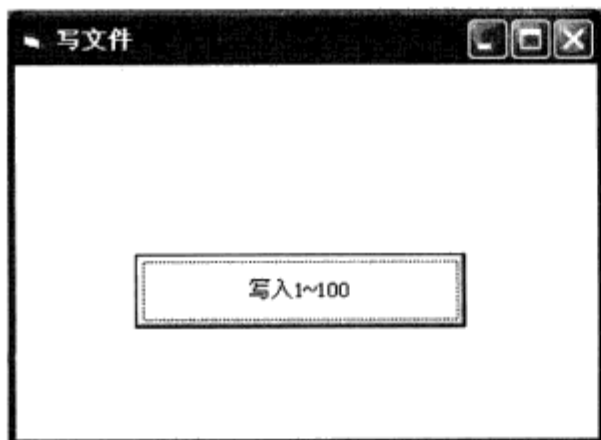


图 6-6 写文件窗体

代码如下。

```
Private Sub Command1_Click()
Open App.Path & "\f1.txt" For Output As #1'APP.Path是取当前文件所在的路径
For i = 1 To 100
    'Print #1, i      '换行写入文件
    'Write #1, i     '换行写入文件与Print #1, i没有本质的区别
    'Print #1, i;    '不换行写入文件,数据间用空格分隔
    Write #1, i;     '不换行写入文件,数据间用逗号分隔
Next
Close #1
End Sub
```

【实例 6-7】 打开 f1.txt 文件，将 101 到 200 追加到 f1.txt 中。效果如图 6-7 所示。
代码如下：

```
Private Sub Command1_Click()
Open App.Path & "\f1.txt" For Append As #1
For i = 101 To 200
    Write #1, i;
Next
Close #1
End Sub
```



图 6-7 追加数据窗体

【实例 6-8】 从 f2.txt 中读取若干个数，求出其中奇数个数，偶数个数。f2.txt 文件中的数据如图 6-8 所示。
运行效果如图 6-9 所示。

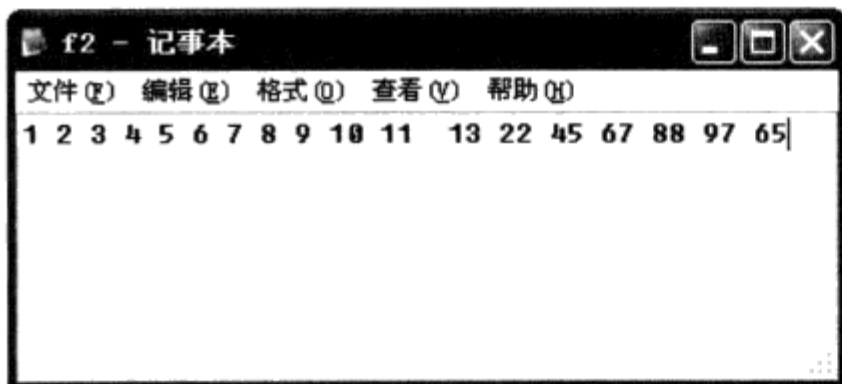


图 6-8 文件 f2.txt



图 6-9 数据奇偶性判断窗体

代码如下：

```
Private Sub Command1_Click()
Dim n As Integer, cnt1 As Integer, cnt2 As Integer
Open App.Path & "\f2.txt" For Input As #2
Do While EOF(2) = False
    Input #2, n
```

```

    If n Mod 2 = 0 Then
        cnt2 = cnt2 + 1
    Else
        cnt1 = cnt1 + 1
    End If
Loop
Text1.Text = cnt1
Text2.Text = cnt2
End Sub

```

【实例 6-9】 把 1~100 被 3 和 7 同时整除的数存放到文本框中，通过模块及参数传递把文本框中的数据存入到 f3.txt 中，效果如图 6-10 所示。

代码如下。

```

Private Sub Command1_Click() '计算
For i = 1 To 100
If i Mod 3 = 0 And i Mod 7 = 0 Then
Text1.Text = Text1.Text & i & " "
End If
Next
End Sub

```

```

Private Sub Command2_Click() '写文件
'Call putdata("f3.txt", Text1.Text)
putdata "f3.txt", Text1.Text
End Sub

```

```

Public Sub putdata(filename As String, t As Variant) '过程
Open App.Path & "\" & filename For Output As #512
Print #512, t
Close #512
End Sub

```

【实例 6-10】 从 f4.txt 文本文件中读取若干行英文，显示在文本框中，效果如图 6-11 所示。

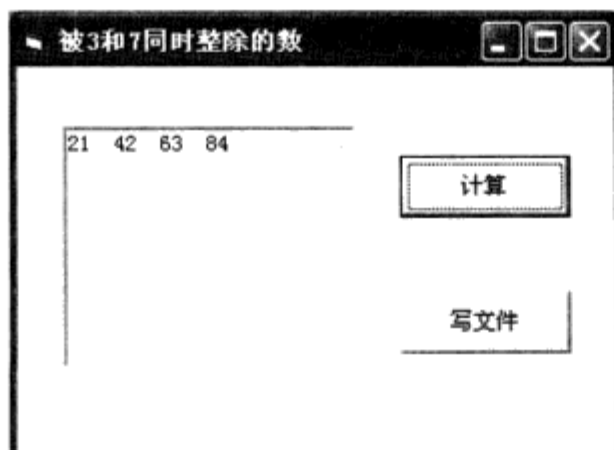


图 6-10 被 3 和 7 同时整除的数

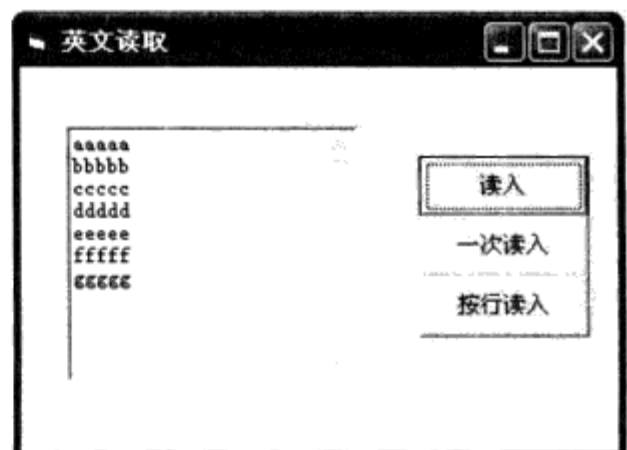


图 6-11 文件读取窗体

代码如下。

```
Private Sub Command1_Click() '读入
Open App.Path & "\f4.txt" For Input As #1
Dim k As String
Do While Not EOF(1) '判断是否到文件尾
    Input #1, k
    Text1.Text = Text1.Text & k & vbCrLf
Loop
Close #1
End Sub
```

```
Private Sub Command2_Click() '一次读入
Open App.Path & "\f4.txt" For Input As #1
Dim k As String
k = Input(LOF(1), #1) '读取函数, 一次读入
Text1.Text = k
Close #1
End Sub
```

```
Private Sub Command3_Click() '按行读入
Open App.Path & "\f4.txt" For Input As #1
Dim k As String
Do While Not EOF(1)
    Line Input #1, k '一次读取一行
Text1.Text = Text1.Text & k & vbCrLf
Loop
Close #1
End Sub
```

6.2.2 随机文件

随机文件是指通过记录号来进行存取文件, 存取文件时不必考虑记录的先后顺序和位置, 可以根据需要访问任意一个记录。

知识要点分析

1. 随机文件数据类型的定义

通过 Type...End Type 进行定义, 例如:

```
Private Type stu
xh As String * 4 '学号宽度为 4
xm As String * 10 '姓名宽度为 10
cj As Single '成绩宽度为 4
End Type
```

其中学号、姓名、成绩相当于数据库表中的字段, 一般在通用部分进行定义。

2. 随机文件的打开

```
Open "文件名" for Random As #文件号 len=记录长度
```

与顺序文件的打开方式一样, 将 For 后面的打开方式改为 Random, 就变成了随机文件的打开。

3. 随机文件的写入

Put #文件号,记录号,变量

4. 随机文件的读取

Get #文件号,记录号,变量

其中 Put 和 Get 中的变量如果为 a,各个数据的表示为: a.xh a.xm a.cj。

5. 随机文件的关闭

随机文件的关闭和顺序文件相同。

Close #文件号

应用实例

【实例 6-11】 将表 6-1 的记录写入随机文件 f5.txt 中。效果如图 6-12 所示。

表 6-1 学生成绩

学号	姓名	成绩
1001	张三	70.5
1002	李四	90
1003	王五	60

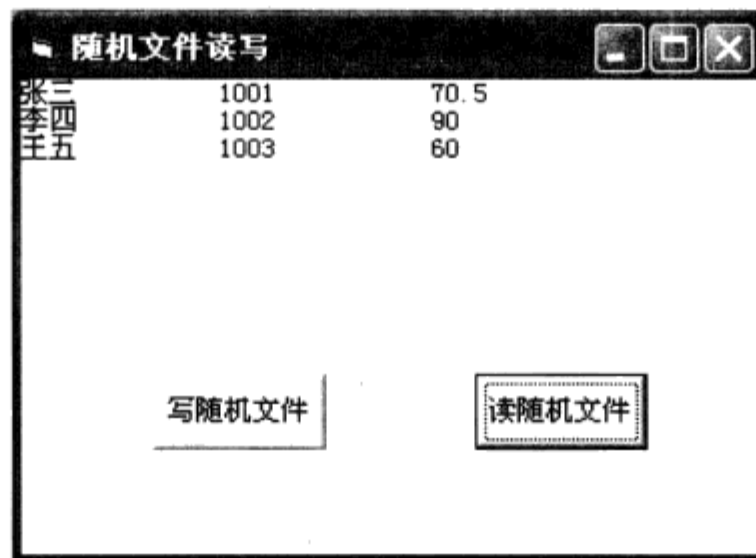


图 6-12 随机文件窗体

代码如下。

```
Private Type stu '定义数据类型
    xh As String * 4 '学号宽度为 4
    xm As String * 10 '姓名宽度为 10
    cj As Single '成绩宽度为 4
End Type

Private Sub Command1_Click() '写随机文件
    Dim a As stu, b As stu, c As stu
    Open App.Path & "\F5.txt" For Random As #1 Len = 18
    a.xh = "1001"
    a.xm = "张三"
```

```
a.cj = 70.5
Put #1, 1, a
b.xh = "1002"
b.xm = "李四"
b.cj = 90
Put #1, 2, b
c.xh = "1003"
c.xm = "王五"
c.cj = 60
Put #1, 3, c
Close #1
End Sub

Private Sub Command2_Click() '读随机文件
Dim d As stu
Open App.Path & "\F5.txt" For Random As #1 Len = Len(d)
n = LOF(1) / Len(d)
For i = 1 To n
    Get #1, i, d
    Print d.xm, d.xh, d.cj
Next
Close #1
End Sub
```

6.2.3 文件函数

☎ 知识要点分析

文件读写操作过程中经常使用一些函数，下面把文件中常用的几个函数做一下介绍。

1. EOF（文件号）函数

该函数是判断是否到文件尾，如果到文件尾函数值为 True，如果没到文件尾函数的返回值为 False。

2. LOC（文件号）函数

在随机文件中返回当前记录的记录号。

3. LOF（文件号）函数

该函数是返回文件的字节数，即文件的大小。在随机文件中，用文件的大小除以单个记录的大小，可以得到文件的记录总数。

✍ 应用实例

【实例 6-12】 编写一个通用的随机文件录入程序，如图 6-13 所示。

代码如下。

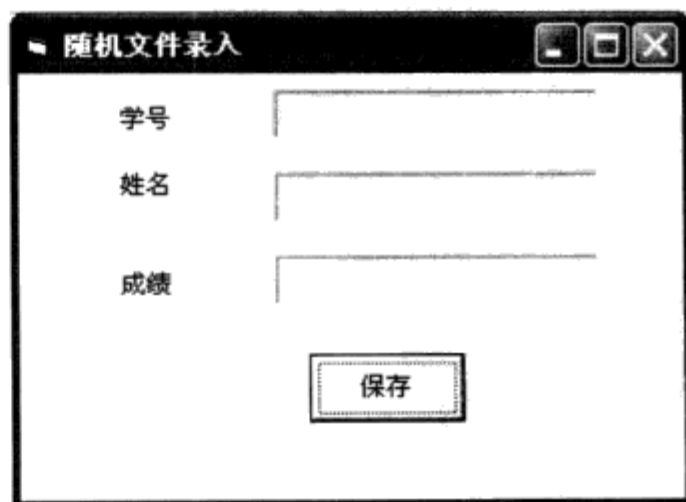


图 6-13 随机文件录入窗体

```

Private Type stu           '定义数据类型
    xh As String * 4       '学号宽度为 4
    xm As String * 10     '姓名宽度为 10
    cj As Single          '成绩宽度为 4
End Type

Private Sub Command1_Click() '写随机文件
    Dim a As stu
    Open App.Path & "\F6.txt" For Random As #1 Len = Len(a)
    n = LOF(1) / Len(a)    '文件的记录总数
    a.xh = Text1.Text
    a.xm = Text2.Text
    a.cj = Text3.Text
    Put #1, n + 1, a       '新记录号是在总记录号的基础上加 1
    Close #1
    Text1.Text = ""       '清空保存的数据
    Text2.Text = ""
    Text3.Text = ""
    Text1.SetFocus        '设置光标停在 Text1 上
End Sub

```

【实例 6-13】 编写一个用记录号查找的随机文件数据的程序，如图 6-14 所示。

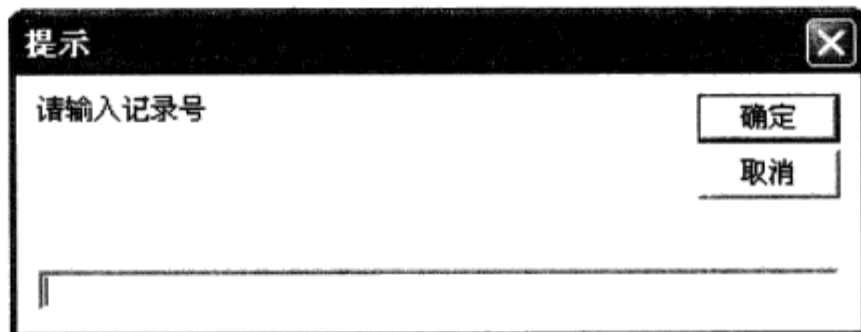


图 6-14 信息输入对话框

如果输入记录号为 3，查找结果如图 6-15 所示。

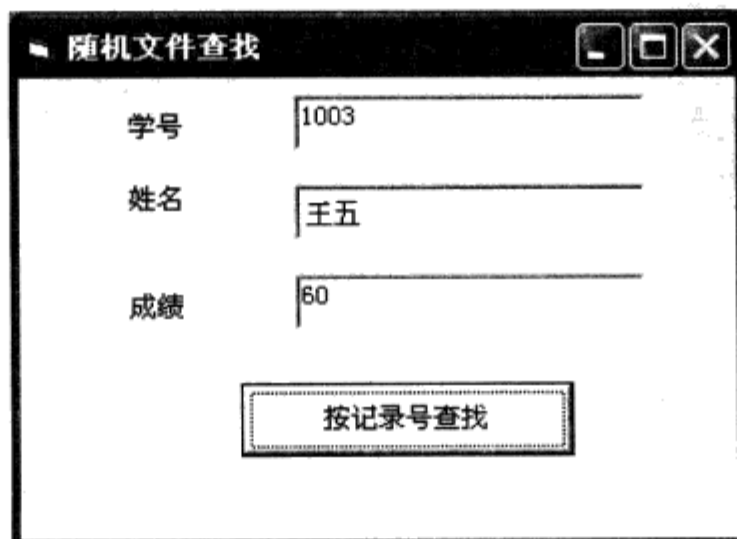


图 6-15 随机文件的查找

如果输入的记录号不存在，如下图 6-16 所示。
程序代码如下。

```
Private Type stu
    xh As String * 4
    xm As String * 10
    cj As Single
End Type

Private Sub Command1_Click() '写随机文件
    Dim a As stu, recno As Integer
    Open App.Path & "\F6.txt" For Random As #1 Len = Len(a)
    n = LOF(1) / Len(a) '文件的记录总数
    recno = InputBox("请输入记录号", "提示")
    If recno = 0 Or recno > n Then
        MsgBox "记录号输入错误", 0, "错误提示"
    Else
        Get #1, recno, a
        Text1.Text = a.xh
        Text2.Text = a.xm
        Text3.Text = a.cj
    End If
    Close #1
End Sub
```



图 6-16 错误提示

6.3 文件控件及使用

☎ 知识要点分析

6.3.1 驱动器列表控件、目录列表控件、文件列表控件

在 VB 中，文件系统控件包括驱动器列表、目录列表、文件列表等。

1. 驱动器列表

驱动器列表主要是返回选中的驱动器，通过 Drive1.Drive 实现。

2. 目录列表

目录列表主要是返回选中驱动器的文件夹所在的路径，通过 Dir1.Path 实现。

3. 文件列表

文件列表主要是返回选中路径下的文件，主要有下列几个用法。

返回文件列表的路径通过 File1.Path 实现。

返回文件列表中选中的文件通过 File1.Filename 实现。

刷新文件列表的方法通过 File1.Refresh 实现。

设置文件是否可以多重选择通过 File1.Multiselect 实现。

设置文件的类型通过 File1.Pattern 实现。

应用实例

【实例 6-14】 通过驱动器列表、目录列表、文件列表实现三者间的关联，在文件列表中选中的文件在文本框中显示出来，如图 6-17 所示。

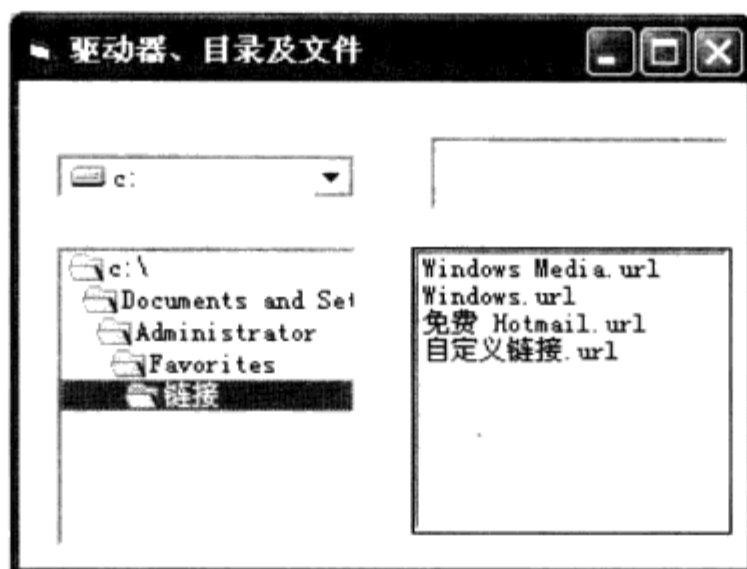


图 6-17 驱动器、目录及文件列表窗体

代码如下。

```
Private Sub Dir1_Change() '目录列表带动文件列表
File1.Path = Dir1.Path
End Sub

Private Sub Drive1_Change() '驱动器列表带动目录列表
Dir1.Path = Drive1.Drive
End Sub

Private Sub File1_Click() '选中的文件在文本框中显示
Text1.Text = File1.FileName
End Sub
```

6.3.2 通用对话框的使用

知识要点分析

通用对话框主要实现文件的打开、保存、字体设置、颜色设置及打印等功能。

1. 通用对话框的添加

在工具箱上右击，然后在快捷菜单中选择部件，会打开部件对话框，然后选择 Microsoft Common Dialog Control 控件，如图 6-18 所示。

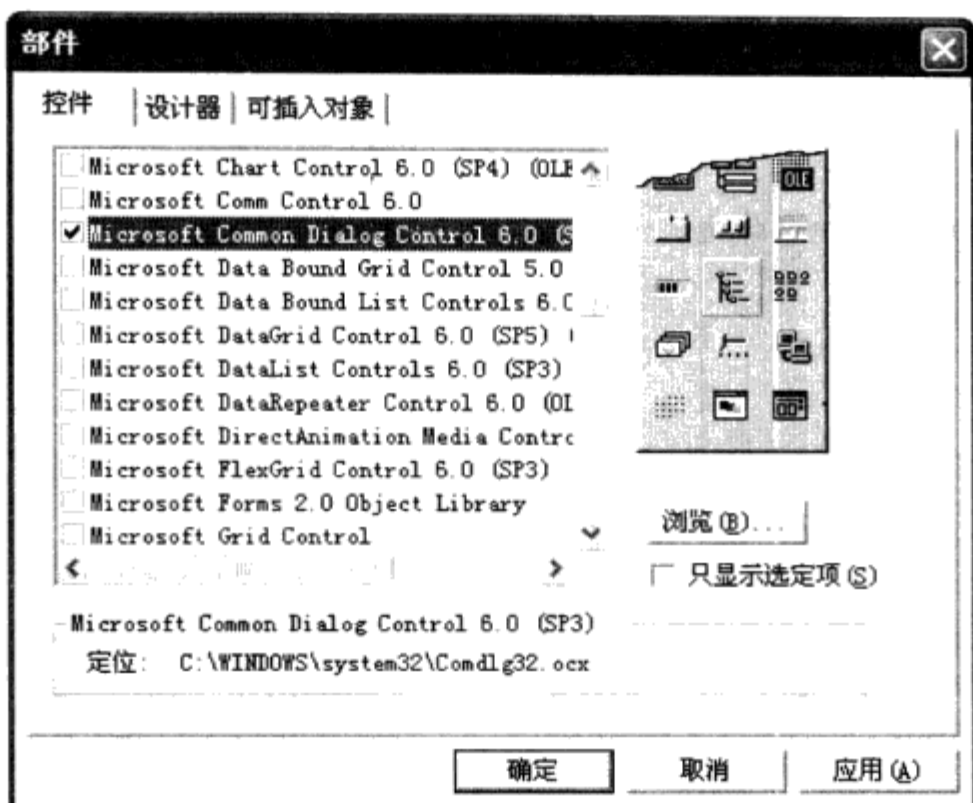


图 6-18 通用对话框的添加

2. 通用对话框 Action 属性值

通用对话框主要是通过 `CommonDialog1.Action=N` 来设置打开方式, 属性值及其含义如表 6-2 所示。

表 6-2 通用对话框 Action 属性值

Action 值	控件方法	含义
1	Showopen	打开
2	Showsave	保存
3	Showcolor	颜色
4	Showfont	字体
5	Showprinter	打印

3. 通用对话框的基本属性 (表 6-3)

表 6-3 通用对话框的基本属性

属性	含义
Dialogtitle	对话框标题
Filename	设置对话框中的文件名初值, 也可返回用户选中的文件
Initdir	设置默认路径
Filter	设置文件类型
Filterindex	设置显示文件类型的缺省类型

应用实例

【实例 6-15】 利用通用对话框, 打开 f 7.txt 文本文件, 然后将其保存为 f 8.txt 文件, 如图 6-19 所示。

单击“打开”如图 6-20 所示。

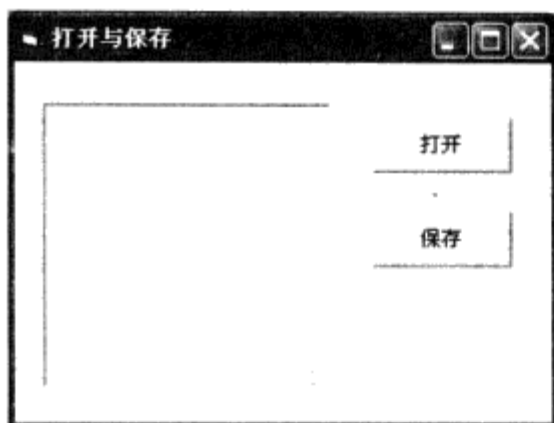


图 6-19 “打开与保存”窗体

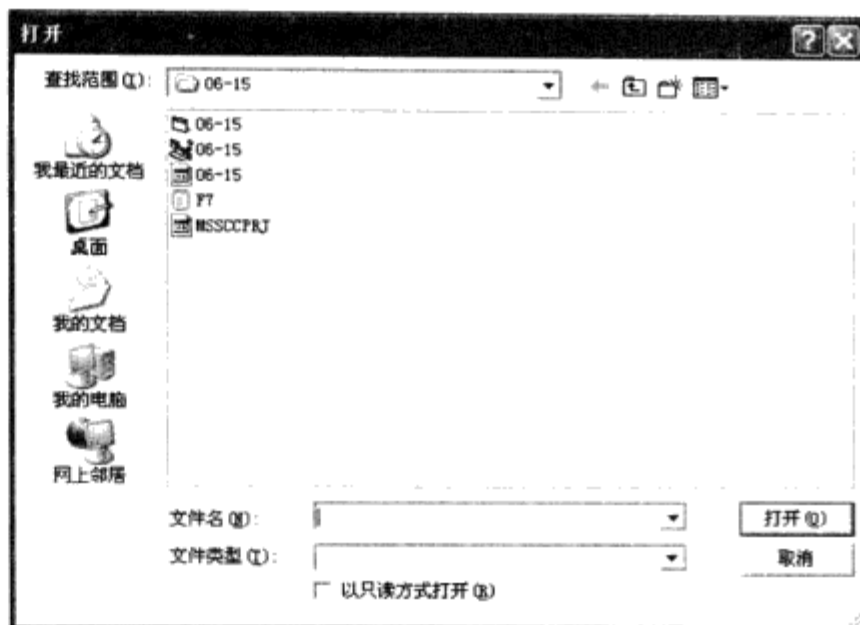


图 6-20 “打开”窗体

选择 f7.txt 文件后，打开的效果如图 6-21 所示。

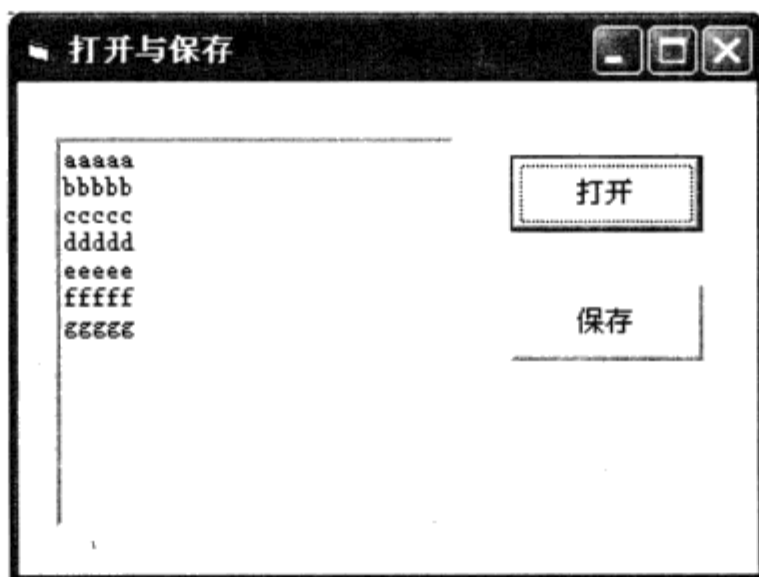


图 6-21 文件 f7.txt

单击“保存”，效果如图 6-21 所示，输入 f8.txt 将文本框的内容保存到 f8.txt 中，如图 6-22 所示。

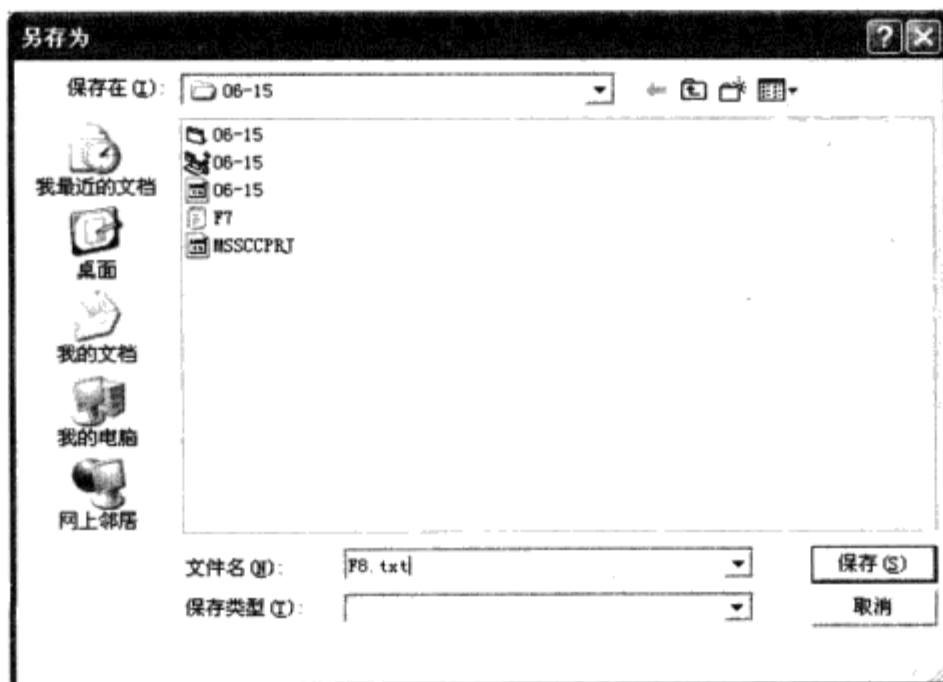


图 6-22 “另存为”窗体

程序代码如下。

```
Private Sub Command1_Click() '打开
    'CommonDialog1.Action = 1
    CommonDialog1.ShowOpen'与 CommonDialog1.Action = 1 等价
    Open CommonDialog1.FileName For Input As #1
    Do While EOF(1) = False
        Line Input #1, A
        Text1.Text = Text1.Text & A & vbCrLf
    Loop
    Close #1
End Sub

Private Sub Command2_Click() '保存
    'CommonDialog1.Action = 2
    CommonDialog1.ShowSave'与 CommonDialog1.Action = 2 等价
    Open CommonDialog1.FileName For Output As #1
    Print #1, Text1.Text
    Close #1
End Sub
```

【实例 6-16】 利用通用对话框设置窗体的背景颜色，如图 6-23 所示。

单击颜色会弹出如图 6-24 所示窗口，然后选择颜色即可以设置窗体的背景颜色。



图 6-23 通用对话框窗体

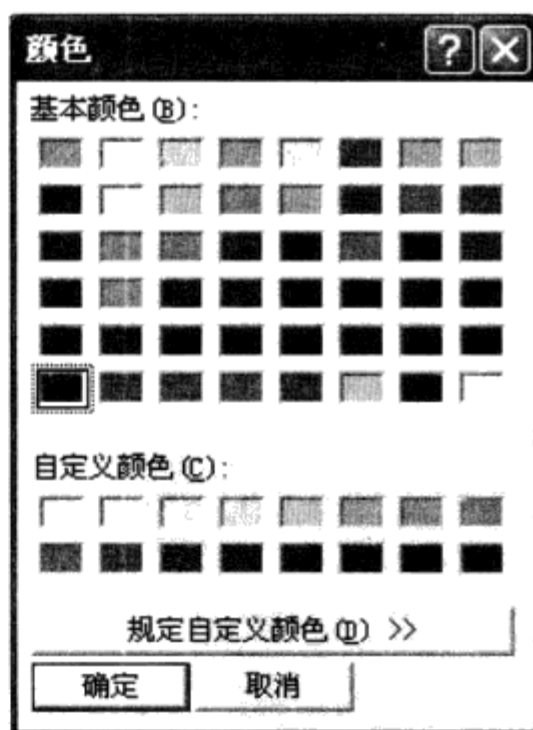


图 6-24 背景颜色

程序代码如下。

```
Private Sub Command1_Click() '颜色
    'CommonDialog1.Action = 3
```

```

CommonDialog1.ShowColor           '与 CommonDialog1.Action = 3 等价
Form1.BackColor = CommonDialog1.Color
End Sub

```

本章小结

本章通过 Sub 过程和 Function 过程对 Visual Basic 过程的创建和调用进行了分析。通过顺序文件和随机文件对 Visual Basic 文件操作进行讲解，通过驱动器列表实现对文件进行管理，通过文件操作函数实现对数据管理。

习 题 6

一、判断题

1. 在 Sub 过程中可以调用 Function 过程。()
2. 在 Sub 过程中可以嵌套定义 Function 过程。()
3. Function 过程必须有返回值。()
4. 只有函数过程能够将过程中处理的信息传回到调用的程序中。()
5. 用数组作为过程的参数时，使用的是“按地址传递”方式。()
6. 如果不指明函数过程参数的类型，则该参数没有数据类型。()
7. Sub 过程与 Function 过程最根本的区别是两种过程参数的传递方式不同。()
8. 在某个 Sub 过程中定义的局部变量可以与其他事件过程中定义的局部变量同名，但其作用域只限于该过程。()
9. 建立过程时，过程名后的称为形参，调用过程时，过程名后的称为实参，形参与实参之间保持类型一致即可。()
10. 在声明过程时，形参可以使用关键字 ByVal 或 ByRef 声明，若使用 ByRef 声明，则在调用此过程时，该参数将按值传递。()
11. 以 Append 的方式打开文件可以进行数据的追加，若文件不存在则不能追加数据。()
12. 设打开了 5 个文件，文件号为 1,2,3,4,5 要关闭所有文件可以用一条 Close 语句实现。()
13. Bof 函数可以判断是否可以判断是否到达文件尾。()
14. 随机文件中记录的长度是固定不变的，可以按任意顺序访问随机文件中的数据。()
15. 设有语句 Open “C:\Test.Dat” For Input As #1，该语句可以打开 C 盘根目录下下一个已存在的文件 Test.Dat，若文件不存在则创建新文件。()
16. 在随机文件中记录号是通过随机函数随机产生的。()
17. 获得打开文件的长度（字节数）应使用 LEN 函数。()

18. 在 Visual Basic 中, 文件的操作步骤是打开或新建, 进行读、写操作, 关闭文件。()

19. Eof 函数的返回值是记录号。()

20. 文件号的范围介于 1~512 之间, 使用时有先后顺序。()

二、选择题

1. 下面过程定义语句中合法的是 ()。

- A. Sub P1(ByVal x As Integer) B. Sub P1(x%) As Single
C. Function P1(P1) D. Function P1(ByVal P1)

2. 要想在过程调用后返回两个结果, 下面的过程定义语句合法的是 ()。

- A. Sub PP(ByVal x%,ByVal y%) B. Sub PP(x%,ByVal y%)
C. Sub PP(x%,y%) D. Sub PP(ByVal x%,y%)

3. 定义两个过程 Private Sub Sub1(Sr() As String)和 Private Sub Sub2(Ch() As String*6), 在调用中用 Dim S(3) As String*6, A(3) As String 定义了两个字符串数组, 下面调用语句中正确的有 ()。

- ①Call Sub1(S) ②Call Sub1(A) ③Call Sub2(A) ④Call Sub2(S)
A. ①, ② B. ①, ③ C. ②, ③ D. ②, ④

4. 在过程定义中, Private 表示 ()。

- A. 此过程可以被其他过程调用
B. 此过程不可以被任何其他过程调用
C. 此过程只可以被本工程中的其他过程调用
D. 此过程只可以被本窗体模块中的其他过程调用

5. 若在应用程序的标准模块、窗体模块和过程 Sub1 的说明部分, 分别用 Public G As Integer、Private G As Integer, Dim G As Integer 语句声明了 3 个同名变量 G。如果在过程 Sub1 中使用赋值语句 G=35, 则该语句是在给 () 部分定义的变量 G 赋值。

- A. 标准模块 B. 过程 Sub1 C. 窗体模块 D. 以上 3 个都是

6. 在窗体上画 1 个名称为 Command1 的命令按钮, 并编写如下程序。

```
Private Sub Command1_Click()
    Dim x As Integer, y As Integer
    x = 10: y = 5
    Call f1(x, y)
    Print x; y
End Sub
Private Sub f1(ByVal x1 As Integer, y1 As Integer)
    x1 = x1 + 2
    y1 = y1 + 2
End Sub
```

程序运行后, 单击命令按钮, 在窗体上显示的内容是 ()。

- A. 10 5 B. 12 5 C. 10 7 D. 12 7

7. 有如下的程序。

```

Private Sub Form_Click()
    Dim a As Integer, b As Integer, c As Integer
    a = 6: b = 8: c = 3
    Call test(a, b, c + 1)
    Print "主程序"; 6; b; c
End Sub
Private Sub test(x As Integer, y As Integer, ByVal z As Integer)
    Print "子过程"; x; y; z
    x = 2
    y = 4
    z = 9
End Sub

```

当运行程序后，在窗体上显示的结果是（ ）。

- A. 子过程 6 4 3 B. 主程序 6 4 3 C. 主程序 6 8 4 D. 子过程 6 8 4
 主程序 6 8 4 子过程 6 8 4 子过程 6 4 3 主程序 6 4 3

8. 如下程序的运行结果是（ ）。

```

Private Sub Form_Click()
    Dim m As Integer, i As Integer, x(10) As Integer
    For i = 0 To 4
        x(i) = i + 1
    Next i
    For i = 1 To 2
        Call proc(x)
    Next i
    For i = 0 To 4
        Print x(i)
    Next i
End Sub
Public Sub proc(a() As Integer)
    Static i As Integer
    Do
        a(i) = a(i) + a(i + 1)
        i = i + 1
    Loop While i < 2
End Sub

```

- A. 3 4 7 5 6 B. 3 5 7 4 5 C. 2 3 4 4 5 D. 4 5 6 7 8

9. 窗体上有 2 个文本框 Text1, Text2 及 1 个命令按钮 Command1, 编写下列程序。

```

Dim y As Integer
Private Sub Command1_Click()
    Dim x As Integer
    x = 2
    Text1.Text = p2(p1(x), y)

```

```

    Text2.Text = p1(x)
End Sub
Private Function p1(x As Integer) As Integer
    x = x + y
    y = x + y
    p1 = x + y
End Function
Private Function p2(x As Integer,y As Integer) As Integer
    p2 = 2 * x + y
End Function

```

当程序运行时，单击命令按钮后，文本框 Text1 和 Text2 中的值分别是 ()。

- A. 2 4 B. 8 4 C. 10 10 D. 4 4

10. 在窗体上画 1 个文本框 Text1 和 1 个命令按钮 Command1，然后编写如下事件过程。

```

Private Sub Command1_Click()
    Text1.Text = ""
    Text1.SetFocus
    S = Fc(2) + Fc(3) + Fc(4)
    Text1.Text = S
End Sub
Private Function Fc(x As Integer) As Long
    If x = 1 Then
        Fc = 1
    Else
        Fc = Fc(x - 1) * x
    End If
End Function

```

上述程序的运行结果是 ()。

- A. 30 B. 32
C. 28 D. 本框 Text1 中输出不定值

11. 在顺序文件中 ()。

- A. 按每条记录的记录号从小到大排序
B. 按每条记录的长度从小到大排序
C. 按记录的某个关键数据项的排序顺序组织文件
D. 记录按写入的先后顺序存放，并按写入的先后顺序读出

12. 在随机文件中，每条记录必须 ()。

- A. 内容不一样 B. 长度不相等
C. 长度相等 D. 排序

13. 顺序文件的读操作通过下列 () 语句可以实现。

- A. Input #和 Read # B. Read #和 Get #
C. Get #和 Input # D. LineInput #和 Input #

14. 随机文件写数据和读数据的语句分别是 ()。
- A. Put 和 Get B. Get 和 Put C. Print 和 Input D. Write 和 Input
15. 执行语句 `open "Tel.Dat" For Random As #1 Len=50` 后, 对文件 Tel.Dat 中的数据能够执行的操作是 ()。
- A. 只能写, 不能读 B. 只能读, 不能写
C. 既可以读, 也可以写 D. 不能读, 不能写
16. 下面叙述中不正确的是 ()。
- A. 对顺序文件中数据的操作只能按一定的顺序执行
B. 顺序文件结构简单
C. 能同时对顺序文件进行读写操作
D. 顺序文件的数据是以字符 (ASCII 码) 的形式存储的
17. 下面叙述中不正确的是 ()。
- A. 随机文件中记录的长度不是固定不变的
B. 随机文件由若干条记录组成, 并按记录号引用各个记录
C. 可以按任意顺序访问随机文件中的数据
D. 可以同时打开的随机文件进行读写操作
18. 要在 C 盘当前文件夹下建立一个名为 StuDatA.dat 的顺序文件, 应先使用 () 语句。
- A. `Open"StuDatA.dat" For Output As #2`
B. `Open"C:StuDatA.dat" For Input As #2`
C. `Open"C:StuDatA.dat" For Output As #2`
D. `Open"StuDatA.dat" For Input As #2`
19. 执行语句 `Open"C:StuDatA.dat" For Input As #2` 后, 系统 ()。
- A. 将 C 盘当前文件夹下名为 StuDatA.dat 的文件的内内容读入内存
B. 在 C 盘当前文件夹下建立名为 StuDatA.dat 的顺序文件
C. 将内存数据存放在 C 盘当前文件夹下名为 StuDatA.dat 的文件中
D. 将某个磁盘文件的内容写入 C 盘当前文件夹下名为 StuDatA.dat 的文件中
20. 下面叙述中不正确的是 ()。
- A. 若使用 `Write#` 语句将数据输出到文件, 则各数据项之间自动插入逗号, 并且将字符串加上双引号
B. 若使用 `Print#` 语句将数据输出到文件, 则各数据项之间没有逗号分隔, 且字符串不加双引号
C. `Write#` 语句和 `Print#` 语句建立的顺序文件格式完全一样
D. `Write#` 语句和 `Print#` 语句均实现向文件中写入数据

第7章

数据库基础知识

数据库技术是计算机科学的一个重要分支，是现代数据管理技术的核心技术，由于其具有数据结构化、冗余度低、容易扩充和容易操作等特点，目前被广泛使用在各种信息系统中。随着计算机技术的不断发展，数据库应用也从最初的一般管理渗透到各行各业，在国民经济中发挥着越来越大的作用。

7.1 数据库系统的基本概念

信息是指现实世界各种事物的特征、形态以及不同事物间的联系等在人脑中的抽象反映。对这些经抽象而形成的概念，人们可以理解，可以加工传播，可以进行推理，从而达到认识世界、改造世界的目的。人类的社会活动是受信息支配的，信息在人类社会活动中有着非常重要的意义。数据是表达信息的载体，是一切文字、符号、声音、图像等有意义元素的组合。在计算机中数据的存储和处理都是以二进制的形式进行的。

现实世界事物的存在特征和变化状态都有其自身的规律，各种事物之间也存在着种种联系。人们可从已知的信息中，经过加工推理得到新的信息，并以此作为社会活动的决策依据。这一过程就叫做信息处理。由于信息是用数据表示的，所以信息处理具体体现在数据处理上。数据处理是对数据的收集、记载、整理、组织、存储、检索、计算加工、维护、传送等一系列活动的总称。作为数据处理过程中的一个基本环节，数据管理的优劣直接影响到数据处理的效率。

7.1.1 数据管理技术的发展

知识要点分析

数据管理是指数据的组织、编目、定位、存储、检索和维护。数据管理技术的发展大体上经历了三个阶段：人工管理阶段、文件系统管理阶段和数据库管理阶段。

早期数据的处理只能用手工的方式来进行，随着计算机技术的不断成熟和发展，产生了文件系统。数据的存放和管理都以数据文件的形式来进行。文件中最小的数据单位是数据项，数据项用于描述事物的某个特征，若干个相关数据项的集合称为记录，一个记录能够表达一个具体的事物，若干个记录的集合称为文件。用户通过对文件的访问实现对记录

的存取。虽然文件系统将人们从手工管理数据繁重的工作中解脱出来，但随着数据管理规模的日渐扩大和数据量的不断增加，其发展已越来越不能满足人们对于数据处理的需要，于是数据库系统应运而生。

1968年，美国IBM公司推出层次模型的IMS（Information Management System），它是第一个被商品化了的数据库系统。1969年，美国数据系统语言协会（CODASYL）的数据库任务组（DBTG）发表了网状模型的DBTG报告；1970年，E.F.Codd连续发表论文，奠定了关系型数据库的理论基础。从此数据库系统成为数据管理新的主流和方向。

与文件系统相比，数据库系统具有以下特点。

1. 数据结构化——用数据模型表示数据结构

数据的结构化是数据库系统与文件系统的本质区别。在文件系统中，由于文件彼此之间没有联系，其数据在整体上是没有任何结构的。而在数据库系统中，数据是按照一定的数据模型来组织和存放的，数据模型不仅描述数据本身，同时还描述了数据间的联系，因而数据文件间是有联系的。这种联系借助于具有相同含义的公共关键字段通过关联或连接加以实现。

2. 有较高的数据独立性

数据独立性一直是数据库技术努力追求的目标。简单地说，数据独立性就是令数据与程序无关，数据存储方式的改变不会影响到应用程序。由于数据库系统提供了映射功能，确保了应用程序对数据结构和存取方法有较高的独立性。数据库系统的数据独立性分两级，即物理独立性和逻辑独立性。当数据库物理结构（包括数据的组织和存储、存取办法、外部存储设备等）发生变化时，修改整体逻辑结构和物理结构之间的映射，使数据库整体逻辑结构不受影响，进而用户的逻辑结构及应用程序也不用改变，此时称数据库达到了物理数据独立性。同样，当数据库的整体逻辑结构发生改变时，只需修改用户逻辑结构与整体逻辑结构之间的映射，使用户的逻辑结构及应用程序不受影响，此时称数据库达到了逻辑数据独立性。数据的独立性增加了数据库系统的稳定性，提高了应用系统的开发效率，降低了程序维护的开销。

3. 数据共享

所有用户的数据都包含在数据库中，每个用户（应用程序或终端用户）通常只用到其中的一部分数据。数据库的不同用户所使用的数据可以重叠，一个数据库中的同一数据可供多个用户使用，这就是数据共享。数据共享不仅是文件系统与数据库系统的重要区别，同时也是数据库技术先进性的一个重要体现。

例如，高校学生成绩管理数据库，除了面向各级教务管理部门和教师以外，学生也可以查询，同时其中的学生数据不但可以在本系统中使用，同样也可以用在学生宿舍管理、图书馆图书管理、学校餐饮管理等诸多方面，如图7-1所示。

4. 较少的数据冗余度

在数据库系统中，用户不是自己建立文件，而是取数据库中的数据子集。用户使用的逻辑文件是数据库管理系统从数据库中映射而成的。同一个数据可能在物理存储上只存一次，但可以把它映射到不同的逻辑文件中。这就是为什么数据可以减少冗余，可以实现共享的根本原因。减少数据冗余，不仅可以提高存储的效率，而且也保证了数据访问的一致性。

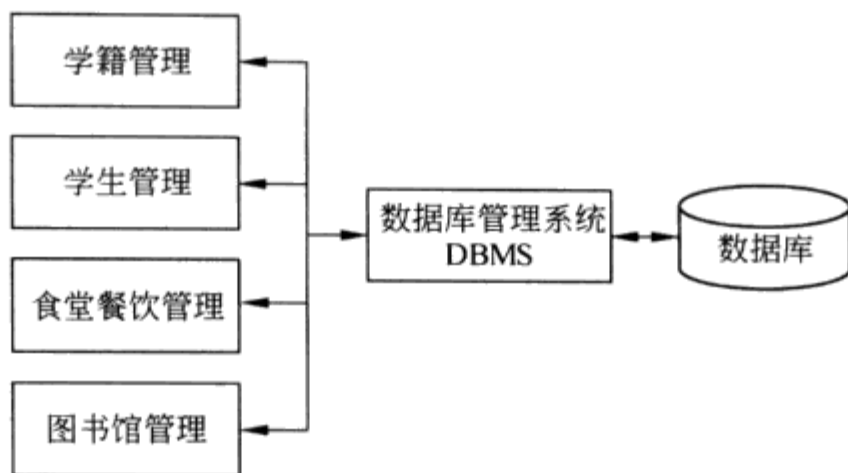


图 7-1 学生数据应用示意图

5. 为用户提供方便的用户接口

数据库管理系统作为用户与数据库的接口，提供了数据库定义、数据库运行、数据库维护等多种功能。用户可以使用查询语言或终端命令操作数据库，也可以使用程序方式操作数据库。

6. 对数据实行集中统一的管理

数据库系统可提供数据控制功能，包括数据库的恢复、并发控制、数据安全性和数据完整性，以保证数据库中的数据是安全的、正确的和可靠的。

7.1.2 数据库系统的基本概念

☞ 知识要点分析

1. 数据库

数据库 (Database, DB) 是长期存储在计算机内的有组织的可共享的数据的集合。数据库中的数据是按照一定的数据模型组织、描述和存储的，具有较小的冗余度、较高的程序数据独立性及易扩展性等特点，并且可供各种用户、各个应用程序所共享。

2. 数据库管理系统

数据库管理系统 (Database Management System, DBMS) 是用户与数据库的接口，用户对数据库的一切操作，包括定义、查询、更新及各种控制等，都需要通过 DBMS 来进行。无论开发还是运行数据库系统，都需要 DBMS 的支持。

DBMS 总是基于某种数据模型的。按照数据模型，DBMS 可以分为层次型、网状型、关系型、面向对象型等。目前主流的数据库管理系统，如 Oracle、Sybase、DB2、SQL Server、Access、Visual FoxPro 等，都是以关系型数据模型作为主要的数据模型。

数据库管理系统应具有以下主要功能。

(1) 数据定义功能：DBMS 提供数据定义语言 (Data Definition Language, DDL)，用以定义数据库中的数据对象。

(2) 数据操作功能：对数据进行检索和查询，是数据库的主要应用。为此，DBMS 提供数据操作语言 (Data Manipulation Language, DML)，支持用户对数据库中的数据进行查询、更新 (包括插入、删除和修改) 等操作。

(3) 数据库的运行管理：DBMS 对数据库的运行管理主要包括数据的安全性控制、数

数据库的完整性控制、数据库的并发控制以及发生故障后的数据库恢复。

(4) 数据库的维护功能：数据库的维护功能包括数据库的输入、转换、转储和备份，数据库的重组、重构造以及性能监视和分析功能等。

3. 数据库应用系统

数据库应用系统 (Database Application System, DBAS) 是程序员根据用户的需要，在数据库管理系统的支持下开发的，并能够在数据库管理系统支持下运行的程序和数据库的总称，如财务管理系统、档案管理系统等。

4. 数据库系统

数据库系统 (Database System, DBS) 是指在计算机系统中引入数据库后的系统，通常由数据库、数据库管理系统 (及其开发工具)、数据库应用系统、数据库用户等构成。图 7-2 展示了数据库系统的构成，图 7-3 体现了数据库系统在计算机系统中的地位。

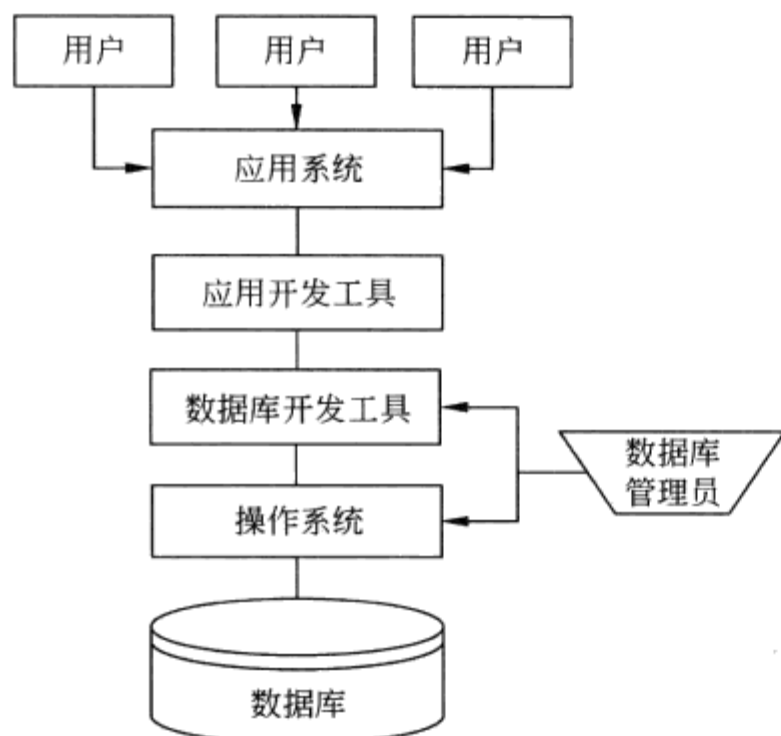


图 7-2 数据库系统的构成

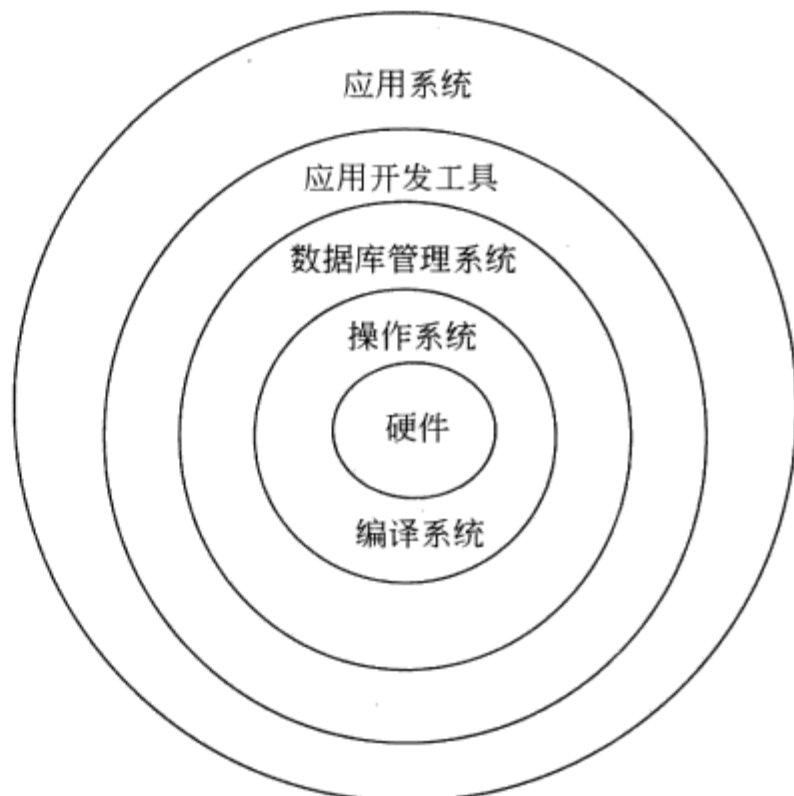


图 7-3 数据库系统在计算机系统中的地位

7.2 数据模型

数据模型是对客观事物及其联系的数据描述，它不仅要表示存储了哪些数据，而且要用某种结构形式表示不同数据之间的联系。数据库总是基于某种数据模型的，数据模型是数据库的框架。这个框架形式化地描述数据库的数据组织形式，在框架约束下填入具体数据就形成了实际的数据库。

7.2.1 常用术语

☞ 知识要点分析

数据模型具有两方面的功能，即数据描述与数据联系描述。在这些描述中，可能会

用到以下一些术语。

1. 实体

客观存在并可相互区分的事物称为实体，一个职工、一张桌子、一个学生、一门课程、一次考试、一场比赛都是一个实体。

2. 属性

实体所具有的特性称为属性，例如，学生实体具有学号、姓名、性别、年龄等属性；每个实体都有自己的一组属性值，例如，("200304110","王斌","男","19")。不同的实体可以根据各自不同的属性值来区分。

3. 域

属性的取值范围和取值类型被称为域，例如，性别属性的域是男或女的字符型数据。

4. 实体集

具有相同性质的同类实体的集合称为实体集。例如，一个班的学生、图书馆的所有书籍等，都可以看做一个实体集。由于在数据库设计中常常不关心每一个具体的实体，而是关心整个的实体集，所以在本书的讨论中，常常把实体集和实体不加区别地使用。

5. 实体标识符

能够唯一标识每个实体的属性或属性集称为实体标识符。例如，学生的学号可以作为学生的实体标识符。

上述概念是数据在信息世界中的一些称谓，当数据被存储到计算机中后，属性就成为数据项（或称字段），实体就成为记录，实体集就成为表文件，实体标识符就成为候选关键字或主关键字。

6. 联系

两个不同实体集的实体之间的联系简称为联系，可分为一对一联系、一对多联系和多对多联系三种。

一对一联系：对于实体集 A 中的每个实体，实体集 B 中至多有一个实体与之有联系；反之亦然，则称实体集 A 与实体集 B 具有一对一联系，记为 1:1。

一对多联系：若对于实体集 A 中的每一个实体，实体集 B 中有多个实体与之有联系；反之，对于实体集 B 中的每一个实体，实体集 A 中至多只有一个实体与之有联系，则称实体集 A 与实体集 B 具有一对多联系，记为 1: N 。

多对多联系：若对于实体集 A 中的每一个实体，实体集 B 中有多个实体与之有联系；反之，若对于实体集 B 中的每一个实体，实体集 A 中也有多个实体与之有联系，则称实体集 A 与实体集 B 是多对多联系，记为 $M:N$ 。

7.2.2 数据模型的分类

知识要点分析

常用的数据模型有两类，一类是概念数据模型，也称信息模型；另一类是逻辑数据模型，也称结构数据模型，如图 7-4 所示。

1. 概念数据模型

概念数据模型用来描述信息的结构，它只强调信息的特性和语义表达，完全不涉及信息在计算机内的表示和处理，是现实世界到计算机世界的第一层抽象。

比较著名和常用的概念数据模型是由 P.P.Chen 于 1976 年提出的实体联系模型 (Entity Relationship Model)，简称 ER 模型。ER 模型通过 ER 图表示实体及实体间的联系。在设计数据库时，先用 ER 图准确地反映信息，再从 ER 图出发，结合具体的计算机系统和 DBMS，构造实际的数据模型。

在 ER 图中，用矩形框表示实体型，用菱形框表示联系型，用椭圆形框表示属性，用直线来连接上述三种图框。作图时，需要把相应的命名记入框中；对组成关键字的属性，标记下划线；在描述关系的菱形框的引线上要标记联系的方式（如 1:N 等），有时为了突出实体间的联系，可以不在图中划出属性，属性用另外的表格列出。

建立 ER 图的步骤依次为：确定实体型，确定联系型，用连线组合实体型和联系型，确定实体型和联系型的属性，确定并标记键。需要指出的是，由于人们看问题的角度、强调的侧面各不相同，因此他们绘制的 ER 图也可能各不相同，即 ER 图不是唯一的。图 7-5 给出了学生与课程联系的 ER 图。

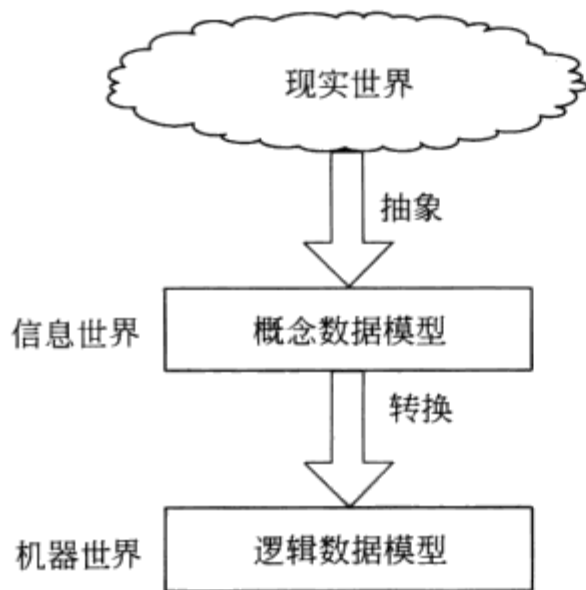


图 7-4 数据模型分类

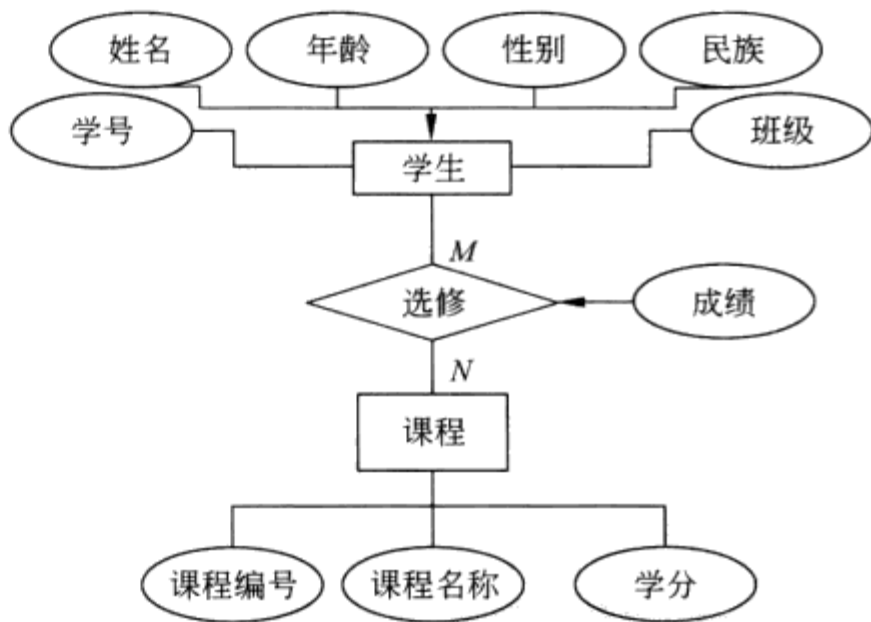


图 7-5 学生课程 ER 图

2. 逻辑数据模型

逻辑数据模型是能够在计算机中真正实现的模型，它直接面向数据库的逻辑结构，有着严格的形式化定义，便于在计算机系统中实现，是对现实世界的第二层抽象。逻辑数据模型由数据结构、数据操作和数据完整性约束三部分组成。其中，数据结构用于解决实体及实体间联系的表达和实现；数据操作用于实现数据库的检索和更新（包括插入、删除和修改）；数据完整性约束用于给出数据及联系所应具有制约和依赖规则，以保证数据的一致性和完整性。

目前比较成熟的逻辑数据模型有层次模型、网状模型、关系模型和面向对象模型。

7.3 关系模型

知识要点分析

关系模型是1970年由IBM公司San Jose研究室研究员E.F.Codd首次提出的,是目前使用最广泛的数据模型。它具有两个特点:第一,数据结构简单,采用二维表格表示,数据视图形象直观,易理解,易操作;第二,以深厚的数学理论基础为背景,包括关系运算理论和关系规范化理论,所以从其问世之时开始长盛不衰。

关系模型由数据结构、数据操作和完整性约束规则三部分组成。

1. 数据结构

关系模型由一组关系组成,每个关系都是一张扁平的二维表,如表7-1学生表就是一个关系,多个关系有机地组合在一起,就构成了关系模型的数据结构。关系模型的数据结构可以从ER图中导出。

表 7-1 学生表

学号	姓名	性别	年龄	民族	班级
200304110	王斌	男	20	汉	03041
200305245	金丽	女	21	蒙	03052
200307108	王涛	男	19	汉	03071
200406201	王峰	男	20	汉	04062
200412103	张和平	男	19	蒙	04121
200503112	赵敏	女	18	汉	05031
200504321	张云鹏	男	18	回	05043
200506101	李娜	女	19	汉	05061
200506102	张利民	男	20	汉	05061
200506220	贾云	男	19	蒙	05062

下面是在关系模型中经常会使用到的一些概念。

1) 元组

二维表中的一行称为关系的一个元组。

2) 属性

二维表中的一列称为关系的一个属性,属性的取值范围称为域,一个元组中的一个属性值称为元组分量。

3) 关系模式

一个关系的名称及其全部属性名的集合简称为该关系的模式,可形式化地表示为:关系名(属性1,属性2,⋯,属性n)。关系模式描述了一个关系的结构,可形象地理解为一个二维表的框架,在这个框架下填入数据,就构成了关系,称为关系模式的一个实例。

4) 关键字

关键字,简称键,是关系中的一个属性或属性组。常用到的键有以下几种。

- (1) 超键：在一个关系中能够唯一地标识元组的属性组合称为该关系的超键。
- (2) 候选键：不含多余属性的超键称为候选键，一个关系中可能有多个候选键。
- (3) 主键：候选键可能有多个，用户选作元组标识的那个候选键称为主键，一个关系中的主键只能有一个。若不加说明，一般情况下，键是指主键。
- (4) 外键：如果关系 R 中的某个属性组合 A 不是 R 的候选键，而是另一个关系 S 的候选键，则称 A 为 R 的外键。通过外键可实现关系间的联系。

例如，有两个关系：学生（学号，姓名，班级）和成绩（学号，课程号，成绩）。在成绩关系中，由于一个学生可以选修多门课程，可能有多个元组的学号属性值相同，因此学号不能成为候选键，但它是另一个关系学生中的候选键，在学生关系中，学号属性可以唯一地确定一个元组。因此学号是成绩关系的外键。利用这个外键，可以建立成绩关系和学生关系的联系。

2. 数据操作

关系模型的操作主要包括检索和更新（包括插入、删除和修改）数据。关系模型提供了完备的 DML，以支持对数据库实施各种数据操作，按其查询的方式可分为关系代数和关系演算两类，利用关系运算关系模式可以推导出多个关系外模式，此外，关系 DML 的存取方式面向集合，非过程化的级别较高。

3. 完整性约束

数据库的完整性包括数据的正确性、有效性和相容性。关系模式是相对稳定的。随着数据库的数据插入、删除和修改，关系模式下的关系却是千变万化的。为了防止错误数据的出现，维护数据库中的数据与现实世界的一致性，关系数据库必须遵循以下三类规则。

1) 实体完整性约束

实体完整性约束是指关系中不允许出现相同的元组，在组成主键的属性上也不允许取空值。

2) 参照完整性约束

参照完整性约束是指不能引用不存在的实体，例如，在学生关系中删除了某个学生元组，那么成绩关系中的相关元组也必须删除。参照完整性是不同关系间建立关联、防止错误引用的保证。

3) 用户定义的完整性约束

用户定义的完整性约束是用户根据实际情况针对某些具体数据制定的约束规则，例如，成绩关系中的成绩属性，其值范围应在 0~100 之间。

7.4 Access 2007 概述

7.4.1 Access 2007 简介

知识要点分析

Microsoft Access 2007 是 Microsoft 公司推出的面向办公自动化、功能强大的桌面数据

库管理系统，是 Office 系列办公软件之一，主要适用于中小型应用系统或作为客户机/服务器系统中的客户端数据库。因其界面友好、易学易用、开发简单、接口灵活，无须深厚的数据库知识就可以灵活地操作数据库，所以很受数据库初学者的欢迎。

1. Access 2007 的主要特点

(1) 完善地管理各种数据库对象，具有强大的数据组织、用户管理、安全检查等功能。

(2) 强大的数据处理功能，在一个工作组级别的网络环境中，使用 Access 开发的多用户数据库管理系统具有传统的 XBase 数据库系统所无法实现的客户服务器 (Client/Server) 结构和相应的数据库安全机制，同时 Access 具备许多先进的大型数据库管理系统所具备的特征，如事务处理/出错回滚能力等。

(3) 可以方便地生成各种数据对象，利用存储的数据建立窗体和报表，可视性好。

(4) 作为 Office 套件的一部分，可以与 Office 集成，实现无缝连接。

(5) 能够利用 Web 检索和发布数据，实现与 Internet 的连接。

2. Access 2007 的新特色

(1) 使用 Office Fluent 用户界面更快地获得更好的结果。Access 2007 通过其 Office Fluent 用户界面、新的导航窗格和选项卡式窗口视图为用户提供全新的体验。即使用户没有数据库经验，也可以开始跟踪信息并创建报表。

(2) 使用预制的解决方案快速入门。为了方便用户，程序中已经建立了一些表单和报表，用户可以轻松地自定义这些表单和报表以满足其业务需求。

(3) 针对同一信息创建具有不同视图的多个报表。在 Access 2007 中创建报表真正能体验到“所见即所得”。用户可以根据实时可视反馈修改报表，并可以针对不同观众保存不同的视图。

(4) 可以迅速创建表，而无须担心数据库的复杂性。借助自动数据类型检测，在 Access 2007 中创建表就像处理 Excel 表格一样容易，甚至可以将整个 Excel 表格粘贴到 Access 2007 中，以便利用数据库的强大功能开始跟踪信息。

(5) 使用全新字段类型，实现更丰富的方案。Access 2007 支持附件和多值字段等新的字段类型，可以将任何文档、图像或电子表格附加到应用程序中的任何记录中。使用多值字段，可以在每一个单元格中选择多个值。

(6) 直接通过源收集和更新信息。通过 Access 2007，用户可以使用 Microsoft Office InfoPath 2007 或 HTML 创建表单来为数据库收集数据，然后用户可通过电子邮件向队友发送此表单，并使用队友的回复填充和更新 Access 表，而无须重新键入任何信息。

(7) 使用 Windows SharePoint Services 和 Access 2007 与工作组中的其他成员共享 Access 信息。借助这两种应用程序的强大功能，工作组成员可以直接通过 Web 界面访问和编辑数据以及查看实时报表。

(8) 使用 Access 2007 的富客户端功能跟踪 Windows SharePoint Services 表。可将 Access 2007 用作富客户端界面，通过 Windows SharePoint Services 列表分析和创建报表。甚至还可以使列表脱机，然后在重新连接到网络时对所有更改进行同步处理，从而让用户可以随时地、轻松地处理数据。

(9) 将数据移动到 Windows SharePoint Services，增强可管理性。将数据移动到 Windows SharePoint Services，使数据更透明。用户可以定期备份服务器上的数据、恢复垃

圾箱中的数据、跟踪修订历史记录以及设置访问权限，从而可以更好地管理信息。

(10) 访问和使用多个源中的信息。通过 Access 2007，可以将其他 Access 数据库、Excel 电子表格、Windows SharePoint Services 网站、ODBC 数据源、Microsoft SQL Server 数据库和其他数据源中的表链接到用户数据库，然后可以使用这些链接的表轻松地创建报表，从而根据更全面的信息来做出决策。

7.4.2 Access 2007 的数据库对象

知识要点分析

Access 2007 是一种关系型数据库，它的数据库由一系列表组成，表又由一系列行和列组成，每一行是一个记录，每一列是一个字段，每个字段有一个在表中唯一的字段名，表和表之间可以通过关联或连接建立联系，以便进行信息的综合查询。Access 2007 数据库以文件形式保存，文件的扩展名是 .accdb，早期 Access 格式创建的数据库的文件扩展名为 .mdb。

Access 2007 数据库由 6 种对象组成，它们是表、查询、窗体、报表、宏和模块，分别用于实现对数据的保存、检索、显示和更新。

1. 表 (Table)

表即关系，是基于关系数据模型的数据集合，是数据库的基本对象，是创建其他 5 种对象的基础。表由记录组成，记录由字段组成，表用来存储数据库的数据，故又称数据表。

2. 查询 (Query)

查询是数据库中应用最多的对象，最常用的功能是从表中检索特定的数据，查询结果不仅有多种去向而且还可以用作窗体或报表的记录源，某些查询还是“可更新的”，即利用查询可以更新数据源。查询还可以对表中的数据进行汇总等操作。查询分为选择查询和动作查询两种。使用选择查询可从指定的表中获取满足给定条件的记录，使用动作查询可以生成一个新表或者对指定表的记录进行更新、添加或删除操作。

3. 窗体 (Form)

窗体对象用于建立基于 Access 数据库的应用程序界面，为用户提供浏览、输入及更改数据的窗口，窗体也称表单。

4. 报表 (Report)

报表的功能是将数据库中的数据分类汇总，然后打印出来，以便分析。表对象允许用户不用编程仅通过可视化的直观操作就可以设计报表打印格式。报表可用来汇总和显示表中的数据。报表可在任何时候运行，而且将始终反映数据库中的当前数据。通常将报表的格式设置为适合打印的格式，但是报表也可以在屏幕进行查看、导出到其他程序或者以电子邮件的形式发送。

5. 宏 (Macro)

宏是用来自动执行任务的一个操作或一组操作，在 Access 2007 中，宏可以包含在宏对象中，也可以嵌入在窗体、报表或控件的事件属性中，嵌入的宏成为所嵌入对象或控件

的一部分。

宏对象是一个或多个宏操作的集合，其中的每一个宏操作执行特定的单一功能。用户可以将这些宏操作组织起来形成宏对象，以执行特定的任务。宏对象在导航窗格中的“宏”下可见，嵌入的宏则不可见。

6. 模块 (Module)

与宏一样，模块是可用于向数据库中添加功能的对象，但模块定义的操作比宏更精细和复杂，用户可以根据自己的需要用宏语言 (Visual Basic for Application, VBA) 编写模块。

模块是声明、语句和过程的集合，可分为类模块和标准模块，类模块可附加到窗体或报表，而且通常包含一些特定于所附加到的窗体或报表的过程。标准模块包括与任何其他对象无关的常规过程。在导航窗格的“模块”下列出了标准模块，但没有列出类模块。

Access 提供的上述 6 种对象分工极为明确，从功能和彼此间的关系角度考虑，这 6 种对象可以分为三个层次：第一层次是表和查询，它们是数据库的基本对象，用于在数据库中存储数据和查询数据。第二层次是窗体和报表，它们是直接面向用户的对象，用于数据的输入输出和应用系统的驱动控制。第三层次是宏和模块，它们是代码类型的对象，用于通过组织宏操作或编写程序来完成复杂的数据库管理工作并使得数据库管理工作自动化。

7.4.3 Access 的集成开发环境

☞ 知识要点分析

单击 Windows 的“开始”按钮或使用桌面快捷方式启动 Office Access 2007 时，将首先出现“开始使用 Microsoft Office Access”窗口，如图 7-6 所示。

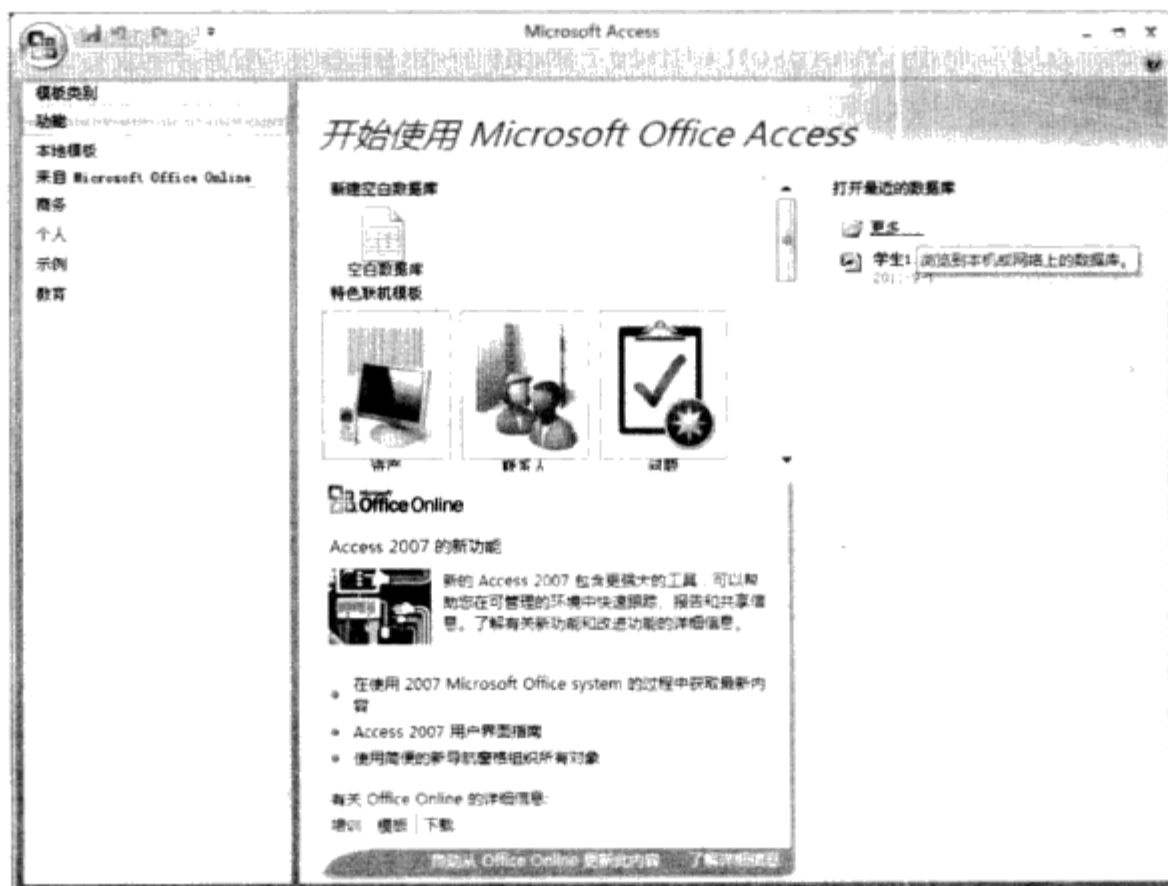


图 7-6 “开始使用 Microsoft Office Access”窗口

此时，用户可以选择创建一个新的空白数据库或通过模板创建数据库，也可以选择打开最近使用过的数据库或者已有的数据库，将会出现图 7-7 所示的程序窗口。

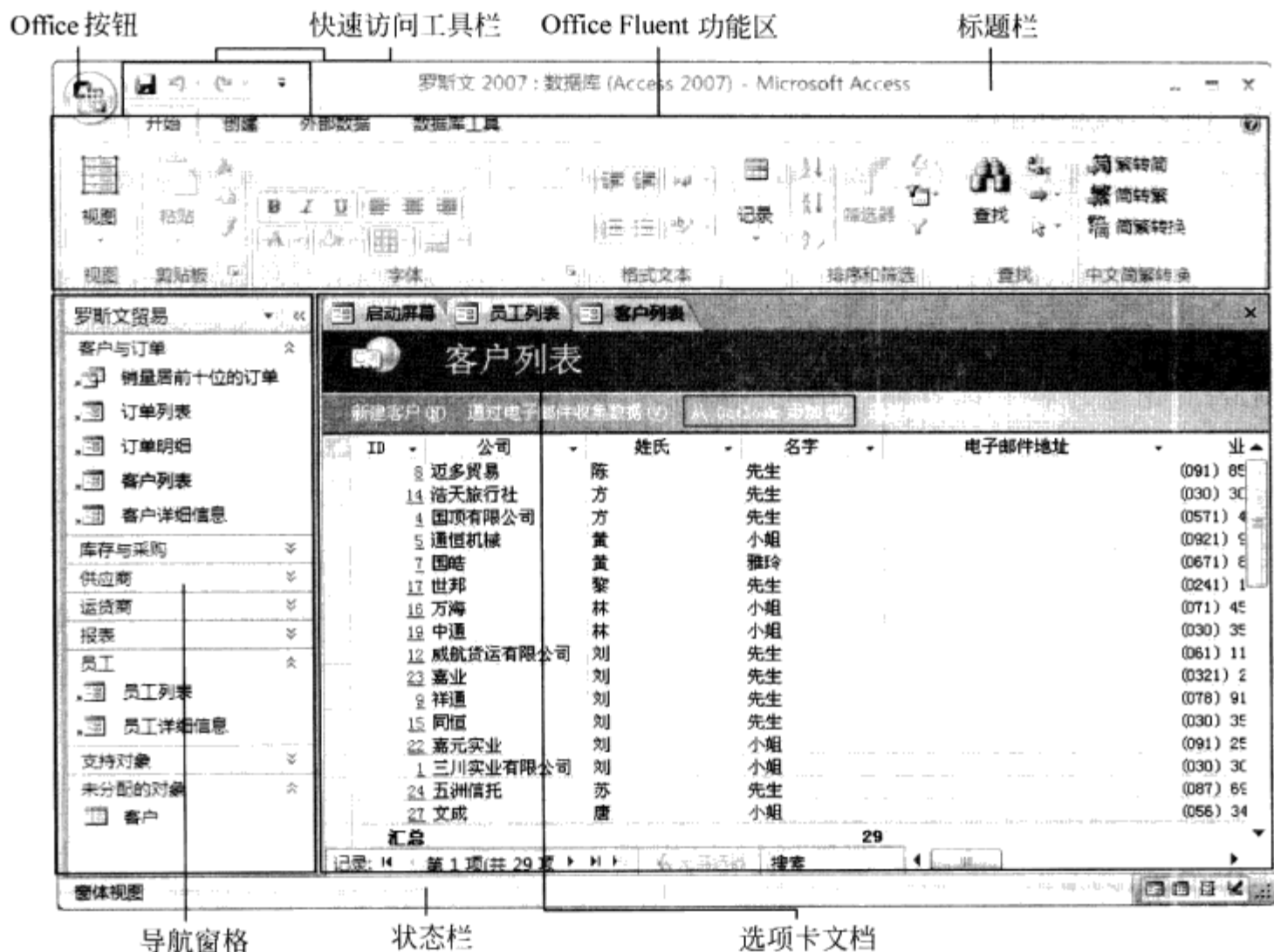


图 7-7 Office Access 2007 的用户界面

Office Access 2007 中的 Microsoft Office Fluent 界面由多个元素构成，这些元素定义了用户与产品的交互方式。这些用户界面元素主要包括以下几种。

1. 快速访问工具栏

显示在功能区上方的常用工具栏，单击一次即时访问最需要的命令，如“保存”和“撤销”。

2. 功能区

功能区是 Microsoft Office Fluent 用户界面的一部分，由“开始”、“创建”、“外部数据”和“数据库工具”等命令选项卡组成，每个选项卡都包含多组相关命令。特别是有一种上下文命令选项卡，它是根据上下文显示的一种命令选项卡。所谓上下文，就是用户正在着手处理的对象或正在执行的任务。

3. 导航窗格

导航窗格列出了对当前打开数据库中所有对象的轻松访问方式。导航窗格按对象类型、创建日期、修改日期、相关表（基于对象相关性）等组织对象。为节省显示空间，导航窗格还可以折叠。

4. 选项卡式对象

表、查询、窗体、报表和宏在 Access 窗口中都显示为选项卡式对象。通过单击对象选

项卡，可以在各种对象间轻松切换。

5. 状态栏

显示状态信息并可在视图间切换或调整显示比例。

6. 浮动工具栏

透明地显示在所选文本的上方，可以轻松地设置应用格式。

7.5 数据库设计基础

建立数据库的目的是为了实现对数据进行高质量、高效率 and 低成本的管理，除此之外，一个设计良好的数据库还应具有生命周期长、易维护和易扩充等特征。设计合理的数据库可以让用户访问到最新的、准确的信息。由于正确的设计对于实现使用数据库的目标非常重要，因此有必要投入时间学习良好设计的相关原则。这样，最终就更有可能获得一个既能满足用户的需要又能轻松适应变化的数据库。

☞ 知识要点分析

在设计数据库时，应遵循两个基本原则：

(1) 要尽量避免重复信息，因为重复信息不仅会浪费存储空间，而且会增加出错和数据不一致的可能性；

(2) 必须确保数据库中数据的正确性和完整性，因为一旦数据库中包含了不正确的信息，则从数据库中提取信息的报表也会随之发生错误，最终将导致基于这些报表所做决策的失误。

一个好的数据库设计应该是这样的：

- 通过将信息划分到基于主题的表中，以减少冗余数据；
- 设置联接表时必需的信息，以方便地实现查询；
- 支持和确保信息的准确性和完整性；
- 可满足数据处理和报表需求。

为实现上述目的，数据库的设计过程分如下步骤进行：

- (1) 确定数据库的用途。
- (2) 查找和组织所需的信息。
- (3) 将信息划分到表中。
- (4) 将信息项转换为列。
- (5) 指定表的主键。
- (6) 建立表间关系。
- (7) 优化设计。
- (8) 应用规范化规则。

注意，这里谈到的是数据库的设计，第8章将会具体看到设计工作在计算机中的实现。

7.5.1 确定数据库的用途

知识要点分析

确定数据库的用途包括分析和确定数据库的用途、预期使用方式及使用者，可以简单地分为一段或多段描述性内容，包括各种用户将在何时及以何种方式使用数据库，目的是为了获得一个好的任务说明，作为整个设计过程的参考。任务说明可以帮助用户在进行决策时将重点集中在目标上。

例如，高校学生成绩管理数据库的用途，简单地讲，是为了保存学生的成绩，以方便各类人员的查询，但细化后可以发现该数据库有三类用户，每类用户都有他们对数据库不同的需求：教师在每门课结课之后需要录入所教授课程的成绩，并对成绩进行分析，生成学生成绩上报表；学生可随时查看自己各门课程的成绩，并对所修学分进行统计；学籍管理人员负责整个数据库的维护，在学生入校时录入学生信息、班级信息，在每个学期的开始录入课程信息、教师信息，并能随时对班级的开课情况、学生的修课情况、教师的代课情况进行修改和查询以及汇总等。

7.5.2 查找和组织所需的信息

知识要点分析

从现有的信息着手，收集一切与数据库有关的纸质表单，并列出现上面所显示的每一种信息，如果没有任何现成的纸质表单，则需要根据用户的描述临时绘制一个。同时，考虑将来可能创建的报表和邮件以及希望数据库进行回答的问题，也有助于确定将来数据库中可能需要的各个项。

在进行信息提取时，要做到详尽，尽量列出所有可能想到的项，并且将每条信息分为最小的有用单元，如果还有其他人使用该数据库，也应向他们征求意见，不要一开始就试图追求完美，在后续步骤中会对收集到的信息列表进行优化。

例如，对高校学生成绩管理的过程中，可以收集到学生登记表，上面记录着每个学生的信息，包括学生的学号、姓氏、名字、民族、籍贯、年龄、出生日期、入学日期、所在院系、所在班级、家庭住址等，这些项都有可能成为表中的一个字段。

7.5.3 将信息划分到表中

知识要点分析

要将信息划分到表中，就必须明确主要实体或主题。

首次检查待管理的项目的初步列表时，可能非常想将所有的项目都放入一个表中，例如，将高校学生成绩管理收集上来的全部信息组织成如表 7-2 所示的情况，但这样的做法不一定好。

表 7-2 学生全部信息表

学号	姓名	性别	班级	班级名称	课程号	课程名	学分	成绩
200307108	王涛	男	03071	03 电商 1 班	423403	计算机基础	3	91
200307108	王涛	男	03071	03 电商 1 班	566305	企业策划	2	87
200412103	张和平	男	04121	04 国贸 1 班	566305	企业策划	2	79.5
200506101	李娜	女	05061	05 法学 1 班	233351	大学语文	2	67
200506102	张利民	男	05061	05 法学 1 班	566305	企业策划	2	82
200307108	王涛	男	03071	03 电商 1 班	423403	计算机基础	3	91
200307108	王涛	男	03071	03 电商 1 班	566305	企业策划	2	87
200412103	张和平	男	04121	04 国贸 1 班	566305	企业策划	2	79.5
200506101	李娜	女	05061	05 法学 1 班	233351	大学语文	2	67
200506102	张利民	男	05061	05 法学 1 班	566305	企业策划	2	82

在上述的表中，每行同时记录了有关学生、班级、课程及其成绩各个方面的信息。由于一个学生可能同时选修多门课程，因此在表中该学生的信息就不得不多次重复，这样就形成了数据冗余，不但浪费了磁盘空间，而且也为后续的操作带来不便，增加了数据维护的开销。例如，当需要修改有关学生的信息时，不得不在多条记录上进行重复的修改，而且一旦忘记了修改某条记录，就会产生信息不一致的错误。又如，某门课只有一个学生选修，由于记录中既包含了有关学生的事实，也包含了有关课程的事实，因此删除该学生的信息势必会导致唯一的课程信息的丢失。

为了解决上述矛盾和问题，将表 7-2 进行分拆不失为一种好的解决方案，即将表 7-2 拆分成两个表，学生表存储学生信息，课程表存储课程信息，然后将学生表连接到课程表上。由于在学生表中学生信息仅被记录一次，所以修改学生信息也只需要做一次，删除了某个学生信息，也仅限于学生表中记录的删除，而不会影响到课程信息。

在选择了用表来表示的主题后，该表中的列就应仅存储有关该主题的事实。例如，课程表应仅存储有关课程的事实，由于学生的姓名是有关学生的事实，而不是有关课程的事实，因此姓名应仅属于学生表。

同理，班级信息和成绩信息也应从表 7-2 中分离出来，成为单独的实体。这样，高校学生成绩管理数据库中就存在 4 个主题，对应的需要建立 4 张表来保存对应的信息。

7.5.4 将信息项转换为列

知识要点分析

要确定表中的列，就必须明确在表中需要记录主题的哪些信息。例如，对于学生表，需要记录每位学生的学号、姓名、年龄、性别、民族、家庭地址、联系方式等，它们都有可能成为将来学生表中的列。

在为每个表确定了初始的列后，可以对列进行进一步优化。优化有以下两个基本原则。

1. 不要包含已计算的数据

尽量不要在表中设定用来存储计算结果的列。在希望查看相应结果时，可以让 Access

去执行计算。例如，在学生表中如果设定了出生日期列，就没有必要再设定年龄列，年龄完全可以用出生日期计算出来。

2. 将信息按照其最小的逻辑单元进行存储

在确定列时，应该充分考虑用户的检索需要。例如，在学生表中，可将学生姓名分为姓氏和名字两个列，以便于用户单独地对姓氏和名字进行检索的需要。类似地，家庭地址也可以设定为地址、城市、省/直辖市/自治区、邮政编码和国家/地区 5 个独立的组成部分。是否分拆取决于用户的操作需要，可在充分咨询用户后进行决策。

7.5.5 指定主键

知识要点分析

主键是一个字段或字段组合，利用主键可以唯一地标识一条记录。如在学生表中，学号可以唯一地确定一条记录，学号就可以用作主键。需要注意的是，一个表中只能有一个主键。利用主键字段可以将多个表中的数据关联起来，从而可以实现综合信息的查询。

由于要在表中唯一地标识一条记录，所以主键字段中不能有重复的值。例如，一般不使用姓名来作为主键，因为同名同姓的人实在是太多了。另外，主键字段也不允许为空值。如果某条记录的某个字段值可以在一个特定的时间段内未分配或未知，则该字段不能作为主键的组成部分。

主键字段应该由那些其值始终不会更改的字段来充当。因为在一个多表的数据库中，一个表的主键可能会被其他表所引用。如果主键一旦发生更改，则必须还要将此更改应用到其他任何引用该主键的位置上。使用不会更改的主键可降低出现主键与其他引用该键的表不同步的概率。

如果尚未确定可能成为好的主键的一个或一组列，则不妨考虑使用具有“自动编号”数据类型的字段。使用“自动编号”数据类型时，Access 将自动给该字段分配一个值，这样的标识符不包含描述它所表示的行的事实信息，非常适合作为主键使用，因为它们不会更改。而那些包含有关某一行的事实数据的主键则很有可能会改变。

在某些情况下，可能需要使用字段组合来作为表的主键。例如，在成绩表中，可以使用学号和课程号这个字段组合作为主键，来唯一地标识其中的记录。由字段组合来充当的主键称为复合键。

7.5.6 创建表关系

知识要点分析

在关系数据库中，同属于一个数据库的表之间不是彼此孤立的，应通过一种有意义的方式再次将各个表中的信息组织到一起，形成综合查询。这就是创建表关系。

前面已经提到表和表之间的关系有一对一关系、一对多关系和多对多关系三种。由于一对一关系可以看做是特殊的一对多关系，所以在此仅以高校学生成绩管理数据库为例，介绍后两种关系的建立。

1. 创建一对多关系

在高校学生成绩管理数据库中，一个班级可能有多名学生，而一名学生只能从属于一个班级，因此班级表和学生表之间应该是一种一对多的关系，班级表为这种关系的“一”方，学生表为这种关系的“多”方。

为了在数据库设计中表示一对多关系，应将关系“一”方的主键作为附加的一列或多列，添加到关系“多”方的表中。例如在本例中，可将“班级”表中的“班级编号”列添加到“学生”表中，这样就可以使用“学生”表中的班级号来查找每个学生所在的班级名称。“班级编号”是班级表的主键，同时它也是“学生”表中的外键。通过建立主键和外键的配对提供了联接相关表的基础。

2. 创建多对多关系

在学校的现实生活中，一个学生可以选修多门课程（意味着有多门课程的成绩），一门课程也可以为多个学生所选修。因此在高校学生成绩管理数据库中，对于学生表的每条记录，都可能与课程表的多条记录相对应；反之，对于课程表的每条记录，都可能与学生表的多条记录相对应，因而学生表与课程表之间是一种多对多的关系。

如果仍然采用建立一对多关系的方法，将课程表中的“课程编号”列添加到学生表中，则需要为每个学生添加多条记录，以反映其所修的每门课程的信息（包括成绩信息）。这些记录中有关学生的信息内容完全相同，从而产生可能导致数据不准确的低效设计。

解决这个问题的方案是，创建第三个表（通常称为联接表），利用该表将多对多关系分解为两个一对多关系。将这两个表的主键都插入到第三个表中，用第三个表去记录关系的每个匹配项或实例。这就是成绩表。

成绩表中包含了学号（学生表的主键）和课程号（课程表的主键），其主键就由这两个字段构成的字段组合来充当。在学生成绩数据库中，学生表和课程表并不直接彼此关联，它们是通过成绩表间接关联的。学生表和课程表之间的多对多关系是通过使用两个一对多关系在数据库中得到表示的：学生表和成绩表具有一对多关系，每个学生可以具有多个行项目，而每个行项目仅与一个学生相关；课程表和成绩表具有一对多关系。每个课程有多个与之关联的行项目，而每个行项目仅引用一个课程。通过成绩表，可以确定特定学生中的所有课程成绩，也可以确定特定课程的所有学生成绩。

7.5.7 优化设计

知识要点分析

在确定表、字段和关系后，数据库设计就可以进入测试阶段。在创建表并使用示例数据填充表以后，可以通过创建查询、添加新记录等操作，来发现数据库设计中潜在的问题。以下是要检查的事项。

(1) 是否有遗漏的信息项？是则判断该信息是否属于现有的某一个表，如果是有关其他主题的信息，则可能需要新建一个表，并添加相关的信息。

(2) 是否存在可通过现有字段计算得到的不必要的列（例如通过出生日期计算出的年龄）？是则进行计算通常会更好，并能够避免创建新列。

(3) 是否在某个表中重复输入相同的信息? 是则可能需要将这个表拆分为两个具有一对多关系的表。

(4) 是否存在这样的表: 具有很多字段, 但记录数量有限, 且各个记录中有很多空字段? 是则重新考虑对该表的设计, 使其包含更少的字段和更多的记录。

(5) 每个信息项是否已拆分为最小的有用单元? 如果需要对某个信息项进行报告、排序、搜索或计算, 则请将该项放入其单独的列中。

(6) 每一列是否包含有关所属表的主题的事实? 如果某一列不满足此条件, 则该列属于其他表。

(7) 表之间的所有关系是否已经都由公共字段或第三个表加以表示? 一对一和一对多关系要求使用公共列, 而多对多关系要求使用第三个表来表示。

7.5.8 应用规范化规则

知识要点分析

在数据库设计中, 应用数据规范化规则的目的是为了确认表结构的设计是否正确。将数据规范化规则应用到数据库设计的过程称为数据库规范化。通过数据库规范化, 可以确保将信息项划分到恰当的表中。需要在每个步骤中持续应用这些规则, 以确保设计达到范式的要求。广泛接受的范式有 5 个: 第一范式到第五范式。本书只介绍前面的三个范式。

1. 第一范式

第一范式规定, 表中每个行和列的交叉处只存在一个值, 而决不是值的列表。例如, 不能在一个名为“价格”的字段中放置多个“价格”。如果将行与列的每个交叉点看作一个单元格, 则每个单元格中只能包含一个值。

2. 第二范式

第二范式要求每个非键列完全依赖于整个主键, 而不仅仅依赖于主键的一部分。当主键由多个列组成时, 适用此规则。例如, 高校学生成绩管理数据库中的成绩表, 由“学号”和“课程号”构成其主键, 如果再在此表中加入“学生姓名”就违反了第二范式, 因为“学生姓名”依赖于“学号”, 但并不依赖于“课程号”, 因此并不依赖于整个主键。必须将“学生姓名”从表中删除, 它属于不同的表, 即属于“学生”表。

3. 第三范式

第三范式要求不仅每个非键列依赖于整个主键, 而且非键列要互相独立, 即每个非键列必须且只能依赖于主键。例如, 高校学生成绩管理数据库的学生表, 其主键为“学号”, 如果在学生表中既有“出生日期”又有“年龄”的话, 那么就违反了第三范式, 因为非键列“年龄”依赖于另一个非键列“出生日期”, 如果更改了“出生日期”字段中的值, 那么“年龄”字段中的值将相应地发生改变。

7.6 本章小结

本章围绕 4 方面的内容展开。

(1) 介绍了数据、数据库管理系统和数据库系统的一些相关概念。

(2) 介绍了数据模型中常用的概念以及数据模型类型的划分, 对其中的关系数据模型进行了重点的阐述。

(3) 介绍了 Office Access 2007 的特点、组成对象和集成用户界面。

(4) 介绍数据库设计的基本原则和设计步骤。

通过本章的学习旨在为 Office Access 2007 的学习打下牢固的理论基础, 希望读者在后面的学习中不断地体会和印证这些理论。

习 题 7

一、判断题

1. 数据管理技术发展过程经过人工管理、文件系统和数据库系统三个阶段。()
2. 数据库系统的核心是数据库。()
3. 关系模型由数据结构、数据操作和完整性约束规则三部分组成。()
4. 在一个关系中, 如果某个属性或属性组能唯一地标识一个元组, 就称其为关键字。()
5. 在关系数据库模型中, 二维表的行称为字段。()
6. 要求主键属性不能为空的完整性是参照完整性。()
7. 实体联系模型的 ER 图都是唯一的。()
8. 用实体-联系方法表示的概念模型称为 ER 模型。()
9. 在一个关系中, 候选键可能有多个, 而主键只能有一个。()
10. 关系数据库系统中的关系是一个文件。()
11. Access 数据库是一种网状数据库。()
12. Access 2007 数据库文件的扩展名为.mdb。()

二、选择题

1. 数据独立性是数据库技术的重要特点之一, 所谓数据独立性是指 ()。
A. 数据与程序独立存放
B. 不同的数据被存放在不同的文件中
C. 不同的数据只能被对应的应用程序所使用
D. 以上三种说法都不正确
2. 下列描述中正确的是 ()。
A. 程序就是软件
B. 软件开发不受计算机系统的限制
C. 软件既是逻辑实体, 又是物理实体
D. 软件是程序、数据与相关文档的集合
3. 下面 () 是非结构化数据。
A. Word 文档 B. 学籍表 C. 通讯录 D. 以上都是
4. 表示二维表中的行的数据库术语是 ()。
A. 数据表 B. 记录 C. 域 D. 属性

5. 表示二维表中的列的数据库术语是 ()。
- A. 数据表 B. 记录 C. 域 D. 字段
6. 数据库管理系统主要的任务是 ()。
- A. 生成报表 B. 信息检索 C. 更新数据 D. 以上都包括
7. 目前成为数据库主流的数据模型是 ()。
- A. 层次模型 B. 网状模型 C. 关系模型 D. 面向对象模型
8. 下列说法正确的是 ()。
- A. DB 包含 DBS 和 DBMS B. DBMS 包含 DB 和 DBS
C. DBS 包含 DB 和 DBMS D. 没有任何关系
9. 用二维表表示实体及实体之间联系的数据模型是 ()。
- A. 实体联系模型 B. 网状模型 C. 层次模型 D. 关系模型
10. 在 E-R 图中, 用 () 表示实体型, 用 () 表示联系型, 用 () 表示属性, 用 () 来连接上述三种图框。
- A. 矩形框 B. 椭圆形框 C. 菱形框 D. 直线
11. 在学生关系中删除了某个学生元组, 那么成绩关系中的相关元组也必须删除, 这属于 () 规则。
- A. 实体完整性 B. 参照完整性
C. 用户定义的完整性 D. 以上都不对
12. 在成绩关系中的成绩属性, 限定其值应在 0~100 之间, 这属于 () 规则。
- A. 实体完整性 B. 参照完整性
C. 用户定义的完整性 D. 以上都不对
13. 在一个设计合理的关系型数据库中, 以下哪种关系不应存在。()
- A. 一对一 B. 一对多 C. 多对多 D. 以上都不对
14. () 不是 Access 2007 的数据库对象。
- A. 页 B. 报表 C. 窗体 D. 表
15. 在 Access 2007 数据库中, 下列对象应用最多的是 ()。
- A. 查询 B. 报表 C. 窗体 D. 宏

第 8 章

创建数据库及数据表

在第 7 章中，我们学习了数据库理论知识，也学习了数据库的简要设计方法，在本章中要把前面一章的理论知识用到实际数据库的创建过程中。

使用数据库技术来解决大量数据的管理任务必须先建立数据库。数据库在计算机里表现为由数据库管理系统建立、使用、控制的一个或若干个文件。Access 2007 的数据库就是计算机中扩展名为 .accdb 的文件。创建 Access 数据库就是利用 Access 2007 软件建立 .accdb 的文件。建立了数据库后，再应用 Access 建立数据库中的表。

8.1 创建 Access 数据库

8.1.1 利用模板创建 Access 数据库

知识要点分析

Access 提供了一些数据库模板方便初学者创建数据库。利用这些模板初学者可以迅速有效地建立数据库，但是模板是有限的，不能也不可能解决所有数据库管理问题。微软公司网站上也提供了一些数据库模板，供初学者使用（<http://office.microsoft.com/zh-cn/templates/FX100595492052.aspx?pid=CL100632982052>）。

应用实例

【实例 8-1】 利用“联系人”模板，创建一个名为“联系人 123”的 Access 数据库。创建数据库步骤如下。

(1) 启动 Access 2007，出现“新建文件”环境，选择左侧“模板类别”中的“本地模板”，如图 8-1 所示。

(2) 选择“本地模板”中的“联系人”模板，在右侧文件名文本框中输入新文件名“联系人 123”，然后选择存储路径，再单击“创建”按钮，即可生成新数据库“联系人 123”，如图 8-2 所示。

由以上操作可知，数据库模板数量不能也不可能满足现实生活数据管理需要，因此大量数据库不能依靠数据库模板完成。

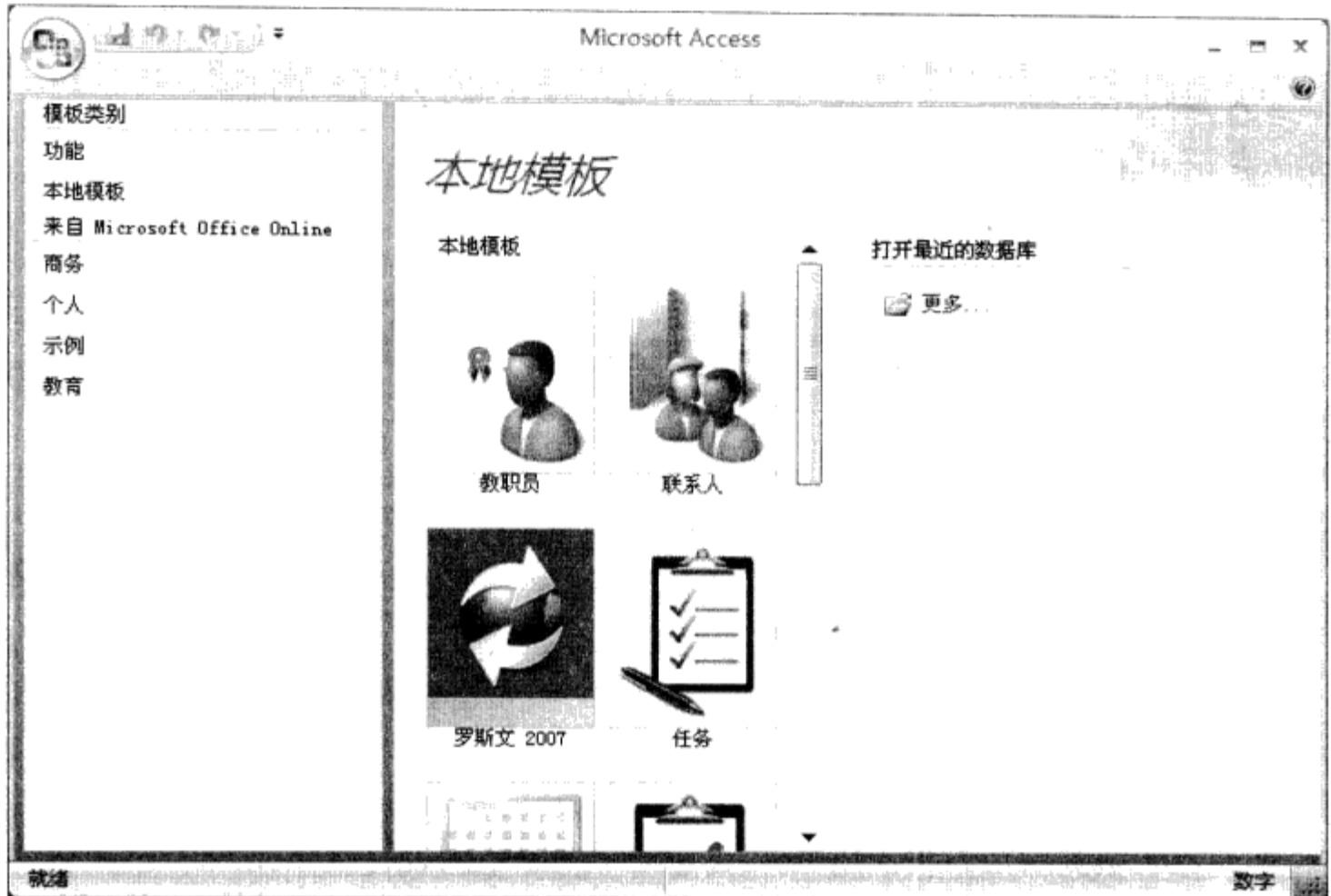


图 8-1 新建文件



图 8-2 “联系人 123” 数据库窗口

8.1.2 创建空 Access 数据库

☎ 知识要点分析

创建空 Access 数据库是为了针对现实生活中数据管理任务进一步创建各种数据库对

象，包括表、查询、窗体、报表、模块等。这样数据库就成为了一个大容器。

应用实例

【实例 8-2】 启动 Access 2007，创建一个名为“高校学生成绩管理系统”的空的 Access 数据库。

创建空数据库步骤如下。

(1) 启动 Access，选择“新建空白数据库”中的“空白数据库”模板，如图 8-3 所示。

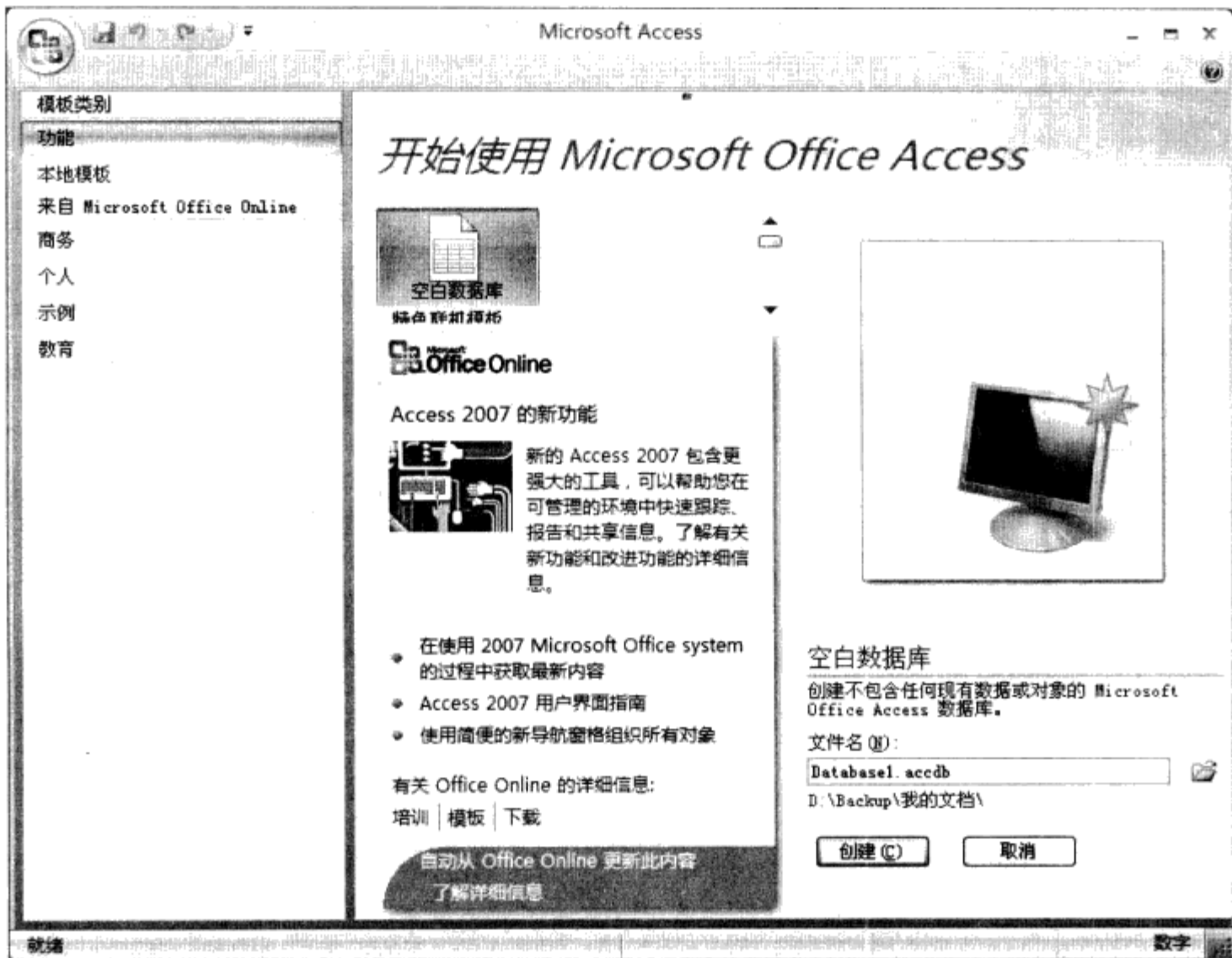


图 8-3 新建空白数据库

(2) 在右侧文件名文本框中输入数据库文件名“高校学生成绩管理系统”，然后选择存储路径，再单击“创建”按钮，即可生成空数据库文件“高校学生成绩管理系统.accdb”，如图 8-4 所示。

这一步中应选择数据库文件的存储位置，以及定义数据库文件名。在位置选择上应考虑到数据库可能的大小、病毒影响等因素，尽量放在安全、可靠的位置。名称定义应考虑数据库内容等因素，尽量不使用默认名称如 Database1 等。

(3) 可以看到上述数据库中所有数据对象为空，在操作系统的文件夹下可以看到图 8-5 高校学生成绩管理系统数据库文件。

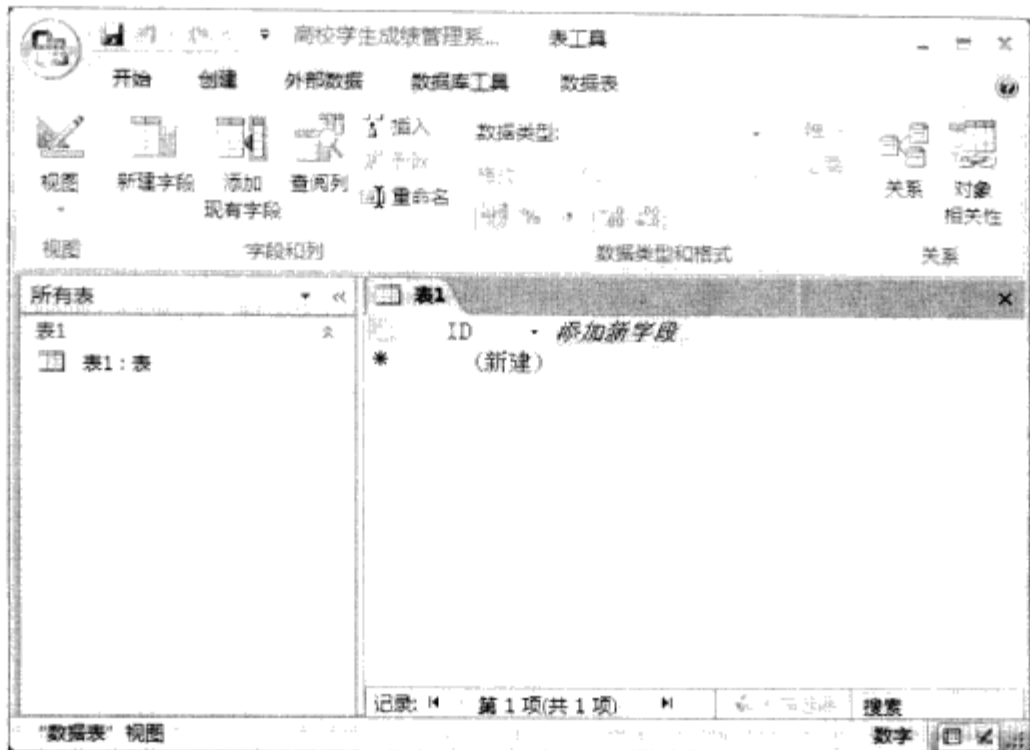


图 8-4 高校学生成绩管理系统窗口

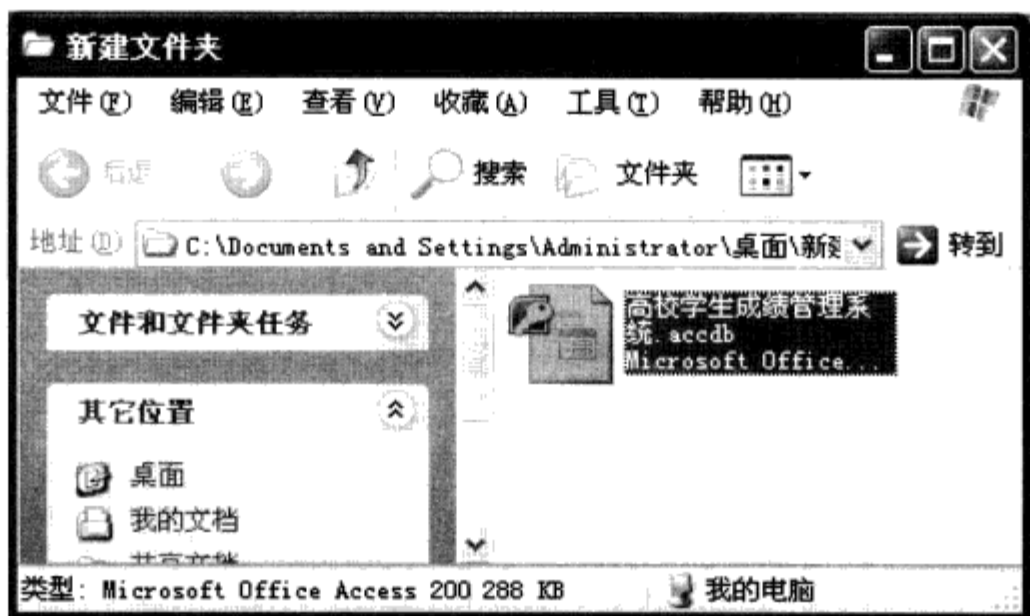


图 8-5 高校学生成绩管理系统数据库文件

8.2 创建数据表

表是存储数据的数据库对象，也就是在第 7 章中讲到的关系。表是数据库的基础。

☞ 知识要点分析

表的结构是扁平的二维形式，每列称为一个字段，每行称为一个记录。定义表就要定义表中的每个字段的字段名称，类型和字段属性。在第 7 章中，根据我们的研究，高校学生成绩管理系统数据库有三张表，即“学生”表、“课程”表和“成绩”表。表 8-1~表 8-3 分别是“学生”表、“课程”表、“成绩”表的结构。

表 8-1 “学生”表

字段名	类型	长度	字段名	类型	长度
学号	文本	9	年龄	数字	整型
姓名	文本	20	民族	文本	2
性别	文本	1	班级	文本	6

表 8-2 “课程”表

字段名	类型	长度	字段名	类型	长度
课程号	文本	6	课程类别	文本	2
课程名	文本	40	学分	数字	整型

表 8-3 “成绩”表

字段名	类型	长度	字段名	类型	长度
学号	文本	9	成绩	数字	整型
课程号	文本	6			

这一节中，将把这三个表建立在数据库中。

8.2.1 利用表模板创建 Access 数据库表

Access 提供了很多的表模板帮助初学者建立表。启动 Access，打开数据库“高校学生成绩管理系统”，选择“创建”选项卡中“表”组中“表模板”中的“联系人”模板，即可生成表，如图 8-6 所示。

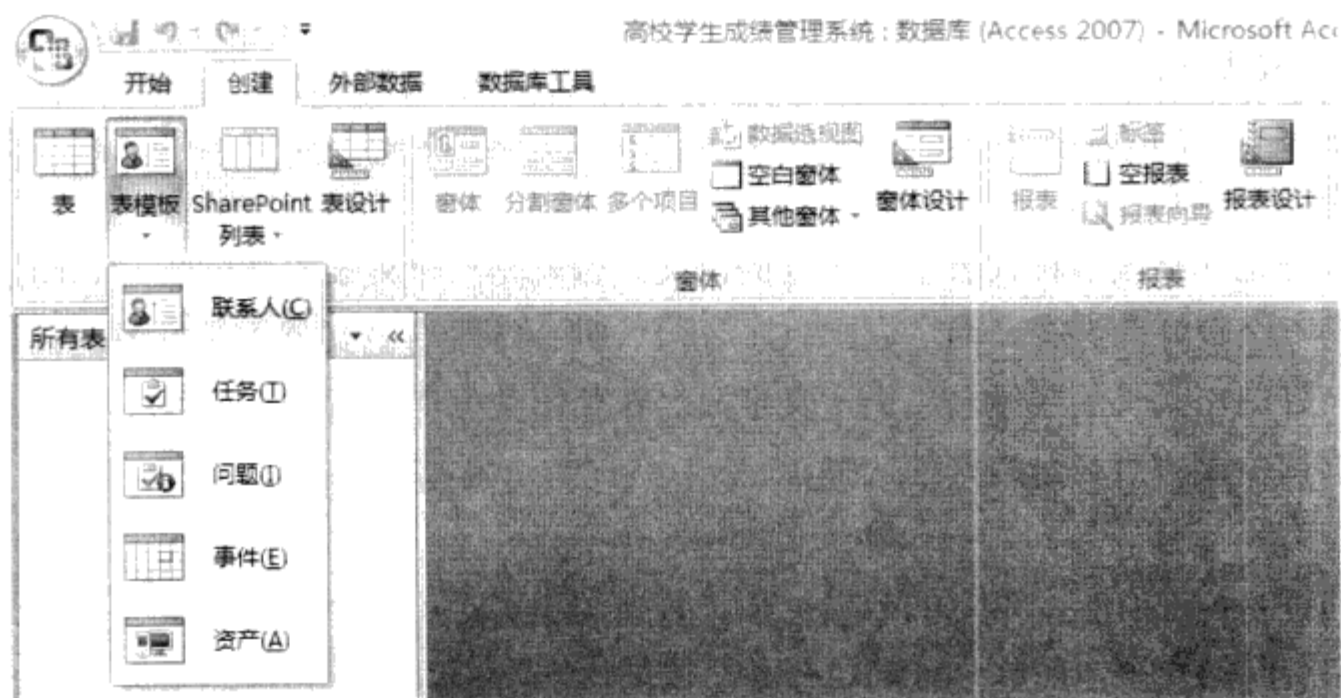


图 8-6 表模板建表窗口

选择表模板，可以快速建立表。但是和数据库模板一样，表模板同样不能满足现实生活中的千差万别的数据管理需求，大量的表需要设计人员自行建立。

8.2.2 利用表设计器创建 Access 数据表

利用表设计器建表，只需选择“创建”选项卡中“表”组中“表设计”，即可打开表设计器，如图 8-7 所示。

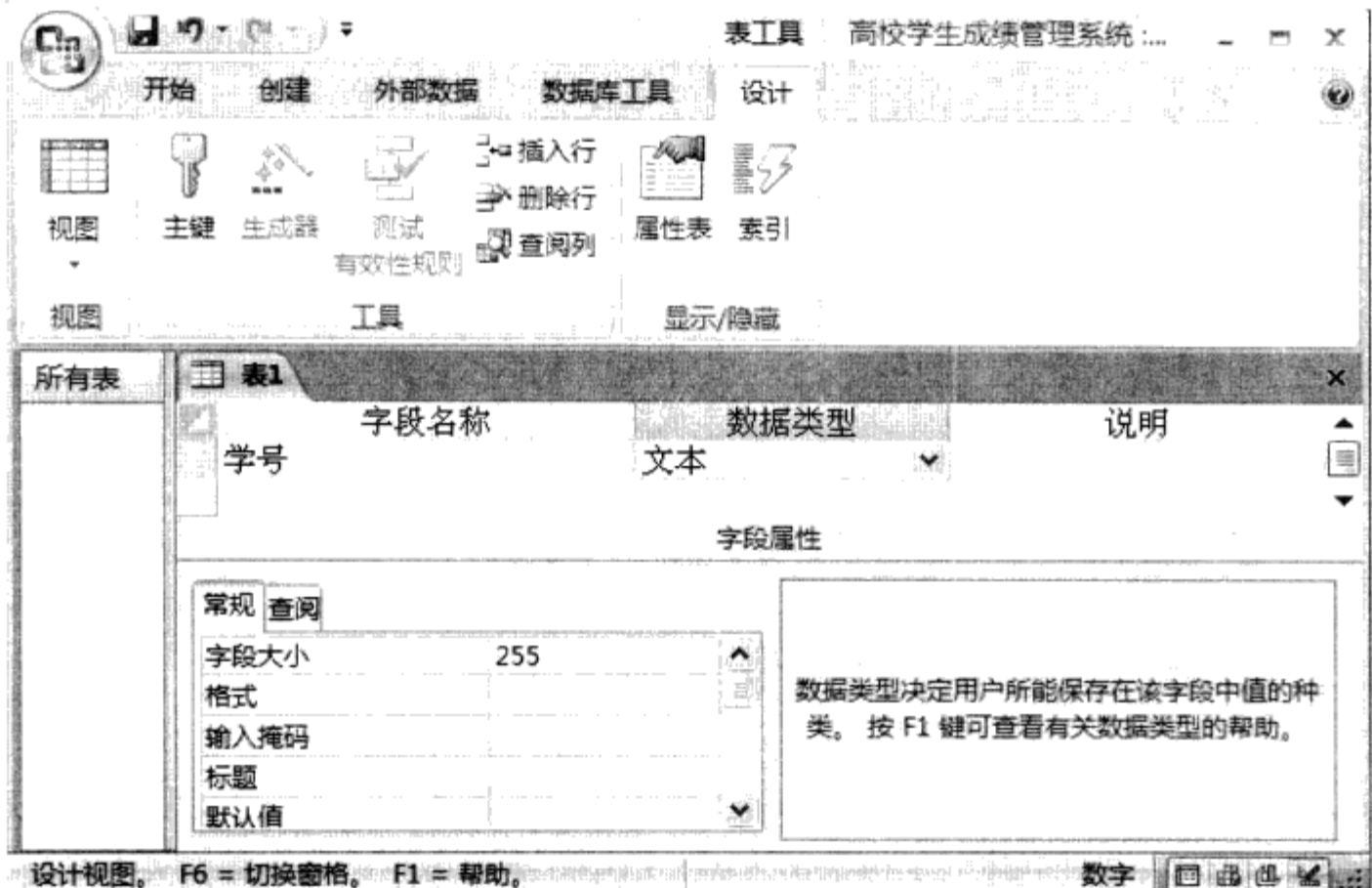


图 8-7 表设计器窗口

1. 字段名定义

表是由若干列组成，每列称为一个字段。每个字段都必须有字段名称。定义字段名称时应考虑如下原则。

(1) 好记。字段名称代表了一个列的语义，也就是一个列要表达的含义。简短有意义的词语是一个较好的选择，而 A、B、C、列 1、列 2 等没有意义就不是好的选择。

(2) 好用。太长的字段名、无意义字段名在使用数据库编程时容易发生错误，因此选择英语单词，汉语拼音是较好的选择。

2. 字段类型选择

表的每个列都有一个确定的数据类型，数据类型确定了一个字段能够存储什么样的数据，存储空间的大小，能够进行的运算和字段的用法。Access 数据库主要提供了 9 种数据类型。表 8-4 是 Access 数据库提供的主要的数据类型。

在表 8-1~表 8-3 中分别罗列了学生表、课程表、成绩表的字段名和数据类型。选择数据类型要根据现实情况满足字段计算的需要。如：“学号”字段选择了“文本”类型而不是“数字”类型，原因是“学号”数据是不需要进行计算的。

表 8-4 数据类型

数据类型	作用特点	字段举例
文本	用于文本或文本与数字的组合。最多存储 255 个字符	姓名, 地址, 电话号码, 零件编号或邮编
备注	用于长文本和数字, 最多存储 65 536 个字符	注释、说明
数字	用于要进行计算的数据。根据“字段大小”属性定义具体的数字类型	年龄, 学分, 成绩
日期/时间	用于日期和时间	出生日期, 入学日期, 登录时间
货币	用于存储货币值, 并且计算期间禁止四舍五入。存储 8 个字节	学费
自动编号	在表添加记录时自动插入的唯一顺序号(每次递增 1)或随机编号	Id
是/否	用于只可能是两个值中的一个。不允许 Null 值	
OLE 对象	用于使用 OLE 协议在其他程序中创建的 OLE 对象, 最多存储 1 GB	照片
超链接	用于超链接	个人主页

3. 字段属性

设置字段属性是为了对字段进行进一步的改进以求在数据的完整性、安全性以及提高数据库使用性能上有重要的意义。常用的字段属性有字段大小、格式、输入掩码、有效性规则、有效性文本等。但不是所有字段上都有这些字段属性, 字段属性与字段的类型相关。下面介绍字段属性的用法和应用范围。

1) 字段大小

可以应用在文本、数字、自动编号类型的字段上设置这些数据类型字段上存储数据的最大大小。

对于文本类型字段, 可以输入介于 1 到 255 的值, 也就是该字段最大存储的字符数。例如姓名字段, 数据类型为文本, 字段大小设置为 20, 则表示该字段最多可以存储 20 个汉字。

对于数字类型字段, 可以选择下列选项之一。

- 字节——用于范围在 0 到 255 之间的整数。存储为 1 个字节。
- 整型——用于范围在-32 768 到+32 767 之间的整数。存储为 2 个字节。
- 长整型——用于范围在-2 147 483 648 到+2 147 483 647 之间的整数。存储为 4 个字节。
- 单精度型——用于范围在 -3.4×10^{38} 到 $+3.4 \times 10^{38}$ 之间且最多具有 7 个有效位数的浮点数值。存储为 4 个字节。
- 双精度型——用于范围在 -1.797×10^{308} 到 $+1.797 \times 10^{308}$ 之间且最多具有 15 个有效位数的浮点数值。存储为 8 个字节。
- 同步复制 ID——用于存储同步复制所需的全局唯一标识符。存储为 16 个字节。
- 小数——可以指定小数位数的精确数值, 可以存储-1028 到+1028 之间的值。存储为 12 个字节。字段大小选择为小数后还需设定精度和数值范围两个属性。精度表

示该字段小数点前后数字个数,不包括小数点。数值范围表示小数点后的数字位数。例如分数字段,如果要求该字段能够存储最大 100 分,最小 0 分,同时小数点后保留两位的话,可以设置该字段为小数,同时设定该字段精度为 5,数值范围为 2。

2) 格式

格式属性设置字段值显示打印的默认显示方式。可以应用在各种数据类型的字段上,但在数字、日期时间、是否、货币类型的字段上应用最普遍。需要指出的是,格式属性不涉及数据的存储,只是规定字段的显示打印方式。

在数字类型字段上,可以选择下列选项之一。

- 常规数字——按输入时的样子显示数字。
- 货币——使用千位数分隔符显示数字。并且应用“控制面板”中“区域和语言选项”中的设置设定货币符号、负数金额、小数点及小数位数的区域要求。例如把 1234.567 显示为 ¥1,234.57。
- 固定——至少显示一位数,应用“控制面板”中“区域和语言选项”中的设置设定货币符号、负数金额、小数点及小数位数的区域要求。
- 标准——使用千位数分隔符显示数字,此格式不显示货币符号。例如,1234.567 显示为 1,234.57。
- 百分比——将数值乘以 100 并在得到的数字末尾追加一个百分号进行显示。例如,0.123 显示为 12.30%。
- 科学记数——以标准科学记数显示值。例如,1234.567 显示为 1.23E03。

在日期类型字段上可以选择下列选项之一。

- 常规日期——使用“短日期”和“长时间”设置的组合显示值。
- 长日期——使用“控制面板”中“区域和语言选项”中的“长日期”设置显示值。
- 短日期——使用“控制面板”中“区域和语言选项”中的“短日期”设置显示值。
- 长时间——使用“控制面板”中“区域和语言选项”中的“时间”设置显示值。
- 短时间——使用格式 HH:MM 显示值,其中 HH 为小时,MM 为分钟。

在是否类型字段上可以选择下列选项之一。

- 真/假——将值显示为 True 或 False。
- 是/否——将值显示为 Yes 或 No。
- 开/关——将值显示为 On 或 Off。

对于货币类型字段、自动编号字段的格式属性与数字型字段格式属性相同。

3) 输入掩码

输入掩码是为了控制数据库使用人员在字段中输入哪些内容。例如“学号”字段是文本类型,但是在本章中该字段只能输入 9 位的数字。输入掩码和格式可以同时使用,但掩码用来控制输入,而格式用来控制显示方式。

输入掩码由三段控制字符组成,三段字符之间用分号分隔。第一段定义掩码字符串,由占位符和字面字符组成。第二段定义是否将掩码字符和所有显示数据一起存储到数据库中。如果希望同时存储掩码和数据,这一段写 0。如果只希望存储用户输入的数据,这一段写 1。第三部分定义用来指示需输入数据位置的占位符。默认情况下,Access 使用下划线“_”作为占位符。如果希望使用其他字符,在掩码的第三部分输入该字符,比方“*”。

因此掩码中最重要的是编写第一段的掩码控制字符，表 8-5 是 Access 提供的掩码控制字符，表 8-6 是输入掩码的实例。

表 8-5 Access 提供的掩码控制字符

掩码控制字符	说明
0	数字 (0 到 9, 必选项; 不允许使用加号 [+] 和减号[-])。
9	数字或空格 (非必选项; 不允许使用加号和减号)
#	数字或空格 (非必选项; 空白将转换为空格, 允许使用加号和减号)
L	字母 (A 到 Z, 必选项)
?	字母 (A 到 Z, 可选项)
A	字母或数字 (必选项)
a	字母或数字 (可选项)
&	任一字符或空格 (必选项)
C	任一字符或空格 (可选项)
.,:;- /	十进制占位符和千位、日期和时间分隔符 (实际使用的字符取决于 Microsoft Windows 控制面板中指定的区域设置)
<	使其后所有的字符转换为小写
>	使其后所有的字符转换为大写
!	使输入掩码从右到左显示, 而不是从左到右显示。输入掩码中的字符始终都是从左到右填入。可以在输入掩码中的任何地方包括感叹号
\	使其后的字符显示为原义字符。可用于将该表中的任何字符显示为原义字符 (例如, \A 显示为 A)
密码	将“输入掩码”属性设置为“密码”, 以创建密码项文本框。文本框中输入的任何字符都按字面字符保存, 但显示为星号 (*)

表 8-6 输入掩码的实例

字段	输入掩码	可以输入的数据	掩码含义
学号	000000000;0;*	200601001	在本例中, 您必须输入 9 位数字字符, 不可省略
固定电话号码	(9999) 00000009;0;*	(0471) 5555-1234 () 555-0199	在本例中, 区号部分使用控制符 9, 因此区号是可选项的。此外, 感叹号 (!)会导致从左到右填充掩码
普通数字	#999;0;*	-20 2000	任何正数或负数, 不超过四个字符, 不带千位分隔符或小数位
书籍编号	ISBN 0-&&&&&&&&&-0;0;*	ISBN 1-55615-507-7	书号, 其中包含文本、第一位和最后一位 (这两位都是强制的)、第一位和最后一位之间字母和字符的任何组合
产品型号	>LL00000-0000;0;*	DB51392-0493	强制字母和字符的组合, 均采用大写形式。例如, 使用这种类型的输入掩码可以帮助用户正确输入部件号或其他形式的清单

4) 有效性规则、有效性文本

有效性规则是一个条件表达式，控制用户必须在该字段上输入满足条件允许的数据。使用输入掩码是强制用户以特定方式输入值。例如，日期型字段的输入掩码强制用户必须以掩码规定的方式输入日期，如 2008-12-01。而有效性规则可以要求用户输入某个范围的日期。

有效型规则、掩码都是对数据完整性的控制。表 8-7 是有效性规则的实例。

表 8-7 有效性规则的实例

有效性规则	含义
<>0	输入非零值
>=0	值不得小于零
0 or >100	值等于 0 或者大于 100
<#01/01/2007#	输入 2007 年之前的日期
>=#01/01/2007# AND <#01/01/2008#	必须输入 2007 年的日期
<Date()	小于当前日期
LIKE "[A-Z]*@[A-Z].com" OR "[A-Z]*@[A-Z].net" OR "[A-Z]*@[A-Z].org"	输入有效的 .com、.net 或 .org 电子邮件地址
[要求日期]<=[订购日期]+30	输入在订单日期之后的 30 天内的要求日期
[结束日期]>=[开始日期]	输入不早于开始日期的结束日期

有效性规则必须是一个条件表达式，关于条件表达式将在查询一章中学习。有效性文本是在用户输入数据不满足有效性规则需提醒用户注意的文本。

5) 其他属性

默认值属性可以用在各种类型的字段上用于添加新记录时自动向字段分配值。

必填属性可以用在除自动编号类型之外的所有数据类型字段，选择“是”则要求该字段在每条记录中都必须有值。

允许空字符串属性可以用在文本、备注、超链接类型的字段上，设定字段是否允许输入零长度字符串("")。

Unicode 压缩属性可以用在文本、备注、超链接类型的字段上，当存储了不到 4 096 个字符时压缩字段中的数据（对文本字段而言始终为 True）。

输入法模式属性可以用在文本、备注、日期/时间、超链接，控制当用户把屏幕焦点移到该字段上时是否打开输入法。例如姓名字段应该打开输入法，而年龄字段应该关闭输入法。

8.2.3 主键和索引

1. 主键

主键是一个或一组字段，它的值能够唯一标识一个记录。原则上讲，每个表中都应有且仅有一个主键，为的是记录不重复。比方说学生表，在几个字段中能够标识一个记录字

段是“学号”字段，因为每个学生的学号是一定不相同的，只要学号不同，那么一定是不同的学生。而在成绩表中，每个学生都会有多个成绩记录，因此“学号”就不能做主键。同样，一门课程也有多个学生成绩，因此“课程号”字段也不能做主键。而“学号”+“课程号”的组合在不考虑补考的情况下，可以做主键，因为每个学生每门课程的成绩是唯一的。

应用实例

【实例 8-3】 打开“高校学生成绩管理系统”数据库，创建“学生”表、“课程”表和“成绩”表，并给每个表定义合适的主键。

创建“学生”表步骤如下。

(1) 打开“高校学生成绩管理系统”数据库，选择“创建”选项卡中“表”组中“表设计”，即可打开表设计器，在表设计器中依次输入“学生”表中的6个字段，并设置字段类型与长度，保存表名称为“学生”，如图 8-8 所示。

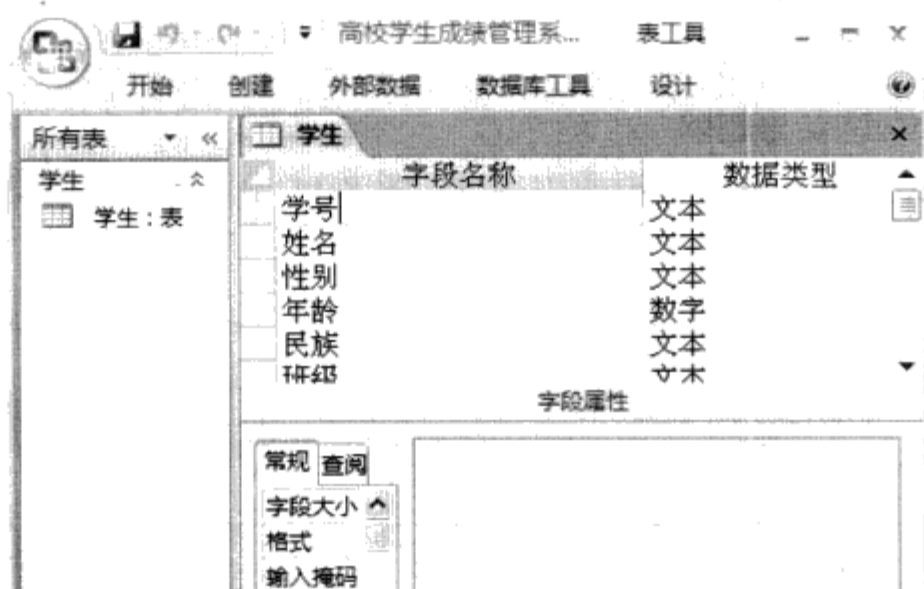


图 8-8 学生表

(2) 选中学号字段，单击“表工具”中“设计”选项卡中“主键”按钮，在“学号”字段前就会出现“钥匙”标志，说明此时学号被设定为主键。如图 8-9 所示。

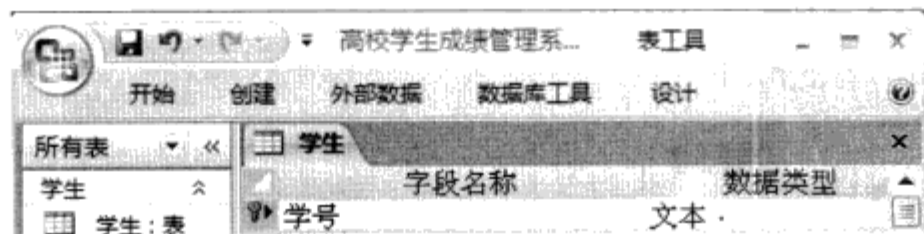


图 8-9 学生表定义主键窗口

成绩表如图 8-10 中所示，“学号”+“课程号”联合主键而不是有两个主键。

知识要点分析

2. 索引

Access 使用索引来提高查找和记录排序的速度。索引根据选择创建索引的字段来确定

存储记录的位置。Access 通过索引获得记录存储位置后，即可通过直接移到正确的位置来检索数据。这样一来，使用索引查找数据可以比扫描所有记录查找数据快很多。但不是所有字段都要建立索引，索引在提高查询速度的同时降低了表插入、更新、删除记录的速度。一般地讲，经常用于查询的字段、进行排序的字段，以及在多个表查询用于联接的字段要建立索引。例如，学生表中“学号”、“姓名”是最经常用于查询的字段，要建立索引。不过需要提醒的是，“学号”字段已经被设定为主键，主键由 Access 自动创建索引，无须人工建立。索引分为单字段索引和多字段索引。

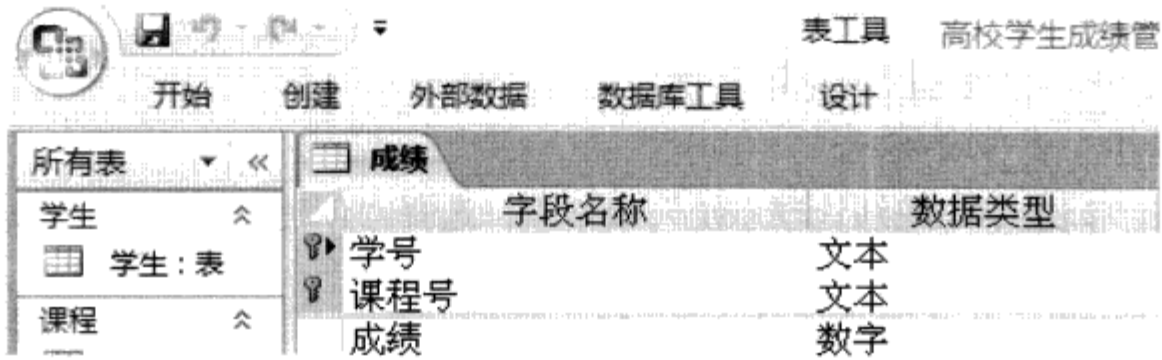


图 8-10 成绩表主键

1) 单字段索引的建立与删除

单字段索引可以直接在表设计视图中完成。图 8-11 中是建立学生表“姓名”字段索引。表 8-8 是索引选项。

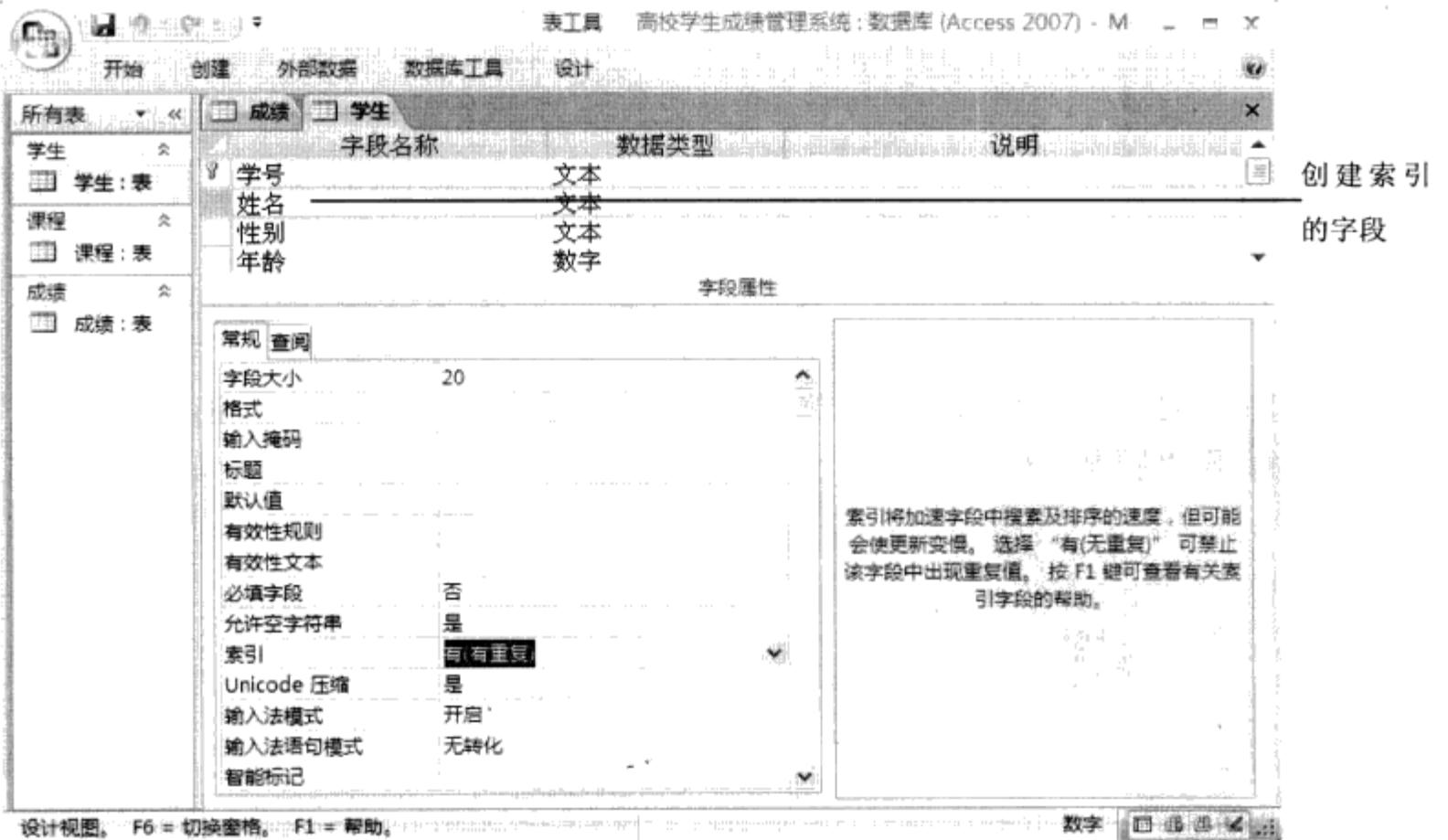


图 8-11 姓名字段索引

如果创建唯一索引，Access 将不允许在该字段中输入重复的值。Access 自动为主键创建的索引就是唯一索引。

表 8-8 索引选项

“索引”属性的设置	含义
无	不在此字段上创建索引（或删除现有索引）
有(有重复)	在此字段上创建索引
有(无重复)	在此字段上创建唯一索引

2) 多字段索引的建立与删除

建立多字段索引需要使用索引窗口，在表设计视图下，单击“索引”快捷按钮或选择“视图—索引”项进入索引窗口，如图 8-12 所示的索引窗口。

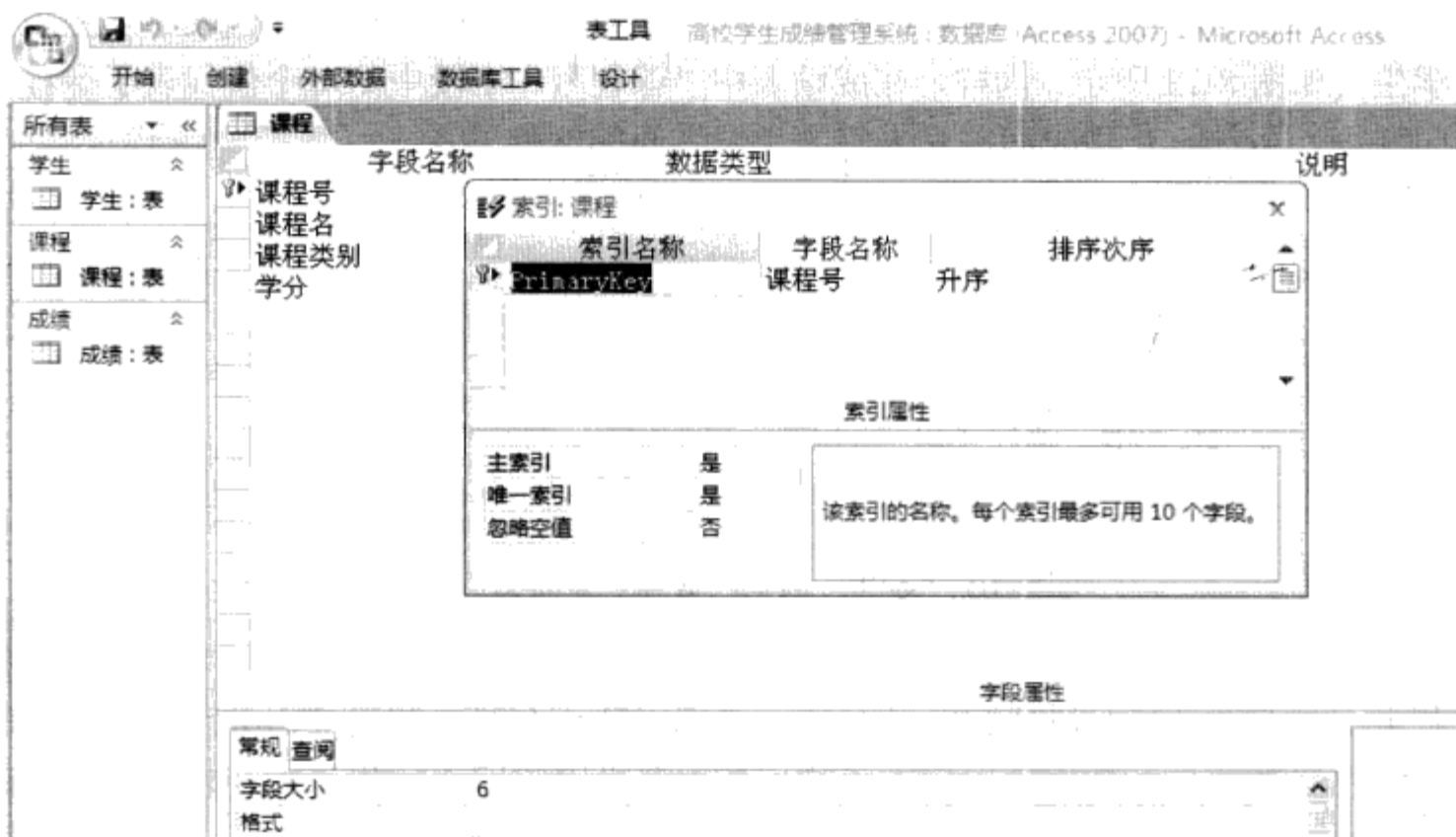


图 8-12 索引视图窗口

在“索引名称”列中，在第一个空白行内键入索引的名称。可以按照某一个索引字段的名称来命名索引，也可以使用其他名称。

在“字段名称”列中，单击箭头，然后单击想用于索引的第一个字段。

在下一行中，将“索引名称”列留空，然后，在“字段名称”列中单击索引的第二个字段。重复此步，直至选择了要包含在索引中的所有字段为止。

在“索引”窗口中的“索引属性”下，为“索引名称”列中包含索引名称的行设置索引属性。表 8-9 是索引属性列表。

表 8-9 索引属性

标签	值
主索引	如果为“是”，则索引为主键
唯一索引	如果为“是”，则索引中的每个值必须唯一
忽略空值	如果为“是”，则在索引字段中具有空值的记录被排除在索引之外

8.3 建立表间关系

8.3.1 关系简介

☞ 知识要点分析

Access 利用表格存储数据，在学生成绩管理数据库中，建立了学生、课程、成绩三个表，但这三个表不应该是独立的。在成绩表中“学号”字段中的每个值代表了学生表中的一个学生，如图 8-13 所示。同样，在成绩表中“课程号”字段中的每个值代表了课程表中的一门课程。Access 使用表间关系来定义了表之间的这种关系。

学号	姓名	性别	年龄	民族	班级	学号	课程号	成绩
200304210	王斌	男	20	汉	03工商1班	200304210	233351	87.5
200305145	金丽	女	21	蒙	03经济2班	200304210	301432	76.7
200307108	王涛	男	19	汉	03电商1班	200307108	423403	91
200406101	王峰	男	20	汉	04市销2班	200307108	566305	87
200412103	张和平	男	19	蒙	04国贸1班	200412103	566307	79.3
200503112	赵敏	女	18	汉	05经济1班	200505220	302445	88
200504121	张云鹏	男	18	回	05工商3班	200505220	566305	64.5
200505220	贾云	男	19	蒙	05法学2班	200505220	753204	59
200506101	李娜	女	19	汉	05法学1班	200506101	233351	67
200506102	张利民	男	20	汉	05法学1班	200506102	423403	82

图 8-13 “学生”表与“成绩”表间关系示意

定义表之间的联系后，表间的数据就有了一定的关系。学生表中的一个记录在成绩表中有若干记录通过学号字段值相等的对应。要使用这两个表中的记录，必须创建联接学生表和成绩表的查询。而查询的工作方式是将学生表主键字段中的值与成绩表的学号字段的值进行匹配。可以得到图 8-14 查询结果。

学号	姓名	性别	年龄	民族	班级	学号	课程号	成绩
200304210	王斌	男	20	汉	03工商1班	200304210	233351	87.5
200305145	金丽	女	21	蒙	03经济2班	200304210	301432	76.7
200307108	王涛	男	19	汉	03电商1班	200307108	423403	91
200406101	王峰	男	20	汉	04市销2班	200307108	566305	87
200412103	张和平	男	19	蒙	04国贸1班	200412103	566307	79.3
200503112	赵敏	女	18	汉	05经济1班	200505220	302445	88
200504121	张云鹏	男	18	回	05工商3班	200505220	566305	64.5
200505220	贾云	男	19	蒙	05法学2班	200505220	753204	59
200506101	李娜	女	19	汉	05法学1班	200506101	233351	67
200506102	张利民	男	20	汉	05法学1班	200506102	423403	82

学生.学号	姓名	年龄	成绩.学号	课程号	成绩
200505220	贾云	19	200505220	753204	59
200505220	贾云	19	200505220	566305	64.5
200505220	贾云	19	200505220	302445	88

图 8-14 基于关系的查询结果

8.3.2 创建关系

☞ 知识要点分析

关系类型有三种：一对一关系、一对多关系、多对多关系。

1. 一对一关系

在一对一关系中，第一个表中的每条记录在第二个表中只有一个匹配记录，而第二个表中的每条记录在第一个表中至多只有一个匹配记录。这种关系并不常见，因为多数以此方式相关的信息都存储在一个表中。

2. 一对多关系

在学生成绩数据库中，学生表的一个记录在成绩表中对应若干个记录，而成绩表中的一个记录在学生表中只能对应一个记录。这样的关系是一对多关系。“一”端就是主键端，“多”的一端为外键端。

3. 多对多关系

“学生”表和“课程”表之间的关系。每个学生选修多门课程。另一方面，一门课程可能被多个学生选修。因此，对于“学生”表中的每条记录，都可能与“课程”表中的多条记录对应。此外，对于“课程”表中的每条记录，都可以与“学生”表中的多条记录对应。这种关系称为多对多关系。

Access 不能直接表示多对多关系，必须创建第三个表，该表通常称为联接表，在学生成绩数据库中就是成绩表，它将学生与课程的多对多关系划分为两个一对多关系。

参照完整性选项是确定在表间关系中是否实施参照完整性。

参照完整性是指外键的值必须参照主键表中的已有的值。学生表和成绩表的关系实施参照完整性后，那么成绩表的“学号”字段的值必须是学生表中“学号”字段中的一个值。

在对成绩表增加记录时，“学号”字段的值必须来自学生表中“学号”字段的值。在学生表中删除一个记录或修改记录的“学号”字段的值，需要考虑成绩表中是否已有记录参照了要删除学生记录的学号值。如果有记录参照了要删除或修改记录的“学号”值，那么 Access 将按照“实施参照完整性”选项的子选项“级联删除记录”，“级联更新记录”来操作。如果选择了“级联删除记录”，那么删除了一个学生记录，将删除成绩表中参照该记录学号的所有成绩记录。如果选择了“级联更新记录”，那么修改了一个学生记录的“学号”值，将修改成绩表中参照该记录学号的所有成绩记录的“学号”。

应用实例

【实例 8-4】 打开“高校学生成绩管理系统”数据库，创建表之间关系。

设计步骤如下。

(1) 在数据库窗口选择“数据库工具”选项卡中“显示/隐藏”组中的“关系”按钮，打开如图 8-15 所示的关系窗口。

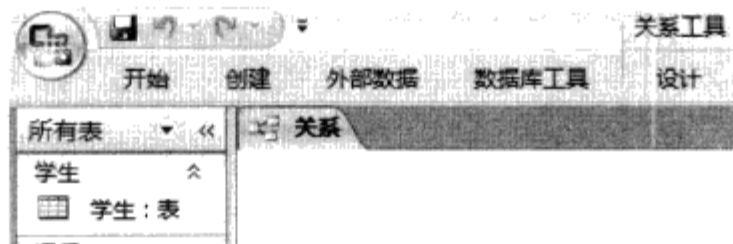


图 8-15 关系窗口

(2) 选择显示要创建关系的主键表和外键表。

外键指在本表中不是主键，在另外表中是主键的字段。例如成绩表的“学号”字段是

外键，因为“学号”字段在学生表是主键。同样成绩表的“课程号”字段也是外键。

(3) 鼠标拖拽主键字段到外键字段上，显示关系选项对话框。创建关系操作如图 8-16 所示。

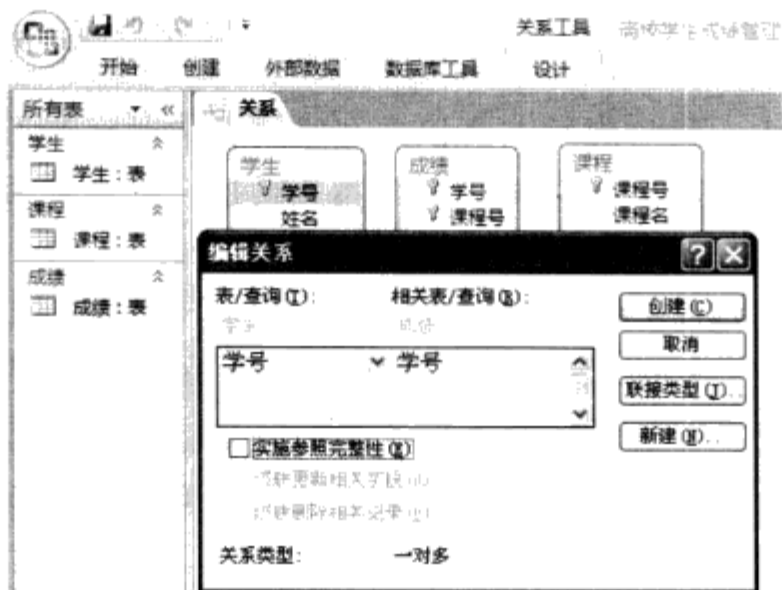


图 8-16 创建关系

选定选项后，关系创建完成。图 8-17 所示为关系图。

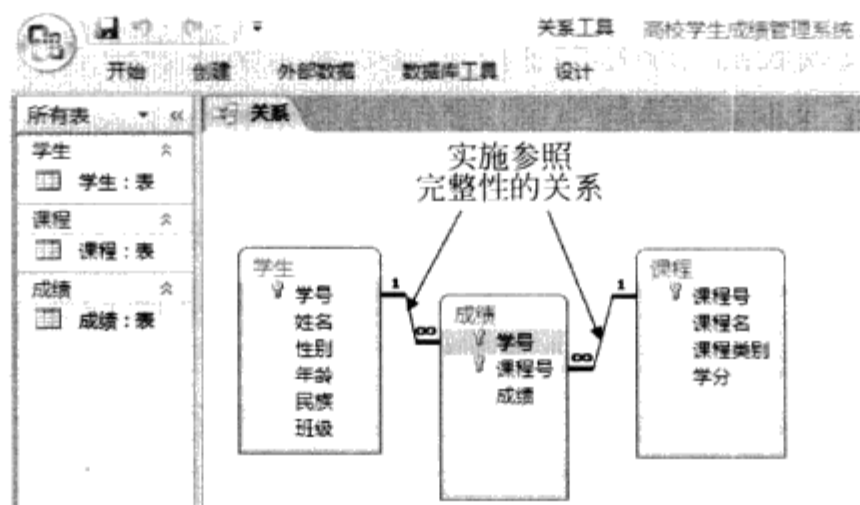


图 8-17 关系图

8.4 数据表的使用

表建立完成后，可对数据进行基本操作。这些操作包括对记录的更新操作、一般查找和筛选排序等。

8.4.1 增加、修改或删除记录

☎ 知识要点分析

数据表最基本的功能就是数据的更新操作，数据表视图如图 8-18 所示。

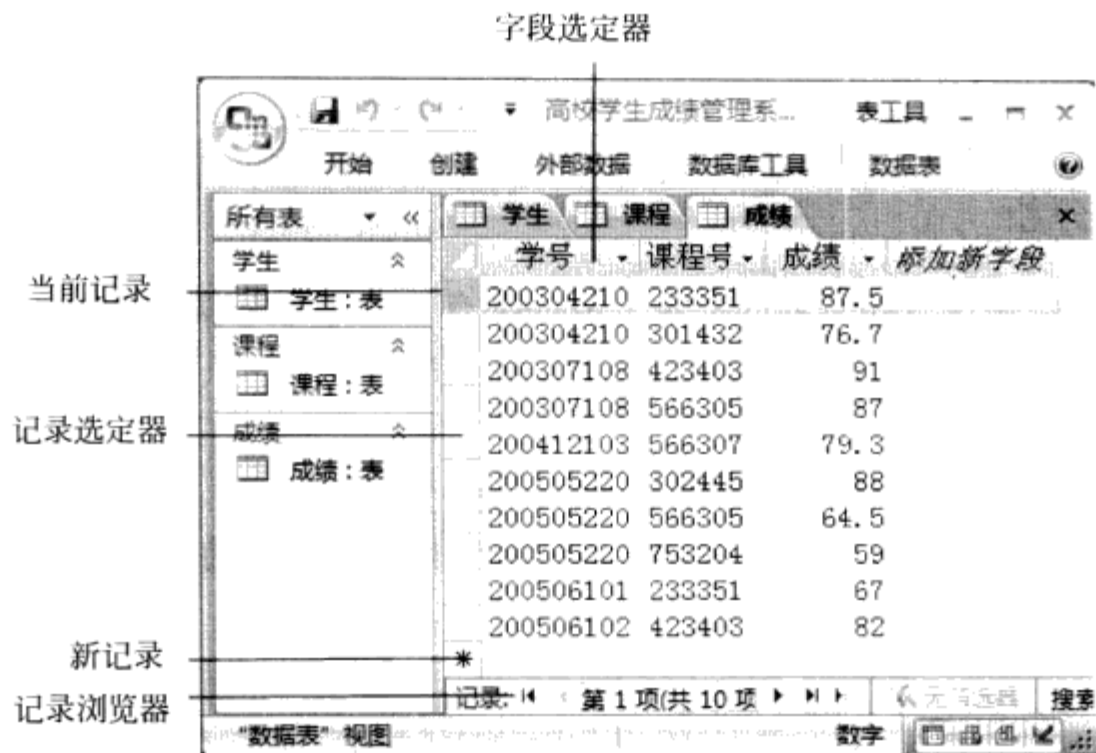


图 8-18 数据表视图

单击字段选定器可以选择一列。

单击记录选定器可以选择一行。

在新记录中可以添加一个新记录，图标“*”表示为新记录。

在选定的当前记录中可以修改记录值，用 Delete 键可以删除当前记录。

利用记录浏览器可以定位记录。

8.4.2 排序

☎ 知识要点分析

单击字段选定器，选择排序字段，可以多选。在右键菜单中选择排序选项，如图 8-19 所示。多字段排序时，按从左至右的顺序排列。即左边字段优先级高于右边的字段。



图 8-19 记录排序

8.4.3 筛选

☎ 知识要点分析

通过筛选可以快速查找需要的记录。Access 有“按选定内容筛选”、“内容排除筛选”、“按窗体筛选”、“高级筛选”几种筛选。筛选并不改变数据，只是将满足筛选条件的记录显示。

1. 按选定内容筛选及内容排除筛选

按选定内容筛选是在有选定内容的字段中，查找出相同值的记录。如图 8-20 中，想要查找所有蒙古族同学。

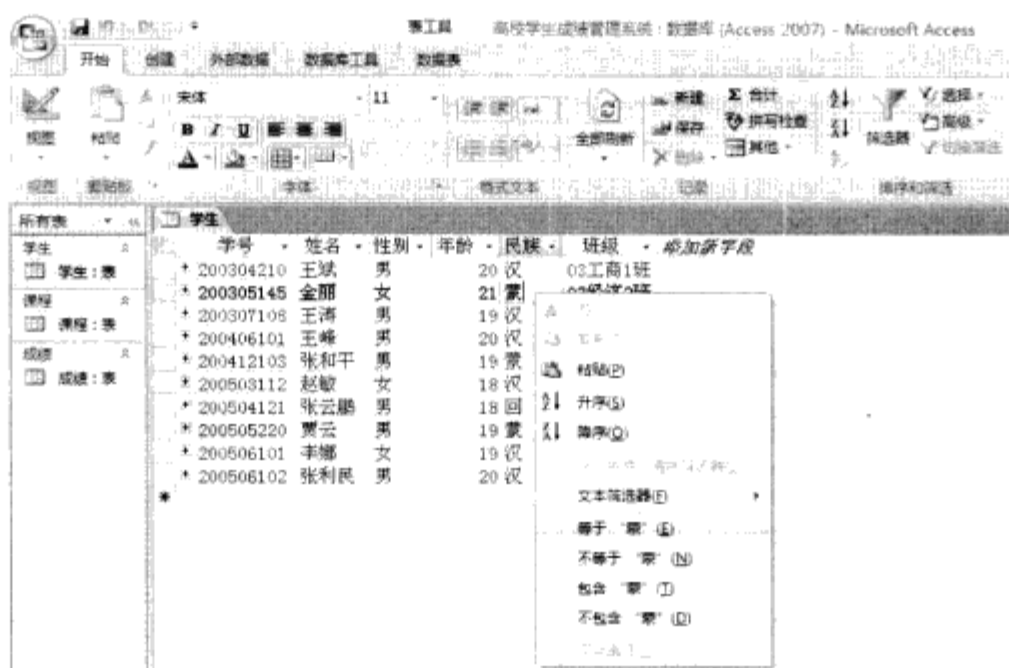


图 8-20 按选定内容筛选

单击菜单项后，所有蒙古族同学显示出来，如图 8-21 所示。

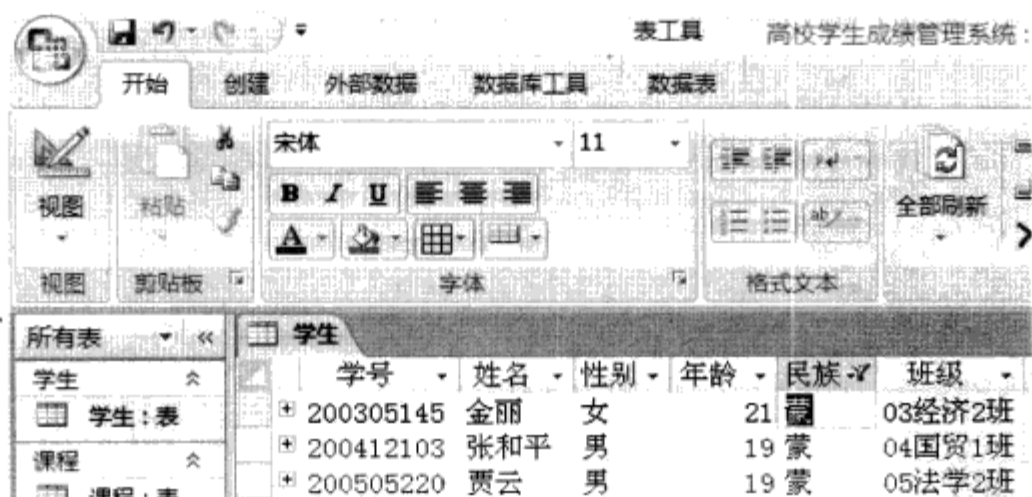


图 8-21 按选定内容筛选结果

2. 按窗体筛选

选择“开始”→“排序和筛选”→“高级”→“按窗体筛选”，可以看到窗体筛选对话框，如图 8-22 所示。

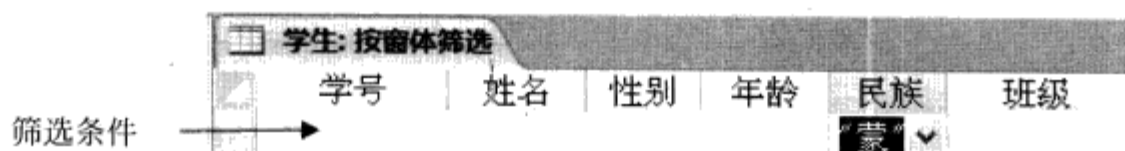


图 8-22 按窗体筛选

在筛选条件里，按字段输入筛选条件，单击“应用筛选”按钮，可以进行筛选，结果如图 8-21 所示。

3. 高级筛选

高级筛选类似于第 9 章讲到的“查询”。下章中将详细介绍。

习 题 8

一、判断题

1. 算术运算符、字符运算符、关系运算符、逻辑运算符中优先级最低的是关系运算符。()
2. 在 Access 中，参照完整性规则不包括查询规则。()
3. 文本类型的数据不能包括数字。()
4. 一个表中的主键可以是一个或多个字段。()
5. 利用 Access 2007 创建的数据库文件，其扩展名为 .accdb。()
6. 表中的主键的数据类型必须是文本或自动编号。()
7. 不同的记录可以具有重复的主键值。()
8. 掩码字符 A 表示必须输入一个数字。()
9. 在数据库中，建立索引的主要作用是防止数据丢失。()
10. 不能确定任何单字段的值的唯一性时，可以将两个或更多的字段组合成为主关键字。()

二、单选题

1. 要使打开的数据库不能被修改，必须以 () 方式打开数据库。
A. 独占 B. 打开 C. 只读 D. 独占只读
2. 设置关系时，如果选中了实施参照完整性准则，那么 ()。
A. 不能将记录添加到相关的表中
B. 不能在相关的表中存在匹配的记录时删除主表中的记录
C. 不能在相关的表中有相关记录时更改主表中的主关键字
D. 以上都包括
3. 下列选项中，不属于逻辑运算符的是 ()。
A. NOT B. IN C. AND D. OR
4. “商品”与“顾客”两个实体集之间的联系一般是 ()。
A. 一对一 B. 一对多 C. 多对一 D. 多对多
5. 整型数字类型的大小为 () B。

- A. 1 B. 2 C. 4 D. 8
6. Access 常用的数据类型有 ()。
- A. 文本、数值、日期和浮点数 B. 数字、字符串、时间和自动编号
C. 数字、文本、日期 / 时间和货币 D. 货币、序号、字符串和数字
7. Access 中, () 字段类型的长度由系统决定。
- A. 是 / 否 B. 文本 C. 货币 D. 备注
8. 数据表视图中, 不可以 ()。
- A. 修改字段的名称 B. 修改字段的类型
C. 删除一个字段 D. 删除一条记录
9. 在关系型数据库中, () 是属性的物理表示。
- A. 文件 B. 表 C. 报表 D. 字段
10. 数据表中的“行”叫做 ()。
- A. 字段 B. 数据 C. 记录 D. 数据视图

三、上机练习

1. 创建一个名为“图书资料管理”的数据库, 建立“书籍”表、“借阅”表和“学生”表, 表结构如表 8-10、表 8-11 所示。(其中“学生”表用“高校学生成绩管理系统”数据库中的“学生”表)

表 8-10 “书籍”表

字段名	类型	长度	字段名	类型	长度
书籍编号	文本	6	总册数	数字	整型, 有效性规则为 0~100
书籍名称	文本	50	出版社	文本	100
购入日期	日期时间	默认值为 2000-1-1	作者	文本	20

表 8-11 “借阅”表

字段名	类型	长度
书籍编号	文本	6
学号	文本	20, 与书籍编号字段联合作主键
借书日期	日期时间	
还书日期	日期时间	

2. 建立表之间的关系。
3. 录入部分数据。

第9章

数据库查询

数据库不仅用来记录各种各样的数据，而且还要对数据进行管理。其中最基本、最具有灵活性的操作就是查询。本章学习查询的基本知识、创建各种类型的查询、修改查询、在查询中执行计算等。

9.1 查询概述

9.1.1 查询的概念

知识要点分析

在设计一个数据库时，为了节省存储空间，常常把数据分类，并分别存放在多个表里。尽管在数据表中可以进行许多操作，如浏览、排序、筛选和更新等，但很多时候还是需要检索一个或多个表（或查询）中符合条件的数据，将这些数据集合在一起，执行浏览、计算等各种操作。查询实际上就是将这些分散的数据再集中起来。即查询是依据一定的查询条件，对数据库中的数据信息进行查找。它与表一样都是数据库的对象。使用查询可以执行一组选定的数据记录集合，虽然这个记录在数据库中实际上并不存在，只是在运行查询时，Access 才从数据源表中提取数据创建它，但正是这个特性，使查询具有了灵活方便的数据操纵能力。

查询的基本作用如下：

- 通过查询浏览表中的数据，分析数据或修改数据。
- 利用查询可以使用户的注意力集中在自己感兴趣的数据上，而将当前不需要的数据排除在查询之外。
- 将经常处理的原始数据或统计计算定义为查询，可大大简化数据的处理工作。用户不必每次都在原始数据上进行检索，从而提高了整个数据库的性能。
- 查询的结果集可以用于生成新的基本表，可以进行新的查询，还可以为窗体、报表以及数据访问页提供数据。由于查询是经过处理的数据集合，因而适合于作为数据源，通过窗体、报表或数据访问页提供给用户。

9.1.2 查询的种类

知识要点分析

Access 2007 提供了多种查询方式,极大地方便了用户的查询工作。查询的类型包括以下几种。

- 选择查询。选择查询是从一张或多张表中检索数据,并且在可以更新记录的数据表中显示结果。可以使用选择查询对记录进行分组、计算总计、平均值等。
- 参数查询。参数查询是在执行时显示对话框,通过用户输入的信息作为查询的一个条件来生成查询表。此方式的查询比较灵活,作为窗体和报表的基础很方便。
- 交叉表查询。此种查询方式是将表中数据分组,一组作为数据表的左侧,另一组作为数据表的上部,表中显示某个字段的总计、平均值等。
- 操作查询。操作查询是通过使用一个操作来更改数据表的多条记录。包括生成、更新、追加、生成表查询。此种查询适合于对数据表大批量的数据修改需要。
- SQL 查询。SQL 即结构化查询语言,是关系型数据库的应用语言。Access 中的所有查询都可以认为是一个 SQL 查询。

查询有 5 种视图:设计视图、数据表视图、SQL 视图、数据透视表视图和数据透视图视图。

在 Access 中将查询与数据表作为同类型的对象,因此,一个数据库中的数据表与查询表的名称不能相同。另外,查询表和数据表都可作为生成记录至窗体、报表、数据页等对象的数据来源。

9.2 创建查询

创建查询可以使用查询向导,也可以在设计视图中由用户指定查询条件。可以对单表创建查询,也可以对多表创建查询。

9.2.1 使用向导创建单表查询

应用实例

【实例 9-1】 单表查询是指对单个表中的数据创建查询表。例如通过“学生”表创建只包括“学号”、“姓名”、“班级”字段的查询表。

创建查询步骤如下:

(1) 打开“高校学生成绩管理系统”后,单击“创建”选项卡中的“其他”组中的“查询向导”按钮,如图 9-1 所示。

(2) 在打开的“新建查询”对话框中,选择简单查询向导,即可打开“简单查询向导”对话框,如图 9-2 所示。

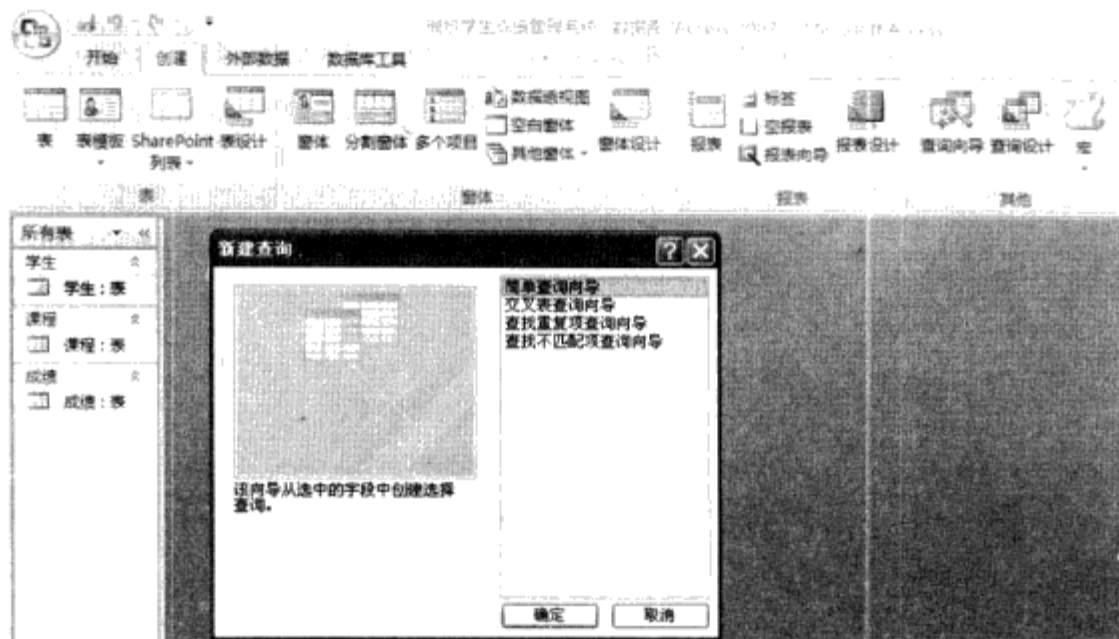


图 9-1 “新建查询”对话框

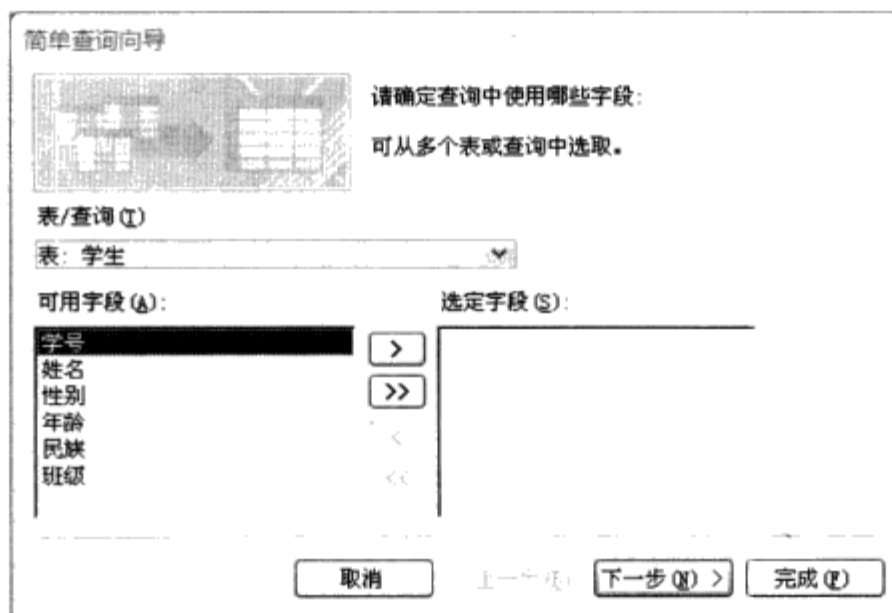


图 9-2 “简单查询向导”对话框 1

(3) 在“表/查询”下拉列表框中选择要创建查询的表，如“学生”，然后在下面的“可用字段”列表中双击要生成查询表的字段，如“学号”、“姓名”、“班级”，使这些字段显示在“选定的字段”列表框中，如图 9-3 所示。

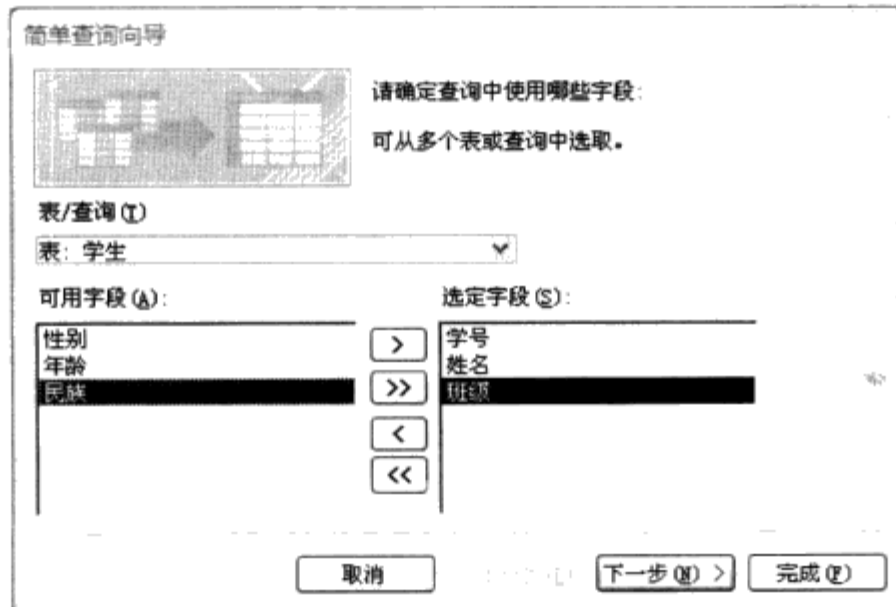


图 9-3 “简单查询向导”对话框 2

(4) 单击“下一步”按钮，在接下来的对话框中设置查询表的名称，本例中输入“学生查询”，如图 9-4 所示。

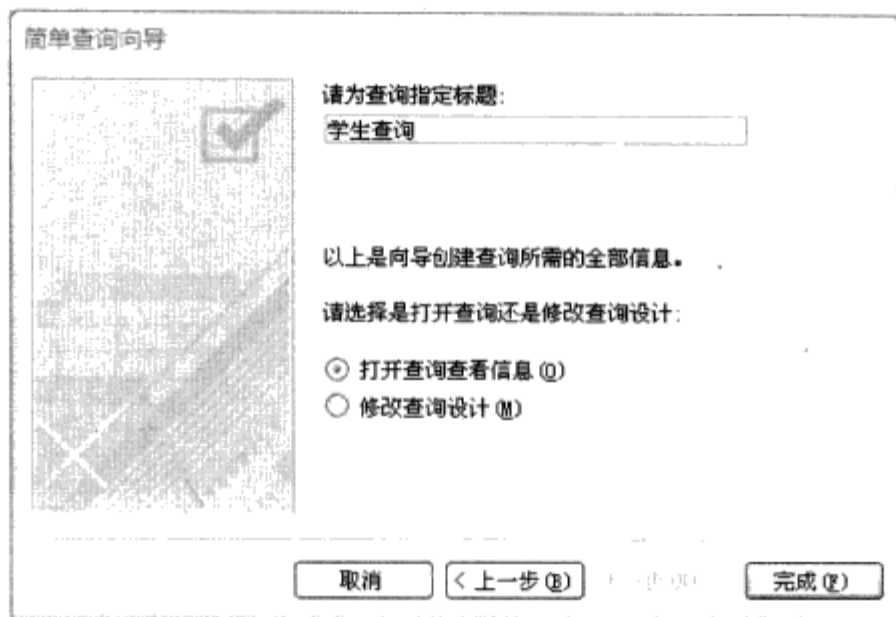


图 9-4 “简单查询向导”对话框 3

(5) 单击“完成”按钮，生成的查询表如图 9-5 所示。

所有表	学生查询		
学生	学号	姓名	班级
学生:表	200304210	王斌	03工商1班
学生查询	200305145	金丽	03经济2班
课程	200307108	王涛	03电商1班
课程:表	200406101	王峰	04市销2班
成绩	200412103	张和平	04国贸1班
成绩:表	200503112	赵敏	05经济1班
	200504121	张云鹏	05工商3班
	200505220	贾云	05法学2班
	200506101	李娜	05法学1班
	200506102	张利民	05法学1班

图 9-5 “学生查询”的查询结果

9.2.2 使用向导创建多表查询

使用向导可以非常方便地创建多表查询。方法类似创建单表查询。只是在查询步骤中重复步骤 (3)，选择多个表中的多个查询字段。这里就不再多述。

9.2.3 用查询设计器创建查询

☞ 知识要点分析

使用查询向导只能进行一些简单的查询，或者进行某些特定的查询，如查找重复项等。Access 2007 还提供了一个功能更加强大的查询设计器。通过查询设计器不仅可以从头设计

一个查询，而且还可以用来对一个已有的查询进行编辑和修改。当用户明白一个查询是怎样组成的和有一个什么样的结构后，就会发现在设计器中创建查询更容易，所创建的查询作用更大。

要使用查询设计器设计查询，首先需要认识查询设计器。

查询设计器如图 9-6 所示，该对话框分为两个部分：上部是数据表/查询显示区，下部是查询设计区。数据表/查询显示区用来显示查询所使用的基本表或查询（可以是多个表/查询），查询设计区用来指定具体查询条件。

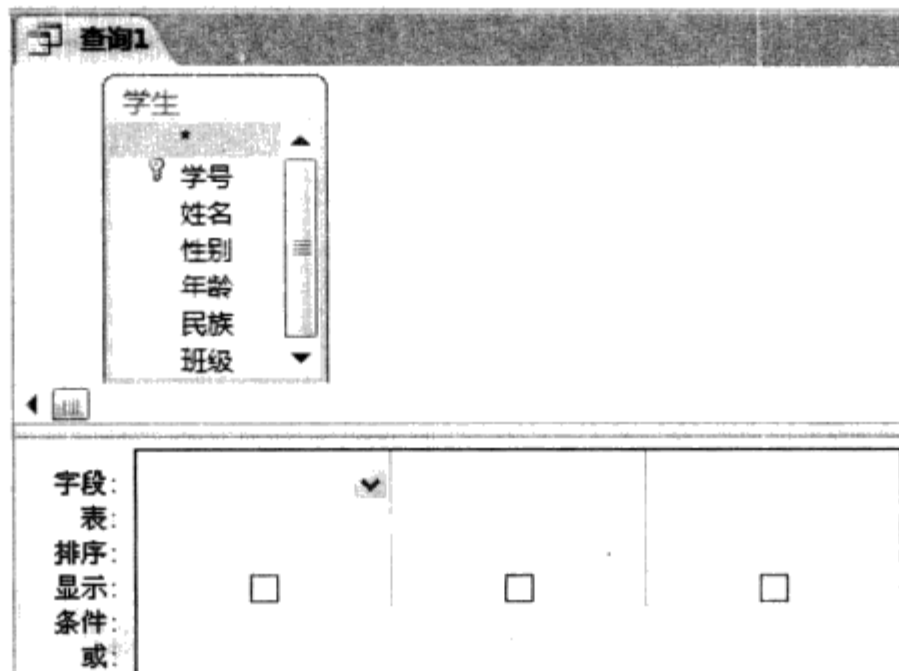


图 9-6 查询设计器

查询设计区中网格的每一列都对应着查询结果集中的一个字段，网格的行标题表明字段的属性及要求。

- 字段：查询工作表中所使用的字段名称。
- 表：该字段所来自的数据表。
- 排序：确定是否按该字段排序以及按何种方式排序。
- 显示：确定该字段是否在查询工作表中显示。
- 条件：用来指定该字段的查询条件。
- 或：用来提供多个查询条件。

应用实例

【实例 9-2】 通过查询设计器查询学号是“200505220”的学生选修了几门课程，要求显示“学号”、“姓名”、“课程号”、“成绩”字段。

创建查询步骤如下：

- (1) 打开“高校学生成绩管理系统”数据库，单击“创建”选项卡中的“其他”组中的“查询设计”按钮，打开如图 9-7 所示的“查询设计”窗口和“显示表”对话框。
- (2) 在“显示表”对话框中的“表”选项卡中，双击要创建查询的数据表，如“学生”表和“成绩”表，将其添加到查询设计窗口中。
- (3) 单击“关闭”按钮，关闭“显示表”对话框。
- (4) 在“查询设置”窗口中，双击或从表中拖动创建查询表所需的字段，使这些字段

出现在相应的“字段”单元格中。



图 9-7 查询设置窗口

(5) 在下面的“显示”设置单元格中选择是否显示该字段。

(6) 在“学号”字段的“条件”单元格中输入查询的条件“200505220”。单击其他单元格时，会为设置的文本条件自动添加双引号，如图 9-8 所示。

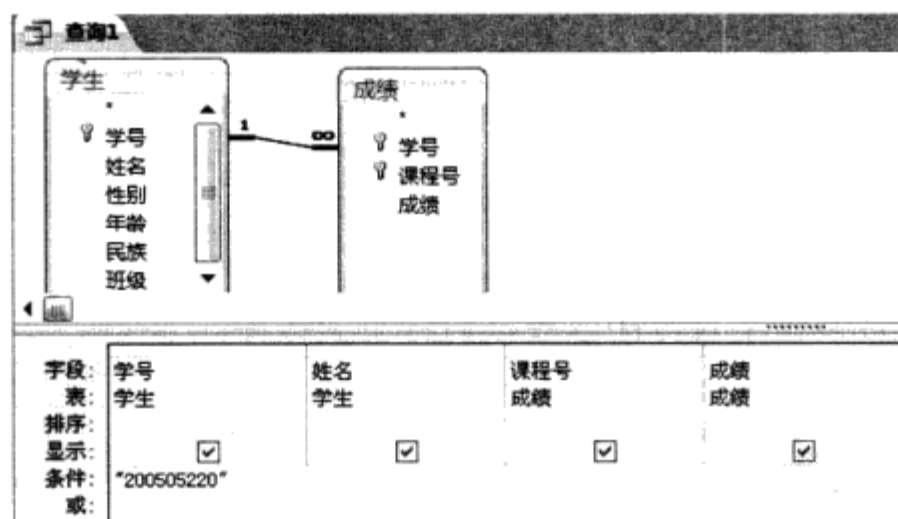



图 9-8 查询设置

(7) 设置完毕后，单击“设计”选项卡中“结果”组中“运行”按钮, 生成的查询表如图 9-9 所示。

学号	姓名	课程号	成绩
200505220	贾云	753204	59
200505220	贾云	566305	64.5
200505220	贾云	302445	88

图 9-9 “学生选课查询”的查询结果

(8) 关闭该窗口,并在弹出的提示框中单击“是”按钮,然后在继续弹出的“另存为”对话框中设置查询表的名称为“学生选课查询”。

如果需要保留该查询表,单击“保存”按钮,在弹出的“另存为”对话框中为其命名。这时,在数据库对话框的“查询”对象列表框中会看到保存的查询表。

对于创建的查询,还可以在设计视图中进行修改,如插入新的字段、改变字段的位置、隐藏字段等。

9.3 创建条件查询

创建查询时,可以通过在“条件”单元格中输入条件表达式来限制结果中的记录。如何正确地构建条件表达式,是在设计查询时必须解决的基本问题之一。

9.3.1 查询条件的设置

☞ 知识要点分析

查询条件是指创建查询时输入的条件。条件可以是运算符、常量、字段值、函数以及字段名和属性的任意组合。条件中的字段名必须用方括号括起来,窗体或表对象在后面加感叹号“!”。

条件可以在文本、数字、日期/时间、备注、是否等类型的字段中设置。

如果在查询中要设置多个条件,条件与条件间的关系可以是 And 或 Or。用 And 连接的条件为前、后的条件必须同时具备,用 Or 连接的条件为只要具备前面的条件或后面的条件即可。

条件的表达式可以包括各种算术运算符、关系运算符、逻辑运算符、函数和特殊运算符。特殊运算符见表 9-1。

表 9-1 特殊运算符及其含义

特殊运算符	说明	举例	举例说明
In	用于指定一个字段值的列表	In("内蒙古","吉林")	表示查询籍贯是内蒙古或吉林的学生记录
&	连接字符串	"a"&"b"	"ab"
Between And	用于指定一个字段的范围	Between #2001-02-01# And #2001-04-30#	查询 2001 年 2 月 1 日到 2001 年 4 月 30 日的记录
Is Like	用于查找类似文本字段的字符模式	Is Like"内蒙*"	查询开头为“内蒙”的记录
Is Not Like		Is Not Like"内蒙*"	查询开头不为“内蒙”的记录
Is Null	空记录的查找	Is Null	查找空记录
Is Not Null		Is Not Null	查找非空记录

1. 条件表达式中的运算符

1) And 运算符

And 运算符称为“逻辑与”运算符。条件表达式 <A> And 限定查询结果记录集

中的记录必须同时满足由 And 所连接的两个条件 A 和 B。

2) Or 运算符

Or 运算符称为“逻辑或”运算符。条件表达式 <A> Or 限定查询结果记录集中的记录只需要满足由 Or 所连接的两个条件 A 和 B 中的一个。

3) In 运算符

用于指定某一系列值的列表。例如 In (“西安”，“南京”，“广州”) 等价于“西安” Or “南京” Or “广州”。当表达式中包含的值很多时，使用 In 运算符要简短得多，而且意义也更为明晰。

4) Between And

用于指定一个范围。主要用于数字型、货币型、日期型字段。条件表达式 Between <A> And 限定查询结果记录集中的记录值介于 A、B 之间。

5) Like

用于查找指定样式的字符串，可使用一些通配符来实现模糊查询。例如 Like “Smith” 指定查找包含字符串“Smith”的记录，而 Like “Sm*” 则指定查找包含以“Sm”开头的字符串的记录。

2. 通配符

在指定字符串的样式时，可以使用的通配符如表 9-2 所示。

表 9-2 字符串样式中的通配符

通配符	匹配的内容
?	一个字符
*	零个或多个字符
#	一个数字 (0~9)
[字符表]	字符表中的一个字符
[!字符表]	字符表中没有的一个字符

可用方括号 “[]” 为字符串中该位置的字符设置一个范围，如 [a-z]、[0-9]、[!0-9] 等，用连字符 “-” 来隔开范围的上下界。例如，表达式 Like “P[A-F]###” 的含义为查找以字母 P 开头，后跟 A 到 F 之间的任何字母和 3 个数字的数据。又如，表达式 Like “a?[a-f]#[!0-9]*” 的含义为查找的字符串中第一个字符为 a，第二个字符为任意字符，第三个字符为 a~f 中的任意一个字符，第 4 个为数字，第 5 个为非 0~9 的任意字符，其后为任意字符串。

方括号中的一组字符可以匹配表达式中任何的单一字符，而且字符表中几乎可以包含 ANSI 字符集中的任何字符，包括数字。事实上，特殊字符，如左括号 ([)、问号 (?)、井号 (#) 和星号 (*)，当它们括在括号内时，可以直接和它们自己匹配。一组字符内的右括号 (]) 不能匹配它自己，但是如果它是一组之外的单一字符，就能用来匹配。

3. 条件表达式示例

表 9-3 是一些条件表达式的例子。

表 9-3 表达式示例

表达式	意义
Len([公司名称])>Val(30)	公司名称在 30 个字符以上
In("美国", "英国", "加拿大")	美国、英国或加拿大

续表

表达式	意义
Between #2/2/2005# And #12/1/2005#	2005年2月2日到2005年12月1日
订购日期 DateSerial(Year([订购日期]),Month([订购日期])+1,1)-1	订购日期为每个月最后一天
[货主地区] Is Not Null	“货主地区”字段不能取空值
[货主名称] Like "*Imports"	“货主名称”以“Imports”结尾

9.3.2 数值条件的设置


设置数值类型数据的条件时，可使用算术运算符和比较运算符以及布尔运算符直接输入表达式。

应用实例

【实例 9-3】 创建成绩在 70~90 之间的学生查询。要求包括“学号”、“姓名”、“成绩”字段，保存为“70~90 的学生”查询。

创建查询步骤如下：

(1) 在“查询设计”窗口中，设置查询条件“>=70 and <=90”，如图 9-10 所示。

(2) 设置完毕，单击“设计”选项卡中“结果”组中“运行”按钮，生成的查询表如图 9-11 所示。

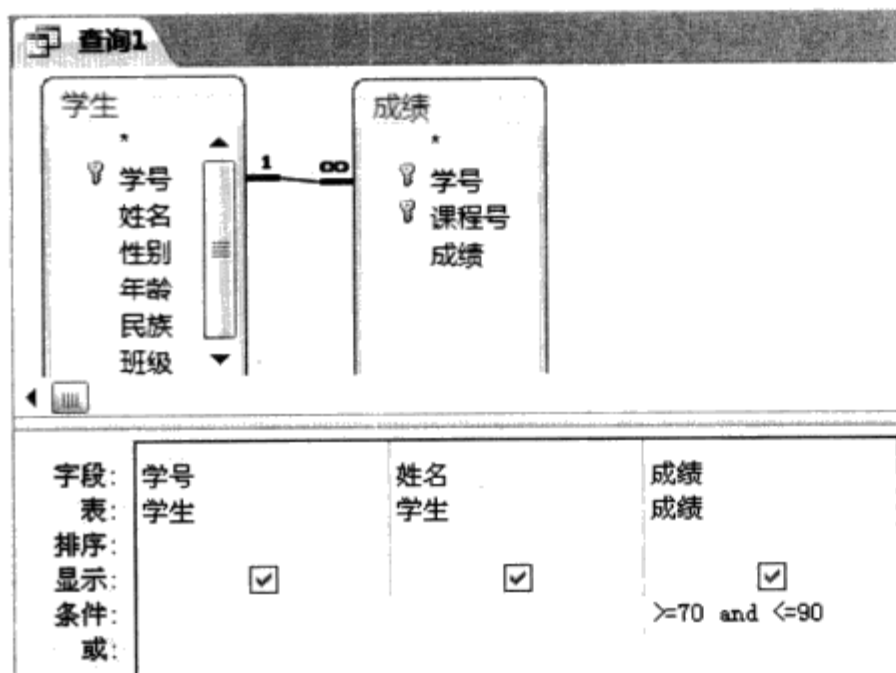


图 9-10 设置数值条件

学号	姓名	成绩
200304210	王斌	87.5
200304210	王斌	76.7
200307108	王涛	87
200412103	张和平	79.3
200506102	张利民	82
200505220	贾云	88

图 9-11 生成的数值条件查询表

(3) 设置查询表的名称为“70~90 之间的学生”。

9.3.3 生成新字段的计算

在查询时，有时需要通过计算生成新字段，其中的表达式可直接在设计窗口的字段名单元格中输入，也可以通过表达式生成器进行设置。注意表达式中字段名和公式之间用“:”

分隔。

应用实例

【实例 9-4】 创建学号是“200505220”的学生的“期中各科成绩”查询，要求包括“学号”、“姓名”、“期中各科成绩”字段（期中各科成绩=[成绩]*0.3）。

创建查询步骤如下：

- (1) 在查询设计视图中，选择“学生”和“成绩”表。
- (2) 添加“学号”、“姓名”字段。
- (3) 为创建“期中各科成绩”字段，在后面单元格中输入“期中各科成绩：[成绩]*0.3”。
- (4) 选中该字段的“显示”选项，如图 9-12 所示。
- (5) 切换到数据表视图状态，生成的查询表如图 9-13 所示。

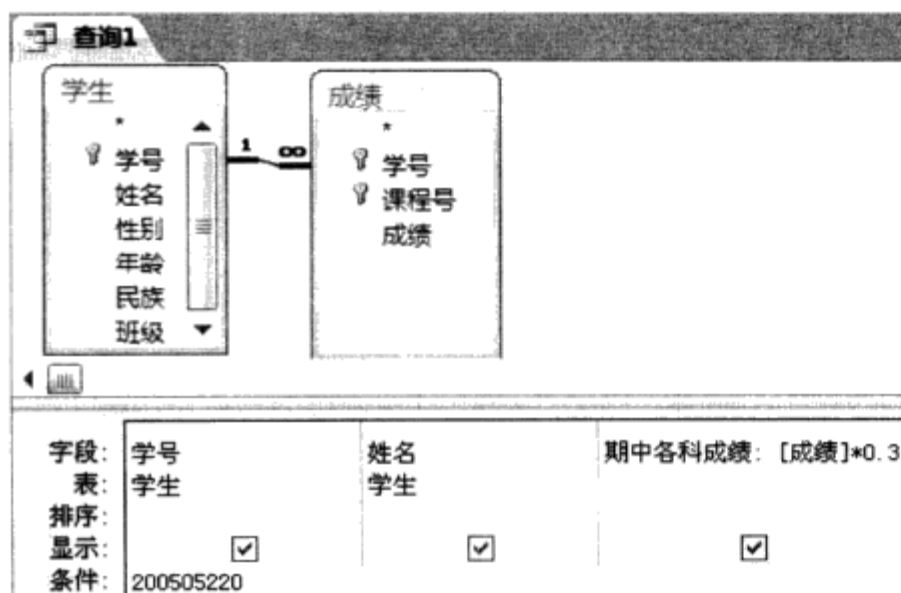


图 9-12 设置新字段

学号	姓名	期中各科成绩
200505220	贾云	17.7
200505220	贾云	19.35
200505220	贾云	26.4

图 9-13 生成的计算查询结果

- (6) 将生成的查询表保存为“期中各科成绩”，并关闭。

9.4 创建交叉表查询

知识要点分析

交叉表查询是一种特殊的合计查询类型，可以使数据按电子表格的方式显示查询结果集，这种显示方式在水平与垂直方向同时对数据进行分组，使数据的显示更为紧凑。比如用户要从基本表中得到这样的信息：某一学生选修了哪些课程及课程成绩。从查询的类型可知这种查询应该用交叉表查询。

在交叉表的设计视图图 9-14 中，“交叉表”栏的下拉列表中 4 个选项的含义分别如下。

- 行标题：设置为行标题字段中的数据将作为交叉表的行标题，一个交叉表查询中可以有不超过 3 个的行标题。

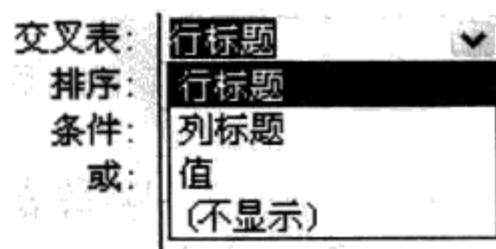


图 9-14 “交叉表”栏

- 列标题：设置为列标题字段中的数据将作为交叉表的列标题，一个交叉表查询中只能有一个字段作为列标题。
- 值：设置为值的字段是交叉表中行列标题相交单元格的显示内容，在一个交叉表查询中只能有一个字段作为值。
- 不显示：设置为不显示的字段内容将不会出现在交叉表查询结果集中。如果用户不希望显示作为查询的筛选条件的字段，那么该字段所对应的总计栏一般设置为条件，交叉表栏设置为不显示。在一个交叉表查询中可以有多多个字段设置为不显示。

应用实例

【实例 9-5】 查询学生表中每个学生选修了哪些课程及课程成绩。

创建查询步骤如下：

- (1) 打开“查询设计”窗口，并选择“学生”表、“成绩”表和“课程”表。
- (2) 添加“学号”、“姓名”、“课程名”和“成绩”字段。
- (3) 单击“设计”选项卡中的“查询类型”组中的“交叉表”按钮，则设计网格中的“显示”栏变为“交叉表”栏。

(4) 单击“学号”字段下的“总计”栏，在下拉列表框中选择“分组”，在“交叉表”栏的下拉列表框中选择“行标题”；同样，设置“姓名”字段下“总计”栏的下拉列表框中选择“分组”，在“交叉表”栏的下拉列表框中选择“行标题”；设置“课程名”字段下“总计”栏的下拉列表框中选择“分组”，在“交叉表”栏的下拉列表框中选择“列标题”；设置“成绩”字段下“总计”栏的下拉列表框中选择“总计”，在“交叉表”栏的下拉列表框中选择“值”，如图 9-15 所示。

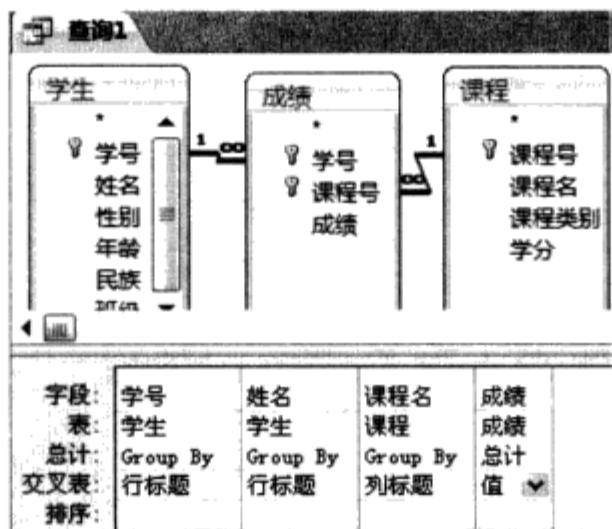



图 9-15 设计好的交叉表查询设计视图

- (5) 单击“运行”按钮, 则交叉表查询的结果集如图 9-16 所示。

学号	姓名	大学语文	高等数学	计算机基础	经营战略	企业策划	市场营销	数据库原理
	王斌	87.5	67.5					
200307108	王涛			91		87		
200412103	张和平				79.5			
200505220	贾云					64.5	59	88
200506101	李娜	67						
200506102	张利民			82				

图 9-16 交叉表查询结果集

9.5 操作查询

操作查询是复制或更新数据的查询。操作查询包括追加查询、删除查询、更新查询和生成表查询 4 种类型。追加查询和生成表查询复制数据，删除查询和更新查询则更改数据。

9.5.1 生成表查询

生成表查询是通过对多个表的数据进行组织以生成一个新的数据表。

应用实例

【实例 9-6】 生成一个课程名为“企业策划”的数据表，其中包括“学生”表中的“学号”、“姓名”字段，“课程”表中的“课程名”字段和“成绩”表中的“成绩”字段。创建查询步骤如下。

(1) 在新建的查询设计窗口，如图 9-17 所示设置其中的各项。

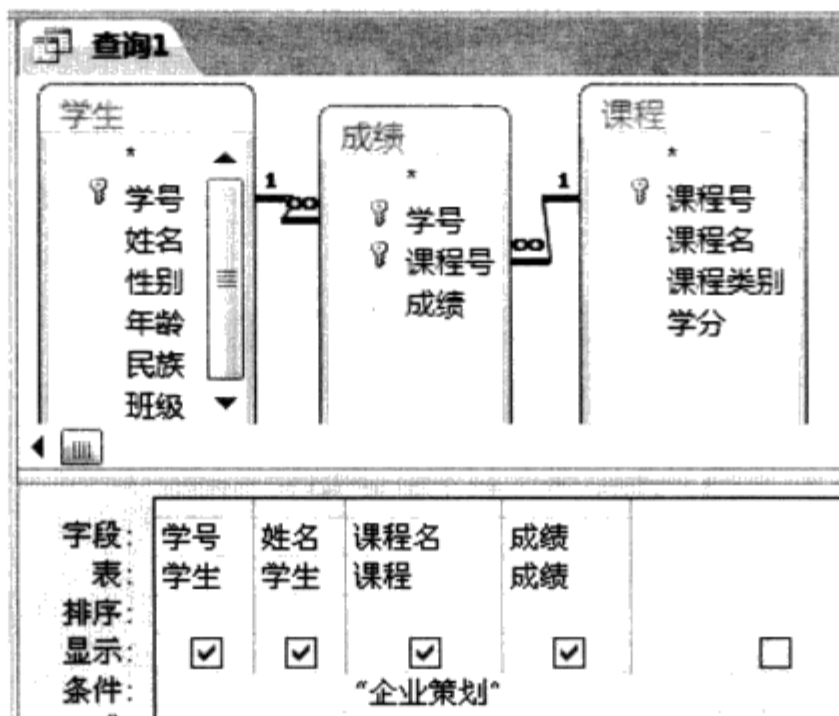


图 9-17 生成表查询设置

(2) 切换到数据表视图，可以看到如图 9-18 所示查询结果。

学号	姓名	课程名	成绩
200307108	王涛	企业策划	87
200412103	张和平	企业策划	79.5
200506102	张利民	企业策划	82
200505220	贾云	企业策划	64.5
*			

图 9-18 查询结果

(3) 回到设计视图。为了形成生成表查询，只要单击“设计”选项卡中的“查询类型”组中的“生成表”按钮即可。

(4) 在出现的对话框中选择将生成的表保存在当前的数据库中，名字为“企业策划”，如图9-19所示，单击“确定”按钮。

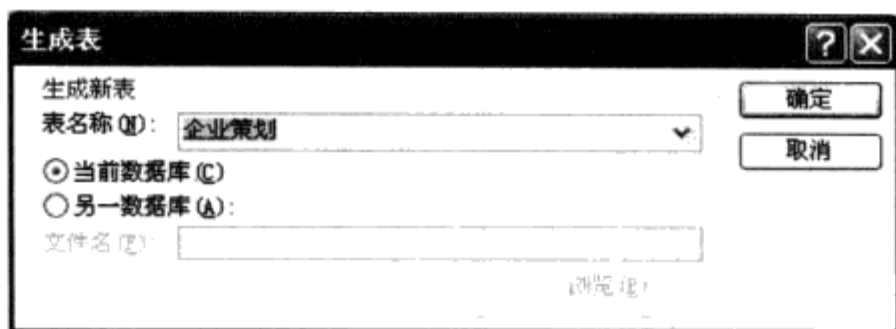


图 9-19 “生成表”对话框

(5) 单击“运行”按钮，屏幕提示如图9-20所示的对话框。

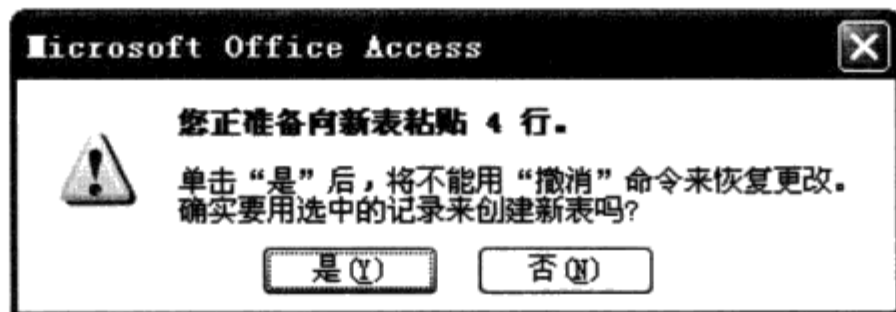


图 9-20 “生成表”提示框

(6) 单击“是”按钮，完成表的创建。

(7) 关闭设计视图窗口，保存设计的查询表为“生成企业策划”。

(8) 在数据库的左侧“所有表”列表框中，可以看到新生成的名为“企业策划”的数据表。

9.5.2 创建追加查询

有时需要将某个表中符合一定条件的记录添加到另一个表中，这类工作可以通过追加查询操作来完成。

应用实例

【实例 9-7】 对“企业策划”的数据表，生成一个追加课程名为“大学语文”的追加查询。

创建查询步骤如下：

(1) 在“所有表”列表框中，选择“生成企业策划”查询表，并单击“设计”按钮，在设计视图中打开该对象。

(2) 将其另存为“生成企业策划和大学语文”。

(3) 修改课程名称的条件为“大学语文”。

(4) 单击“设计”选项卡中的“查询类型”组中的“追加”按钮，在打开的“追加”对话框中进行如下设置，如图 9-21 所示。

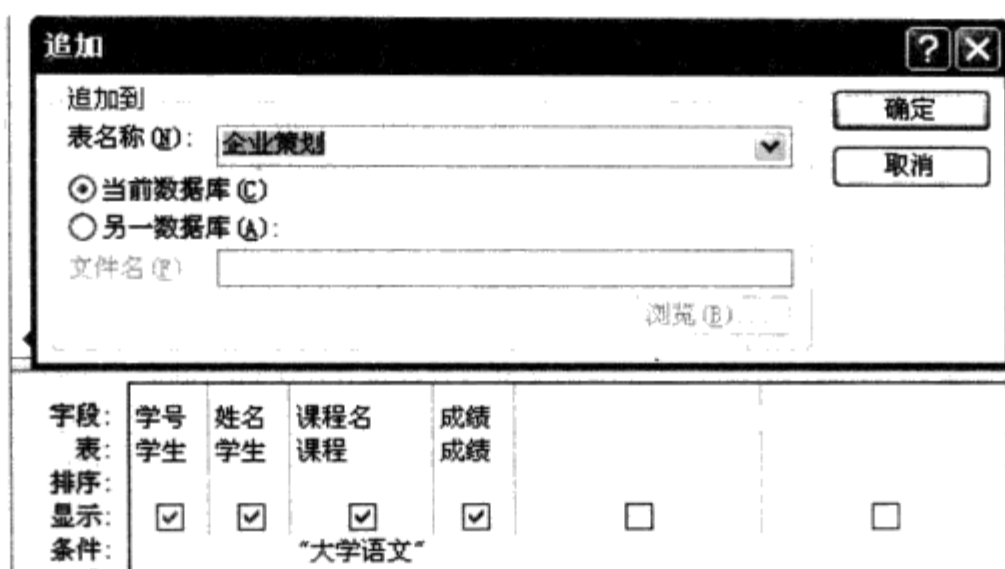


图 9-21 “追加”对话框的设置

(5) 单击“确定”按钮后，创建的追加查询设计如图 9-22 所示。

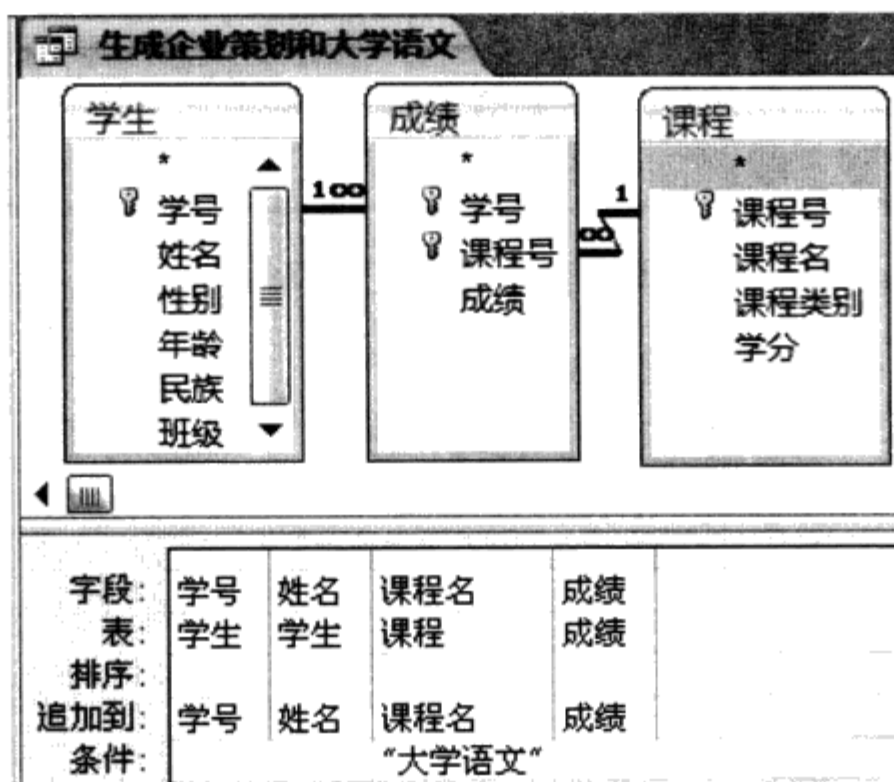



图 9-22 追加查询设置

(6) 单击“运行”按钮，在弹出的“追加”提示框中，单击“是”按钮，将符合条件的一组记录追加到指定的表中。

(7) 关闭并保存追加查询表。

(8) 在表对象中，双击“企业策划”数据表会看到增加的记录，如图 9-23 所示。

9.5.3 删除查询

删除查询可以成批删除数据表中的某类记录。

学号	姓名	课程名	成绩
200307108	王涛	企业策划	87
200412103	张和平	企业策划	79.5
200506102	张利民	企业策划	82
200505220	贾云	企业策划	64.5
200304210	王斌	大学语文	87.5
200506101	李娜	大学语文	67
*			

图 9-23 追加后的数据表

应用实例

【实例 9-8】 删除“企业策划”表中课程名是“企业策划”的记录。

创建查询步骤如下：

- (1) 在“所有表”列表框中，选择“生成企业策划和大学语文”查询表。
- (2) 单击“设计”选项卡中的“查询类型”组中的“删除”按钮，在设计网格中插入“删除”行。
- (3) 分别添加“*”和“课程名”字段，设置条件为“成本会计”，如图 9-24 所示。

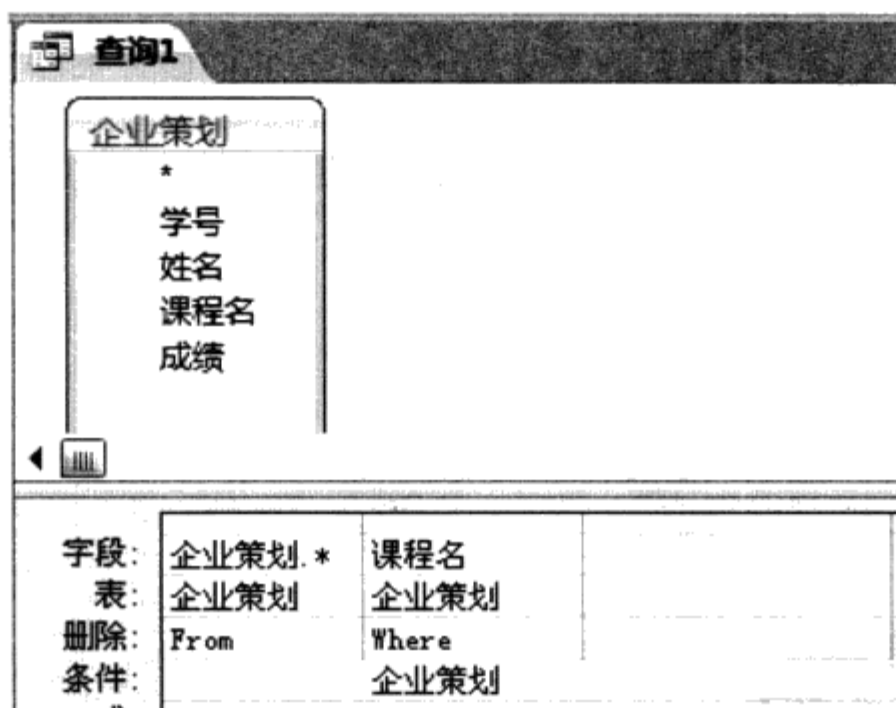



图 9-24 删除查询设置窗口

(4) 单击“开始”选项卡中的“视图”按钮，预览删除查询检索到的一组记录，如图 9-25 所示。

学号	姓名	企业策划.课程名	成绩	字段0
200307108	王涛	企业策划	87	企业策划
200412103	张和平	企业策划	79.5	企业策划
200506102	张利民	企业策划	82	企业策划
200505220	贾云	企业策划	64.5	企业策划
*				

图 9-25 预览删除的记录

(5) 单击“运行”按钮，屏幕显示删除记录的提示框。单击“是”按钮，开始删除设置的记录。

(6) 打开“企业策划”表，会看到课程名为“企业策划”的记录全部被删除。

9.5.4 更新查询

当需要成批修改表中的数据时，可以使用更新查询来提高工作效率。

应用实例

【实例 9-9】 将“高校学生成绩管理系统”数据库中的所有班级是“05 法学 1 班”的学生都改为“05 会计 1 班”。

创建查询步骤如下：

(1) 在“所有表”列表框中，将“学生”表另存为“学生（改班级后）”表。

(2) 在新建的“选择查询”窗口中添加“学生（改班级后）”表。

(3) 将“班级”字段添加到查询字段中后，条件设置为“05 法学 1 班”。

(4) 单击“设计”选项卡中的“查询类型”组中的“更新”按钮，在设计网格中插入“更新到”行。

(5) 将“班级”字段的“更新到”设置为“05 会计 1 班”，如图 9-26 所示。

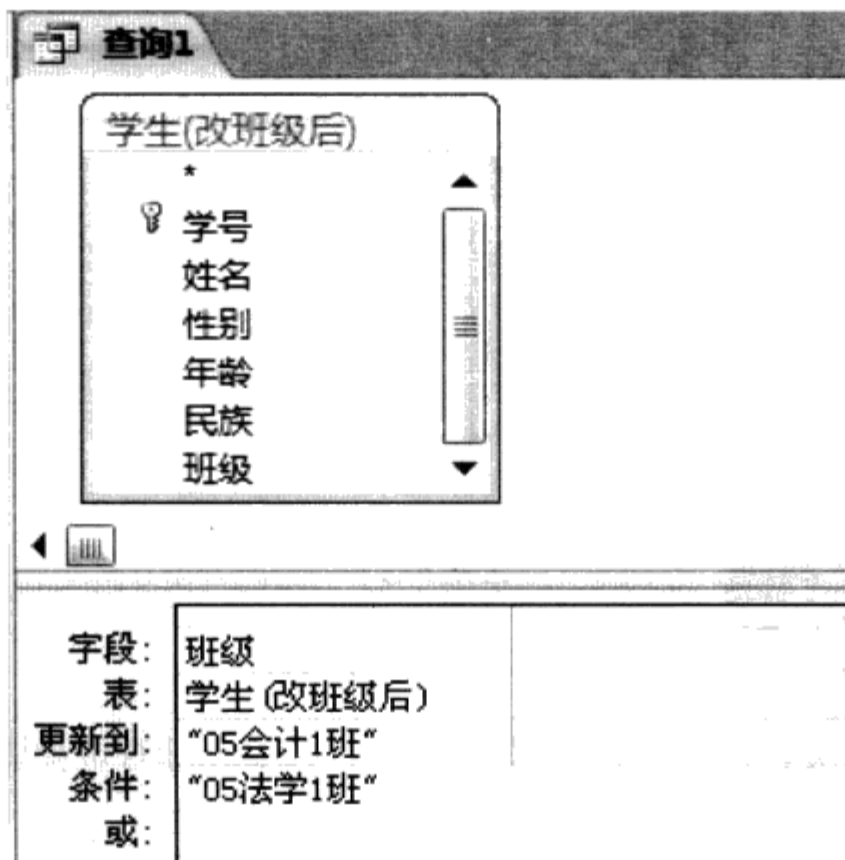


图 9-26 更新查询设计窗口

(6) 单击“运行”按钮，屏幕提示准备更新的提示框。

(7) 单击“是”按钮，就完成了记录的更新。

(8) 打开“学生（改班级后）”表，可以看到原记录已经按照设置的条件更新了，参见图 9-27 和图 9-28。

学号	姓名	性别	年龄	民族	班级	添加新字段
200304210	王斌	男	20	汉	03工商1班	
200305145	金丽	女	21	蒙	03经济2班	
200307108	王涛	男	19	汉	03电商1班	
200406101	王峰	男	20	汉	04市销2班	
200412103	张和平	男	19	蒙	04国贸1班	
200503112	赵敏	女	18	汉	05法学1班	
200504121	张云鹏	男	18	回	05工商3班	
200505220	贾云	男	19	蒙	05法学2班	
200506101	李娜	女	19	汉	05法学1班	
200506102	张利民	男	20	汉	05法学1班	

图 9-27 更新前的数据表

学号	姓名	性别	年龄	民族	班级	添加新字段
200304210	王斌	男	20	汉	03工商1班	
200305145	金丽	女	21	蒙	03经济2班	
200307108	王涛	男	19	汉	03电商1班	
200406101	王峰	男	20	汉	04市销2班	
200412103	张和平	男	19	蒙	04国贸1班	
200503112	赵敏	女	18	汉	05会计1班	
200504121	张云鹏	男	18	回	05工商3班	
200505220	贾云	男	19	蒙	05法学2班	
200506101	李娜	女	19	汉	05会计1班	
200506102	张利民	男	20	汉	05会计1班	

图 9-28 更新后的数据表

9.6 SQL 查询

SQL 是 Standard Query Language（即标准查询语言）的缩写。该语言是关系型数据库通用的标准语言。在使用查询向导或查询设计器创建查询时，Access 自动构造等效的 SQL 语句，可以切换到 SQL 视图查看这个语句。将这两种视图互相对照就会看到：查询设计器中的大多数查询属性都可以在 SQL 视图找到等效的可用子句和选项。因此，对于熟悉 SQL 语言的用户来说，也可以直接在 SQL 视图输入 SQL 语句创建查询。

9.6.1 SELECT 语法略解

☎ 知识要点分析

在 SQL 语言中，最实用的语法莫过于“SELECT...FROM...”。“SELECT”顾名思义是“选取”的意思。“SELECT...”（选取什么呢？）和“FROM...”（从哪里选取？）构成

了 SELECT 的基本语法。SELECT 的语法如下：

```
SELECT [ALL | DISTINCT] [TOP n [PERCENT]] <目标列表表达式> [[AS] <新列名>] [, ...n]
FROM <表名> [[AS] <别名>] [, ...n]
[WHERE <条件表达式>]
[GROUP BY <分组表达式>]
[HAVING <条件表达式>]
[ORDER BY <排序表达式> [ASC | DESC]] [, ...n]
```

参数说明如下。

(1) [ALL | DISTINCT]: 指定在结果集中可否显示重复行。ALL 表示可以, DISTINCT 表示不可以, 缺省时表示结果集中可以出现重复行。

(2) [TOP n [PERCENT]]: 指定只从查询结果集中输出前 n 行。 n 是介于 0 和 4 294 967 295 之间的整数。如果还指定了 PERCENT, 则只从结果集中输出前百分之 n 行, 且 n 必须是介于 0 和 100 之间的整数。

(3) <目标列表表达式> [[AS] <新列名>][, ...n]: 指定为结果集选择的列。选择列表是以逗号分隔的一系列表达式。如果目标表达式是 “*”, 则表示返回表中所有的列。需要的话可以为输出列重新指定新列名。

(4) FROM <表名> [[AS] <别名>][, ...n]: 指定从其中检索数据的表。多表连接查询, 需要指定列所属的表时, 可以用表名, 也可以在 FROM 子句中为表指定别名后, 直接引用其别名。

(5) WHERE <条件表达式>: 指定数据检索的条件, 用于限制返回的行。

(6) GROUP BY <分组表达式>: 实现对结果集的分组。将在 <分组表达式> 上具有相同值的记录放在一个组内, 常用于分组统计。

(7) HAVING <条件表达式>: 指定分组后的筛选条件。HAVING 子句通常与 GROUP BY 子句一起使用, 对 GROUP BY 子句分组的结果进行筛选, 保留满足条件的分组。如果不使用 GROUP BY 子句, HAVING 子句的作用与 WHERE 子句一样。

(8) ORDER BY <排序表达式>: 指定结果集的行按 <排序表达式> 值进行有序的排列。<排序表达式> 可以将排序列指定为列名或列的重命名和表达式, 也可以指定为查询结果集中列的序号。

(9) [ASC | DESC]: 指定查询结果集的行按 <排序表达式> 值的排列顺序排序。ASC 为升序, DESC 为降序, 缺省时按升序排列。

下面举例介绍 SELECT 语句的具体用法。

应用实例

【实例 9-10】 查询学生表中全体学生的姓名。

```
SELECT 姓名 FROM 学生
```

【实例 9-11】 查询学生表中的所有信息。

```
SELECT * FROM 学生
```

【实例 9-12】 查询成绩表中的成绩和增加 10 分后的成绩, 并且将增加 10 分后的成绩

列命名为“新成绩”。

```
SELECT 成绩,成绩+10 AS 新成绩 FROM 成绩
```

【实例 9-13】 显示成绩表中所有选课学生的学号。

```
SELECT ALL 学号 FROM 成绩
```

执行结果中将有重复行出现。省略 ALL 的结果与此一致。

若命令改为：

```
SELECT DISTINCT 学号 FROM 成绩
```

执行结果中将排除所有重复的行。

【实例 9-14】 显示学生表中前 5 个学生的姓名。

```
SELECT TOP 5 姓名 FROM 学生
```

【实例 9-15】 查询成绩表中成绩在 60 分与 80 分之之间的学生的学号和成绩。

```
SELECT 学号,成绩 FROM 成绩 WHERE 成绩 BETWEEN 60 AND 80
```

该命令等价于：

```
SELECT 学号,成绩 FROM 成绩 WHERE 成绩>=60 AND 成绩<=80
```

【实例 9-16】 统计成绩表中最高分和最低分。

```
SELECT MAX(成绩) as 最大成绩,MIN(成绩) as 最小成绩 FROM 成绩
```

在分组统计中最常用的就是聚合函数，常用的聚合函数如表 9-4 所示。

表 9-4 常用的聚合函数

函数名	功能	函数名	功能
SUM	求某一系列值的总和	MAX	求某一系列值中的最大值
AVG	求某一系列的平均值	MIN	求某一系列值中的最小值
COUNT	计数		

【实例 9-17】 统计学生表中男生总人数。

```
SELECT COUNT(*) as 男生总人数 FROM 学生 WHERE 性别='男'
```

【实例 9-18】 统计成绩表中每门课程的平均成绩。

```
SELECT 课程号,AVG(成绩) as 平均成绩 FROM 成绩 GROUP BY 课程号
```

【实例 9-19】 统计学生表中男女生总人数。

```
SELECT 性别,COUNT(*) as 总人数 FROM 学生 GROUP BY 性别
```

带有 GROUP BY 子句的查询语句中，SELECT 子句的列名表中必须有用于分组的列名或是聚合函数。

【实例 9-20】 在成绩表中对成绩大于 65 的学生按学号分组，统计其选课数多于 1 且

平均成绩大于 75 分的学生的学号、选课号和平均成绩。

```
SELECT 学号,COUNT(课程号),AVG(成绩) FROM 成绩
WHERE 成绩>65 GROUP BY 学号
HAVING COUNT(课程号)>1 AND AVG(成绩)>75
```

执行时先过滤成绩小于等于 65 的行，然后按学号分组，再在分组中过滤掉选课数小于等于 1 或平均成绩小于等于 75 的行。

在既有 WHERE 子句，又有 HAVING 子句的分组查询中，其执行顺序为：WHERE 的条件在分组之前起作用，而 HAVING 的条件在分组形成后起作用，正因为如此，聚合函数常出现在 HAVING 条件中。

【实例 9-21】 查询按成绩表中成绩降序排列成绩表的所有信息。

```
SELECT * FROM 成绩 ORDER BY 成绩 DESC
```

知识要点分析

使用 SQL 语句还可以创建操作查询。创建操作查询的 SQL 语句有 INSERT（插入）、UPDATE（更新）和 DELETE（删除），它们的语法结构如下所示。

```
INSERT INTO <表名> [( <属性列 1> [, <属性列 2>] …… )]
VALUES ( <常量 1 > [, <常量 2 >] …… );
```

其功能是将新记录插入到指定表中。其中新记录属性列 1 的值为常量 1，属性列 2 的值为常量 2，……。INTO 子句中没有出现的属性列，新记录在这些列上将取空值。如果 INTO 子句中没有指明任何列名，则新插入的记录必须在每个属性列上均有值。

```
UPDATE <表名>
SET
<列名>=<表达式> [, <列名>=<表达式>] ……
[ WHERE <条件>];
```

其功能是修改指定表中满足 WHERE 子句条件的元组。其中 SET 子句给出<表达式>的值用于取代相应的属性列值。如果省略 WHERE 子句，则表示要修改表中的所有记录。

```
DELETE
FROM <表名>
[WHERE <条件>];
```

其功能是从指定表中删除满足 WHERE 子句条件的所有记录。如果省略 WHERE 子句，表示删除表中全部记录。

应用实例

【实例 9-22】 为“高校学生成绩管理系统”数据库中的“学生”表插入一个新记录。

```
INSERT
INTO 学生(学号,姓名,性别,年龄)
VALUES ('200903010','王刚','男',20)
```

【实例 9-23】 将“课程”表中“课程名”字段的“离散数学”改为“普通物理”。

```
UPDATE 课程
SET 课程名='普通物理'
WHERE 课程名='离散数学'
```

【实例 9-24】 删除“学生”表中“05 法学 1 班”的学生的记录。

```
DELETE
FROM 学生
WHERE 班级='05 法学 1 班'
```

几乎前面几节介绍的所有查询都有对应的 SQL 语句,用户只需用设计视图打开创建好的查询,然后在工具栏中的“视图”按钮的下拉列表中选择“SQL 视图”选项即可查看,这样可以学习到很多关于 SQL 语句的写法。如图 9-29 所示为选择查询的 SQL 设计视图。

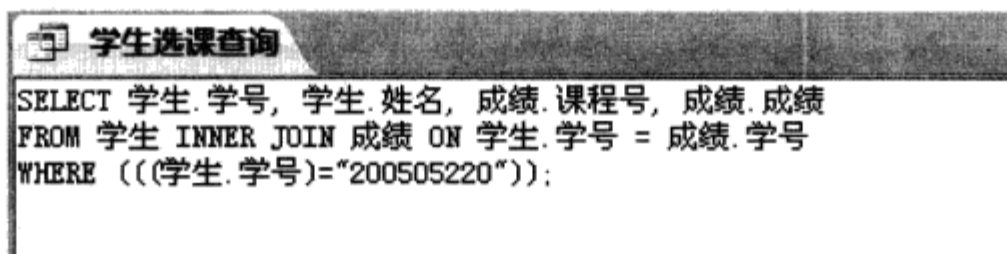


图 9-29 选择查询的 SQL 视图

如图 9-30 所示为交叉表查询的 SQL 设计视图。交叉表查询的 SQL 语法结构为:

```
TRANSFORM...GROUP BY...PIVOT...
```

其中,要统计的字段放在 TRANSFORM 后面,分组字段放在 GROUP BY 后面,决定显示为查询结果的数据表的标题字段放在 PIVOT 后面。

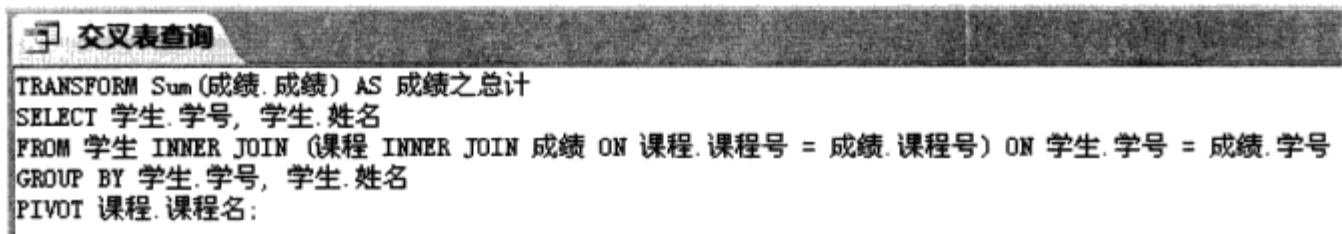


图 9-30 交叉表查询的 SQL 视图

如图 9-31 所示为生成表查询的 SQL 设计视图。生成表查询的 SQL 语法结构为“SELECT...INTO...FROM...”。

其中,将源表中要生成新表的字段放在 SELECT 后面,将生成的新表的名称放在 INTO 后面,将源表放在 FROM 后面。

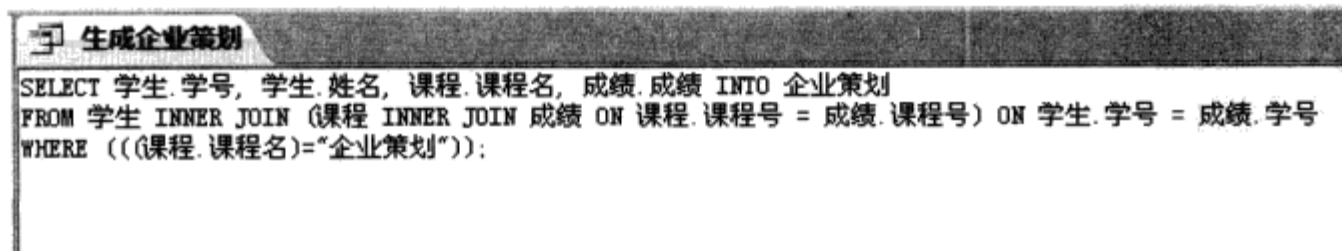


图 9-31 生成表查询的 SQL 视图

如图 9-32 所示为追加查询的 SQL 设计视图。追加查询的 SQL 语法结构为“INSERT INTO…SELECT…FROM…”。其用法为：将要追加查询的目标表放在 INSERT INTO 后面，并将追加的字段用括号括起来后紧跟着放在目标表后面，将要追加的字段放在 SELECT 后面，将源表放在 FROM 后面，表示选择要追加的字段的地方。

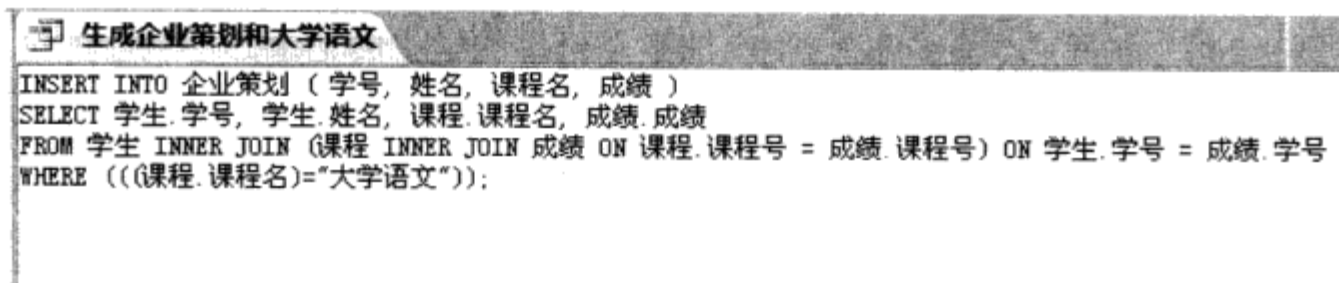


图 9-32 追加查询的 SQL 视图

如图 9-33 所示为删除查询的 SQL 设计视图。删除查询的 SQL 语法结构为“DELETE…FROM…”。将要删除数据的那些字段放在 DELETE 后面，将这些字段所在的数据表放在 FROM 后面。

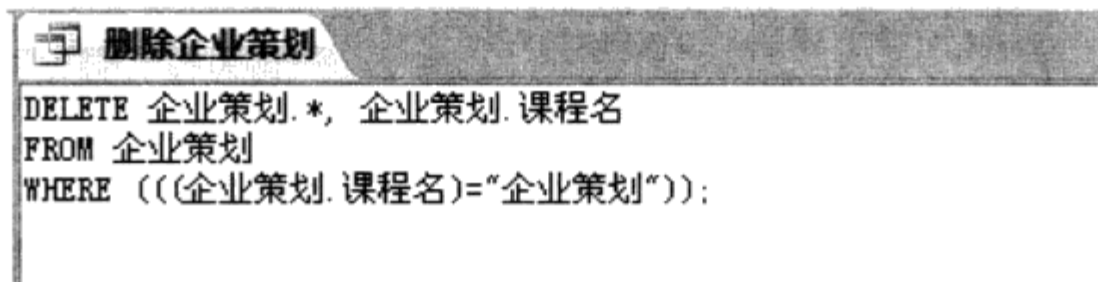


图 9-33 删除查询的 SQL 视图

如图 9-34 所示为更新查询的 SQL 设计视图。它与删除查询基本上相反，其语法结构为“UPDATE…SET…”。该结构将需要更新的数据表放在 UPDATE 后面，而把要更新数据表的具体内容放在 SET 后面。

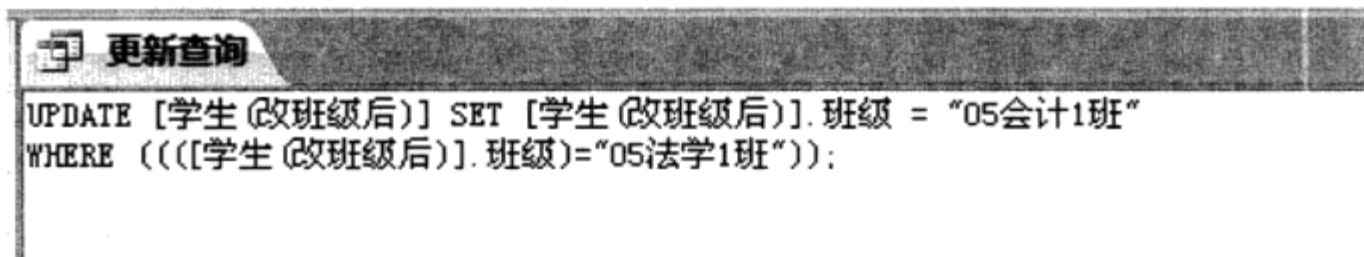


图 9-34 更新查询的 SQL 视图

用户在了解了 SQL 语言的一些简单用法后，就可以尽量少存储和保存对象，而把 SQL 语句用作记录源来简化数据库。

9.6.2 联合查询

☞ 知识要点分析

当用户要把两个或多个含有相同信息的独立表联合为一个表时，要用到联合查询。例

如，学生表中含有学号字段，而成绩表中也含有学号字段，这样就可以把这种具有相同的信息结合起来组成一个新表。

联合查询的语法结构为：

```
SELECT 相同字段 FROM 表 UNION SELECT 相同字段 FROM 表
```

这里每个 SELECT 语句所选取的字段个数必须相同，并以相同的顺序出项。相应的字段还必须具有兼容性数据。

用户可以为两个 SELECT 语句添加 WHERE 子句，以便对联合查询的结果进行限制。为了把返回的数据分组，用户也可以在每个 SELECT 语句中使用 GROUP BY 或 HAVING 子句。另外，还可以添加 ORDER BY 子句作为最后输入并排序合并的表。

如果用户要给某一字段取一个别名，则要在第一个 SELECT 语句中用 AS[别名]子句。

应用实例

【实例 9-25】 下面以学生成绩管理数据库为例来介绍一下用 SQL 语句创建一个联合查询步骤。

创建查询步骤如下：

(1) 在数据库窗口中，单击“创建”选项卡中的“其他”组中的“查询设计”按钮，“显示表”对话框同时打开。

(2) 不添加任何表，直接关闭“显示表”对话框。

(3) 单击“设计”选项卡中的“结果”组中的“SQL 视图”按钮，这时会显示编程设计视图。

(4) 选择“设计”选项卡中的“查询类型”组中的“联合”按钮，此时出现“联合查询”窗口。

(5) 在“联合查询”窗口的空白区域输入：“SELECT 姓名 FROM 学生 WHERE 学号="200506102" UNION SELECT 姓名 FROM 学生 WHERE 学号 in("200304210","200305145","200307108")”的 SQL 语句，如图 9-35 所示。

(6) 单击“运行”按钮，即可看到如图 9-36 所示的运行结果。

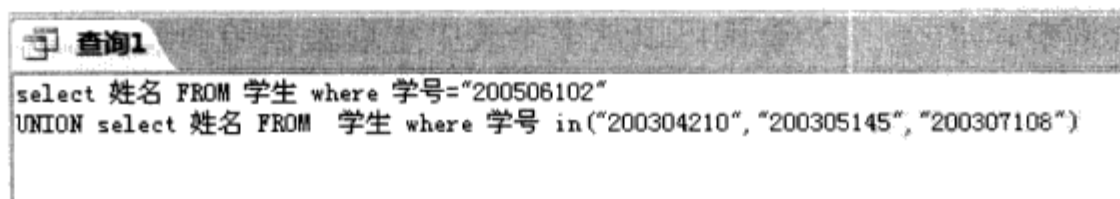


图 9-35 联合查询的 SQL 设计视图

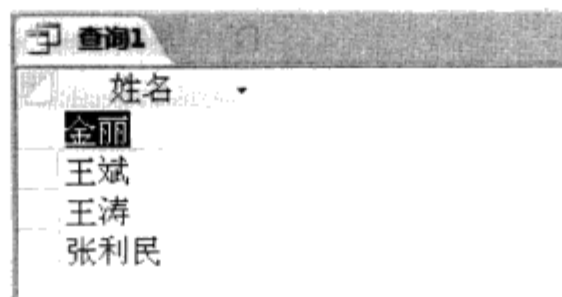


图 9-36 联合查询的结果集

SQL 查询还有“传递查询”、“数据定义查询”、“子查询”等类型，这些内容已超出本书范围，用户可以参阅相关书籍或帮助。

习 题 9

一、判断题

1. 按 Ctrl+Break 键可以中止查询的运行。()
2. 创建交叉表查询时，必须对行标题和值进行分组 (GROUP BY) 操作。()
3. 创建分组统计查询时，总计项应选择 ORDER BY。()
4. 交叉表查询中可以设置一个行标题、多个列标题和一个值。()
5. 在查询设计视图的设计网格中，不包括查询行。()

二、单选题

1. 在查询设计视图的设计网格中，不包括()行。
A. 字段 B. 条件 C. 显示 D. 查询
2. 要统计职工的人数，需在总计行使用()。
A. Sum B. Count C. Avg D. GROUP BY
3. 要查询 30 天以内参加工作的记录，查询条件为()。
A. <Date()-30 B. Between Date() And Date()-30
C. <Date()-31 D. >Date()-30
4. 要查询职务为销售人员或采购人员的查询条件为()。
A. “销售人员” Or “采购人员” B. “销售人员” And “采购人员”
C. “销售人员” 或 “采购人员” D. “销售人员” + “采购人员”
5. 要在查询设计网格中显示“追加”行，则要在查询类型下拉列表框中，选择()
查询。
A. 追加 B. 删除 C. 更新 D. 生成表
6. 下列查询中，() 查询可以从多个表中提取数据组合起来生成一个新表永久保存。
A. 追加 B. 删除 C. 更新 D. 生成表
7. 下列哪种方式不能生成新表。()
A. 生成表查询 B. Create table
C. 表对象下新建表 D. 更新查询
8. 在 Access 中，查询的数据源可以是()。
A. 表 B. 查询 C. 表和查询 D. 表、查询和报表
9. 下列不属于操作查询的是()。
A. 参数查询 B. 生成表查询 C. 删除查询 D. 更新查询
10. 假设某数据库中有一个简历字段，查询简历中最后两个字为“呼市”的记录准则

是 ()。

A. “呼市”

B. $\text{LEFT}([\text{简历}], 2) = \text{“呼市”}$

C. $\text{LIKE} \text{“呼市”}$

D. $\text{RIGHT}([\text{简历}], 2) = \text{“呼市”}$

三、上机练习

利用第8章上机练习创建的“图书资料管理”数据库建立以下查询。

1. 建立“查询1”，查询书籍编号、书籍名称、学生姓名、借书日期和还书日期信息。
2. 建立“查询2”，查询2009年1月1日以后的学生学号、姓名和所借书名。

第10章

设计窗体、报表及模块

窗体作为 Access 2007 数据库的重要组成部分，是联系数据库与用户的桥梁。在实际应用中，经常需要将数据表、查询表中的数据打印出来，这就需要用报表来实现。本章主要学习如何来创建窗体、报表及模块。

10.1 创建窗体

10.1.1 窗体的功能

知识要点分析

窗体和报表都可用于数据库中数据的维护，但两者的目的不同。窗体主要用于数据输入，报表则用于在屏幕或打印输出的窗体中查阅数据。

具体来说，窗体具有以下功能。

1. 数据的显示与编辑

窗体的最基本功能是显示与编辑数据，窗体可以显示来自多个数据表中的数据。用户利用窗体对数据库中的相关数据进行添加、删除、修改以及设置数据的属性等各种操作。

一般地，用一个窗口（运行以后的窗体）显示一条记录（也可以显示多条记录）。使用窗口上提供的移动按钮、滚动条等控件，可以直观地翻看数据库中的任何记录或者记录中的任何字段。

2. 数据输入

窗体经常被用来创建一个填充数据的窗口，作为数据库中数据输入的接口。在这种情况下，窗体利用表或查询作为自身的数据源。窗口的数据输入功能也正是它与报表功能的主要区别。

一个设计优良的窗体能使数据输入更加方便、准确。例如，当数据库中的表比较复杂时，期望所有的数据库用户都能有效地利用数据表视图中的每个表格来输入数据是不大可能的。应该创建一个窗体来从众多的表中选出相关的表，显示希望用户看到的内容。通过仔细地安排输入数据的位置和提供解释性的文字，就能指引用户完成数据输入操作以及其他操作。

3. 应用程序流控制

在使用流行的软件开发工具，如 Visual Basic 等所开发的应用程序中，窗体是重要的组成部分。一般地，窗体提供程序和用户之间信息交互的界面及一些简单的操作任务，而实际的工作主要由程序代码来完成。Access 中窗体也可以与函数、子程序这样的程序代码段相结合。在每个窗体中，都可以使用 VBA 来编写代码，并利用代码执行相应的功能。

4. 信息显示和数据打印

在窗体中，可采取灵活多样的形式显示一些警告或解释的信息，另外，窗体也可以用来打印数据库中的数据。

窗体有 6 种视图：设计视图、窗体视图、数据表视图、布局视图、数据透视表视图和数据透视图视图。

创建窗体的方法包括：

- 对表自动创建窗体；
- 使用向导创建窗体；
- 在窗体的设计窗口中手工创建窗体。

多数窗体都与数据库中的一个或多个表或查询绑定。窗体的记录源引用数据表和查询中的字段。窗体无须包含每个数据表或查询中的所有字段。

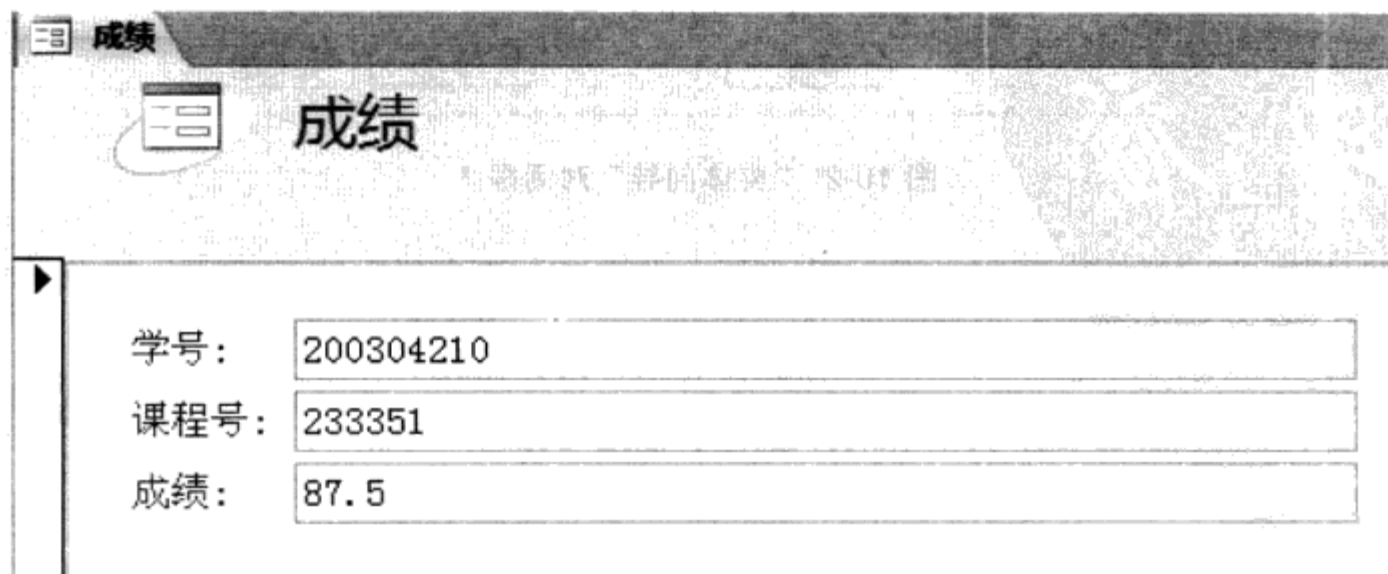
10.1.2 对表自动创建窗体

应用实例

【实例 10-1】 打开“高校学生成绩管理系统”数据库，然后对“成绩”表自动生成窗体。设计步骤如下：

(1) 在数据库窗口中选中要生成窗体的“成绩”表。

(2) 单击“创建”选项卡中的“窗体”组中的“窗体”按钮，即可生成窗体，如图 10-1 所示。



成绩	
学号:	200304210
课程号:	233351
成绩:	87.5

图 10-1 自动生成窗体

10.1.3 使用窗体向导创建窗体

☞ 知识要点分析

尽管使用自动生成窗体来创建窗体的速度很快，但该方法对窗体内容或外观的选择余地不大，也不能满足用户的进一步需求。可以使用窗体向导来创建格式更为丰富的窗体。

打开窗体向导的方式如下：选择“创建”选项卡中的“窗体”组中的“其他窗体”的下拉列表中的“窗体向导”即可。

✎ 应用实例

【实例 10-2】 打开“高校学生成绩管理系统”数据库，利用窗体向导给“学生”表创建一个窗体。

设计步骤如下：

(1) 打开窗体向导，在“表/查询”下拉列表框中选择“学生”表，并在下面的可用字段列表框中双击需要的字段，如图 10-2 所示。

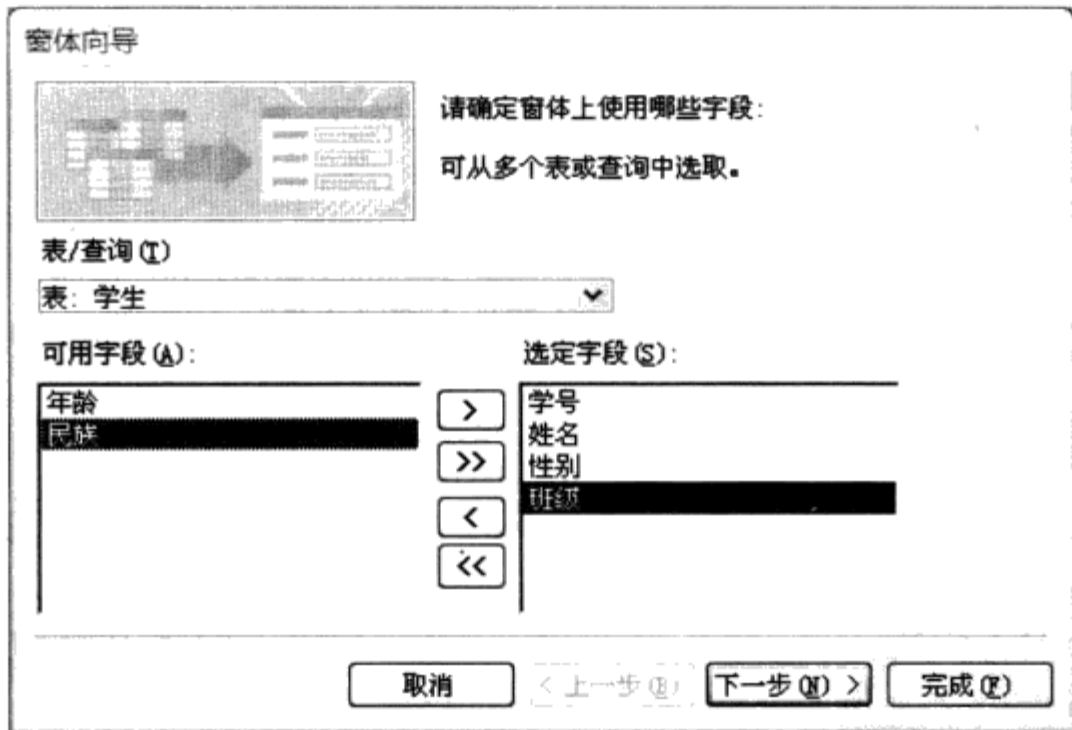


图 10-2 “窗体向导”对话框 1

(2) 单击“下一步”按钮，向导提示在“纵栏表”、“表格”、“数据表”等几种窗体布局中选择一种，选中某个单选按钮即可。本例中选“纵栏表”，如图 10-3 所示。

(3) 单击“下一步”按钮，接下来要选择一种窗体样式，窗体样式包括“城市”、“市镇”、“平衡”、“广场”、“技术”等，在本例中选择“活力”。

(4) 单击“下一步”按钮，在接下来的对话框中选择窗体的标题，输入“学生信息”作为窗体的名称。

(5) 单击“完成”按钮，则“窗体向导”按照上面提供的信息创建纵栏表式的窗体，如图 10-4 所示。

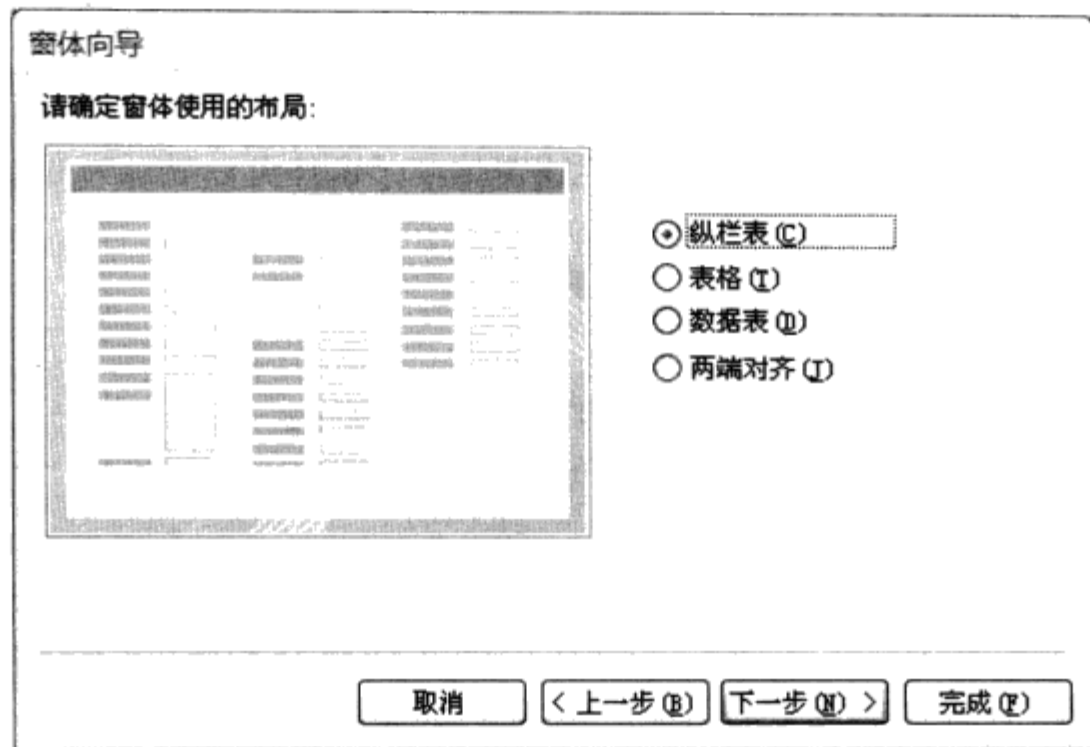


图 10-3 “窗体向导”对话框 2

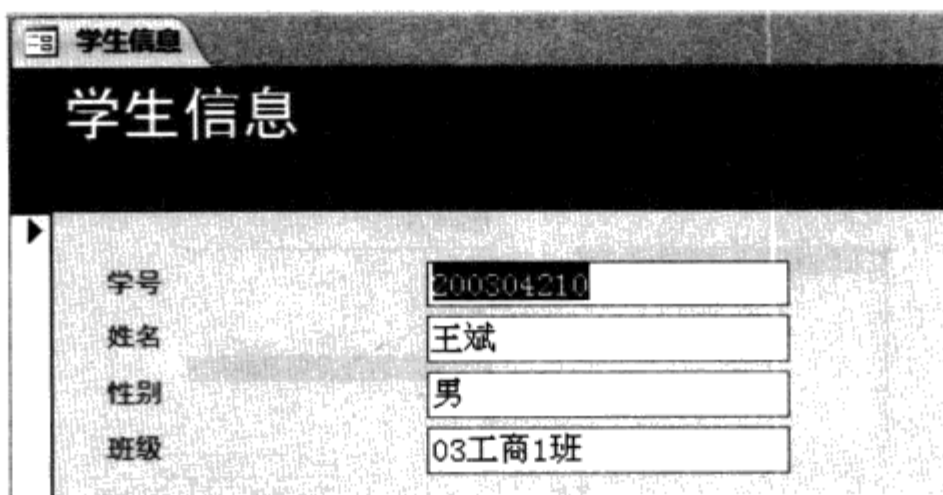


图 10-4 使用向导生成的窗体

10.1.4 创建带有子窗体的窗体

☞ 知识要点分析

子窗体是指在一个窗体中插入的窗体。将多个窗体合并时，其中一个窗体作为主窗体，其余作为子窗体。主窗体和子窗体一般有三种关系。

- 主窗体中多个子窗体的数据来自不相关的记录源。在这种情况下，主窗体只是作为多个子窗体的集合。
- 主窗体和子窗体数据来自相同的数据源。
- 主窗体和子窗体数据来自相关的数据源。

当子窗体只显示与主窗体相关的记录时，意味着主窗体和子窗体是同步的。要实现同步，作为窗体基础的表或查询与子窗体的基础表或查询之间必须是一对多关系。作为主窗体基础的表必须是一对多关系的“一”端，而作为子窗体基础的表必须是一对多关系中的

“多”端。在向主窗体中添加子窗体之前，应首先创建此一对多关系。

使用向导就可以非常方便地创建带有子窗体的窗体，这样，可以在一个窗体中同时查看多个表的数据。这时的主窗体只能是纵栏式窗体，子窗体可以显示为数据表窗体。并且在子窗体中，还可以创建二级子窗体。

应用实例

【实例 10-3】 打开“高校学生成绩管理系统”数据库，使用“学生”和“成绩”表，借助于向导创建一个包括“学生成绩”子窗体的窗体。要求包括“学号”、“姓名”、“性别”、“课程号”、“成绩”字段。

设计步骤如下：

(1) 打开窗体向导，在“表/查询”下拉列表框中选择“学生”表，并在下面选择“学号”、“姓名”、“性别”字段；然后再选择“成绩”表中的“课程号”、“成绩”字段，如图 10-5 所示。

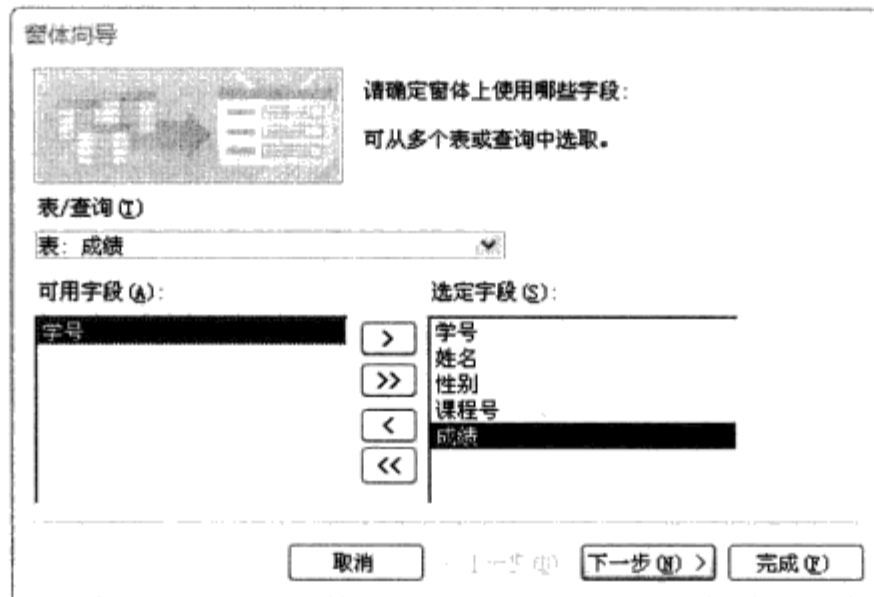


图 10-5 选择字段

(2) 单击“下一步”按钮，在接下来的对话框中选择查看数据的方式为“通过学生”，并选中“带有子窗体的窗体”选项，如图 10-6 所示。

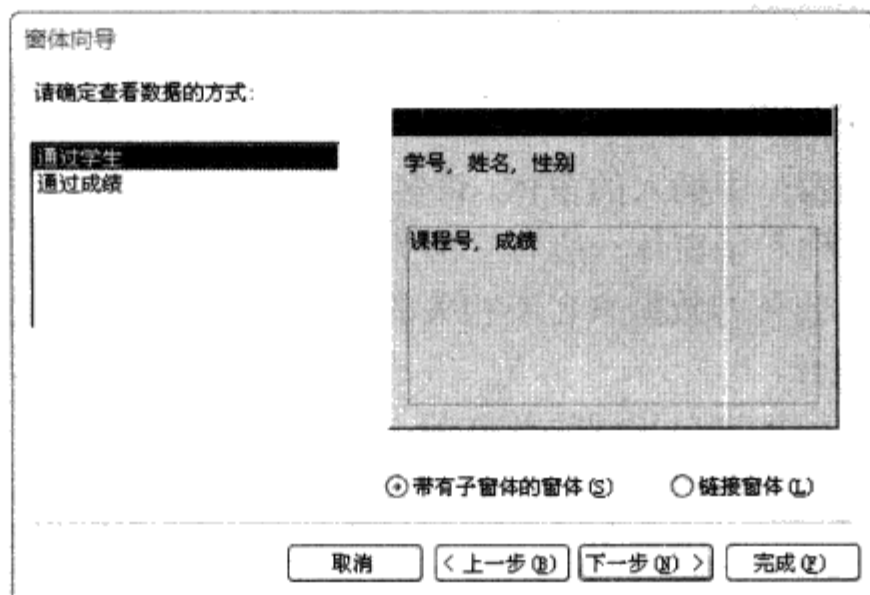


图 10-6 选择查看方式

(3) 单击“下一步”按钮，在接下来的对话框中选择子窗体的布局，本例选“数据表”式子窗体。

(4) 单击“下一步”按钮，在接下来的对话框中选择窗体的样式为“丰富”。

(5) 单击“下一步”按钮，在接下来的对话框中输入窗体的标题为“学生信息与成绩”。

(6) 单击“完成”按钮，创建如图 10-7 所示的带有子窗体的“学生信息与成绩”窗体。

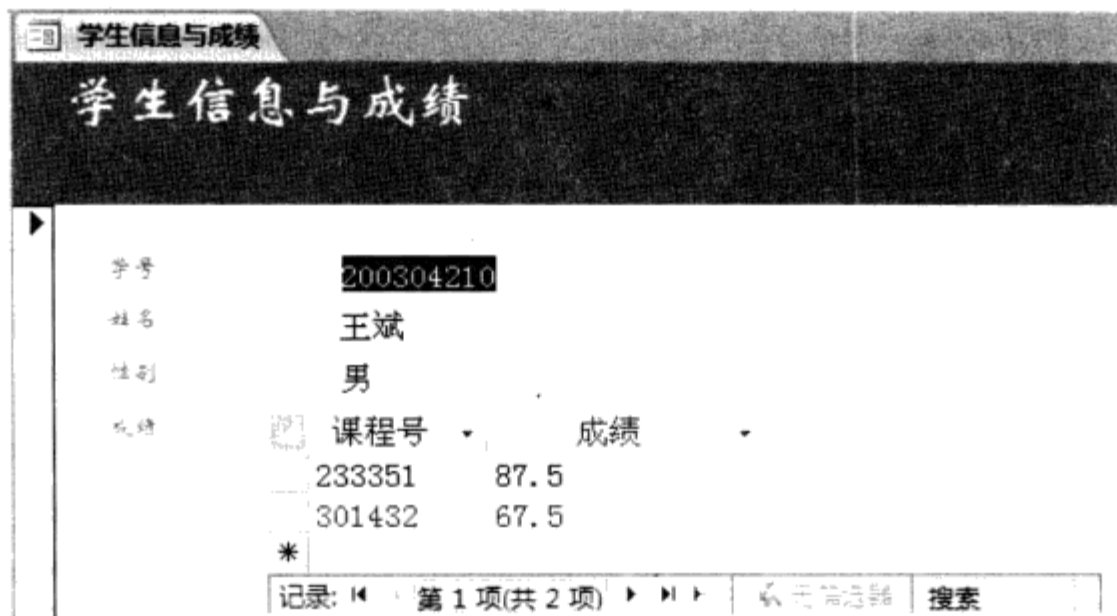


图 10-7 创建的带有子窗体的窗体

10.1.5 使用设计视图创建窗体

在许多情况下，无论是格式还是内容，使用自动创建窗体或窗体向导所生成的窗体都不能满足要求，因此，需要使用设计视图从无到有地创建窗体。

☞ 知识要点分析

窗体的“设计视图”与表、查询等的设计视图窗口的功能相同，也是用来创建和修改设计对象（窗体）的窗口，但其形式与表、查询等的设计视图差别很大；窗体的“窗体视图”是能够同时输入、修改和查看完整的记录数据的窗口，可显示图片、其他 OLE 对象、命令按钮以及其他控件；窗体的“数据表视图”以行列方式显示表、窗体或查询中的数据，可用于编辑字段、添加和删除数据以及查找数据。

1. 窗体的设计窗口如图 10-8 所示

1) 窗体页眉和页脚

窗体页眉用来显示窗体标题等信息，出现在整个窗体的顶部。窗体页脚用来显示窗体说明等信息，出现在整个窗体的底部。

2) 页面页眉和页脚

页面页眉和页脚的作用是用来显示说明、日期和页码等信息，但其中的信息只有在设计视图和打印后才能显示出来，而不会在窗体视图中显示。

3) 主体

窗体的核心部分，用于显示记录，以及用户添加的控件等。

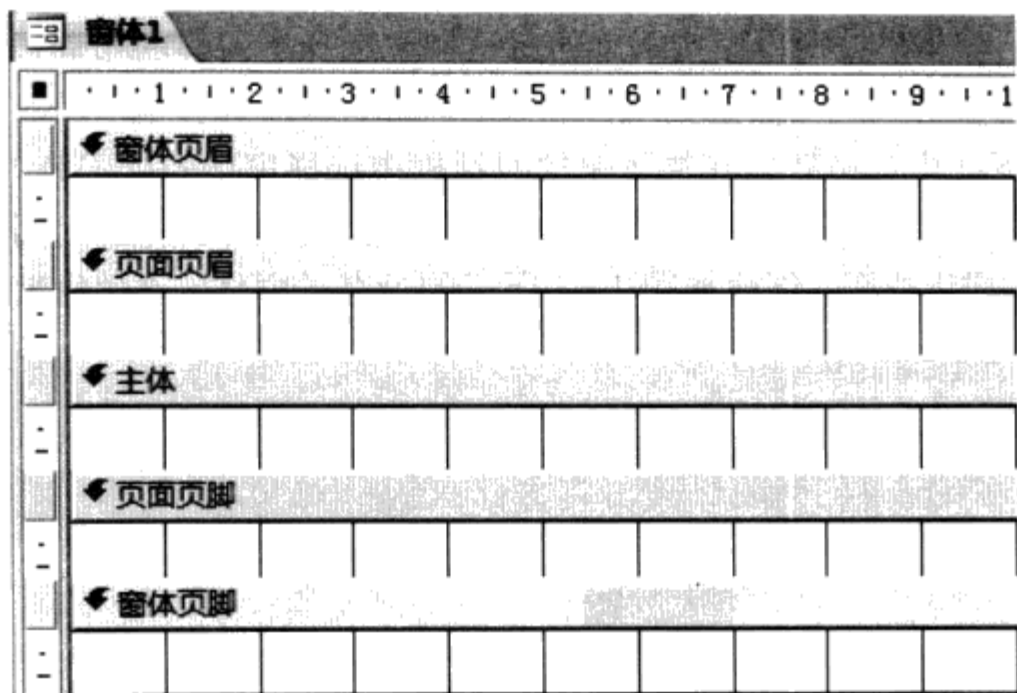


图 10-8 窗体的设计视图窗口

4) 控件

控件是窗体设计的重要组件。控件有不同的类型，其目的是提供各类控件的功能。在工具箱中提供了对窗体设计的各种控件。

窗体中的每个部分都可以放置控件，但在窗体中较少使用页面页眉及页脚，它们常用于报表。

2. 控件

在 Access 2007 中，常用控件放在“窗体设计工具”下面的“设计”选项卡中的“控件”组中，如图 10-9 所示。可以将这些控件拖到窗体中，成为窗体中的新对象。这些控件的名称及作用已经在前面给大家介绍过了，在这里就不做解释了。



图 10-9 常用控件

应用实例

【实例 10-4】 打开“高校学生成绩管理系统”数据库，利用窗体设计视图，创建一个名为“学生信息窗体”的窗体。

设计步骤如下：

(1) 打开数据库，单击“创建”选项卡中的“窗体”组中的“窗体设计”按钮。此时，在窗体的“设计”视图中创建了一个只有主体节的空白窗体。在“设计”视图中，窗体顶部和左侧都有标尺，而且窗体上显示着网格线。

(2) 选择“窗体设计工具”下面的“设计”选项卡中的“工具”组中的“添加现有字

段”按钮，就会打开字段列表。选择“学生”表中的“学号”字段，并将其拖动至设计区的相应位置。用同样的方法将其余所需的绑定控件添加到设计区，如图 10-10 所示。

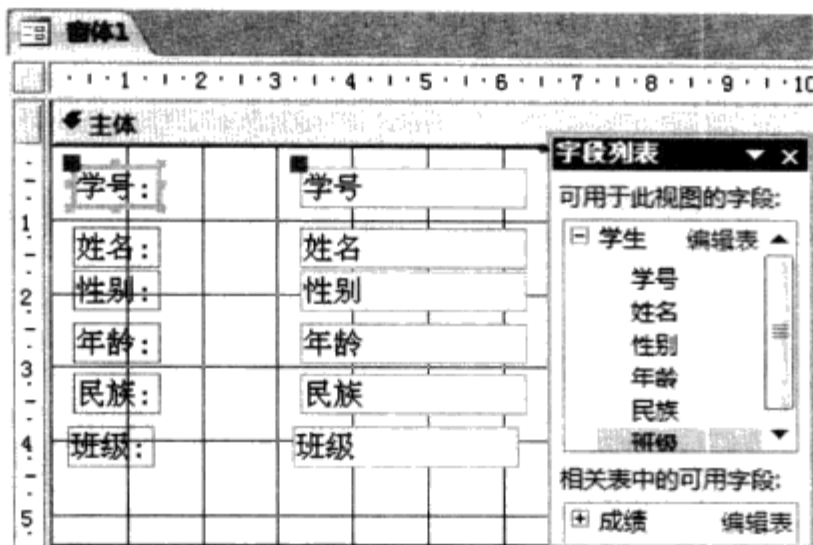


图 10-10 向设计工作区添加字段

还可以在窗体上添加非绑定型控件、创建计算型控件。添加非绑定型控件的方法是：单击某个控件，再在窗体上拖动画出相应控件。创建计算型控件的方法是：如果控件是文本框，则可直接在控件中输入计算表达式。对于各类控件（包括文本框）都可直接使用表达式生成器来创建。选中控件，右击鼠标，在弹出的快捷菜单中选择“事件生成器”下面的“表达式生成器”命令，就会弹出“表达式生成器”对话框，再单击“=”按钮，然后单击使用计算的相应按钮。在这里就不再举例说明了。

(3) 单击“窗体设计工具”下面的“排列”选项卡中的“显示/隐藏”组中的“窗体页眉/页脚”按钮和“页面页眉/页脚”按钮，即可为窗体添加窗体页眉、窗体页脚、页面页眉和页面页脚这 4 节。

(4) 在“窗体页眉”节里添加标签控件，在其中输入“学生信息窗体”，然后用适当的字号进行格式化，如图 10-11 所示。



图 10-11 添加窗体页眉

(5) 用户还可以在页眉中添加图像以增加窗体的美观。方法是单击“图像”控件，然后在“窗体页眉”的相应位置拖拉一个矩形框，这时会出现一个“插入图像”对话框。选中所需图片，然后单击“确定”按钮即可，如图 10-12 所示。



图 10-12 添加图像

(6) 单击工具栏中的“窗体视图”按钮，查看效果，满意后保存退出，如图 10-13 所示。

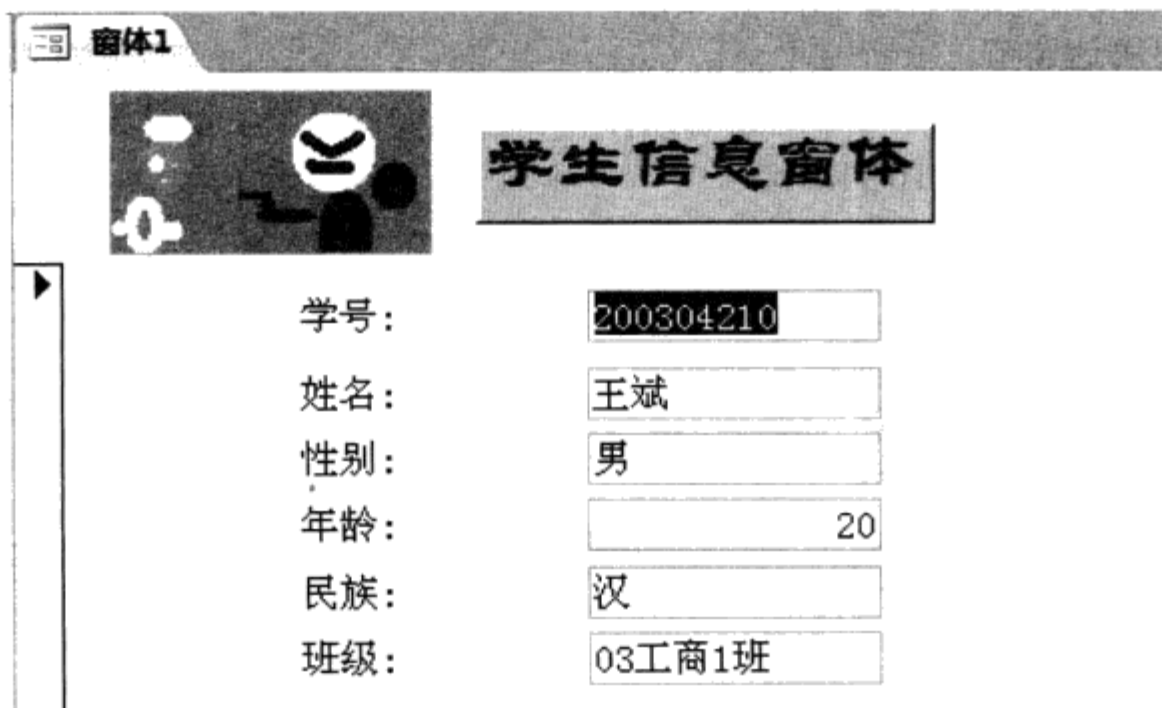


图 10-13 设计好的窗体

10.1.6 创建控制面板窗体

在设计视图窗体中，借助于控件按钮，可以为窗体添加各种控件等，以便对表、查询、

窗体、报表等对象进行快速操作。

应用实例

【实例 10-5】 打开“高校学生成绩管理系统”数据库，创建控制面板窗体，要求在窗体中添加可打开“学生信息”窗体的命令按钮。

设计步骤如下：

(1) 打开数据库，选择“创建”选项卡中的“窗体”组中的“窗体设计”按钮。此时，在窗体的“设计”视图中创建了一个只有主体节的空白窗体。

(2) 单击“命令按钮”控件，并在主体区单击鼠标，绘制一个命令按钮，这时会打开“命令按钮向导”对话框。

(3) 在“类别”对话框列表框中选择命令按钮所对应的操作类别为“窗体操作”，并在“操作”列表框中选择操作的动作为“打开窗体”，如图 10-14 所示。

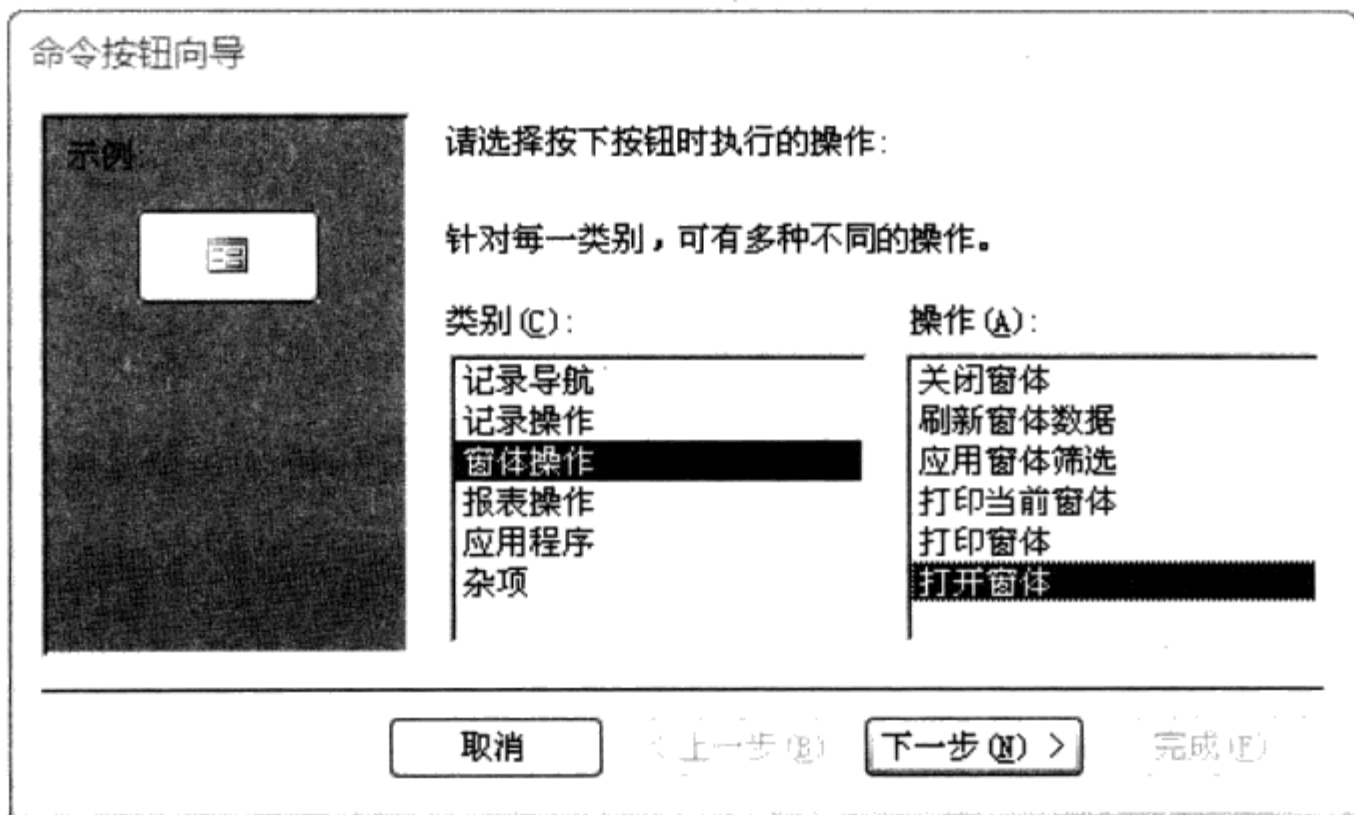


图 10-14 命令按钮向导对话框

(4) 单击“下一步”按钮，在接下来的对话框中选择要打开的窗体，本例中选择“学生信息”窗体作为要打开的窗体。

(5) 单击“下一步”按钮，在接下来的对话框中使用默认设置（打开窗体并显示所有记录）。

(6) 单击“下一步”按钮，在接下来的对话框中选择命令按钮上显示的内容，如图 10-15 所示。

(7) 单击“下一步”按钮，在接下来的对话框中定义命令按钮的名称，可使用默认名称（Command0），此名称常被用到编程代码中。

(8) 单击“完成”按钮，在窗体添加一个名称为“打开学生信息窗体”的命令按钮。

(9) 将窗体保存为“控制面板”后，切换到窗体视图，效果如图 10-16 所示。

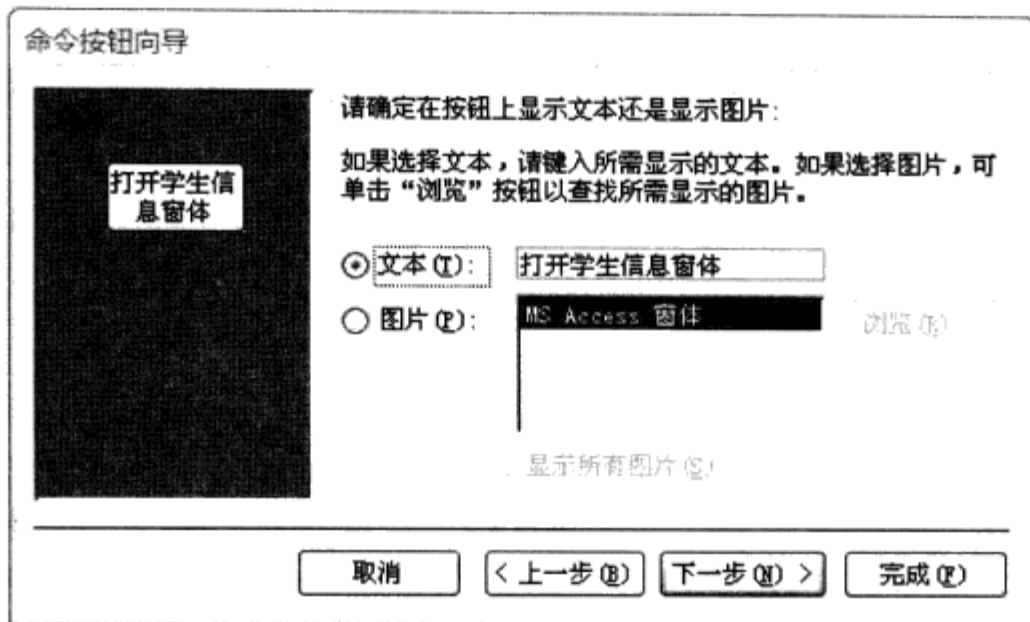


图 10-15 设置按钮内容

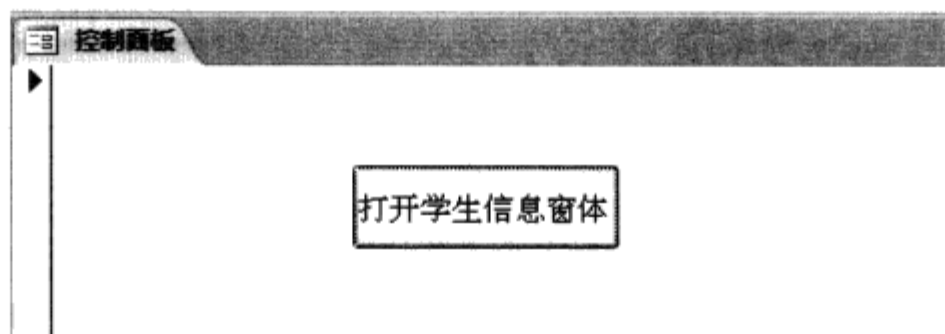


图 10-16 窗体视图效果

(10) 单击“打开学生信息窗体”命令按钮，可以打开对应的“学生信息窗体”，如图 10-13 所示。

10.2 创建报表

10.2.1 报表概述

☎ 知识要点分析

报表用于对数据库中的数据进行计算、分组、汇总和打印，如果希望对表、查询或窗体中的数据进行计算、分析和汇总，并按照指定的格式打印出来，则可以借助于报表来完成。

1. 报表的视图

每个报表均有下列 4 种视图：设计视图、打印预览视图、报表视图和布局视图。其中，设计视图用于创建报表或更改已有报表的结构；打印预览视图用于查看将在报表的每一页上显示的数据。

2. 报表的组成

在设计视图中打开报表时可以看到，报表由报表页眉、页面页眉、主体、页面页脚以及报表页脚等部分组成，每一部分称为一个节。每个节在页面上和报表中具有特定的目的

并按照预期次序打印。

(1) 报表页眉节：用于在报表的开头放置信息，例如标题文字、打印日期或报表说明等。

(2) 页面页眉节：用于在报表的上方放置信息，出现在每一页的上方。

(3) 主体节：用于包含报表的主体内容，可以在报表的主体节中放置控件，以显示数据。

(4) 页面页脚节：用于在报表页面的下方放置信息，出现在每一页的下方。

(5) 报表页脚节：用于在报表页面底部放置信息，例如报表总结、总计数或打印日期等。

10.2.2 自动生成报表

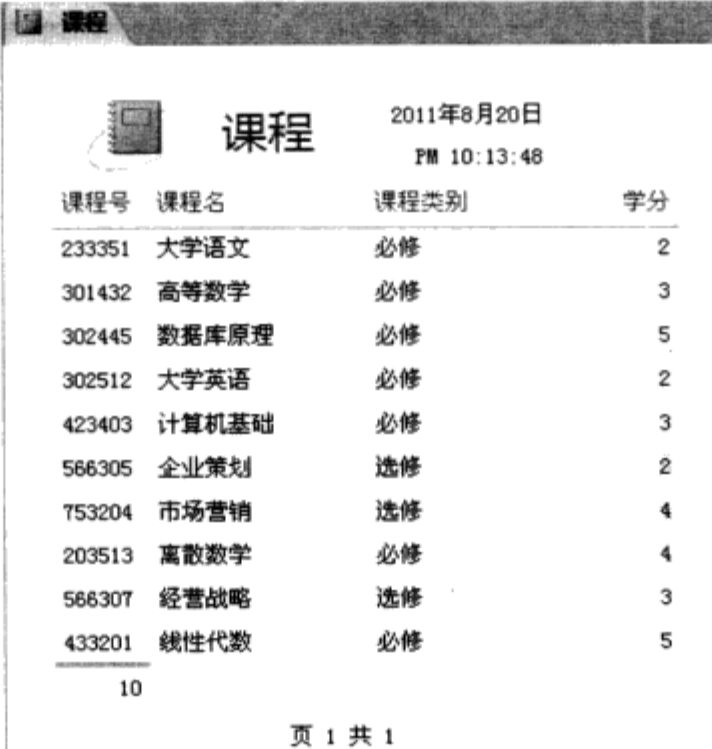
应用实例

【实例 10-6】 打开“高校学生成绩管理系统”数据库，然后对“课程”表自动生成报表。

设计步骤如下：

(1) 在数据库窗口中选中要生成窗体的“课程”表。

(2) 单击“创建”选项卡中的“报表”组中的“报表”按钮，即可生成窗体，如图 10-17 所示。



The screenshot shows a report window titled '课程' (Courses) with a date and time stamp '2011年8月20日 PM 10:13:48'. The report contains a table with the following data:

课程号	课程名	课程类别	学分
233351	大学语文	必修	2
301432	高等数学	必修	3
302445	数据库原理	必修	5
302512	大学英语	必修	2
423403	计算机基础	必修	3
566305	企业策划	选修	2
753204	市场营销	选修	4
203513	离散数学	必修	4
566307	经营战略	选修	3
433201	线性代数	必修	5

At the bottom of the report, it says '10' and '页 1 共 1'.

图 10-17 自动生成报表

10.2.3 使用向导创建报表

创建报表最简单的方法莫过于使用 Access 2007 提供的向导了。报表向导提供了一种

灵活的创建报表的方法,利用报表向导,用户只需回答一系列创建报表的问题,Access 2007 就可以根据用户做出的选择逐步创建所需的报表。

应用实例

【实例 10-7】 打开“高校学生成绩管理系统”数据库,利用报表向导给“成绩”表创建一个报表。

设计步骤如下:

(1) 打开报表向导,在“表/查询”下拉列表框中选择“成绩”表,并在下面的可用字段列表框中双击需要的字段,如图 10-18 所示。

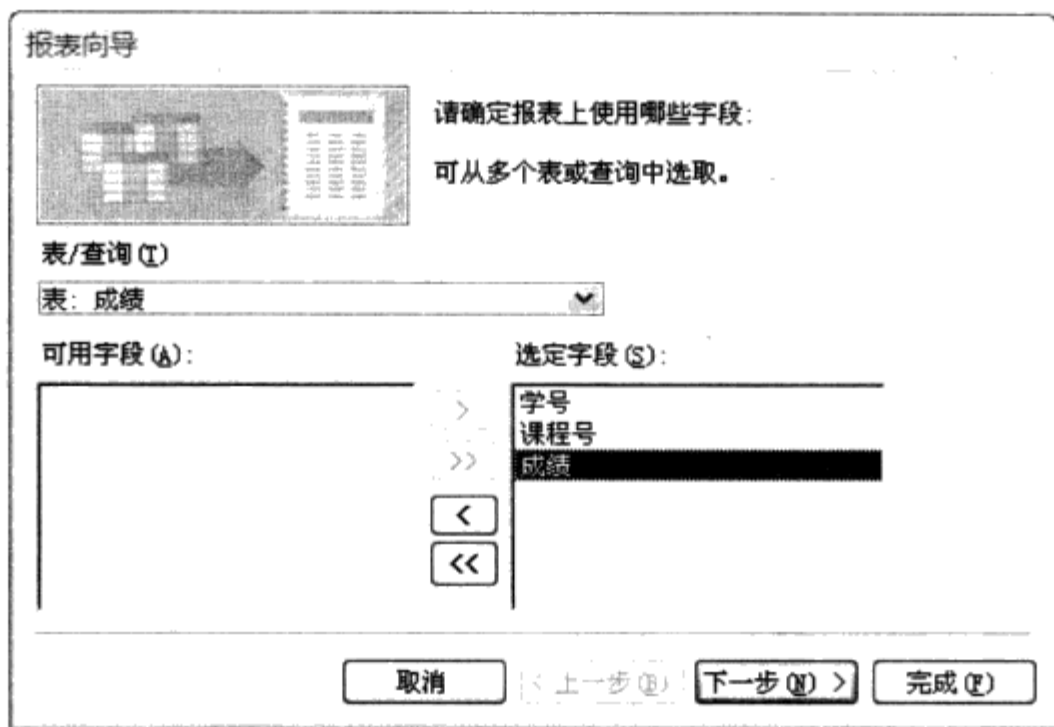


图 10-18 选定字段

(2) 单击“下一步”按钮,在接下来的对话框中,将询问用户是否添加分组级别。用户通过数据分组,可以将某些具有相同属性的记录作为一组进行显示,同时还可以进行数据汇总,本例选“学号”字段作为分组字段,如图 10-19 所示。

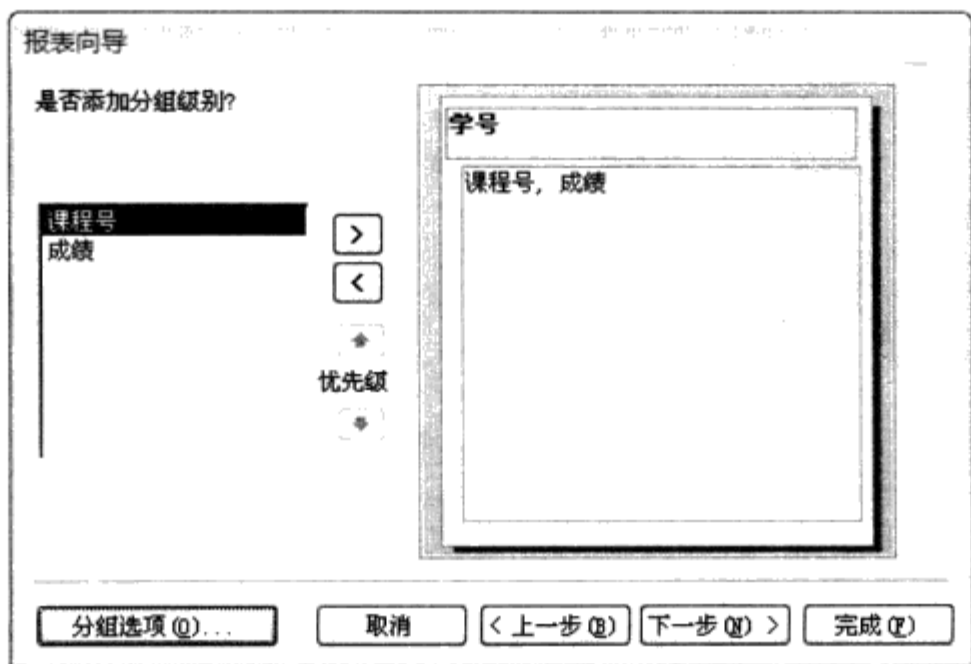


图 10-19 设置分组

(3) 单击“下一步”按钮，在接下来的对话框中，用户可以指定每个组内字段的排序顺序，一次最多对 4 个字段进行排序，单击列表框右侧的按钮，即可选定按升序或降序排列。本例选择按“课程号”进行排序。单击“汇总选项”按钮，即可从“平均值”、“总计值”、“最小值”和“最大值”中选择对结果集进行以上的哪种汇总，如图 10-20 所示。

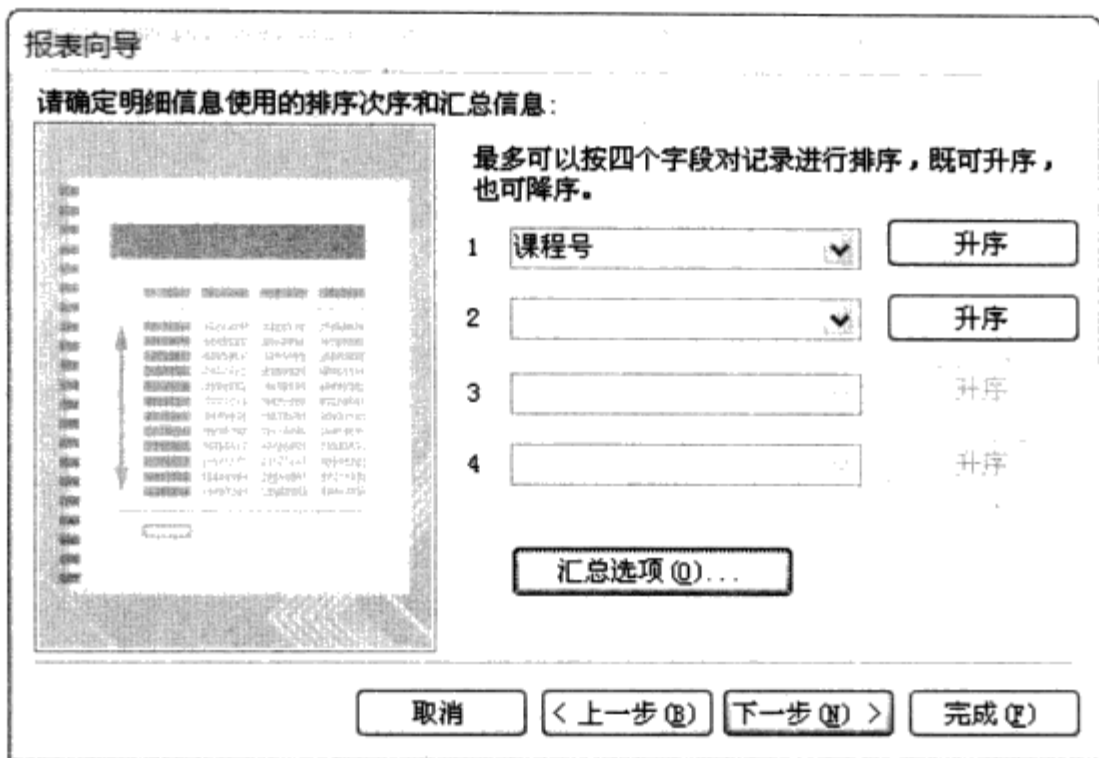


图 10-20 设置排序

(4) 单击“下一步”按钮，在接下来的对话框中，用户可以选择报表的样式布局。在左侧的预览框内进行预览，在“方向”选项组中可以确定报表方向是横向或纵向，如图 10-21 所示。

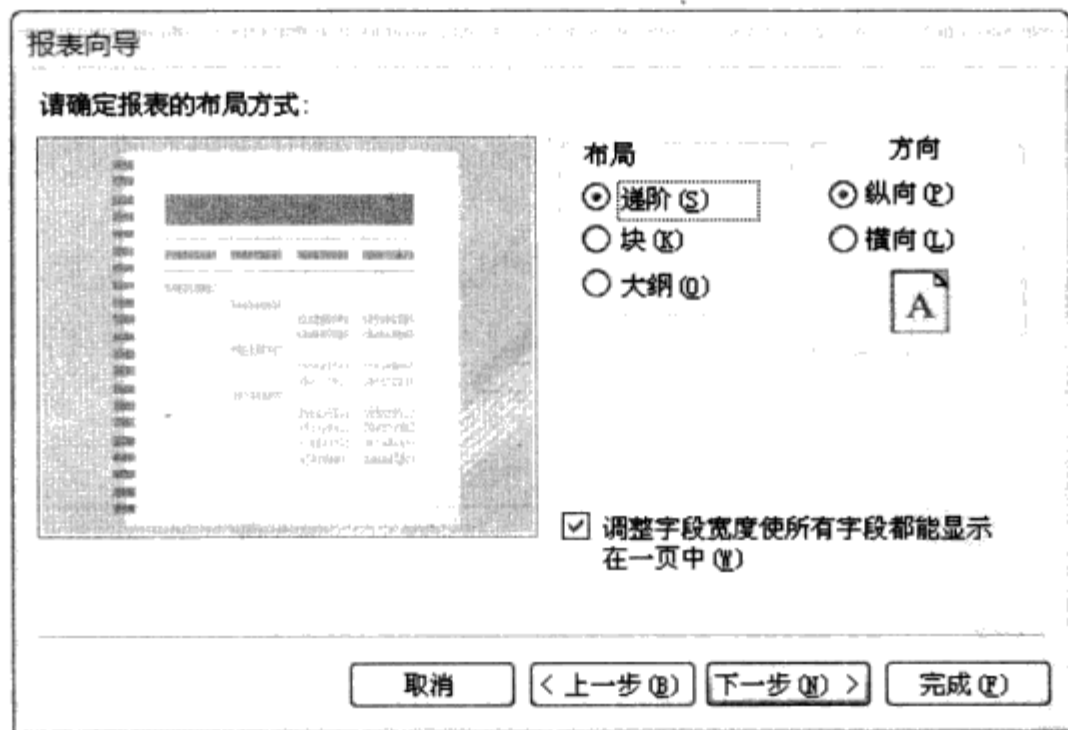


图 10-21 设置报表布局

(5) 单击“下一步”按钮，在接下来的对话框中，可以选择所希望的报表样式。报表样式有“原点”、“地铁”、“城市”等样式，本例选择“模块”样式。

(6) 单击“下一步”按钮，弹出报表向导的最后一个对话框，在此对话框中输入报表的标题“成绩统计”。最后单击“完成”按钮，创建一个如图 10-22 所示的“成绩统计”报表。

学号	课程号	成绩
200304210		
	233351	87.5
	301432	67.5
200307108		
	423403	91
	566305	87
200412103		
	566305	79.5

图 10-22 “成绩统计”报表

10.2.4 使用报表设计视图创建报表

☞ 知识要点分析

利用向导创建的报表可以在设计视图窗口中对其进行修改。另外，一些个性化的报表，必须在设计视图窗口中手工创建。

控件在前面章节已经介绍了，这里就不再多述，它的用法和在设计视图中创建窗体的方法相同。

在报表设计视图中可以通过使用函数，在报表页眉/页脚、页面页眉/页脚或组页眉/页脚中添加日期、页码、计算求和、平均值等。这些函数可通过文本框添加到报表中。表 10-1 为在报表中经常使用的函数及表达式。

表 10-1 报表常用函数和表达式

函数和表达式	结果
= [page]	1、2、3
= "第"& [page] & "页"	第 1 页、第 2 页、第 3 页
= [page] & "/" & [page]	1/5、2/5、3/5、4/5、5/5
= Format([page], "000")	001、002、003
= [数量]*[单价]	数量和单价字段的乘积
= Date()	当前的日期
= Now()	当前的日期和时间
= Sum[数量]	求总数量

应用实例

【实例 10-8】 打开“高校学生成绩管理系统”数据库，利用报表设计视图，创建一个名为“学生信息报表”的报表。

设计步骤如下：

(1) 打开数据库，单击“创建”选项卡中的“报表”组中的“报表设计”按钮。此时，在报表的“设计”视图中创建了一个只有主体节的空白报表。在“设计”视图中，报表顶部和左侧都有标尺，而且报表上显示着网格线。

(2) 选择“报表设计工具”下面的“设计”选项卡中的“工具”组中的“添加现有字段”按钮，就会打开字段列表。选择“学生”表中的“学号”字段，并将其拖动至设计区的相应位置。用同样的方法将其余所需的绑定控件添加到设计区，如图 10-23 所示。



图 10-23 添加控件后的报表设计视图

(3) 添加报表页眉和页脚节，单击“直线”控件，在“页面页眉”中画出一条直线，并对其适当格式化。然后在直线上方再添加一个标签控件，并输入“学生信息报表”字样作为报表的标题，可以对其进行格式化。此标签控件是一个非绑定控件，而前面的控件均是绑定控件，如图 10-24 所示。

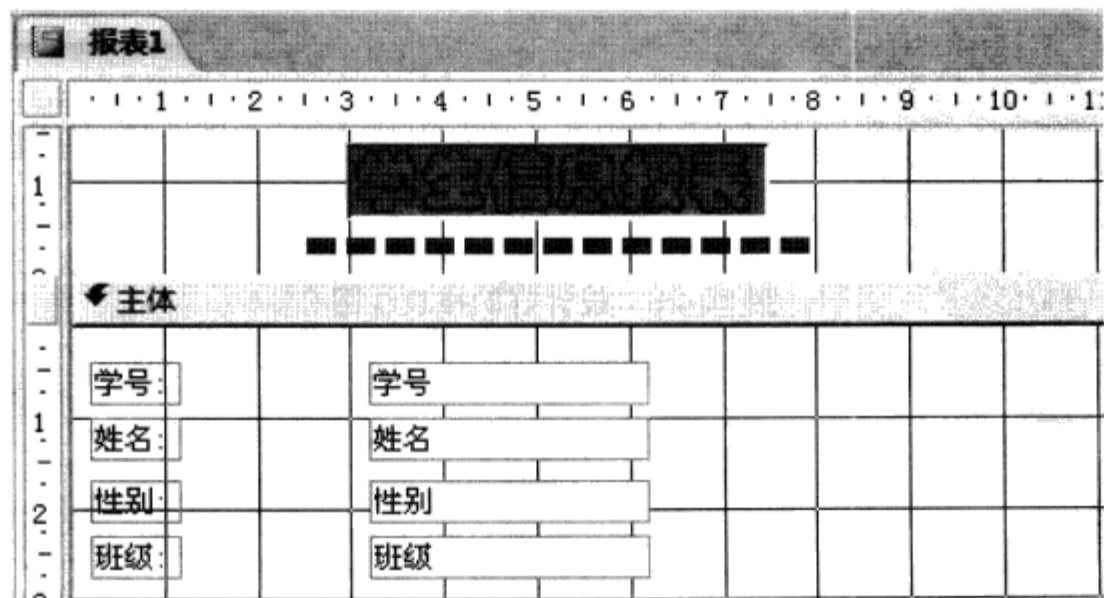


图 10-24 在页眉上添加控件后的视图

(4) 在“页面页脚”中添加一个文本框控件，然后打开其属性对话框，在“数据”选

项卡的“控件来源”中填入“=[“第”]&[page]&[“页”]”，用以标识该页是第几页，如图 10-25 所示。

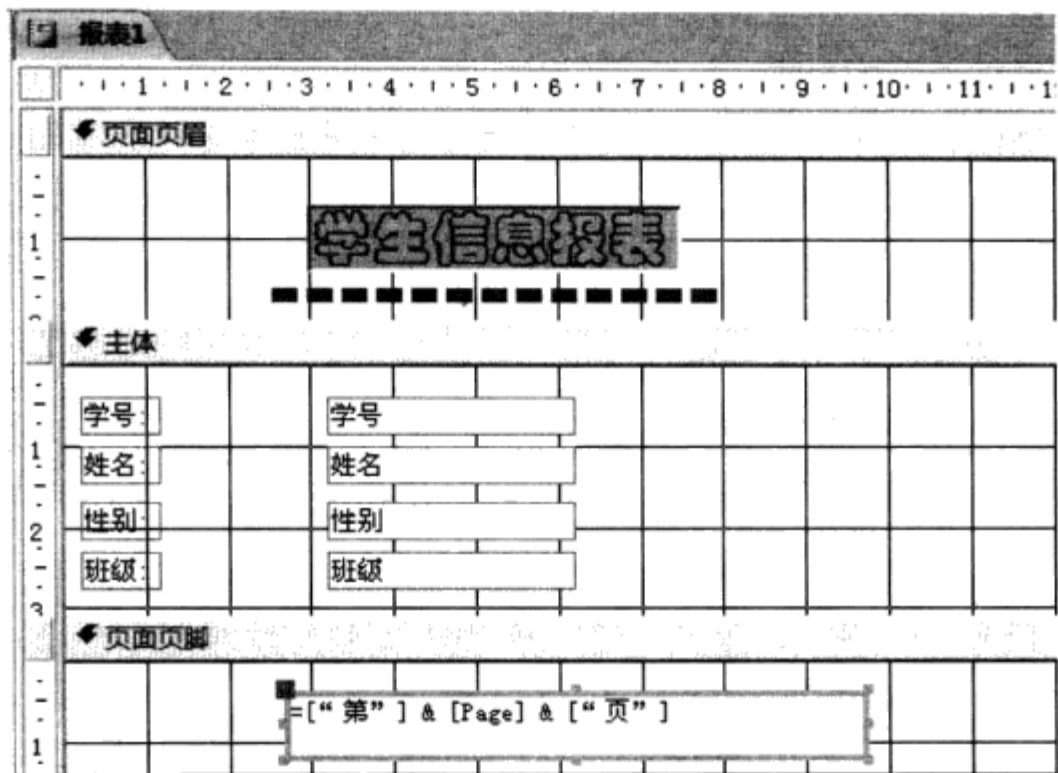


图 10-25 在页脚上添加控件后的报表视图

(5) 选择“视图”按钮下的“打印预览”选项来观看效果，设计好的报表如图 10-26 所示。

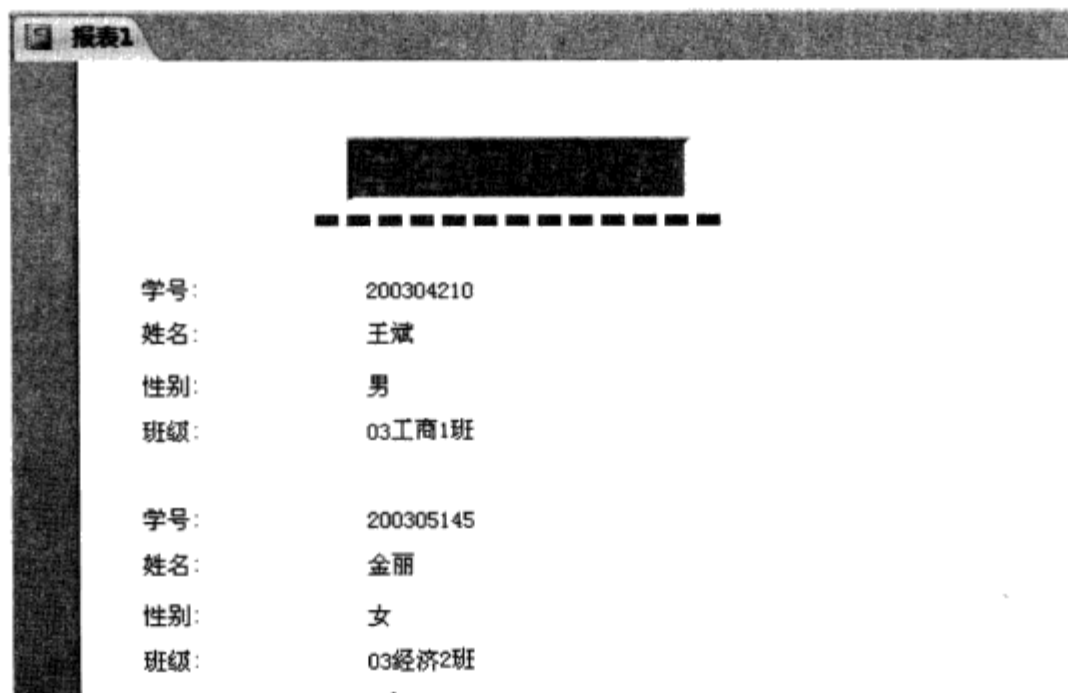


图 10-26 设计好的报表视图

10.2.5 创建带有子报表的报表

☞ 知识要点分析

子报表是包含在其他报表中的报表。在合并多个报表时，其中的一个必然作为主报表，

相对而言，其他称为子报表。主报表可以是绑定型的，也可以是非绑定的。

主报表和子报表的数据来源有以下几种情况：

- 一个主报表内的多个子报表的数据来自不相关的数据源。在这种情况下，非绑定的主报表只能作为合并的无关联子报表的“容器”使用。
- 主报表和子报表的数据来自相同数据源。
- 主报表和多个子报表的数据来自相关数据源。一个主报表也能够包含两个或多个子报表共用的数据。在这种情况下，子报表包含与公共数据相关的详细记录。

应用实例

【实例 10-9】 为“学生信息报表”报表添加“成绩”子报表。

设计步骤如下：

(1) 打开“高校学生成绩管理系统”数据库，选择“学生信息报表”报表，然后单击“设计”按钮，打开“学生信息报表”报表的设计视图，如图 10-25 所示。

(2) 选中“控件向导”按钮，然后单击“子窗体/子报表”控件，在主体区拖动鼠标，创建子报表。这时会弹出“子报表向导”对话框。选择其中的第一个选项（使用现有的表和查询），如图 10-27 所示。

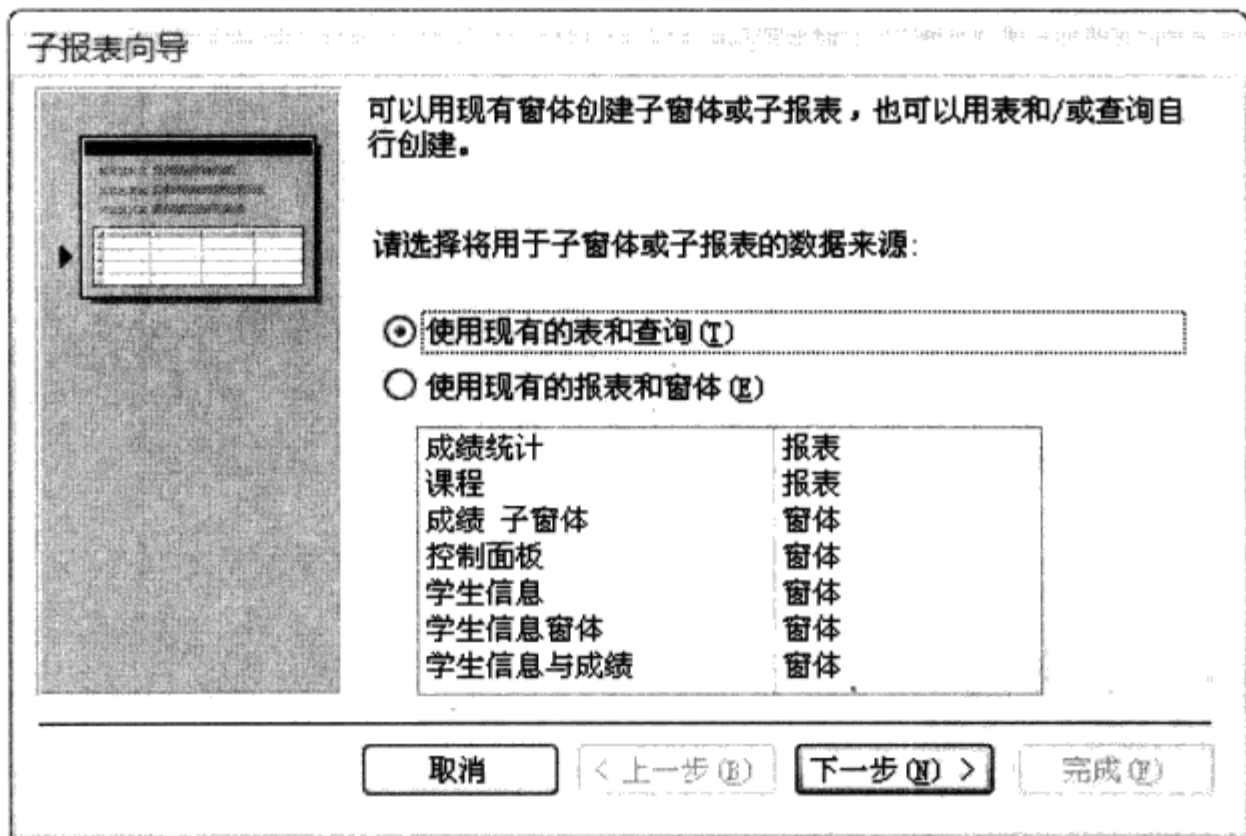


图 10-27 “子报表向导”对话框 1

(3) 单击“下一步”按钮，在接下来的对话框中选择子报表的来源表和字段，在本例中选择“成绩”表以及表中的所有字段。

(4) 单击“下一步”按钮，在接下来的对话框中选择主报表和子报表的连接字段，如图 10-28 所示。

(5) 单击“下一步”按钮，在接下来的对话框中设置子报表的名称，输入“学生成绩子报表”。

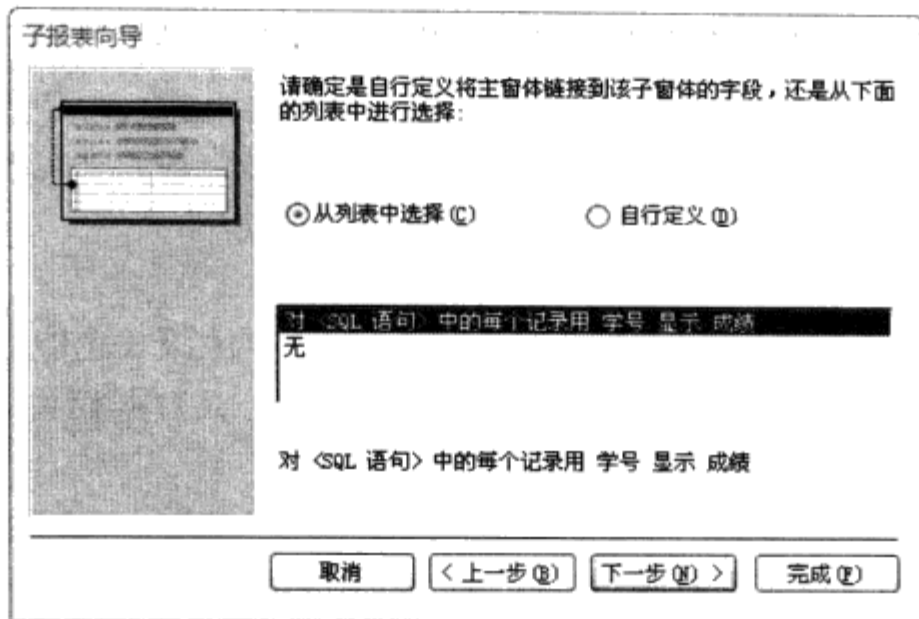


图 10-28 “子报表向导”对话框 2

(6) 单击“完成”按钮，于是在“学生信息报表”报表中添加了子报表。修改子报表后，效果如图 10-29 所示。



图 10-29 添加的子报表

(7) 切换到报表打印预览视图，效果如图 10-30 所示。

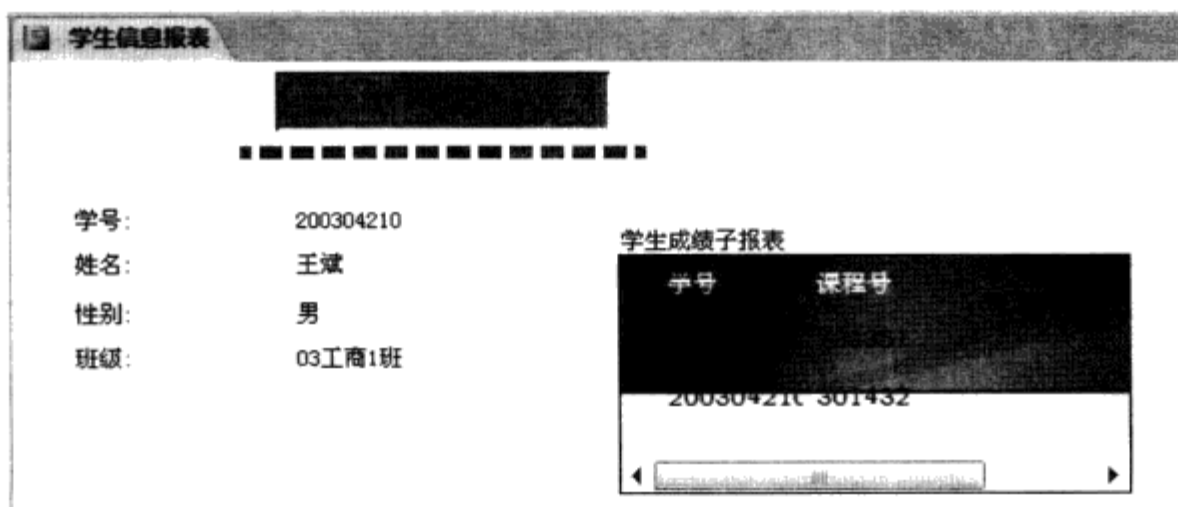


图 10-30 添加的子报表打印预览效果

(8) 保存修改的报表为“学生信息和成绩”报表。

10.3 模块概述

☞ 知识要点分析

模块是一种重要的 Access 数据库对象,是用 Visual Basic for Application 语言编写的程序代码。模块是使用编程语言编写的,与其他数据库对象相比,模块具有更大的灵活性和更强的功能。

1. 模块

模块基本上是由声明、语句和过程组成的集合,它们作为一个已命名的单元存储在一起,对 Visual Basic 代码进行组织。

1) 模块的定义

模块是将 Visual Basic 声明和过程作为一个单元进行存储的集合。在 Access 中,模块分为类模块和标准模块两种类型。

2) 类模块

窗体模块和报表模块都是类模块,它们各自与某一特定窗体或报表相关联。为窗体或报表创建第一个事件过程时,Access 将自动创建与之关联的窗体模块或报表模块。窗体模块和报表模块通常都含有事件过程,而过程的运行用于响应窗体或报表上的事件。使用事件过程可以控制窗体或报表的行为以及它们对用户操作的响应,例如单击某个命令按钮时打开指定的窗体或报表等。

窗体模块和报表模块中的过程可以调用已经添加到标准模块中的过程。

3) 标准模块

标准模块包含与任何其他对象都无关的常规过程以及可从数据库任何位置运行的经常使用的过程。标准模块与某个特定对象无关的类模块的主要区别在于其作用范围和生命周期。在没有相关对象的类模块中,声明或存在的任何变量或常量的值都仅在该代码运行时,该对象中是可用的。

2. 过程

过程是由 Visual Basic 代码组成的单元,它包含一系列执行操作或计算值的语句和方法。过程分两种类型:Sub 过程和 Function 过程。

1) Sub 过程

Sub 过程执行一项操作或一系列操作,但是不返回值。既可以自行创建 Sub 过程,也可以使用 Access 所创建的事件过程模板。

例如,下列事件过程使用 OpenForm 方法打开“学生信息”窗体:

```
Private Sub OpenOrders_Click()  
    DoCmd. OpenForm"学生信息"  
End Sub
```

数据库中每个窗体和报表都有内置的窗体或报表模块，这些模块包含事件过程模板。可以向其中添加代码，使得当窗体、报表或其上的控件中发生相应的事件时，运行这些代码。当 Access 识别出事件在窗体、报表或控件中已经发生时，将自动地运行对象和事件命名的事件过程。

在 Access 2007 中，许多向导（如“命令按钮向导”）在创建对象的同时也创建对象的事件过程。

2) Function 过程

Function 过程通常也称为函数，它将返回一个值，例如计算结果。Visual Basic 包含许多内置函数。例如，Now 函数可返回当前的日期与时间。除了这些内置函数外，也可以自行创建自定义函数。

因为函数有返回值，所以可以在表达式中使用。在 Access 中，很多地方的表达式中都可以使用函数。例如，在 Visual Basic 语句、方法或属性设置中以及在筛选或查询的条件表达式中都可以使用函数。

例如，下列自定义函数 FirstOfNextMonth 返回下个月第一天的日期：

```
Function FirstOfNextMonth()  
    FirstOfNextMonth=_  
    DateSerial(Year(Now),Month(Now)+1,1)  
End Function
```

这个自定义函数包含一个赋值语句，将表达式的结果（位于等号右侧）赋予函数名 FirstOfNextMonth（位于等号左侧）。该函数使用了内置的 Visual Basic 函数 DateSerial、Year、Now 和 Month 来计算结果。DateSerial 函数用于返回包含指定的年、月、日的日期数据，Year 函数返回表示年份的整数，Month 函数返回表示月份的整数。

创建函数后，就可以在 Access 的几乎任何地方的表达式中使用该函数。例如，通过在属性表中将文本框控件的“默认值”属性设置为以下表达式，可以指定下个月的第一天显示为该文本框的默认值：

```
=FirstOfNextMonth()
```

如果要将函数用作属性设置，函数必须位于窗体模块或报表模块中或位于标准模块中。在与窗体或报表无关的类模块中不能将函数用作属性设置。

应用实例

【实例 10-10】 在“高校学生成绩管理系统”数据库中，创建一个名为“打开指定窗体”的窗体，该窗体中只有一个图像控件，单击该图像就可以打开指定的“学生信息与成绩”窗体。

设计步骤如下：

(1) 在 Access 2007 中打开“高校学生成绩管理系统”数据库，创建一个名为“打开指定窗体”的窗体，该窗体中只有一个图像控件，如图 10-31 所示。

(2) 打开图像控件的属性窗口，单击“事件”选项卡。

(3) 选择“单击”事件，然后单击属性框边的“生成器”按钮，如图 10-32 所示。

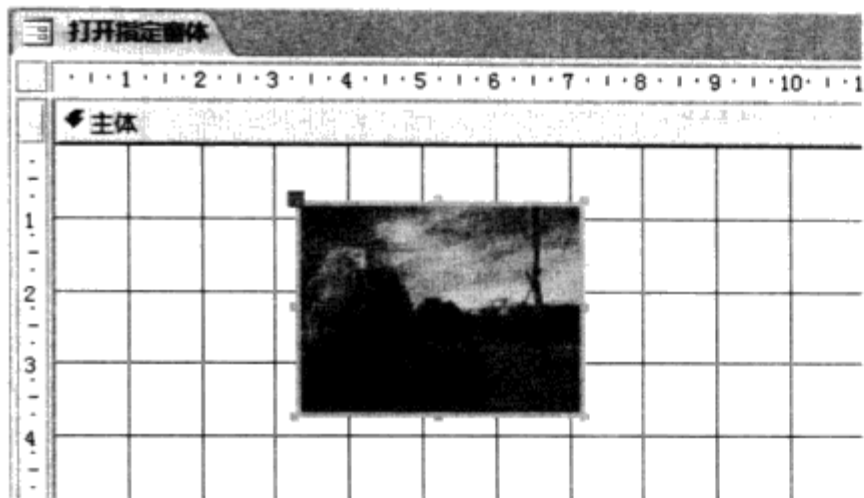


图 10-31 “打开指定窗体”的窗体

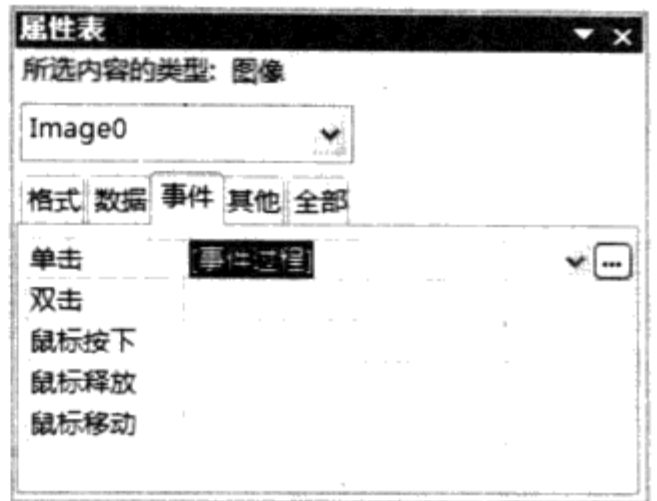


图 10-32 设置控件的“单击”属性

(4) 在如图 10-33 所示的“选择生成器”对话框中，双击“代码生成器”，以显示代码窗口。

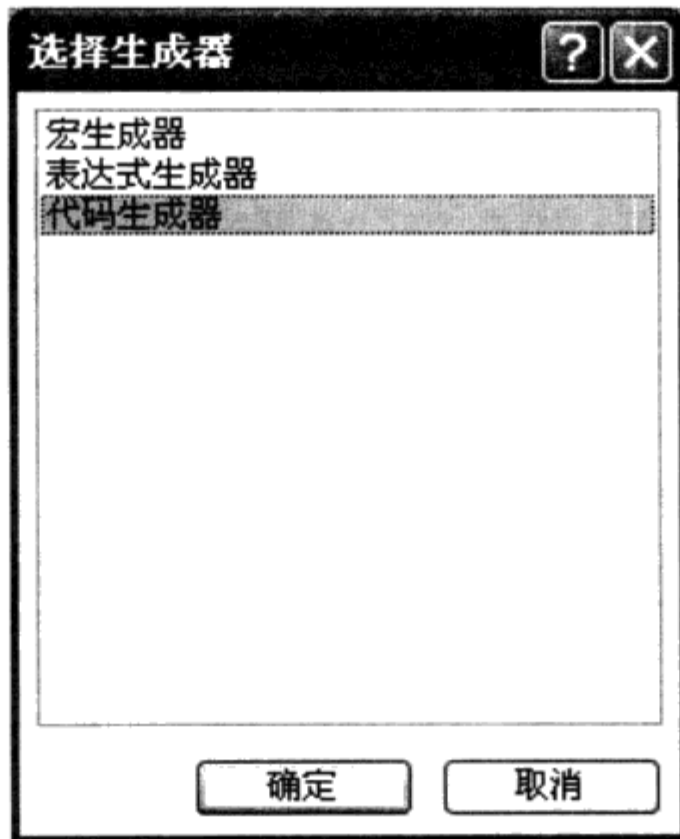


图 10-33 “选择生成器”对话框

(5) 将 Visual Basic 代码添加到 Image0_Click 事件过程中。使得单击图像控件就会打开“学生信息与成绩”窗体，并且弹出“学生信息与成绩窗体已经被打开”的对话框。在 Image0_Click 事件过程中添加以下语句（如图 10-34 所示）。

```
DoCmd.OpenForm "学生信息与成绩"
MsgBox "学生信息与成绩窗体已经被打开。"
```

(6) 若要进行测试，请在“窗体”视图打开“打开指定窗体”，然后单击窗体中的图像控件，就会打开“学生信息与成绩”窗体，并且弹出“学生信息与成绩窗体已经被打开。”的提示信息，如图 10-35 所示。



图 10-34 在 Image0_Click 事件过程中添加代码



图 10-35 测试结果

习 题 10

一、判断题

- 窗体与数据表相同，可以存储数据。()
- 窗体向导只能创建基于单表的窗体。()
- 一个报表中最多可以对 20 个字段或表达式进行分组。()
- 主报表可以是绑定型的，也可以是非绑定的。()
- 窗体有 3 种视图，分别为设计视图、窗体视图和大纲视图。()
- 控件是窗体上用于显示数据、执行操作、装饰窗体的对象。()
- 非结合型的控件没有数据来源，可以用来显示信息、线条、矩形或图像。()
- 控件的类型分为：结合型、非结合型、计算型、非计算型。()
- 若对使用向导生成的报表不满足，可以在版面预览视图进行修改。()
- 报表可以输入和输出数据。()

二、单选题

- 在窗体设计视图中，至多可以使用 () 种节。

- A. 3 B. 4 C. 5 D. 6
2. 在窗体中添加 () 控件, 可以设置打开另一个窗体。
A. 标签 B. 文本框 C. 命令按钮 D. 选项按钮
 3. 在 Access 中, 在窗体中的一个窗体称为 ()。
A. 主窗体 B. 子窗体 C. 数据表窗体 D. 表格式窗体
 4. 若对使用向导生成的报表不满足, 可以在 () 视图进行修改。
A. 窗体设计 B. 版面预览 C. 打印预览 D. 报表设计
 5. 绑定控件的数据来源是 ()。
A. 记录内容 B. 字段值 C. 表达式 D. 文本框
 6. 用来显示整份报表的汇总说明的是 () 区。
A. 报表页脚 B. 页面页脚 C. 组页脚 D. 主体
 7. 要设置在报表每一页的顶部都输出的信息, 需要设置 ()。
A. 报表页眉 B. 报表页脚 C. 页面页眉 D. 页面页脚
 8. 在报表设计区, () 区用来处理每一条记录。
A. 报表页眉 B. 页面页眉 C. 组页眉 D. 主体
 9. 如果加载一个窗体, 先被触发的事件是 ()。
A. Load 事件 B. Open 事件 C. Click 事件 D. DblClick 事件
 10. 窗体是 Access 数据库中的一种对象, 以下哪项不是窗体具备的功能。()
A. 输入数据 B. 编辑数据 C. 输出数据 D. 显示和查询表中的数据

三、上机练习

利用第 8 章上机练习创建的“图书资料管理”数据库及第 9 章建立的查询完成以下问题。

1. 建立学生借书录入窗体。
2. 建立学生借书信息报表。

第 11 章

Visual Basic 的数据库访问技术

作为微软旗下一款优秀的计算机编程开发工具，VB 在数据库应用开发方面的能力十分强大，微软设计了多种数据库访问方法，在以往开发环境的基础上，增加了功能强大的 ADO 作为存取数据的新标准，不但能够灵活、方便地实现数据库的各种操作，同时还支持多种数据库类型。目前，VB 被广泛地用作大型公司数据库和客户机/服务器应用程序的前台开发工具，与后台的数据库技术相结合，提供了一个高性能的客户机/服务器方案。

11.1 数据访问技术基础

☞ 知识要点分析

在使用 VB 开发数据库应用程序过程时，通常的做法是，先使用数据库管理系统或 VB 可视化数据管理器建立数据库和数据表，然后在 VB 程序中通过使用各种数据控件或引用对象模型与数据库建立连接，再借助于数据绑定控件对数据库中的数据进行各种操作。

1. 数据库的分类

VB 可以访问的数据库有以下三类。

(1) Jet 数据库：Jet 数据库由 Jet 数据库引擎直接生成和操作，不仅灵活而且速度快，如 Microsoft Access 97 和 VB 使用相同的 Jet 数据库引擎。

(2) ISAM 数据库：即索引顺序访问方法数据库，如 Dbase、FoxPro、Paradox、TextFiles。在 VB 中可以生成和操作这些数据库。

(3) ODBC 数据库：即遵守 ODBC 标准的客户/服务器数据库，如 SQL Server、Oracle、Sybase 等，VB 可以使用任何支持 ODBC 标准的数据库。

2. 数据引擎

VB 的数据引擎是 VB 程序与数据库连接的桥梁，分为三种。

(1) Jet 数据库引擎：由微软提出专门针对本地数据访问需求的数据引擎，适用于对 Access、Excel 等各种本地数据源的访问。

(2) ODBC（开放数据库互联）：ODBC 是微软公司开放服务结构中有关数据库的一个组成部分，它建立了一组规范，并提供了一组对数据库访问的标准 API（应用程序编程接口）。这些 API 利用 SQL 来完成其大部分任务。ODBC 本身也提供了对 SQL 语言的支持，用户可以直接将 SQL 语句送给 ODBC。一个基于 ODBC 的应用程序对数据库的操作不依

赖任何 DBMS (数据库管理系统), 不直接与 DBMS 打交道, 所有的数据库操作由对应的 DBMS 的 ODBC 驱动程序完成。因此 ODBC 能以统一的方式处理所有的基于 SQL 的数据库。

(3) OLE DB: 作为微软的组件对象模型 (COM) 的一种设计, OLE DB 是一组读写数据的方法。OLE DB 能够以统一的方式访问存储在不同信息源中的数据, 包括关系和非关系数据库、电子邮件和文件系统、文本和图形、自定义商业对象等。与 ODBC 相比, OLE DB 提供了更好的访问数据的灵活性, 且 OLE DB 对 ODBC 具有兼容性, 允许 OLE DB 访问现有的 ODBC 数据源。符合 ODBC 标准的数据源要符合 OLE DB 标准, 还必须提供相应的 OLE DB 服务程序, 这种服务程序称为 ODBC OLE DB Provider。OLE DB 主要由数据提供者 (Data Providers)、数据使用者 (Data Consumers)、服务组件 (Service Components) 三部分组成。

3. 数据访问接口

数据访问接口是一个对象模型, 代表了访问数据的各个方面。在 VB 中可用的数据访问接口有三种: ActiveX 数据对象 (ADO)、远程数据对象 (RDO) 和数据访问对象 (DAO)。

1) 数据访问对象 (DAO)

DAO 是一种应用程序编程接口 (API), 存在于微软的 Visual Basic 中, 它允许程序员请求对微软的 Access 数据库的访问。DAO 是微软的第一个面向对象的数据库接口。DAO 最适用于单系统应用程序或在小范围本地分布使用。其内部已经对 Jet 数据库和 ISAM 数据库的访问进行了加速优化, 而且其使用起来也是很方便的。所以如果数据库是 Access 数据库且是本地使用的话, 建议使用这种访问方式。VB 已经把 DAO 模型封装成了 Data 控件, 分别设置相应的 DatabaseName 属性和 RecordSource 属性就可以将 Data 控件与数据库中的记录连接起来。以后就可以使用 Data 控件对数据库进行操作。

2) 远程数据对象 (RDO)

RDO 是一个到 ODBC 的面向对象的数据访问接口, 利用 RDO 和 MSRDC, 应用程序无须使用本地的查询处理程序即可访问 ODBC 数据源, 因此在访问远程数据库引擎时, 可以获得更好的性能与更大的灵活性。RDO 的不足之处在于, 在访问 Jet 数据库或 ISAM 数据库方面仍然有一定的限制, 并且它只能通过现存的 ODBC 驱动程序来访问关系数据库。尽管如此, RDO 仍不失为大型关系数据库开发者经常选用的最佳接口。RDO 提供了用来访问存储过程和复杂结果集的更多和更复杂的对象、属性以及方法。和 DAO 一样, 在 VB 中也把其封装为 RDO 控件了, 其使用方法与 DAO 控件的使用方法完全相同。

3) ActiveX 数据对象 (ADO)

ADO 是一种建立在 OLE DB 上的轻量的、高性能的应用程序层接口, 同时也支持 ODBC 访问。ADO 可以处理任何类型的本地或远程数据, 因此 ADO 技术得到了广泛的应用。ADO 扩展了 DAO 和 RDO 所使用的对象模型, 这意味着它包含较少的对象, 更多的属性、方法和事件。微软已经明确表示今后把重点放在 ADO 上, 对 DAO 和 RDO 不再作升级, 所以 ADO 已经成为了当前数据库开发的主流。

4. 使用 Access 2007 作为后台数据库

本书仅以 Access 2007 数据库为例阐述 VB 数据库应用系统的开发。为在 VB 中可以访问 Access 2007 数据库, 用户需在微软站点上下载并安装 2007 Office System 驱动程序之数

据连接组件 (AccessDatabaseEngine.exe), 非 Microsoft Office 应用程序可以使用这一组件从 2007 Office System 文件中读取数据, 例如从 Microsoft Office Access 2007 (mdb 和 accdb) 文件以及 Microsoft Office Excel 2007 (xls、xlsx 和 xlsb) 文件中读取数据。这些组件还支持与 Microsoft Windows SharePoint Services 和文本文件建立连接, 此外, 还会安装 ODBC 和 OLEDB 驱动程序, 供应用程序开发人员在开发与 Office 文件格式连接的应用程序时使用。

11.2 ADO Data 控件

ADO Data 控件是一个 ActiveX 控件, 通过在设计时设置 ADO Data 控件的属性, 可以快速建立与数据库的连接, 可以用最少的代码创建数据库应用程序。利用 ADO Data 控件可以实现以下功能。

- (1) 连接一个本地数据库或远程数据库。
- (2) 打开一个指定的数据库表, 或定义一个基于结构化查询语言 (SQL) 的查询、存储过程或该数据库中的表的视图的记录集合。
- (3) 将数据字段的值传递给数据绑定控件, 可支持在这些控件中显示或更改这些值。
- (4) 添加新的记录, 或根据对显示在绑定的控件中的数据的数据的任何更改更新数据库。

由于 ADO Data 控件不是 VB 的内部控件, 因此在使用前必须将其添加到工具箱中: 在工具箱任意空白处右击, 选择快捷菜单中的“部件”命令, 在弹出的“部件”对话框选中“Microsoft ADO Data Control 6.0 (SF6)”复选框, 如图 11-1 所示。单击“确定”按钮, 就可以将 ADO Data 控件 (Adodc) 添加到工具箱中。

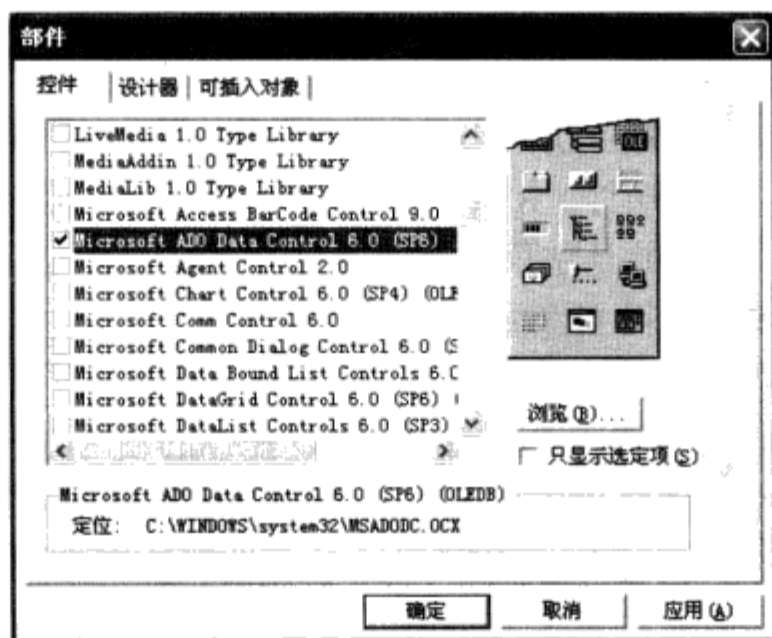


图 11-1 “部件”对话框

在窗体上绘制 ADO Data 控件, 其外观如图 11-2 所示。

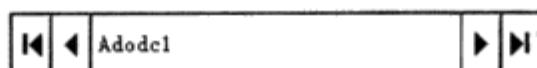


图 11-2 ADO Data 控件外观

两端的 4 个按钮分别用于控制其对应记录集记录指针的移动，从左至右依次为 MoveFirst（移动到首记录）、MovePrevious（移动到前一条）、MoveNext（移动到下一条）和 MoveLast（移动到最后一条），中间的区域为标题区，用以显示 ADO Data 控件的标题（Caption）。

11.2.1 ADO Data 控件的常用属性

将 ADO Data 控件添加到窗体后，就可以设置它的属性，进行数据库应用程序的开发。ADO Data 控件的核心属性有三个：ConnectionString、CommandType 和 RecordSource。其中，ConnectionString 属性用于定义 ADO Data 控件和数据库连接的连接字符串，通过这个连接字符串可以实现 ADO Data 控件和 OLE DB Provider 支持的数据库的连接；RecordSource 指出了可以操作的数据来源，即记录集的内容；CommandType 用于说明这个内容的类型，可以是一张表，一个存储过程或者 SQL 命令生成的结果集。明确了这些属性后，就得到了 ADO Data 控件的 RecordSet 对象。通过 RecordSet 对象的各种属性、方法和事件，就可以管理、维护和使用数据库中的数据。

☞ 知识要点分析

1. ConnectionString 属性

ConnectionString 属性用来建立与数据源的连接。当创建连接时，可以使用三种数据源：OLE DB 文件（.UDL）、ODBC 数据源（.DSN）或连接字符串。创建连接的方法是：在窗体中选择 ADO Data 控件后，右击，在弹出的快捷菜单中选择“ADODC 属性”命令，将会打开“属性页”对话框，选择“通用”选项卡，如图 11-3 所示。

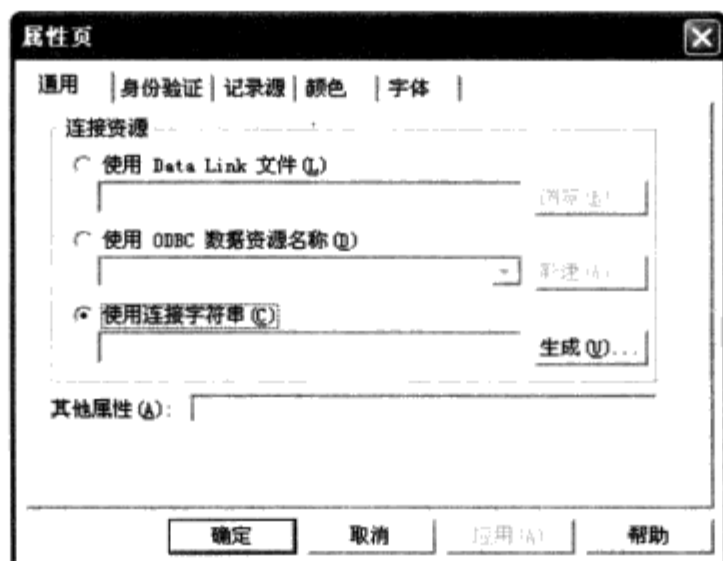


图 11-3 “属性页”对话框“通用”选项卡

(1) 如果使用连接字符串，则在图 11-3 中单击“使用连接字符串”选项按钮，单击“生成”按钮，打开“数据链接属性”对话框；选择“提供程序”选项卡，如图 11-4 所示。在“OLE DB 提供程序”列表中选择“Microsoft Office 12.0 Access Database Engine OLE DB Provider”，单击“下一步”按钮。将会自动切换到“连接”选项卡，如图 11-5 所示；在“数据源”文本框中输入要连接的 Access 2007 数据库的路径和文件名，单击“测试连接”按钮。如果测试成功，那么到数据源的连接便已建立，单击“确定”按钮，就会返回图 11-3

所示的“属性页”对话框，并根据用户的选择自动填充“使用连接字符串”文本框，例如，“Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\GXGL\高校学生成绩管理系统.accdb;Persist Security Info=False”。

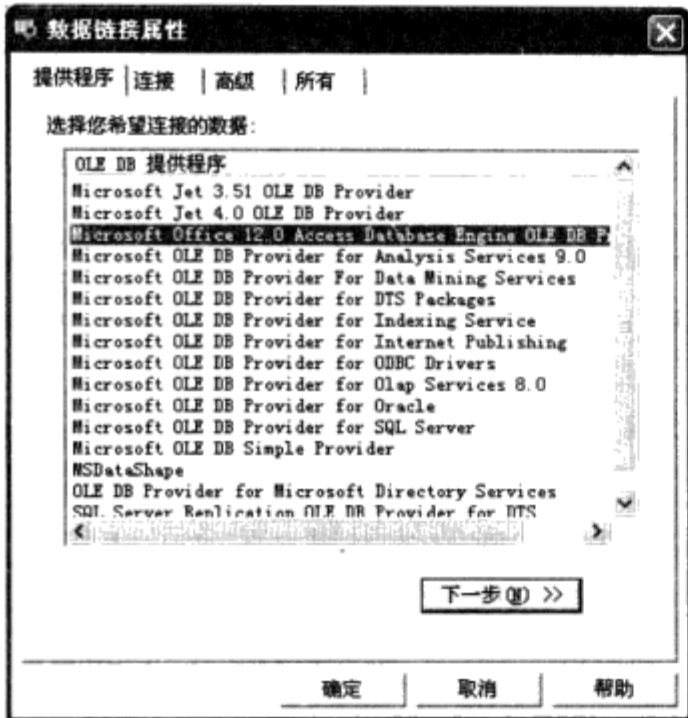


图 11-4 “提供程序”选项卡

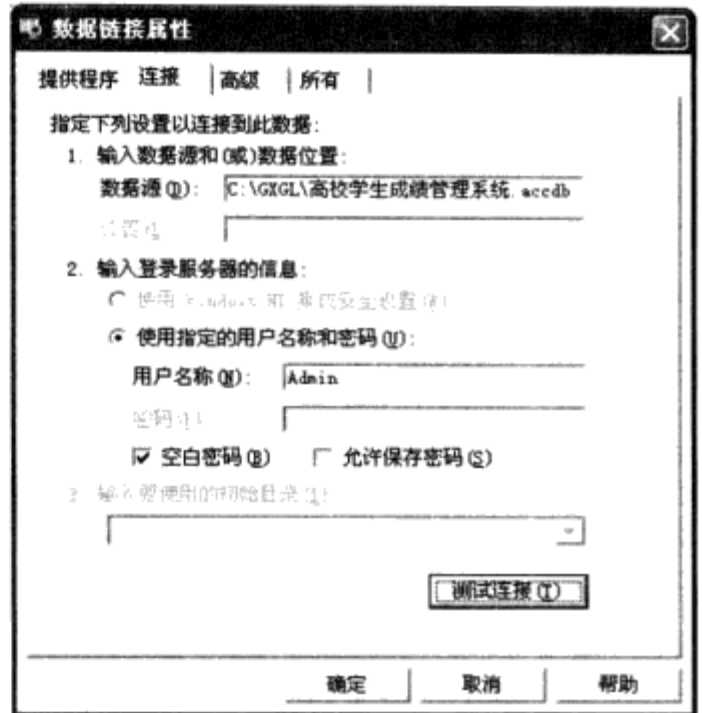


图 11-5 “连接”选项卡

(2) 如果要建立 OLE DB 连接，则在图 11-3 中单击“使用 Data Link 文件”选项按钮。单击“浏览”按钮后，在弹出的“选择数据链接文件”对话框中，选择并打开对应的.udl 文件，如图 11-6 所示。



图 11-6 “选择数据链接文件”对话框

如果事先没有 .UDL 文件, 则可按如下步骤建立: ①在“我的电脑”中切换至指定的路径; ②利用“工具”菜单中的“文件夹选项”命令, 在“查看”选项卡中设置“显示已知文件类型的扩展名”; ③新建一个文本文件, 并修改其扩展名为“.UDL”; ④在对应的文件上右击, 在弹出的快捷菜单中选择“属性”命令, 将会弹出如图 11-7 的对话框; ⑤单击“提供程序”选项卡(同图 11-4)和“连接”选项卡(同图 11-5)分别进行设置, 完成后单击“确定”按钮, 数据链接文件设置完毕。

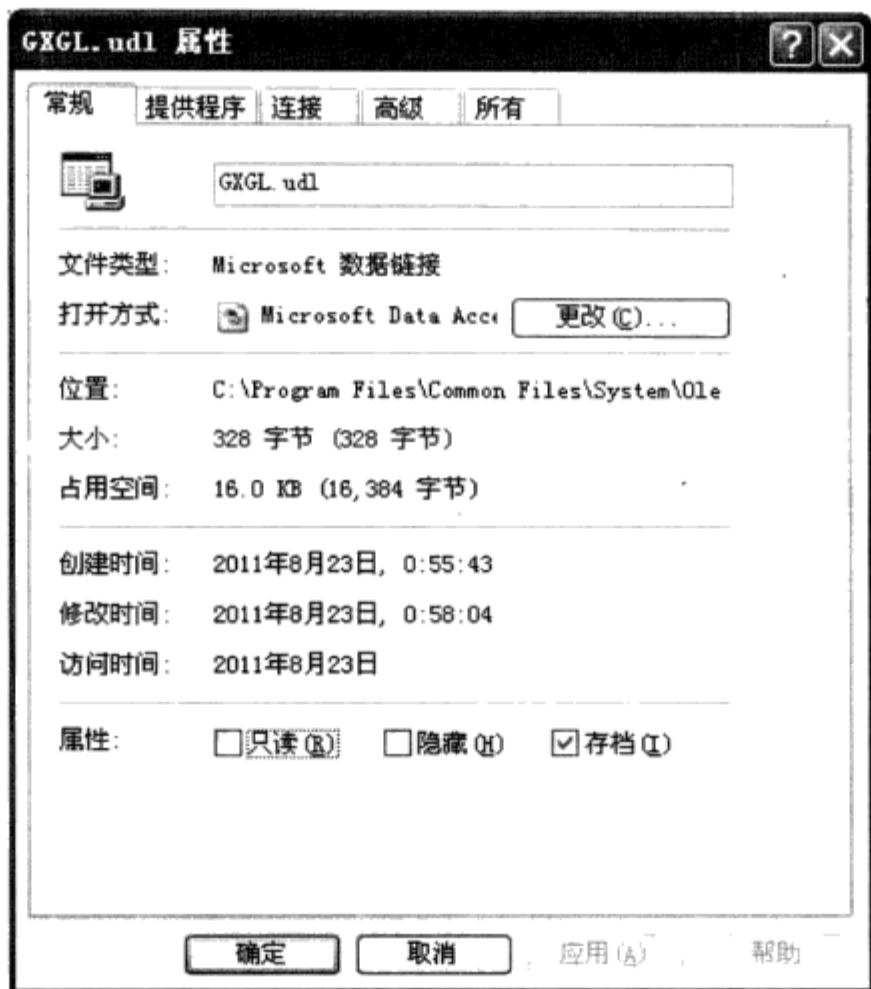


图 11-7 设置数据链接文件

(3) 如果要建立 ODBC 数据连接, 则在图 11-3 中单击“使用 ODBC 数据资源名称”选项按钮。对于已经配置过的 ODBC 数据源, 可以直接从下拉列表中选择, 否则单击“新建”按钮创建一个 ODBC 数据源, 步骤如下:

① 单击“新建”按钮, 将会打开“创建新数据源”对话框, 如图 11-8 所示。

② 首先选择 ODBC 数据源的类型, 可以是系统数据源、用户数据源、文件数据源。选择系统数据源将创建该机器专用的数据源, 且登录到这台机器的任何用户都可以使用。选择用户数据源将创建该机器专用的数据源, 且只有创建者才能够使用。选择文件数据源将创建基于文件的数据源, 所有能够访问该数据库的用户都可以共享该数据源。这里选择“系统数据源”, 单击“下一步”按钮。

③ 在弹出的如图 11-9 的对话框中, 指定该数据源的数据库驱动程序为 Microsoft Access Driver(*.mdb,*.accdB), 单击“下一步”按钮, 将会显示如图 11-10 所示的对话框。

④ 单击“完成”按钮, 将会弹出如图 11-11 所示的对话框。在“数据源名”中为新创建的数据源命名, 然后单击“选择”按钮。

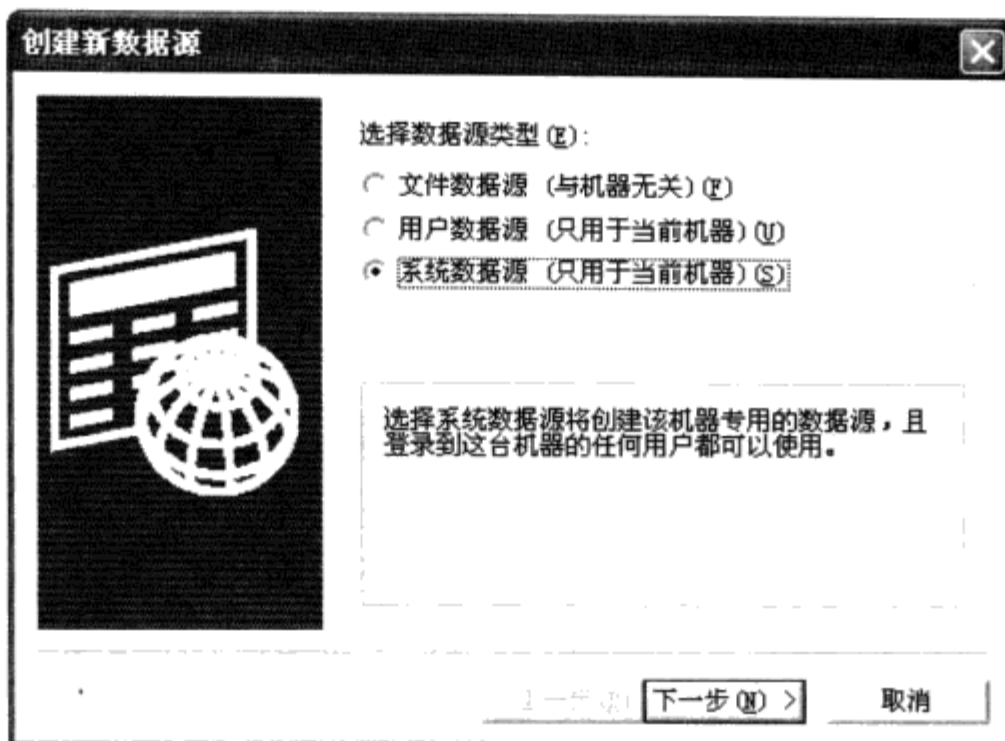


图 11-8 “创建新数据源”对话框



图 11-9 选择数据源驱动程序



图 11-10 创建新数据源完成

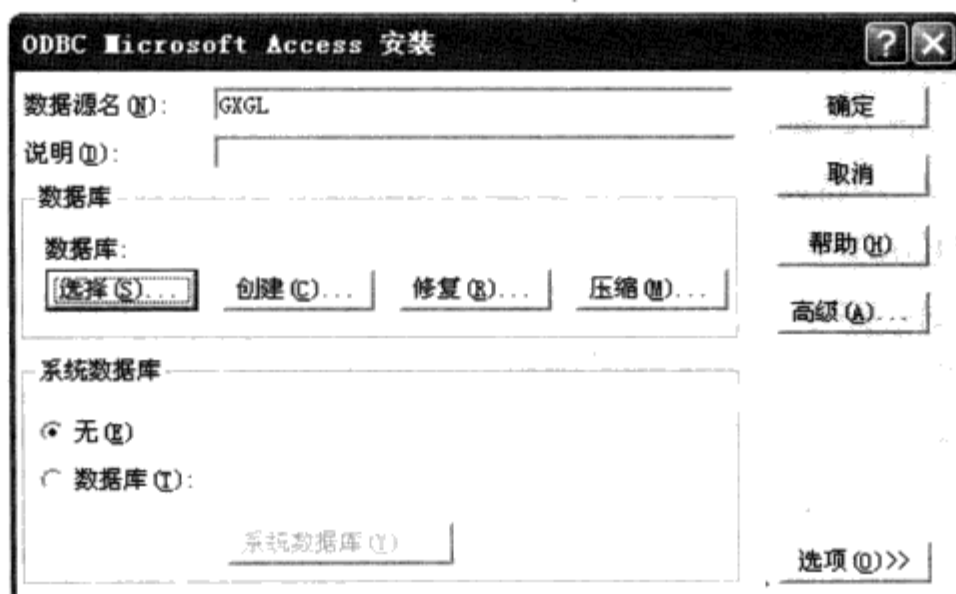


图 11-11 ODBC 数据源参数设置

⑤ 在弹出的如图 11-12 所示的对话框中选择数据库，完成后单击“确定”按钮，返回

图 11-11 所示的对话框。

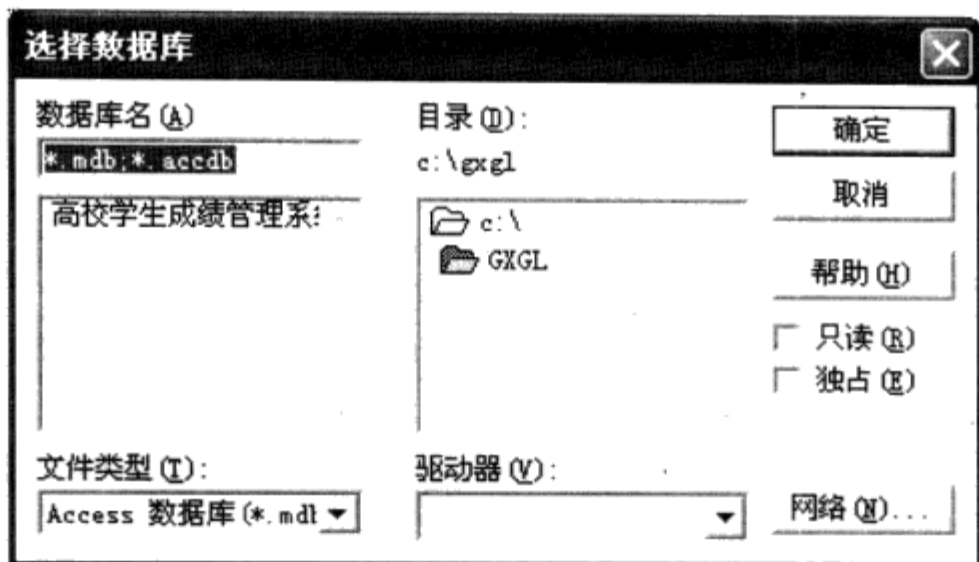


图 11-12 选择数据库

⑥ 单击“确定”按钮，返回到“属性页”对话框，如图 11-13 所示。在“使用 ODBC 数据资源名称”下拉列表中，选择新创建的 ODBC 数据源。

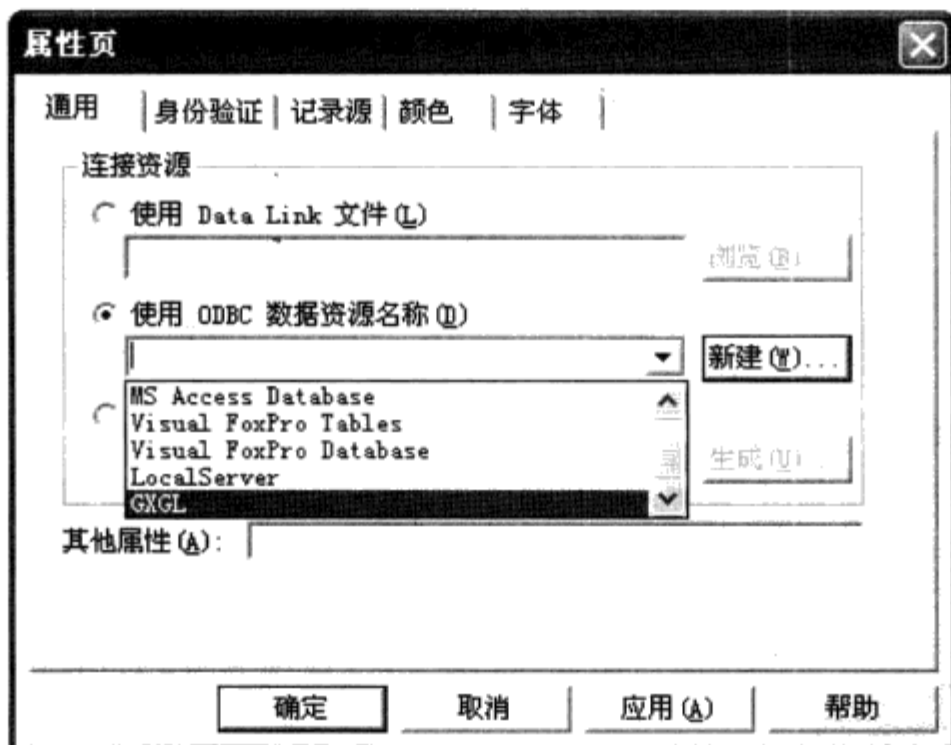


图 11-13 选择新创建的数据源

2. RecordSource 属性和 CommandType 属性

RecordSource 属性用于设置 ADO Data 控件的记录源，以确定具体访问的数据，这些数据将构成 ADO Data 控件的记录集 (Recordset) 对象。

CommandType 属性用于指定 RecordSource 的取值类型，其值参见表 11-1。如果 RecordSource 为数据库中的表，则命令类型 (CommandType) 应取值为“2-AdCmdTable”；如果 RecordSource 为数据库中的存储过程，则命令类型 (CommandType) 应取值为“4-AdCmdStoreProc”；如果 RecordSource 是一个查询，则 CommandType 应取值为“1-AdCmdText”。

表 11-1 CommandType 属性的取值

值	说明
1-AdCmdText	命令类型为命令文本，通常为 SQL 语句
2-AdCmdTable	命令类型为一个存在的表的名称
4-AdCmdStoreProc	命令类型为一个存在的存储过程的名称
8-AdCmdUnknown	命令类型无法确定，缺省值

RecordSource 属性、CommandType 属性均在 ADO Data 控件的“属性页”对话框的“记录源”选项卡中进行设置，如图 11-14 所示。

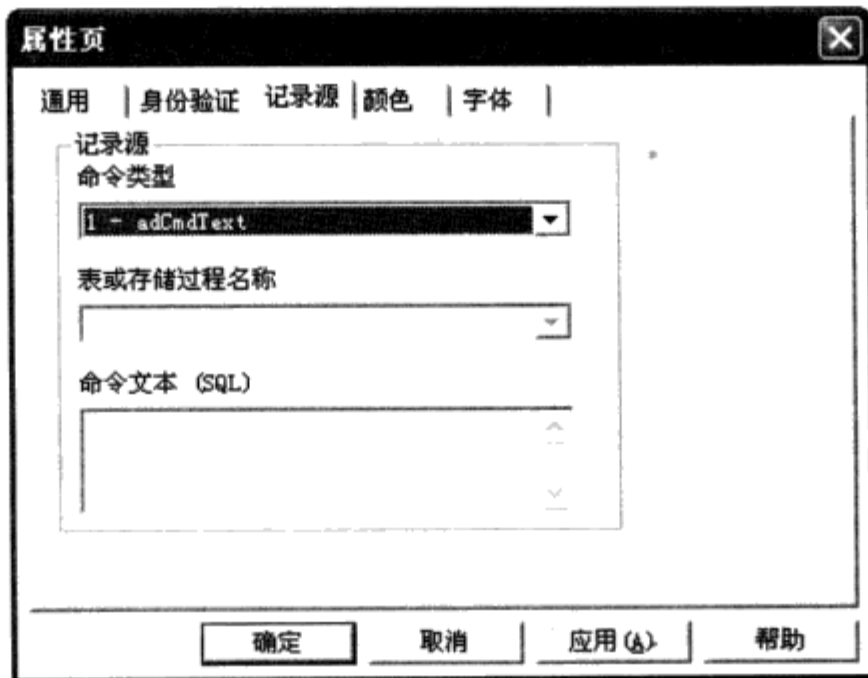


图 11-14 “属性页”对话框“记录源”选项卡

3. Recordset 属性

Recordset 属性用于返回由 ADO Data 控件所确定的记录集，可以是一个表，也可以是一个查询结果集。利用 Recordset 对象的属性、方法和事件可以实现对数据库的访问。在任何时候，记录指针总是指向记录集中的一个记录，该记录被称之为当前记录。以名为 Adodc1 的 ADO Data 控件为例，如果想要访问其记录集中当前记录指定的字段的内容，可以采用如下的两种方法。

(1) Adodc1.Recordset.Fields(字段索引)，其中字段索引为字段在记录集上的顺序编号，注意编号从 0 开始。

(2) Adodc1.Recordset.Fields("字段名")。

4. UserName 属性和 Password 属性

当数据库受密码保护时，使用 ADO Data 控件访问该数据库，需设置 UserName 属性和 Password 属性。UserName 属性用于指定用户的名称，Password 属性用于指定密码。

5. Caption 属性

Caption 属性用于设置或返回显示在 ADO Data 控件标题区上的文本。

6. BOFAction 和 EOFAction 属性

BOFAction (或 EOFAction) 属性用于控制当记录集的记录指针指向 BOF 位置 (或 EOF 位置) 时，ADO Data 控件所要采取的操作，仅当使用 ADO Data 控件上的记录指针移动按

钮时有效，取值及含义如表 11-2 所示。

表 11-2 BOFAction 和 EOFAction 属性的取值

属性名	值	含义
BOFAction	0-adDoMoveFirst	回拨记录指针，将记录指针指向第一条记录，默认值
	1-adStayBOF	将记录指针保持在 BOF 位置不动，BOF 属性值为 True，此刻禁止 ADO Data 数据控件上的 MovePrevious 按钮
EOFAction	0-adDoMoveLast	回拨记录指针，将记录指针指向最后一条记录，默认值
	1-adStayEOF	将记录指针保持在 EOF 位置不动，EOF 属性值为 True，此刻禁止 ADO 数据控件上的 MoveNext 按钮
	2-adDoAddNew	自动追加一条新记录，并将记录指针指向它

11.2.2 ADO Data 控件的常用方法和事件

知识要点分析

1. Refresh 方法

Refresh 方法用于刷新 ADO Data 控件的连接属性，并重建记录集。在程序运行过程中，如果修改了 ADO Data 控件的数据源连接属性（ConnectionString 属性、CommandType 属性或 RecordSource 属性），必须使用 Refresh 方法来重建记录集。

语法：

```
Adodc1.Refresh
```

2. WillMove 和 MoveComplete 事件

WillMove 事件是在记录集中记录指针的位置即将发生变化时被触发，而 MoveComplete 事件则是在记录集中记录指针的位置发生改变后被触发。

3. WillChangeField 和 FieldChangeComplete 事件

WillChangeField 事件是在即将完成对记录集中当前记录的某个字段的修改时被触发，而 FieldChangeComplete 则是在完成字段修改之后被触发。

4. WillChangeRecord 和 RecordChangeComplete 事件

WillChangeRecord 事件是在即将完成对记录集中当前记录的修改时被触发，而 RecordChangeComplete 事件则是在完成记录修改之后被触发。

11.2.3 ADO Data 控件的 Recordset 对象

知识要点分析

当 ADO Data 控件的 ConnectionString 属性、RecordSource 属性和 CommandType 属性设置完成后，就可以唯一确定其记录集（Recordset 对象）。对 Recordset 对象的编辑、修改操作最终会通过 ADO 传送给数据库中相应的数据表。

1. Recordset 对象的常用属性

1) AbsolutePosition

AbsolutePosition 属性为只读属性，用于返回记录集中当前记录指针的位置。

2) RecordCount 属性

RecordCount 属性为只读属性，用于返回记录集中记录的个数。

3) BOF 和 EOF 属性

BOF 和 EOF 属性用于判断记录集中当前记录指针是否指向文件头（BOF 位置）或者文件尾（EOF 位置），是则返回 True，否则返回 False。

2. Recordset 对象的常用方法

以下的讲解均以名为 Adodc1 的 ADO Data 控件为例。

1) AddNew 方法

用于在记录集中追加一条新记录，并将记录指针指向它。一般用于为表类型的记录集添加记录。

语法：

```
Adodc1.Recordset.AddNew
```

2) Delete 方法

用于在记录集中删除当前记录。一般用于为表类型的记录集删除记录。

语法：

```
Adodc1.Recordset.Delete
```

说明：

(1) 使用记录集的 Delete 方法删除记录时，系统不进行任何提示，记录将会被直接删除。因此为了避免误操作，建议在编程时通过分支结构让用户确认其操作无误后再行删除。

(2) 使用记录集的 Delete 方法删除记录后，应使用记录集的 MoveNext 方法将记录指针下移 1 位，以指向一个有效的记录。

3) Move 方法

Move 方法用于在记录集中移动记录指针。如果在空的记录集中调用 Move 方法，将会产生错误。

语法：

```
Adodc1.Recordset.Move NumRecord, [Start ]
```

说明：

NumRecord 为记录指针移过的记录数，长整型，值为负代表向上（BOF 方向）移动，值为正代表向下（EOF 方向）移动。

Start 为可选参数，代表记录指针移动的起始位置，其值可以为以下几种。

(1) adBookmarkCurrent，表示当前记录，默认值。

(2) adBookmarkFirst，表示第一条记录。

(3) adBookmarkLast，表示最后一条记录。

4) Move 方法群组

Move 方法群组由 4 种移动记录指针的方法构成，它们分别是 MoveFirst、MoveLast、MoveNext 和 MovePrevious，用于将记录集的记录指针移动到第一条记录、最后一条记录、下一条记录和上一条记录。

语法：

```
Adodcl.Recordset.MoveFirst  
Adodcl.Recordset.MoveLast  
Adodcl.Recordset.MoveNext  
Adodcl.Recordset.MovePrevious
```

说明：

对 ADO Data 控件的 BOFAction 和 EOFAction 属性的设置，当且仅当使用 ADO Data 控件上的记录指针移动按钮时有作用，对记录集的 Move 方法及 Move 方法群组这个设置是无效的。因此在使用记录集的 Move 方法及 Move 方法群组时，要特别注意记录指针越界的问题，即在记录指针已指向头一条记录（或者最后一条记录）的情形下，如果此时还试图向上（或下）移动记录指针，将会导致记录指针指向一个没有实体记录的位置，这样可能会导致错误的产生。

为此在记录指针移动完毕以后，一定要判断记录指针是否已指向 BOF 或者 EOF 位置，如果是，记录指针一定要回拨，让它指向有实际意义的记录。

例如，当前记录指针已指向记录集的最后一条记录，下面的代码可以有效避免记录指针的越界：

```
Adodcl.Recordset.MoveNext  
If Adodcl.Recordset.EOF Then Adodcl.Recordset.MoveLast
```

5) Find 方法

Find 方法用于在记录集中查找符合指定条件的第一条记录，并使之成为当前记录。

语法：

```
Adodcl.Recordset.Find ("条件")
```

说明：

使用 Find 方法前，应使用记录集的 MoveFirst 方法，将记录指针指向第一条记录；使用 Find 方法后，如果在记录集中没有找到满足指定条件的记录，记录指针将会指向 EOF 位置，因此可以用记录集的 EOF 属性是否为 True 来判断 Find 方法的结果。

6) Update 方法和 CancelUpdate 方法

Update 方法用于保存对记录集当前记录所做的所有更改，一经保存这些更改便不能被撤销。事实上，更改记录后调整记录指针的位置同样可以达到保存的效果。

CancelUpdate 方法用于取消对记录集中记录的添加或对当前记录内容的修改，恢复修改前的状态。CancelUpdate 方法用于取消目前尚未完成的编辑修改操作。

语法：

```
Adodcl.Recordset.Update
```

```
Adodc1.Recordset.CancelUpdate
```

11.3 数据绑定控件

在 VB 中,用以实现与数据库连接的数据控件种类有很多,ADO Data 控件只是其中较为常用的一种。利用这些数据控件,程序设计人员只需要编写少量的代码就可以访问各种类型数据库中的数据,美中不足的是这些数据控件不具备数据显示的能力,不能独立地进行数据的浏览,所以就需要把具有数据绑定功能的控件同数据控件结合起来使用,共同完成数据的显示、编辑和查询等操作。

在 VB 中专门提供了一些与数据控件相绑定,用于显示由数据控件所确定的记录集中的数据,具有数据感知功能的控件,它们被称为数据绑定控件。当一个控件被绑定到数据控件后,VB 会把从当前数据库记录中取出的字段应用于该控件,利用这个控件显示数据并接受用户的修改,移动记录指针后,这些改变就会自动写入到数据库相应的表中。

所有的数据绑定控件都具有两个标准属性,DataSource 属性和 DataField 属性。DataSource 属性用来设置或返回数据源,即要绑定的数据控件的名称,以实现与其对应记录集的连接。DataField 属性用来设置或返回记录集中被绑定的字段。

11.3.1 内部数据绑定控件

常见的内部数据绑定控件有文本框、标签、列表框、组合框、复选框、图像框和图片框等,只要设置它们的 DataSource 属性和 DataField 属性,就可以实现与数据库的绑定。

应用实例

1. 浏览查询的结果

【实例 11-1】 编程实现对高校学生成绩管理系统数据库中的学生成绩的浏览,设计界面如图 11-15 所示。

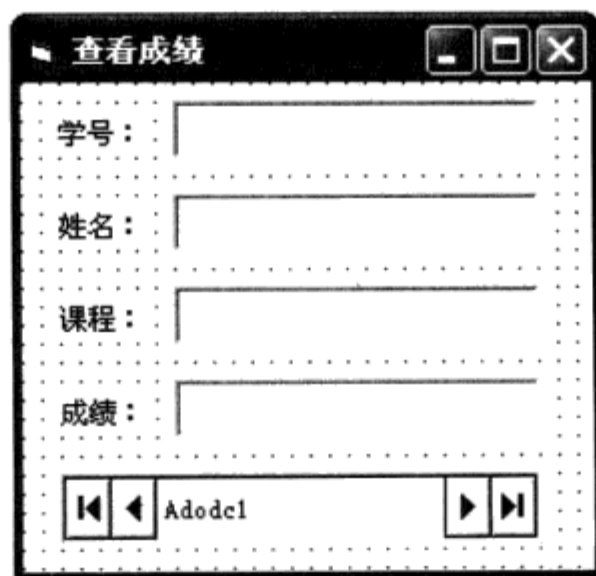


图 11-15 实例 11-1 设计界面

界面设计:

(1) 新建工程和窗体, 在工具箱中添加 ADO Data 控件。

(2) 添加 4 个标签 (Label1、Label2、Label3 和 Label4)、4 个文本框 (Text1、Text2、Text3 和 Text4) 和 1 个 ADO Data 控件 (Adodc1), 设置窗体和标签的 Caption 属性, 如图 11-15 所示, 文本框的 Text 属性值为空。

(3) 设置 Adodc1 的属性: 在 Adodc1 上右击, 在弹出的快捷菜单中选择“Adodc 属性”命令, 打开“属性页”对话框。在“通用”选项卡中设置连接字符串 (ConnectionString 属性), 使之连接到“高校学生成绩管理系统.accdb”数据库; 在“记录源”选项卡中, 设置“命令类型” (CommandType 属性) 为 adCmdText, 并在“命令文本 (SQL)”文本框 (RecordSource 属性) 中输入 SQL 语句: “select 成绩.学号,学生.姓名,课程.课程名,成绩.成绩 from 学生,课程,成绩 where 学生.学号=成绩.学号 and 课程.课程号=成绩.课程号”。

(4) 设置 4 个文本框的 DataSource 属性值为 Adodc1, 设置 DataField 属性值分别为学号、姓名、课程名称和成绩。

编写代码:

```
Private Sub Adodc1_MoveComplete(ByVal adReason As ADODB.EventReasonEnum,
    ByVal pError As ADODB.Error, adStatus As ADODB.EventStatusEnum, ByVal
    pRecordset As ADODB.Recordset)
```

```
    '在 ADO Data 控件的标题区显示当前记录的记录号和记录集中的记录总数
    Adodc1.Caption = Adodc1.Recordset.AbsolutePosition & "/"
        & Adodc1.Recordset.RecordCount
```

```
End Sub
```

2. 对表进行管理, 包括浏览、添加、删除和修改记录

【实例 11-2】 编程实现对高校学生成绩管理系统数据库中的成绩表的管理, 设计界面如图 11-16 所示。



图 11-16 实例 11-2 设计界面

界面设计:

(1) 新建工程和窗体, 在工具箱中添加 ADO Data 控件。

(2) 添加 3 个标签 (Label1、Label2 和 Label3)、3 个文本框 (Text1、Text2 和 Text3)、2 个框架 (Frame1 和 Frame2)、8 个命令按钮 (CmdFirst、CmdLast、CmdNext、CmdPrevious、CmdAdd、CmdSave、CmdDelete、CmdQuit) 和 1 个 ADO Data 控件, 窗体、标签、框架和命令按钮的 Caption 属性值如图 11-16 所示, 文本框的 Text 属性值为空。

(3) 设置 Adodc1 属性: 打开“属性页”对话框, 在“通用”选项卡中设置连接字符串, 使之连接到“高校学生成绩管理系统.accdb”数据库; 在“记录源”选项卡中, 设置“命令类型”为 adCmdTable, 设置“表或存储过程名称”为“成绩”。

(4) 设置 3 个文本框的 DataSource 属性值为 Adodc1, 设置 DataField 属性值分别为学号、课程号和成绩。

(5) 为了界面的整洁, 改变 Form1 的大小, 将 Adodc1 隐藏起来。

编写代码:

```
Private Sub CmdFirst_Click()           '移动到第一条记录
    Adodc1.Recordset.MoveFirst
End Sub
Private Sub CmdLast_Click()           '移动到最后一条记录
    Adodc1.Recordset.MoveLast
End Sub
Private Sub CmdPrevious_Click()       '移动到上一条记录
    With Adodc1.Recordset
        .MovePrevious
        If .BOF Then .MoveFirst
    End With
End Sub
Private Sub CmdNext_Click()           '移动到下一条记录
    With Adodc1.Recordset
        .MoveNext
        If .EOF Then .MoveLast
    End With
End Sub
Private Sub CmdAdd_Click()            '添加记录
    Adodc1.Recordset.AddNew
    Text1.SetFocus
End Sub
Private Sub CmdSave_Click()           '保存修改
    Adodc1.Recordset.Update
End Sub
Private Sub CmdDelete_Click()         '删除记录
    Dim msg As String
    Msg = MsgBox("确实要删除该记录吗?", vbQuestion + vbYesNo, "提示")
    If Msg = vbYes Then
```

```

    With Adodcl.Recordset
        .Delete
        .MoveNext
        If .EOF = True Then .MoveLast
    End With
End If
End Sub
Private Sub CmdQuit_Click()           '退出系统
    End
End Sub

```

11.3.2 ActiveX 数据绑定控件

除了内部数据绑定控件，一些 ActiveX 控件也能够实现与数据控件的绑定，常见的 ActiveX 数据绑定控件有 DataList、DataCombo、DataGrid、MsFlexGrid、DBList、DBCombo、DBGrid 等。

知识要点分析

1. DataList 和 DataCombo

DataList 和 DataCombo 被称为数据列表框和数据组合框，与标准的列表框和组合框相比，DataList 和 DataCombo 不再具有 List 属性和 AddItem 方法，其列表项由它们所绑定的数据控件的数据库字段自动填充，并且 DataList 和 DataCombo 还可以有选择地将一个选定的字段传递给另一个数据控件。

DataList 与 DataCombo 相比，两者的差别仅在于：DataCombo 还具有文本框功能，允许用户输入数据。

DataList 和 DataCombo 是 ActiveX 控件，在使用前必须要添加到 VB 工具箱中。方法是：在工具箱任意空白处右击，选择快捷菜单中的“部件”命令，在弹出的“部件”对话框中，选择“控件”选项卡，选中“Microsoft DataList Control 6.0 (OLEDB)”复选框，就可以将它们添加到工具箱中。

DataList 和 DataCombo 控件的常用属性如表 11-3 所示。

表 11-3 DataList 和 DataCombo 控件的常用属性

属性	属性说明
RowSource	设置用于填充 DataList 或 DataCombo 的下拉列表的数据控件
ListField	设置用于填充下拉列表的 RowSource 属性所指定记录集中的字段
BoundColumn	设置回传字段，当在下拉列表中选择某一字段值后将其回传到 DataField，必须和用于更新列表的 DataField 的类型相同
BoundText	返回在 BoundColumn 属性中所指定的字段的值
DataSource	用于设置 DataList 和 DataCombo 所绑定的数据控件的名称
DataField	用于设置 DataList 和 DataCombo 控件所绑定的字段

由以上的属性说明可以看出，在使用 DataList 控件和 DataCombo 控件时，一般要用

到两个数据控件：一个用来确定填充的数据源 RowSource，另一个用来确定更新的数据源 DataSource。

ListField 和 BoundColumn 是 RowSource 记录集中的两个字段，ListField 用于控制在 DataList 和 DataCombo 中显示的内容，其真实的目的是为了在程序运行时得到用户所选列表项的 BoundColumn 属性值，这个值既可以用做查询的条件，也可以用来更新 DataSource 记录集中的数据。

之所以不在 DataList 和 DataCombo 中直接显示 BoundColumn 的值列表，很大的原因是出于明确数据的考虑。例如，在学生表中进行数据录入时，需要输入学生的班级字段值，但班级字段采用的是编码方式，用户可能只清楚班级的名称，而对班级的代码却一无所知，那么就会在输入时产生障碍。如果在选择时程序提供的是班级的名称，但真实得到的却是班级的代码，那么就会使得所做的应用系统更易于操作。

对于 ListField 字段的选择要遵循两个原则：一是含义明确，二是 ListField 字段为候选关键字，其值可以在 RowSource 记录集中唯一地确定一条记录。

2. DataGrid 控件

DataGrid 控件是一种类似于表格的数据绑定控件，可以通过行列交叉的二维表格的形式来显示记录集中的记录和字段。

作为一种 ActiveX 控件，DataGrid 在使用前必须要添加到工具箱中。在“部件”对话框的“控件”选项卡中，选中“Microsoft DataGrid Control 6.0 (OLEDB)”复选框，就可以将 DataGrid 控件添加到工具箱中。

创建 DataGrid 控件的一般步骤为：①在窗体上添加 ADO Data 控件，并设置其 ConnectionString 和 RecordSource 属性；②在同一个窗体上，添加 DataGrid 控件，并设置其 DataSource 属性为上述 ADO Data 控件；③如果在程序运行过程中，DataGrid 控件的数据源始终具有相同的表结构，则在设计阶段还可以直接指定 DataGrid 控件的列标头并设计其外观。方法为：在 DataGrid 控件上右击，在弹出的快捷菜单中选择相应的命令，菜单命令及功能如表 11-4 所示。

表 11-4 DataGrid 的快捷菜单

菜单命令	命令功能
检索字段	在控件中依照 DataSource 记录集的结构自动设置列标头
编辑	重新设置该网格的大小，添加或删除网格的列
属性	将会弹出“属性页”对话框，用以设置网格的外观和行为
清除字段	在控件中清除检索字段操作设置的列标头

在运行时，可以通过如下方法动态调整 DataGrid 控件显示的数据。

1) 改变 DataGrid 控件的 DataSource 的 RecordSource 属性

例如，DataGrid 控件的 DataSource 为一个 ADO Data 控件，则重写 ADO Data 控件的 RecordSource 后，刷新该 ADO Data 控件，将改变 DataGrid 控件中所显示的数据。

2) 将 DataGrid 控件的 DataSource 属性重新设置为一个不同的数据源

例如，在窗体中有多个 ADO Data 控件，每个控件连接不同的数据库或者具有不同的

RecordSource 属性，可以简单地将 DataSource 从一个 ADO Data 控件重新设置为另一个 ADO Data 控件，来实现 DataGrid 控件中不同数据的显示。

应用实例

1. DataList 和 DataCombo

【实例 11-3】 利用 DataCombo 控件实现对实例 11-2 的改写，使得学号和课程号可以直接从列表中选择，无须手工录入，特别当单击“课程”右侧的下拉箭头时，列表中显示的是课程的课程名。设计界面如图 11-17 所示。

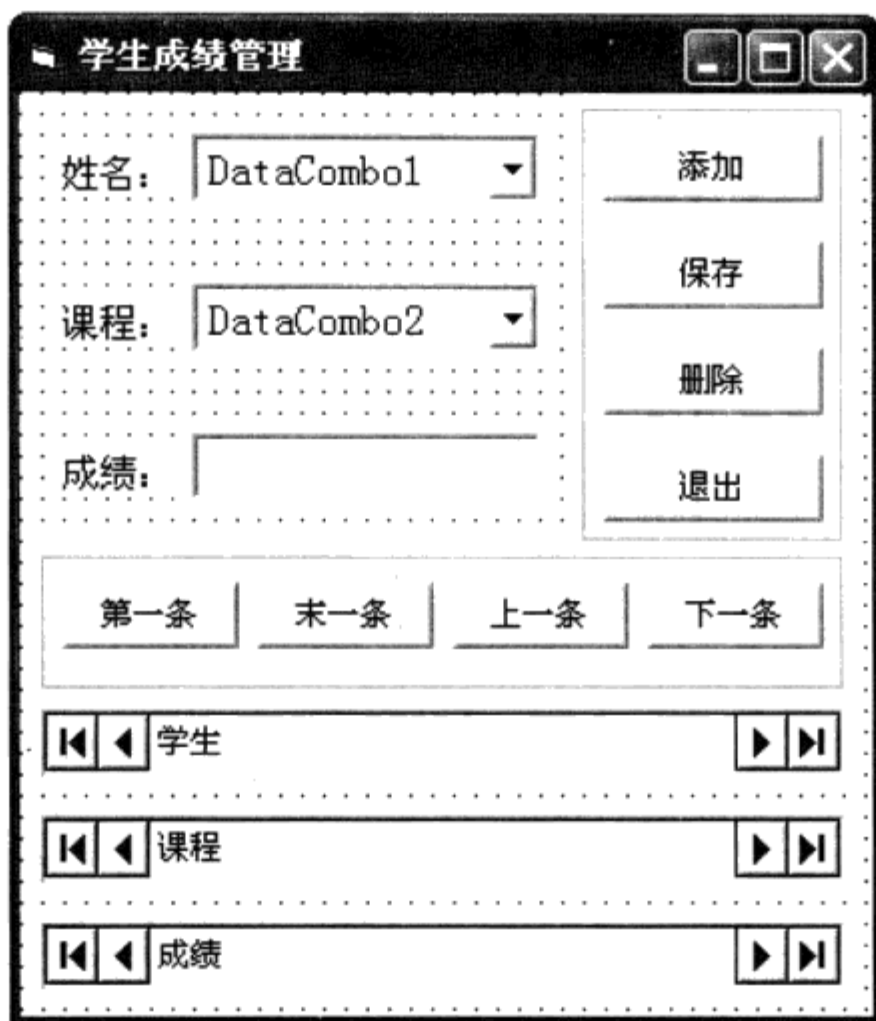


图 11-17 实例 11-3 设计界面

界面设计：

(1) 新建工程和窗体，在工具箱中添加 ADO Data 控件和 DataCombo 控件。

(2) 添加 3 个标签 (Label1、Label2 和 Label3)、2 个 DataCombo (DataCombo1、DataCombo2)、1 个文本框 (Text1)、2 个框架 (Frame1 和 Frame2)、8 个命令按钮 (CmdFirst、CmdLast、CmdNext、CmdPrevious、CmdAdd、CmdSave、CmdDelete、CmdQuit) 和 3 个 ADO Data 控件 (Adodc1、Adodc2 和 Adodc3)，窗体、标签、框架和命令按钮的 Caption 属性值如图 11-17 所示，文本框的 Text 属性值为空。

(3) 设置 Adodc1、Adodc2 和 Adodc3 的 ConnectionString 属性，都与“高校学生成绩管理系统.accdb”数据库相连接，CommandType 均为 adCmdTable，RecordSource 分别设置为“学生”、“课程”和“成绩”。

(4) 设置 DataCombo1 的 RowSource 属性 Adodc1，ListField 属性为“学号”，Boundcolumn

属性为“学号”，DataSource 属性值为 Adodc3，DataField 值为“学号”。

(5) 设置 DataCombo2 的 RowSource 属性 Adodc2，ListField 属性为“课程名”，Boundcolumn 属性为“课程号”，DataSource 属性值为 Adodc3，DataField 值为“课程号”。

(6) 为了界面的整洁，改变 Form1 的大小，将 Adodc1、Adodc2 和 Adodc3 隐藏起来。代码编写参见实例 11-2。

【实例 11-4】 查询学生的单科成绩，设计界面如图 11-18 所示。



图 11-18 实例 11-4 设计界面

界面设计：

(1) 新建工程和窗体，在工具箱中添加 ADO Data 控件和 DataCombo 控件。

(2) 添加 1 个框架 (Frame1 和 Frame2)、3 个标签 (Label1、Label2 和 Label3)、2 个 DataCombo (DataCombo1、DataCombo2)、1 个文本框 (Text1)、1 个命令按钮 (Command1) 和 3 个 ADO Data 控件 (Adodc1、Adodc2 和 Adodc3)，窗体、标签、框架和命令按钮的 Caption 属性值如图 11-18 所示，文本框的 Text 属性值为空。

(3) 设置 Adodc1、Adodc2 和 Adodc3 的 ConnectionString 属性，都与“高校学生成绩管理系统.accdb”数据库相连接，CommandType 均为 adCmdTable，RecordSource 分别设置为“学生”、“课程”和“成绩”。

(4) 设置 DataCombo1 的 RowSource 属性为 Adodc1，ListField 属性为“学号”，Boundcolumn 属性为“学号”。

(5) 设置 DataCombo2 的 RowSource 属性为 Adodc3，ListField 属性为“课程名”，Boundcolumn 属性为“课程号”。

(6) 为了界面的整洁，改变 Form1 的大小，将 Adodc1、Adodc2 和 Adodc3 隐藏起来。编写代码：

```
Private Sub Command1_Click()
    Dim StrQuery As String
    StrQuery = "select * from 成绩 where 成绩.学号 =" & _
        "'" & _DataCombo1.BoundText & "'" & _
        "and 成绩.课程号=" & "'" & DataCombo2.BoundText & "'"
```

```

Adodc3.CommandType = adCmdText
Adodc3.RecordSource = StrQuery
Adodc3.Refresh
If Adodc3.Recordset.RecordCount = 0 Then MsgBox "该生未选修该门课程"
End Sub

```

2. DataGrid

【实例 11-5】 利用数据网格 DataGrid 管理成绩表中的数据。设计界面如图 11-19 所示。

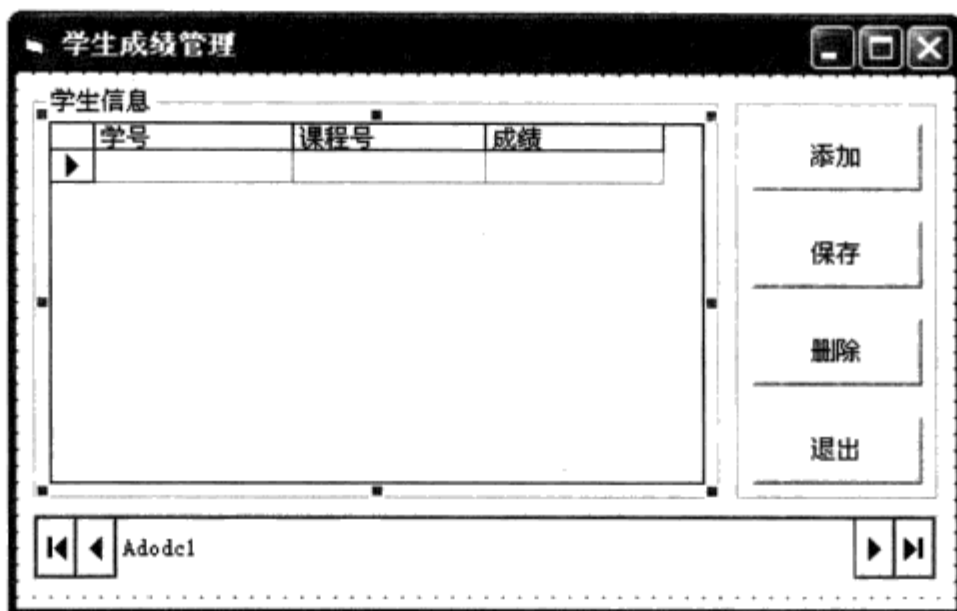


图 11-19 实例 11-5 设计界面

界面设计：

(1) 新建工程和窗体，在工具箱中添加 ADO Data 控件和 DataGrid 控件。

(2) 在窗体上添加 1 个 ADO Data 控件 (Adodc1)、2 个框架 (Frame1 和 Frame2)、1 个数据网格 (DataGrid1)、4 个命令按钮 (CmdAdd、CmdSave、CmdDelete、CmdQuit)，窗体、框架和命令按钮的 Caption 属性设置如图 11-19 所示。

(3) 设置 Adodc1 的 ConnectionString 属性，与“高校学生成绩管理系统.accdb”数据库相连接，CommandType 为 adCmdTable，RecordSource 设置为“成绩”。

(4) 设置数据网格 DataGrid1 的 DataSource 属性值为 Adodc1，右击数据网格，在快捷菜单中选择“检索字段”，用新的字段替换原有的网格布局；右击数据网格，选择“编辑”，调整行高和列宽直到布局合适；右击数据网格，选择“属性”，在打开的“属性页”对话框中设置字体和对齐等。

(5) 为了界面的整洁，改变 Form1 的大小，将 Adodc1 隐藏起来。

代码编写参看实例 11-2。

【实例 11-6】 使用 DataGrid 控件实现男生表和成绩表的切换，设计界面如图 11-20 所示。

界面设计：

(1) 新建工程和窗体，在工具箱中添加 ADO Data 控件和 DataGrid 控件。

(2) 在窗体上添加 1 个 ADO Data 控件 (Adodc1)、1 个数据网格 (DataGrid1)、3 个命令按钮 (CmdStudent、CmdTeacher、CmdQuit)，窗体、框架和命令按钮的 Caption 属性设置如图 11-20 所示。



图 11-20 实例 11-6 设计界面

(3) 设置 Adodc1 的 ConnectionString 属性, 与“高校学生成绩管理系统.accdb”数据库相连接, CommandType 为 adCmdTable, RecordSource 设置为“学生”。

(4) 设置数据网格 DataGrid1 的 DataSource 属性值为 Adodc1。

编写代码:

```
Private Sub CmdStudent_Click()
    Form1.Caption = "男学生表"
    Dim StrQuery As String
    Adodc1.CommandType = adCmdText
    StrQuery = "select * from 学生 where 性别='男'"
    Adodc1.RecordSource = StrQuery
    Adodc1.Refresh
End Sub

Private Sub CmdTeacher_Click()
    Form1.Caption = "成绩表"
    Adodc1.CommandType = adCmdTable
    Adodc1.RecordSource = "成绩"
    Adodc1.Refresh
End Sub

Private Sub CmdQuit_Click()
    End
End Sub
```

本例中有两点需要特殊说明。

(1) ADO Data 控件的 CommandType 属性设置应与 RecordSource 的类型相吻合: 如果 RecordSource 为表, 则 CommandType 应设置为 adCmdTable; 如果 RecordSource 为查询, 则 CommandType 应设置为 adCmdText, 否则将会产生错误。

(2) 如果在运行时需要动态调整 DataGrid 控件显示的数据, 建议在设计时不要进行 DataGrid 控件的检索字段操作。

3. DataList、DataCombo 与 DataGrid 相结合

【实例 11-7】 利用 DataCombo 与 DataGrid, 查询学生的选课成绩, 并统计该生的已修学分。界面设计如图 11-21 所示。

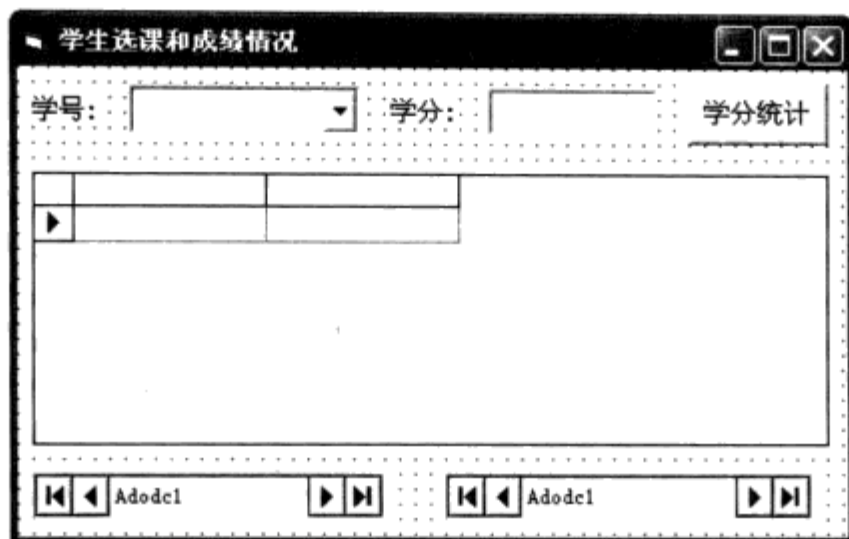


图 11-21 实例 11-7 设计界面

界面设计:

(1) 新建工程和窗体, 在工具箱中添加 ADO Data 控件、DataCombo 控件和 DataGrid 控件。

(2) 在窗体上添加 2 个 ADO Data 控件 (Adodc1、Adodc2)、1 个 DataCombo 控件 (DataCombo1)、1 个文本框 (Text1)、1 个命令按钮 (Command1) 和 1 个 DataGrid 控件 (DataGrid1), 窗体、框架和命令按钮的 Caption 属性设置如图 11-20 所示, 文本框的 Text 属性值为空且 Locked 属性为 True。

(3) 设置 Adodc1 和 Adodc2 的 ConnectionString 属性, 均与“高校学生成绩管理系统.accdb”数据库相连接, CommandType 为 adCmdText, Adodc1 的 RecordSource 设置为命令文本: “Select 成绩.学号,学生.姓名,课程.课程名,课程.学分,成绩.成绩 From 学生,课程,成绩 Where 学生.学号=成绩.学号 And 成绩.课程号=课程.课程号”, Adodc2 的 RecordSource 属性设置为命令文本: “Select Distinct 学号 From 成绩”。

(4) 设置数据组合框 DataCombo1 的 RowSource 属性为 Adodc2, ListField 属性为“学号”, Boundcolumn 值为“学号”。

(5) 设置数据网格 DataGrid1 的 Datasource 属性值为 Adodc1。

(6) 为了界面的整洁, 改变 Form1 的大小, 将 Adodc1、Adodc2 隐藏起来。

```
Private Sub DataCombo1_Click(Area As Integer)
    Dim StrQuery As String
    StrQuery = "Select Distinct 成绩.学号,学生.姓名,课程.课程号,课程.课程名" & _
        ",成绩.成绩,成绩.学分 From 学生,课程,成绩 Where " & _
        "学生学号=成绩.学号 And 成绩.课程号=课程.课程号 And " & _
        "成绩.学号 = " & "'" & DataCombo1.BoundText & "'"
    Adodc1.RecordSource = StrQuery
    Adodc1.Refresh
    Text1.Text = ""
End Sub
Private Sub Command1_Click()
    Dim xf As Integer
    Adodc1.Recordset.MoveFirst
```

```

Do While Not Adodcl.Recordset.EOF
  If Adodcl.Recordset.Fields("成绩") >= 60 Then
    xf = xf + Val(Adodcl.Recordset.Fields("学分"))
    Adodcl.Recordset.MoveNext
  End If
Loop
Text1.Text =xf
End Sub

```

11.4 ADO 对象模型

在实际的编程过程中，ADO 对象模型要比 ADO Data 控件更为常用，使用起来更为灵活、方便。

在使用 ADO 对象模型前，首先要向当前工程添加 ADO 的对象库，添加的方法为：使用“工程”菜单中的“引用”命令，将会打开“引用”对话框，在“可用的引用”列表选中 Microsoft ActiveX Data Object 2.1 Library 复选框，如图 11-22 所示。

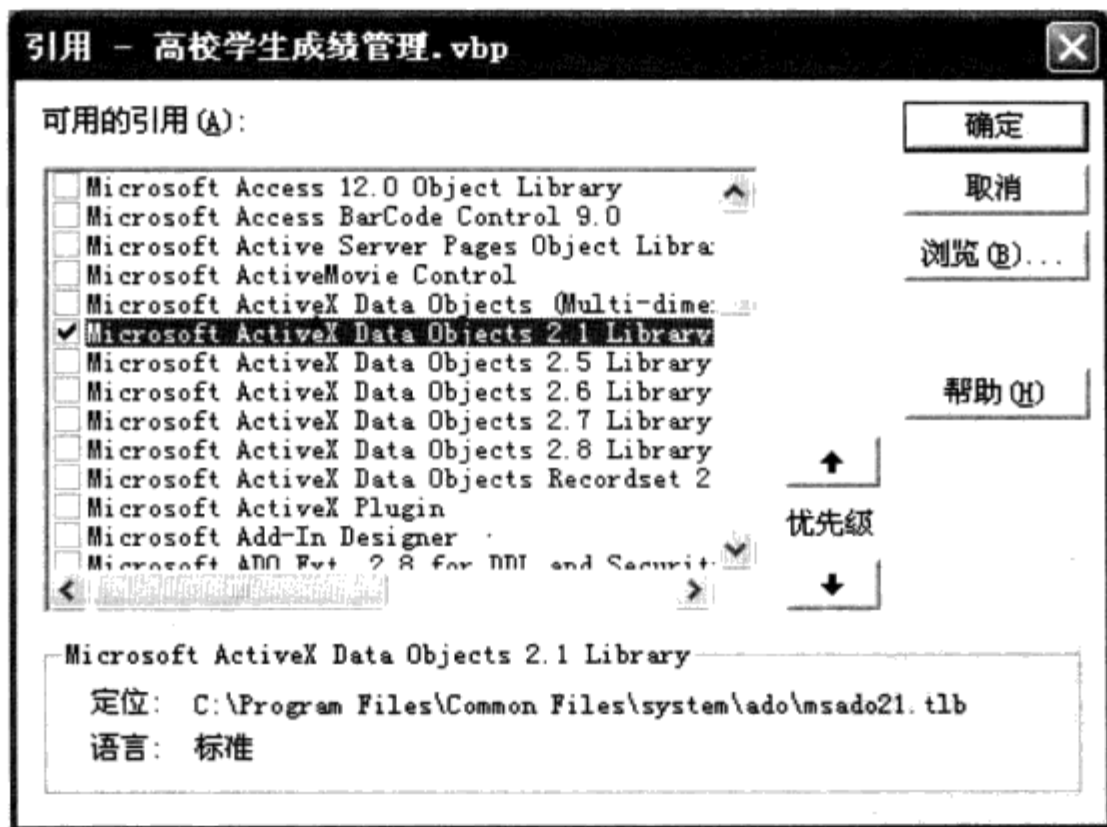


图 11-22 “引用”对话框

☎ 知识要点分析

ADO 对象模型定义了一个可编程的分层对象集合，如图 11-23 所示。

ADO 对象模型的核心是 Connection、Command 和 Recordset 对象。

1. Connection 对象

Connection 对象用于建立与数据源的连接，借以访问数据库。通过设置其 Provider 属性，可以指定 OLE DB 的提供者。通过设置其ConnectionString 属性，可以指定数据源。

利用 `Open` 方法，来建立到数据源的物理连接。利用 `Close` 方法，可以切断到数据源的物理连接。

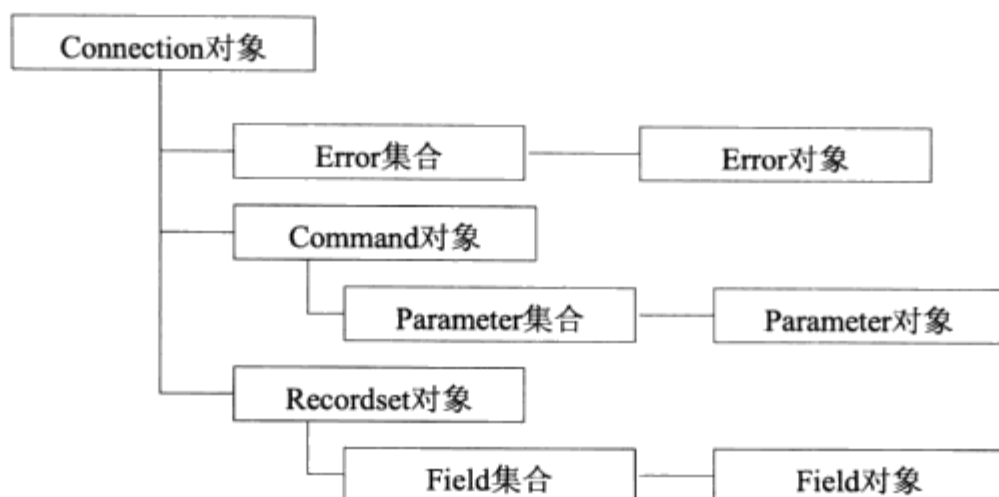


图 11-23 ADO 编程模型的对象关系

例如，使用如下代码可以建立与高校学生成绩管理系统.accdb 数据库的连接：

```
Dim cnnstu As New ADODB.Connection
cnnstu.Provider = "Microsoft.ACE.OLEDB.12.0"
cnnstu.ConnectionString = "Data Source=c:\gxgl\高校学生成绩管理系统.accdb"
cnnstu.Open
```

2. Command 对象

`Command` 对象用于查询数据库，并返回 `Recordset` 对象。通过设置其 `ActiveConnection` 属性，可以设置 `Command` 对象到数据源的连接信息，即所属的 `Connection` 对象。通过设置其 `CommandText` 属性可以指定命令（SQL 语句）的可执行文本，通过其 `Parameter` 对象和 `Parameter` 集合可以定义参数化查询或存储过程参数。利用 `Execute` 方法，执行命令并返回 `Recordset` 对象。

例如，在上例基础上使用 `Command` 对象返回学生表，可以使用如下代码来实现：

```
Dim cmmstu As New ADODB.Command
Dim rsstu As New ADODB.Recordset
Set cmmstu.ActiveConnection = cnnstu
cmmstu.CommandText="Select * From 学生"
Set rsstu=cmmstu.Execute
```

3. Recordset 对象

`Recordset` 对象代表记录集，是基于某一连接的表或者是 `Command` 对象的执行结果。所有对数据源的操作基本上都是通过 `Recordset` 对象来完成的，例如，记录的定位、移动、添加、删除和更改等。通过设置其 `ActiveConnection` 属性：可以设置 `Recordset` 对象所属的 `Connection` 对象。通过设置其 `Source` 属性可以设置或返回 `Recordset` 对象的生成方式。其他属性（如 `RecordCount` 属性、`BOF` 和 `EOF` 属性、`Bookmark` 属性等）含义和功能均与 `ADO Data` 控件的记录集属性相同，在此不再赘述。

使用 `Recordset` 对象的 `Open` 方法，可以打开数据表或查询结果的记录集。使用 `Close` 方法，可以关闭 `Recordset` 对象。其他方法（如 `AddNew` 方法、`Delete` 方法、`Update` 方法、

CancelUpdate 方法、Move 方法、Move 方法群等)的使用与 ADO Data 控件的记录集相同。

例如,上述使用 Command 对象返回学生表的例子,也可以在建立连接的基础上,采用 Recordset 对象来实现,代码如下:

```
Dim rsstu As New ADODB.Recordset
Set rsstu.ActiveConnection = cnnstu
rsstu.Open "select * from BookItem"
```

需要说明的是,本节中涉及了一些有关对象变量声明和使用的方法,限于篇幅本书不再展开,建议读者参看有关的技术文档。

应用实例

【实例 11-8】 利用 ADO 对象模型编程实现对班级表的管理,可以进行记录的添加、删除和浏览,设计界面如图 11-24 所示。

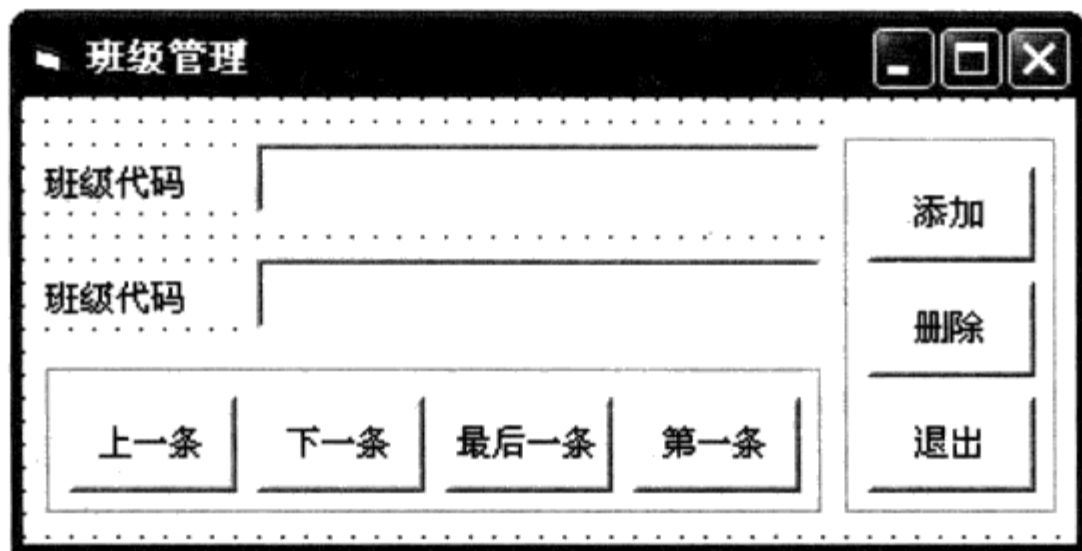


图 11-24 实例 11-8 程序运行界面

界面设计:

- (1) 新建工程和窗体,添加对 ADO 对象函数库的引用。
- (2) 在窗体上添加 2 个标签 (Label1、Label2)、2 个文本框 (Text1、Text2)、2 个框架 (Frame1、Frame2)、3 个命令按钮 (CmdAdd CmdDelete CmdQuit) 和 1 个命令按钮组 (CmdMove, 含 4 个按钮),各控件的 Caption 属性如图 11-24 所示,文本框的 Text 属性值为空。

编写代码:

```
'在窗体的通用声明段中,声明 Connection 和 Recordset 对象变量
Option Explicit
Dim cnnstu As New ADODB.Connection
Dim rsstu As New ADODB.Recordset
Private Sub Form_Load()
    '建立与高校学生成绩管理系统.accdb 数据库的连接
    cnnstu.Provider = "Microsoft.ACE.OLEDB.12.0"
    cnnstu.ConnectionString = "Data Source=c:\gxgl\高校学生成绩管理系统.accdb"
```

```
cnnstu.Open
'定义操作的记录集
rsstu.LockType = adLockOptimistic
rsstu.CursorType = adOpenKeyset
Set rsstu.ActiveConnection = cnnstu
rsstu.Open "select * from 班级"
'设置文本框与数据源中字段的绑定
Set Text1.DataSource = rsstu
Text1.DataField = "班级代码"
Set Text2.DataSource = rsstu
Text2.DataField = "班级名称"
End Sub
Private Sub CmdAdd_Click()
    rsstu.AddNew
End Sub
Private Sub CmdDelete_Click()
    Dim response As Integer
    response = MsgBox("是否删除当前记录?", vbYesNo, "删除记录")
    If response = vbYes Then
        rsstu.Delete
        rsstu.MoveFirst
    End If
End Sub
Private Sub CmdQuit_Click()
    End
End Sub
Private Sub CmdMove_Click(Index As Integer)
    With rsstu
        Select Case Index
            Case 0
                .MovePrevious
                If .BOF Then .MoveFirst
            Case 1
                .MoveNext
                If .EOF Then .MoveLast
            Case 2
                .MoveLast
            Case 3
                .MoveFirst
        End Select
    End With
End Sub
```

本例涉及 Recordset 对象的 LockType 属性和 CursorType 属性。

LockType 属性用于设置 Recordset 对象在编辑过程中使用的锁定类型，值若为

adLockOptimistic, 表示开放式记录锁定, 只在调用 Update 方法时锁定记录; 值若为 adLockedOnly, 表示只读, 无法更改数据, adLockedOnly 是默认值。

CursorType 属性用于设置 Recordset 对象使用的游标类型。值若为 adOpenDynamic, 表示动态游标, 可见其他用户所做的添加、删除和更改, 允许在记录集中进行所有类型的移动, 但不包括提供者不支持的书签操作; 值若为 adOpenKeyset, 表示键集游标, 与动态游标相似, 但无法查看其他用户添加的记录。

11.5 数据报表设计

对于一个完整的数据库应用程序来说, 制作并打印报表是不可缺少的环节。VB 提供了 DataReport 对象作为数据报表设计器, 不仅具有强大的功能, 而且还简单易学。DataReport 对象可以从任何数据源创建报表, 包括数据环境。

11.5.1 数据环境设计器

☎ 知识要点分析

数据环境设计器为建立与数据库的连接和定义命令提供图形接口, 利用数据环境设计器可以创建和修改表、表集和报表的数据环境, 数据环境设计器保存在扩展名为.dsr 的文件中。

1. 向工程中添加数据环境

使用“工程”菜单的“添加 Data Environment”命令, 可在当前工程添加一个数据环境, 并同时打开数据环境设计器窗口, 默认该窗口中包含一个 Connection 对象, 如图 11-25 所示。选择 DataEnvironment 对象或 Connection 对象, 可以在属性窗口为其重新命名。



图 11-25 数据环境设计器

2. 创建与数据源的连接

数据环境设计器中的 Connection 对象用于建立与数据源的连接, 以实现数据的访问。

定义一个 Connection 对象的方法为，在数据环境设计器中，右击 Connection 对象，在弹出的快捷菜单中选择“属性”命令，打开如图 11-26 所示的“数据链接属性”对话框。其响应方法与设置 ADO Data 控件的 ConnectionString 属性完全相同。

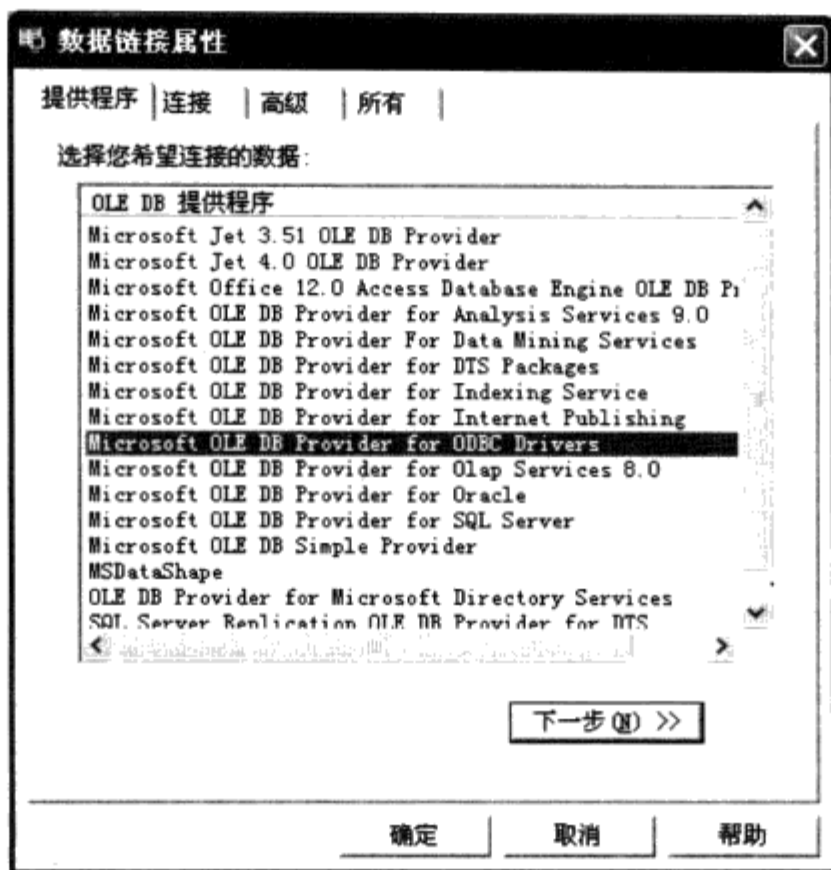


图 11-26 “数据链接属性”对话框

3. 创建命令对象

数据环境设计器中的 Command 对象用于定义将对数据源执行的命令，使用 Command 对象可以查询数据库并返回 Recordset 对象中的记录，从而实现对数据库的操作。在数据环境设计器中，右击 Connection 对象，在弹出的快捷菜单中选择“添加命令”，便可以添加一个命令对象，如图 11-27 所示。选择该 Command 对象，在属性窗口中可以为它重新命名。

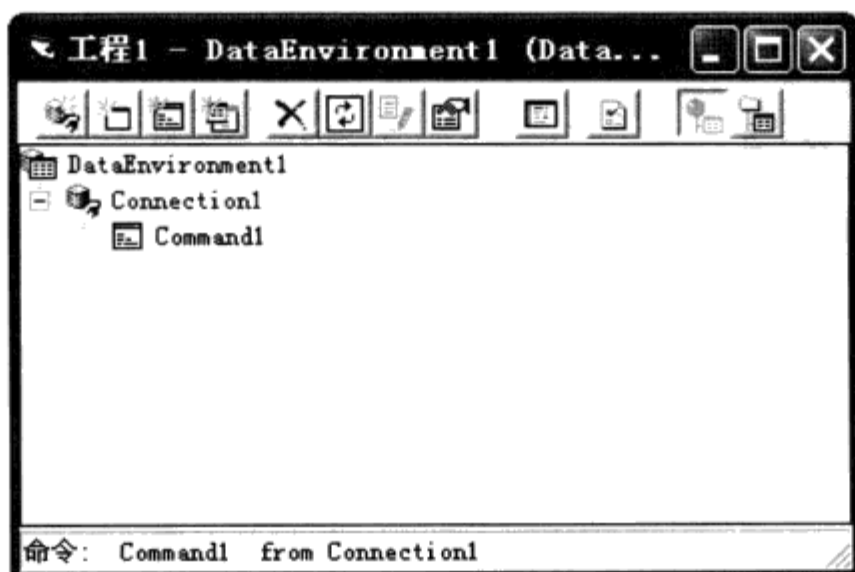


图 11-27 添加了命令对象的数据环境设计器

右击 Command 对象，在弹出的快捷菜单中选择“属性”命令，将会打开“属性”对

对话框，如图 11-28 所示。在“通用”选项卡中，设置其所使用的连接及数据源（数据库或 SQL 语句），为使报表分组打印，还可在“分组”选项卡中，选中“分组命令对象”复选框，并设置“用于分组的字段”。

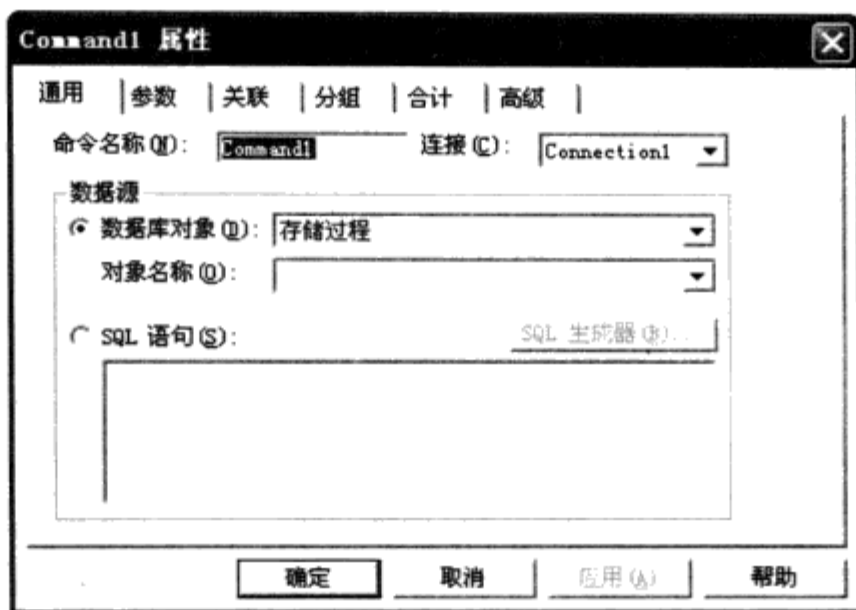


图 11-28 Command 对象的“属性”对话框

4. 创建用户界面

使用数据环境设计器中的 Command 对象创建用户界面非常简单，只要打开窗体，从数据环境设计器中把需要的命令对象或命令对象中的某些字段拖到这个窗体，在该窗体中就会自动添加控件，并自动与该命令对象中定义的所有字段或指定字段实现绑定。

11.5.2 数据报表设计器

☎ 知识要点分析

1. 添加 DataReport 对象

使用“工程”菜单中的“添加 Data Report”，可以在当前工程中添加一个数据报表，同时自动打开报表设计器，如图 11-29 所示。



图 11-29 数据报表 DataReport1 窗口

2. 数据报表设计器的构成

数据报表设计器由以下各部分构成。

(1) 报表标头：指显示在报表开始处的文本，可以用来显示报表标题、作者或数据库名等信息。

(2) 页标头：指在每一页顶部显示的信息，用来显示每页的报表标题。

(3) 细节：指报表显示的最内部的记录数据的“重复”部分，与数据环境层次结构中最底层的 Command 对象相关联，细节区的高度将决定报表的行高。

(4) 页注脚：指在每一页底部显示的信息，常用来显示页码。

(5) 报表注脚：指在报表结束处显示的文本，例如用来显示摘要信息或一个地址或联系人姓名。一个报表只有一个报表注脚，出现在最后一个页注脚之前。

(6) 分组标头、注脚：分组标头与分组注脚成对出现，用于显示分组字段或者分类汇总结果。默认情况下，添加的数据报表没有分组标头和分组注脚两部分，这两个部分只有在数据报表设计器的数据源有分组的情况下才会出现。

3. 数据报表工具箱

当向工程中添加了数据报表设计器后，工具箱上会自动出现一个“数据报表”选项卡，它包含了一些只能在数据报表设计器中使用的特殊控件，如图 11-30 所示。

(1) RptLabel 控件：用于在报表上放置标签，显示静态文本。

(2) RptTextBox 控件：用于在报表上连接并显示字段数据。

(3) RptImage 控件：用于在报表上放置图形，该控件不能被绑定到数据字段。

(4) RptLine 控件：用于在报表上绘制直线。

(5) RptShape 控件：用于在报表上绘制矩形、三角形或圆形（椭圆）。

(6) RptFunction 控件：用于在报表上建立公式，报表生成时自动计算数值。



图 11-30 “数据报表”工具箱

注意：在数据报表设计器中，不能使用 VB 提供的任何内部控件或 ActiveX 控件。

4. DataReport 对象的常用属性

DataReport 核心的属性有两个，DataSource 属性用于指定数据源，DataMember 属性用于指定报表所显示的记录集数据。

应用实例

【实例 11-9】 设计报表按照学号显示学生的成绩和平均成绩。

设计步骤如下所示。

1. 用数据环境设计器设计数据源

新建工程，打开窗体设计器，使用“工程”菜单中的“添加 Data Environment”命令添加一个数据环境（DataEnvironment1），并同时打开数据环境设计器。

设置 Connection1 与“高校学生成绩管理系统.accdb”数据库相连接，并在 Connection1 下创建一个 Command 对象 (Command1)，在 Command1 上右击，选择快捷菜单中的“属性”命令，在弹出的“Command1 属性”对话框中，设置数据对象为“成绩表”，在“分组”选项卡中设置分组字段为“学号”，在“合计”选项卡中添加“平均成绩”汇总项。设置完成后，数据环境设计器如图 11-31 所示。

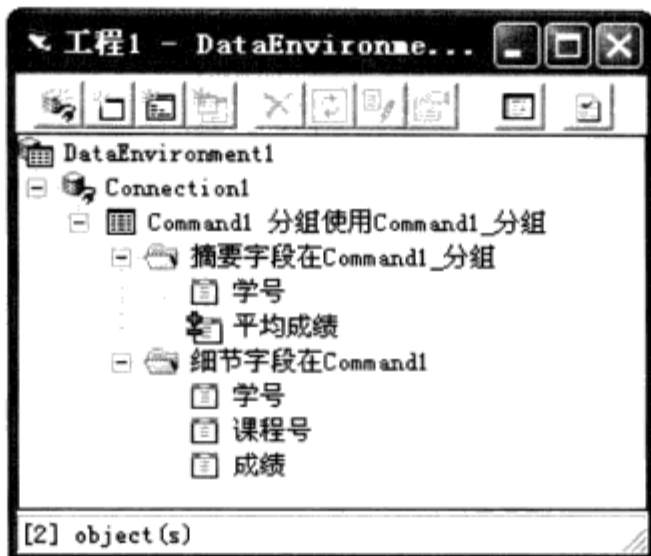


图 11-31 实例 11-9 数据环境设计器

2. 添加数据报表

使用“工程”菜单中的“添加 Data Report”命令，在当前工程中添加一个数据报表 (DataReport1)，同时打开数据报表设计器窗口。

3. 设置 DataReport 对象属性

设置 DataReport1 的 DataSource 的属性为“DataEnvironment1”，DataMember 的属性为“Command1_分组”。

4. 检索结构

在 DataReport1 上右击，在弹出的快捷菜单中选择“检索结构”命令，用新的数据层次代替旧的报表布局。

5. 添加控件

(1) 将数据环境设计器中摘要字段部分的“学号”拖动到数据报表设计器中的分组标头区域，将“摘要字段”部分的“平均成绩”拖动到分组注脚区域，每个字段都会显示标签 (RptLabel 控件) 和文本 (RptTextBox 控件)，前者用于显示字段名，后者用于显示字段值。拖动细节字段部分的“课程号”和“成绩”到细节区域，然后把两个字段的标签部分都拖放到分组标头区域。

(2) 在“报表标头”区域右击，从弹出的快捷菜单中选择“插入控件”中的“报表标题”命令，插入报表标题，其实质为一个 RptLabel 控件，设置其 Caption 属性为“学生成绩分析表”。

(3) 在“页标头”区域右击，从弹出的快捷菜单中选择“插入控件”中的“当前页码”命令，设置其 Caption 属性为“第&p 页”，调整至合适的位置。

6. 设置布局

调整每个控件的大小、位置及格式，调整报表各部分的高度。在格式的反复调整中，

可能需要经常预览报表的效果，为了操作的方便，可以将报表设置为启动对象，具体的方法为：使用“工程”菜单中的“工程属性”命令，在打开“工程属性”对话框中将“启动对象”设置为 DataReport1，每次运行工程后，都会显示数据报表，注意在调试结束后应将启动对象恢复为正确的值。设计完成后的报表设计器界面如图 11-32 所示。

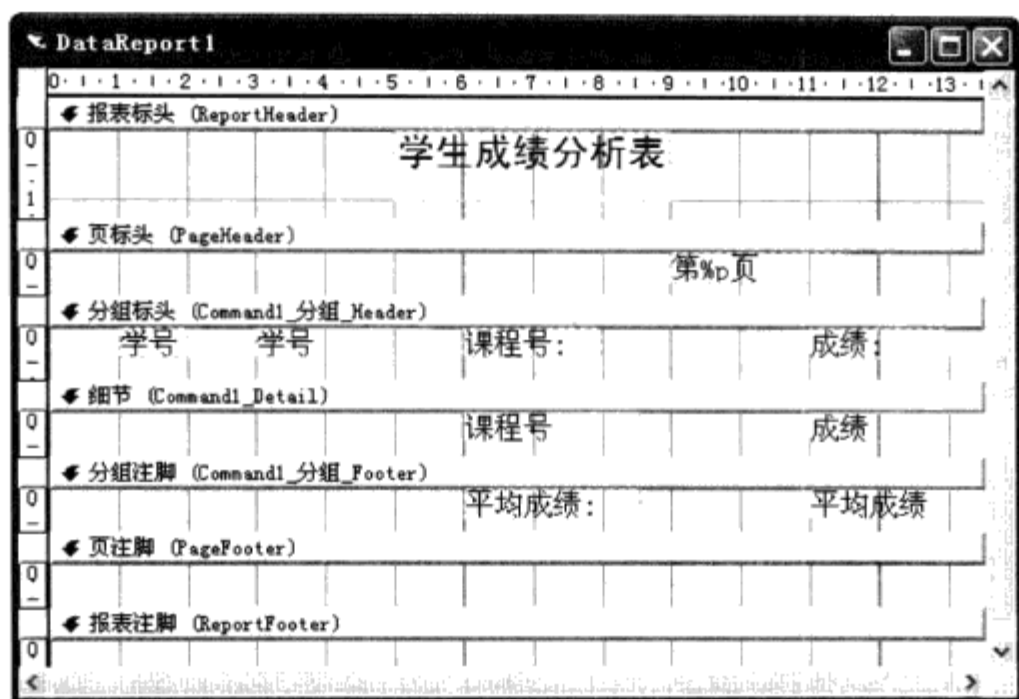


图 11-32 实例 11-9 报表设计器

7. 显示及打印报表

使用 DataReport 对象的 Show 方法可以实现数据报表的显示。

语法：

```
DataReport1.Show
```

报表预览的效果如图 11-33 所示，单击工具栏上的“打印”按钮，可以实现报表的打印。

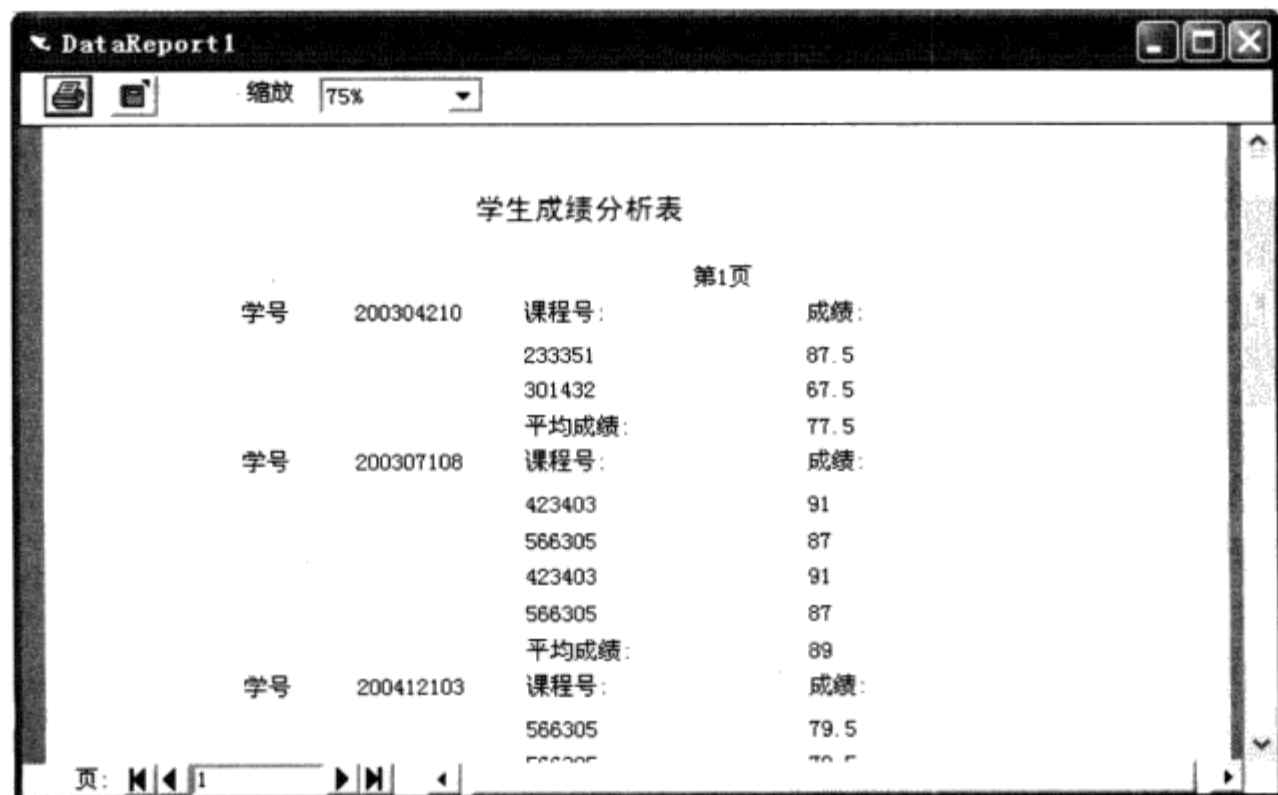


图 11-33 实例 11-9 报表预览

11.6 本章小结

在本章中，首先介绍了 VB 数据库访问技术的基础知识，包括 VB 可访问数据库的类型、VB 数据库访问的底层技术和 VB 的三种数据访问接口，然后结合数据绑定控件详细介绍了 ADO Data 控件和 ADO 对象模型的使用，最后介绍了数据环境设计器在数据报表中的使用和数据报表 (DataReport) 的设计、运行及打印。

习 题 11

一、判断题

1. 使用 ADO Data 控件既可以与数据库建立连接，也可以对数据库中的数据进行显示和编辑。()
2. 设置 ADO Data 控件的数据源可以通过以下三种方法来实现：使用 Data Link 文件、使用 ODBC 数据资源或使用连接字符串。()
3. 在编程过程中可以直接使用数据绑定控件和数据库建立连接。()
4. 数据绑定控件是一些 ActiveX 控件，文本框不能作为数据绑定控件。()
5. ADO Data 控件是 Visual Basic 的内部控件，可直接使用。()
6. ADO Data 控件的 RecordSource 属性只能设置为某个表。()
7. 当记录集对象的 BOF 属性值为 True，表示记录指针指向了第一条记录。()
8. 当对记录集中的记录进行了删除操作以后，应及时使用 Move 方法群组调整记录指针的位置，以保证记录指针指向一个有效的记录实体。()
9. 内部数据绑定控件有两个标准的属性：DataSource 和 DataField 属性。()
10. DataList 和 DataCombo 是 ActiveX 控件，在使用之前需要先添加到控件工具箱中。()
11. 数据网格 DataGrid 控件有两个标准的属性：DataSource 和 DataField 属性。()
12. ADO 对象模型是通过编程来实现对数据库中表的显示和编辑操作。()
13. ADO 对象模型的核心是 Connection、Command 和 Recordset 对象。()
14. 在 Visual Basic 中，数据报表不支持 Show 方法。()
15. 数据环境设计器能够为数据报表 DataReport 提供了数据源。()

二、单选题

1. 在“属性页”对话框中，设置 ADO Data 控件与数据源的连接的方法有 () 种。
A. 1 B. 2 C. 3 D. 4
2. 要利用 ADO Data 控件返回数据库中的记录集，则需设置 () 属性。
A. Connect B. DataSource C. RecordSource D. RecordType
3. 设置 ADO Data 控件的记录源为 SQL 语句的执行结果，则命令类型应选择 ()。
A. adCmdText B. adCmdTable C. adCmdStoreProc D. adCmdUnknown

4. 下列 () 控件不能作为数据绑定控件。
A. Frame B. TextBox C. DataGrid D. Image
5. DataList 和 DataCombo 控件中的列表数据是通过 () 属性自动填充的。
A. DataSource 和 DataField B. RowSource 和 ListField
C. DataSource 和 ListField D. BoundColumn 和 BoundText
6. 文本框的 DataField 属性用来指定文本框所要绑定的 ()。
A. 字段 B. 记录 C. 数据表 D. 数据库
7. 记录集对象的 EOF 属性值为 True, 说明记录指针处于 ()。
A. 第一条记录 B. 第一条记录前 C. 最后一条记录 D. 最后一条记录后
8. 下列关于数据环境设计器的描述中错误的是 ()。
A. 数据环境设计器为数据访问提供了一个交互式环境
B. 它创建的数据环境对象包括 Connection、Command 等 ADO 对象
C. 数据环境设计器只能为数据报表提供数据源
D. 数据环境设计器保存在扩展名为.dsr 的文件中
9. 以下说法正确的是 ()。
A. 使用 ADO Data 可以直接显示数据库中的数据
B. 使用数据绑定控件可以直接访问数据库中的数据
C. 使用 ADO Data 控件可以对数据库中的数据进行操作, 却不能显示数据库中的数据
D. ADO Data 控件只有通过绑定控件才可以访问数据库中的数据
10. 控件 Adodc1 已和高校学生成绩管理数据库建立了连接, 并指定其记录源为学生表, 现使用如下代码实现对男生人数的统计, 请填空完成。

```
Adodc1.Recordset.MoveFirst
Do While ( )
    If Adodc1.Recordset.Fields("成绩") >= 60 Then
        xf = xf + Val(Adodc1.Recordset.Fields("学分"))
        Adodc1.Recordset.MoveNext
    End If
Loop
```

- A. Not Adodc1.Recordset.EOF B. Not Adodc1.Recordset.EOF
C. Adodc1.Recordset.EOF D. Adodc1.Recordset.BOF

[G e n e r a l I n f o r m a t i o n]

书名= V i s u a l B a s i c 程序设计+ A c c e s s 2 0 0 7 数据库应用系统开发案例教程

作者= 曹风华, 张利军主编

页数= 2 5 9

出版社= 北京市: 清华大学出版社

出版日期= 2 0 1 2 . 0 2

S S 号= 1 2 9 3 0 0 7 1

D X 号= 0 0 0 0 0 8 2 4 7 4 3 9

URL= <http://book.szdnnet.org.cn/bookDetail.jsp?dxNumber=000008247439&d=636A354DFB1A454E672D99E009BA5FF5>

7 2 D 9 9 E 0 0 9 B A 5 F F 5