全国计算机等级考试系列教程

二级 Visual FoxPro 程序设计

刘瑞新　主编

汪远征　徐雅静　江　涛　编著

机械工业出版社

本书内容紧扣《全国计算机等级考试二级考试大纲（Visual FoxPro 程序设计）》，通过大量示例，深入浅出地介绍了数据库基础、Visual FoxPro 简介、数据及其运算、数据表与数据库、查询与视图、关系数据库标准语言 SQL、程序设计基础、表单设计基础、常用的标准控件、菜单设计与应用、报表、上机考试技巧等。全书讲解简明扼要，层次分明；理论联系实际，面向应试。各章之后均附有大量的习题，包括等级考试笔试中常见的选择题、填空题以及上机考试中常见的应用（编程）题，且书后配有各章习题参考答案。

本书图文并茂，所有操作都依实际屏幕显示一步一步讲述，读者可以边看书边上机操作，通过范例和具体操作，理解基本概念和学会操作方法。

本书可以作为高等院校非计算机专业程序设计课程的教材，也可以作为各类计算机培训班的教学用书，还可以作为各类应试人员的学习用书。对于计算机应用人员和计算机爱好者，本书也是一本实用的参考书。

图书在版编目（CIP）数据

二级 Visual FoxPro 程序设计/刘瑞新主编. —北京：机械工业出版社，2003.5
（全国计算机等级考试系列教程）

ISBN 7-111-11965-7

Ⅰ. 二… Ⅱ. 刘… Ⅲ. 关系数据库—数据库管理系统，Visual FoxPro—程序设计—水平考试—自学参考资料 Ⅳ. TP311.138

中国版本图书馆 CIP 数据核字（2003）第 028601 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策 划：胡毓坚
责任编辑：田 梅
责任印制：

·新华书店北京发行所发行

2003 年 5 月第 1 版·第 1 次印刷
787mm×1092mm 1/16·23.75 印张·587 千字
0001—5000 册
定价：33.00 元

凡购本图书，如有缺页、倒页、脱页，由本社发行部调换
本社购书热线电话（010）68993821、88379646
封面无防伪标均为盗版

前　言

为了促进我国计算机知识的普及，提高全社会的计算机应用水平，适应国民经济信息化的需要，教育部考试中心举办了“全国计算机等级考试”。为社会提供了一个统一、公正和客观的考核标准，深受社会各界的欢迎。自1994年举办以来，应试人数逐年增加，为计算机的普及和应用起到了十分重要的作用。

为了适应计算机发展和实际应用的需要，教育部考试中心在2002年适时地调整了新的考试大纲并扩大了考试范围。根据教育部考试中心最新制定的《全国计算机等级考试二级考试大纲（Visual FoxPro程序设计）》（简称大纲），作者编写了这本等级考试教程。

Visual FoxPro是Microsoft公司推出的关系型数据库系统，又是一款基于Windows平台的程序开发工具。该系统不仅可以简化数据管理，而且使应用程序的开发流程更为合理。Visual FoxPro提供了一个集成化的开发环境，使组织数据、定义数据库规则和建立应用程序等工作变得简单易行。利用可视化的设计工具和向导，可以快速创建表单、查询和报表。

本书共分12章，内容包括数据库基础、Visual FoxPro简介、数据及其运算、数据表与数据库、查询与视图、关系数据库标准语言SQL、程序设计基础、表单设计基础、常用标准控件、菜单设计与应用、报表、上机考试技巧等。全书内容紧扣《大纲》，讲解简明扼要，层次分明；理论联系实际，面向应试。各章之后均附有大量的习题，包括等级考试笔试中常见的选择题、填空题以及上机考试中常见的应用（编程）题。

本书图文并茂，所有操作都按实际屏幕显示一步一步讲述，读者可以一边看书，一边上机操作，通过范例和具体操作，理解基本概念和学会操作方法。

本书可以作为高等院校非计算机专业程序设计课程的教材，也可以作为各类计算机培训班的教学用书，还可以作为各类应试人员的学习用书。对于计算机应用人员和计算机爱好者，本书也是一本实用的参考书。

本书由刘瑞新主编，汪远征、徐雅静、江涛编写，另外陈东升、刘志都、臧顺娟、秦建国、李黎、董敏红、罗全胜、郭金良、桑继耀、谭瑞梅、王霞、韩献军、朱云、杨杰等也参加了资料整理与校对工作。由于时间仓促，书中不足之处，请广大读者批评指正。

编　者

目　录

第 1 章　数据库基础

数据库是数据库应用程序的核心。本章首先介绍数据库的基本概念，然后介绍数据模型、关系数据库以及 Visual FoxPro 关系数据库管理系统等基础知识。

1.1　数据库的基本概念

数据库是按一定方式把相关数据组织、存储在计算机中的数据集合，数据库不仅存放数据，而且还存放数据之间的联系。

1.1.1　数据与数据处理

数据是指存储在某一种媒体上的能够识别的物理符号。数据的概念有两个方面的涵义：描述事物特性的数据内容以及存储在媒体上的数据形式。数据形式可以是多样的，例如“2003 年 1 月 1 日”是一个数据，它可以表示为“2003-1-1”、“01/01/2003”等形式。

数据的概念在数据处理领域中已经大大地拓宽了，数据不仅包括各种文字或字符组成的文本形式的数据，而且包括图形、图像、动画、影像、声音等多媒体数据。

数据处理是指将数据转换成信息的过程，通过数据处理可以获得信息，如通过商店的进货量和销售量，就可以知道库存量，从而为进货提供依据。

1.1.2　数据库的产生

计算机管理数据随着计算机的发展而不断发展，利用计算机对数据进行处理经历了 4 个阶段。

1．人工管理阶段

计算机诞生之初，外存储器只有纸带、磁带、卡片等，没有像磁盘这样的速度快、存储容量大、随机访问、直接存储的外存储器。软件方面，没有专门管理数据的软件，数据包含在计算或处理它的程序之中。这一阶段的数据管理任务，包括存储结构、存取方法、输入输出方式等完全由程序员通过编程实现。这一阶段的数据管理称为人工管理阶段。

2．文件系统阶段

20 世纪 50 年代后期至 60 年代后期，计算机开始大量地用于各种管理中的数据处理工作。大量数据的存储、检索和维护成为紧迫的需求。此时，在硬件方面，可直接存取的磁盘成为外存储器的主流；软件方面，出现了高级语言和操作系统。

这一阶段的数据处理采用程序与数据分离的方式，有了程序文件与数据文件的区别。数据文件可以长期保存在外存储器上被多次存取，在操作系统的文件系统的支持下，程序使用文件名访问数据文件，程序员只需关注数据处理的算法，而不必关心数据在存储器上如何存取。这一阶段的数据管理称为文件（系统）管理阶段。

文件系统中的数据文件是为了满足特定的需要而专门设计的，为某一特定的程序而使用，数据与程序相互依赖。同一数据可能出现在多个文件中，这不仅浪费存储空间，而且由于不能统一更新，容易造成数据的不一致性。

3．数据库系统阶段

随着社会信息量的迅猛增长，计算机处理的数据量也相应增大，文件系统存在的问题阻碍了数据处理技术的发展，于是数据库管理系统便应运而生。

数据库技术的主要目的是有效地管理和存取大量的数据资源。包括：提高数据的共享性，使多个用户能够同时访问数据库中的数据；减少数据的冗余度，提高数据的一致性和完整性；提供数据与应用程序的独立性，从而减少应用程序的开发和维护费用。

数据库管理系统从 20 世纪 60 年代末问世以来，一直是计算机管理数据的主要方式。

4．分布式数据库系统阶段

20 世纪 70 年代以前，数据库多数是集中式的，网络技术的发展为数据库提供了良好的运行环境，使数据库从集中式发展到分布式，从主机/终端系统结构发展到客户/服务器系统结构。

1.1.3 数据库系统

1．基本概念

① 数据库（DataBase）：是存储在计算机存储器中、结构化的相关数据的集合。它不仅存放数据，而且还存放数据之间的联系。

数据库中的数据面向多种应用，可以被多个应用程序共享。其数据结构独立于使用数据的程序，对于数据的增加、删除、修改和检索由系统软件进行统一的控制。

② 数据库管理系统（DBMS）：是指帮助用户建立、使用和管理数据库的软件系统，主要包括三部分：数据描述语言（DDL）、数据操作语言（DML）以及其他管理和控制程序。

③ 数据库应用系统（DBAS）：是利用数据库系统资源开发的面向某一类实际应用的应用软件系统。一个 DBAS 通常由数据库和应用程序两部分构成，它们都需要在数据库管理系统 DBMS 支持下开发和工作。

④ 数据库系统：是指引进数据库技术后的计算机系统，包括硬件系统、数据库集合、数据库管理系统和相关软件、数据库管理员、用户等五部分。

其中：硬件系统是指运行数据库系统需要的计算机硬件，包括主机、显示器、打印机等。

数据库集合是指数据库系统包含的若干个设计合理、满足应用需要的数据库。

数据库管理系统和相关软件包括操作系统、数据库管理系统、数据库应用系统等相关软件。

数据库管理员是指对数据库系统进行全面维护和管理的专门的人员。

数据库系统最终面对的是用户。

2．数据库系统的特点

与文件系统相比，数据库系统具有以下特点：

① 数据的独立性强，减少了应用程序和数据结构的相互依赖性。

② 数据的冗余度小，尽量避免存储数据的相互重复。

③ 数据的高度共享，一个数据库中的数据可以为不同的用户所使用。

④ 数据的结构化，便于对数据统一管理和控制。

1.2 数据模型

在现实世界中，事物和事物是存在联系的，这种联系是客观存在的，是由事物本身的性

质所决定的。例如，学校教学系统中的教师、学生、课程、成绩等都是相互关联的。通常把表示客观事物及其联系的数据及结构称为数据模型。

1.2.1 基本概念

1．实体

客观存在并且可以相互区别的事物称为实体。实体可以是实际的事物，如教师、职工、部门、单位等；也可以是抽象的事件，如比赛、订货、选修课程等。

2．实体集

实体集是具有相同类型及相同性质（或属性）的实体集合，例如，某个学校的所有学生的集合可以被定义为实体集 Students。

3．属性

实体通过一组属性来表示，属性是实体集中每个成员具有的描述性性质。将一个属性赋予某实体集表明数据库为实体集中每个实体存储相似的信息，例如学生可以用学号、姓名、性别、出生日期等属性描述。但对每个属性来说，各实体有自己的属性，即属性被用来描述不同实体间的区别。

4．联系

实体之间的对应关系称为联系，它反映了现实事物之间的相互联系，例如，一位学生可以选学多门课程；一个部门中可以有多个职工。

1.2.2 实体之间的联系

联系可以归纳为三类：

1．一对一的联系

若对于实体集 A 中的每一个实体，在实体集 B 中都有惟一的一个实体与之联系，则称实体集 A 与实体集 B 具有一对一的联系。例如，一个部门有一个经理，而每个经理只在一个部门任职，则部门和经理之间具有一对一的联系。

2．一对多的联系

若对于实体集 A 中的每一个实体，实体集 B 中有 n（$n > 0$）个实体与之联系，反之，对于实体集 B 中的每个实体，实体集 A 中至多只有一个实体与之联系，则称实体集 A 与实体集 B 具有一对多的联系。例如，一个部门有若干个职工，而每个职工只在一个部门工作，则部门与职工之间是一对多的联系。

3．多对多的联系

若对于实体集 A 中的每一个实体，实体集 B 中有 n（$n > 0$）个实体与之联系，反之，对于实体集 B 中的每个实体，实体集 A 中也有 m（$m > 0$）个实体与之联系，则称实体集 A 与实体集 B 具有多对多的联系。例如，学生和选修课程的联系，某个学生可以选修多门课程，某选修课程也可以被多名学生选修。

1.2.3 数据模型简介

数据库中的数据从整体来看是有结构的，即所谓数据的结构化。各实体以及实体间存在的联系的集合称为数据模型，数据模型的重要任务之一就是指出实体间的联系。按照实体集

间的不同联系方式，数据库分为三种数据模型，即层次模型、网状模型和关系模型。

1．层次模型

层次模型的结构是树型结构，树的节（结）点是实体，树的枝是联系，从上到下为一对多的联系。每个实体由“根”开始沿着不同的分支放在不同的层次上。如果不再向下分支，则此分支中最后的节点称为“叶”。图 1-1 为某系的机构设置，“根”节点是系，“叶”节点是各位教师。

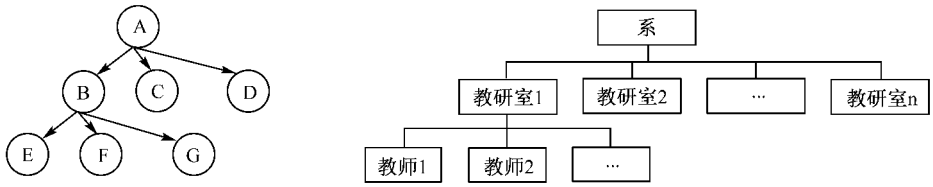


图1-1　树型结构与层次模型

支持层次模型的数据库管理系统称为层次数据库管理系统，其中的数据库称为层次数据库。

2．网状模型

用网形结构表示实体及其之间的联系的模型称为网状模型。在网状模型中，每一个节点代表一个实体，并且允许节点有多于一个的“父”节点。这样网状模型代表了多对多的联系类型，如图 1-2 所示。

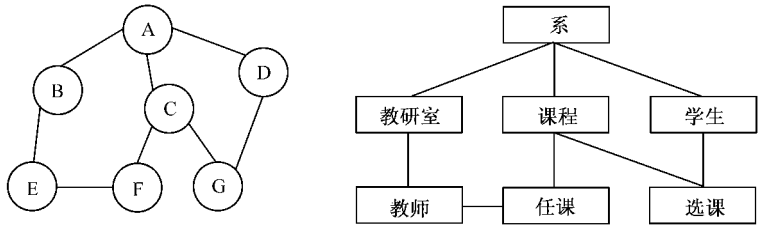


图1-2　网形结构与网状模型

支持网状模型的数据库管理系统称为网状数据库管理系统，其中的数据库称为网状数据库。

3．关系模型

关系模型是以数学理论为基础构造的数据模型，它用二维表格来表示实体集中实体之间的联系。在关系模型中，操作的对象和结果都是二维表（即关系），表格与表格之间通过相同的栏目建立联系。

关系模型有很强的数据表示能力和坚实的数学理论，且结构单一，数据操作方便，最易被用户接受，以关系模型建立的关系数据库是目前应用最广泛的数据库。由于关系数据库的许多优秀功能，层次数据库和网状数据库均已失去其重要性。

1.3　关系数据库

自 20 世纪 80 年代以来，新推出的数据库管理系统几乎都是基于关系模型。Visual FoxPro

就是一种关系数据库管理系统。

1.3.1 基本概念

1．关系与表

关系的逻辑结构就是一张二维表，如学籍表、课程表等。在 Visual FoxPro 中，一个关系就是一个“表”，每个表对应一个磁盘文件，表文件的扩展名为.DBF。表文件名即表的名称，也就是关系的名称。

2．属性与字段

一个关系有很多属性（即实体的属性），对应二维表中的列（垂直方向）。每一个属性有一个名字，称为属性名。对于一张二维表格来说，属性就是表格中的栏（列），同栏的数据应具有相同的性质，如“姓名”这一栏就只能填充姓名数据，而不能是其他数据。

在 Visual FoxPro 中，属性表示为表中的“字段”，属性名即为字段名。

3．关系模式与表结构

对关系的描述称为关系模式，一个关系模式对应一个关系的结构。其格式为：

关系名（属性名 1，属性名 2，…，属性名 n）

在 Visual FoxPro 中对应的表结构为：

表名（字段名 1，字段名 2，…，字段名 n）

4．元组与记录

在一个表格（一个关系）中，行（水平方向）称为“元组”。在 Visual FoxPro 中，元组表示为表中的“记录”。

一个表中可以有多个记录，也可以没有记录，没有记录的表称为“空表”。

5．域

域是属性取值的范围，不同的属性有不同的取值范围，即不同的域。如成绩的取值范围是 0～100，逻辑型属性的取值只能是 .T.（真）或 .F.（假）。

6．码与关键字

用来区分不同元组（实体）的属性或属性组合，称为码。在 Visual FoxPro 中对应的概念是关键字，关键字是字段或字段的组合，用于在表中惟一标识记录。如学生成绩表中的学号字段是关键字，因为学号不可能重复，可以用来惟一标识一个记录，性别字段就不是关键字，因为表中性别可能会在不同记录中出现，即有两个或两个以上的记录该属性相同的。

如果码的任意真子集都不能成为码，这样的“最小码”称为“候选码”。候选码可能有多个，被选中用来区别不同元组的候选码称为主码。在 Visual FoxPro 中，对应的概念是：候选关键字和主关键字。

如果表中的某个字段不是本表的关键字，而是另外一个表中的关键字，则称该字段为外部关键字。

7．关系模型与数据库

从集合论的观点来看，一个关系模型就是若干个有联系的关系模式的集合，一个关系模式是命名的属性集合，另外，关系是元组的集合，元组是属性值的集合。

在 Visual FoxPro 中，把相互之间存在联系的表放到一个数据库中统一管理。例如，在订货管理数据库中可以包含订单表和客户表。数据库文件的扩展名为.DBC。

1.3.2 数据完整性

数据完整性是指数据库中数据的正确性和一致性（或相容性），数据完整性用来防止数据库中存在不合法的数据，防止错误的数据进入数据库中。

数据完整性可以分为实体完整性、域完整性和参照完整性。

1．实体完整性

实体完整性是指数据库表的每一行都有一个惟一的标识。实体完整性由实体完整性规则来定义，完整性规则是指表中的每一行在组成码（关键字）的列上不能有空值或重复值，否则就不能起到惟一标识行的作用。

2．域完整性

域完整性是指数据库数据取值的正确性。它包括数据类型、精度、取值范围以及是否允许空值等。取值范围又可分为静态和动态两种：静态取值范围是指列数据的取值范围是固定的，如年龄小于 150；动态取值范围是指列数据的取值范围由另一列或多列的值决定，或更新列的新值依赖于它的旧值。

3．参照完整性

参照完整性是指数据库中表与表之间存在码（关键字）与外码（外部关键字）的约束关系，利用这些约束关系可以维护数据的一致性或相容性，即在数据库的多个表之间存在某种参照关系。要实现这种参照关系，首先创建表的码与外码：

① 当对含有外码的表进行插入、更新操作时，必须检查新行中外键的值是否在主表中存在，若不存在就不能执行该操作。

② 当对主表中的行进行删除、更新操作时，必须检查被删除行或被更新行中主码的值是否在被一个或多个外码参照引用，若正被参照就不能执行该操作。

1.3.3 对关系数据库的要求

通常生活中的二维表格有多种多样，不是所有二维表格都能被当作“关系”而存放到数据库中。也就是说，在关系模型中对“关系”有一定的规范化要求。

① 关系中的每个属性（列）必须是不可分割的数据单元。如图 1-3 所示左边的复合表不符合要求，不能直接作为关系，应将它改为右边所示的二维表。

姓名	成绩		
	语文	教学	外语

姓名	语文	教学	外语

图1-3　复合表与关系表

② 同一关系中不应有完全相同的属性名，即在同一个表格中不能出现相同的栏（字段）。

③ 关系中不应有完全相同的元组，即在同一个表格中不能出现相同的行（记录）。

④ 元组（记录）和属性名（字段）与次序无关，即交换两行或两列的位置不影响数据的实际含义。

1.3.4 关系运算

关系运算对应于 Visual FoxPro 中对表的操作，在对关系数据库进行查询时，为了找到用户感兴趣的数据，需要对关系进行一定的运算。这些运算以一个或两个关系作为输入，运算的结果将产生一个新的关系。关系的运算主要指选择、投影、连接三种运算。

1．选择运算

选择运算是指从关系中找出满足给定条件的元组，又称为筛选运算。选择的条件以逻辑表达式给出，使得逻辑表达式的值为真的元组被选取。选择是从行的角度进行的运算，即选择部分行，经过选择运算可以得到一个新的关系，其关系模式不变，但其中的元组是原关系的一个子集。

在 Visual FoxPro 中，选择操作使用命令短语 FOR | WHILE 〈条件〉或设置记录过滤器来实现。

2．投影运算

从关系模式中指定若干个属性组成新的关系称为投影。投影是从列的角度进行的运算，经过投影可以得到一个新关系，其关系模式所包含的属性个数往往比原关系少，或者属性的排列顺序不同。投影运算提供了垂直调整关系的手段，体现出关系中列的次序无关的特性。

在 Visual FoxPro 中，投影操作使用命令短语 FIELDS 〈字段 1〉,〈字段 2〉,…，或设置字段过滤器来实现。

选择和投影经常联合使用，从数据库文件中提取某些记录和某些数据项。

3．连接运算

从两个关系中选取满足连接条件的元组组成新关系，称为连接（或链接）。连接是关系的横向结合，连接运算将两个关系模式的属性名拼接成一个更宽的关系模式，生成的新关系中包含满足连接条件的元组。连接过程是通过连接条件来控制的，连接条件中将出现两个关系中的公共属性名，或者具有相同语义、可比的属性。

选择和投影运算都是一目运算，它们的操作对象只是一个关系，相当于对一个二维表进行切割。连接运算是二目运算，需要两个关系作为操作对象，如果需要连接两个以上的关系，应当两两进行连接。

在 Visual FoxPro 中，连接操作相当于对两个二维表进行拼接。有两种意义下的连接操作，用 JOIN 命令实现两个表的连接将得到一个新的表；关联操作命令 SET RELATION 属于逻辑上的连接操作。

4．自然连接和优化

自然连接是指去掉重复属性的等值连接，它是按照属性值对应相等为条件进行的连接操作。自然连接是最常用的连接运算。

系统在执行连接运算时，要进行大量的比较操作，因此执行时比较费时。尤其在包括许多元组的关系之间进行连接时，更加突出。

优化的一般方法是：

① 首先进行选择运算，尽量减少关系中元组的个数，缩小参与连接运算关系的数量，减少访问记录的次数；

② 然后能投影的投影，使关系中的属性个数减少。在投影时必须注意保留连接两个关系

所需要的公共属性或具有相同语义的属性，否则关系之间就失去了联系；

③ 最后再进行连接操作。

利用关系的投影、选择和连接运算可以方便地分解或构造新的关系。

1.4 Fox 系列数据库产品简介

在微机关系数据库中，xBase 家族占有重要的地位。从 dBase 到 FoxBase，FoxPro，Visual FoxPro，这一家族在 PC 平台上始终独占鳌头，拥有最广泛的用户群。

1.4.1 发展历史

1．数据库技术与 xBase 语言

计算机技术的发展带动着数据管理的方法也在不断地发展，20 世纪 70 年代后期，数据库理论的研究已较为成熟。当 IBM-PC 及其兼容机于 80 年代初逐步普及时，美国 Ashton-Tate 公司于 1982 年推出了适合 8 位微机的 dBASE II 关系数据库管理系统。由于它简单、易学、易用，数据库处理能力大大优于其他语言，命令格式与英语自然语法接近，因此很快就随微机的普及而风靡全球，被誉为“大众数据库”。但随着 16 位微机的出现，dBASE II 很快显示出它的不足，如数据类型不够丰富、数据精度不高、各数据表文件中的字段个数和同时打开的数据表文件数过少等。因此，Ashton-Tate 公司于 1984 年 6 月推出了更新版本 dBASE III。dBASE III 在 dBASE II 的基础上增加了日期型、备注型两种数据类型，将数据精度提高到 16 位，改善了报表功能和屏幕输出格式，它允许同时打开更多的数据表文件，增加了数据表文件的字段个数，此外还新增了二十多条命令和十多个函数。因此 dBASE III 较 dBASE II 功能更强、运行速度更快。

dBASE III 虽然具有许多优点，但其缺点也显而易见，其中最突出的一点就是它只能解释执行，从而导致应用程序无法保密、运行速度太慢等种种弊端。为此其他许多公司瞄准市场，纷纷推出 dBASE III 编译程序，其中比较有影响的是美国 Nantucket 公司 1985 年推出的编译 dBASE III 及其后推出的 Clipper 各版本。其中 Clipper 5.0 提供了更为强大、灵活的功能，它可以在网络上运行，其调试功能也相当出色，但它缺少设计工具，因而用户往往需要使用其他公司的产品辅助开发。

1986 年 Ashton-Tate 公司为适应微机联网的需要推出了改进型 dBASE III Plus，它在 dBASE III 基础上增加了 30 多条命令和 30 多个函数，提供了更为友好的用户界面和新的数据目录处理方法。dBASE III Plus 网络版本具有在局域网上运行所需的管理工具，如文件和记录加锁、提供安全保密等。但其安全保密管理是通过一个名为 Protect 的实用程序实现的，而不是由数据库管理系统提供的。由于 dBASE 产品功能一代比一代强，在微机数据库系统中占有统治地位，因此，dBASE 数据库语言被作为微机数据库的一种工业标准。但是 dBASE 仍然存在着不少缺点，如速度慢、不带编译器、人机界面差、命令和函数有限等。

2．从 FoxBASE 到 FoxPro

从事数据库工作之一的美国 Fox Software 公司，正是看到了 dBASE 在性能与速度上存在的不足，也预见到了微型计算机数据库系统应用的巨大潜力，在它成立后的第二年即 1984 年，便推出了与 dBASE 完全兼容的 FoxBASE，其速度大大快于 dBASE，并且在 FoxBASE

中第一次引入了编译器。1986 年，与 dBASE III Plus 兼容的 FoxBASE+推出，并在许多方面有所加强，如提供了编译功能，支持数组，命令与函数更加丰富，拥有众多工具。不久网络版本也投入市场，一时间引起轰动。1987 年 7 月推出了 FoxBASE+2.0，其最高版本是 1988 年 7 月推出的 FoxBASE+2.1。这两大产品不仅速度超越其前期产品，而且还扩充了对开发者极其有用的语言，并提供了良好的界面和较为丰富的工具。FoxBASE+2.1 速度平均比 dBASE III Plus 快 5.9 倍，比 Clipper 快 3.2 倍，可以在 DOS 和 Unix 操作系统上运行等，因此深得微机用户喜爱，直到现在，国内仍在广泛使用 FoxBASE+2.1，但它在安全性方面表现不佳。

1988 年 Ashton-Tate 公司为解决 dBASE 的编译问题推出了 dBASE IV。dBASE IV 除了可以对程序进行编译外，还支持 SQL 语句，具有网络功能，提供了一系列工具，并由数据库管理系统本身提供安全保密功能。

1989 年下半年，FoxPro 1.0 正式推出，它是 FoxBASE+2.10 的升级换代产品。FoxPro 采用了类似 APPLE Machitosh 用户接口那样极友好的图形界面来设计，首次引入了基于 DOS 环境的窗口技术——面向字符的窗口 COM，用户使用的界面再也不是圆点，而是与圆点提示符下等效命令的菜单系统。它支持鼠标，操作方便，是一个与 dBASE、FoxBASE 全兼容的伪编译型集成环境式的数据库开发环境。

FoxPro 1.0 功能强大，运行速度快，在所有基准程序测试中证实，FoxPro 1.0 要比 dBASE IV 快 8 倍，比 dBASE III 快 16 倍，并且比 FoxBASE+2.10 快 2 倍。

1991 年 7 月 FoxPro 2.0 推出，由于使用了 Rushmore 查询优化技术、先进的关系查询与报表技术，以及第四代语言 FGL（Fourth Generation Language）工具，FoxPro 2.0 在性能上大幅度地提高了。它面向对象与事件，其扩展版本能充分使用扩展内存，是一个真正的 32 位产品。除了支持 FoxPro 先前版本的全部功能外，还增加了 100 多条全新的命令与函数，从而使得 FoxPro 的程序设计语言逐步成为 xBase 语言的标准。在与 dBASE IV、Paradox、R:BASE、Clipper 一起参加的基准测试中，FoxPro 以最快的速度大大超过其竞争者，加之它成功的字符环境下的图形用户界面 GUI、第一次引入 SQL 结构化查询语言以及直观的查询设计器，增加了屏幕生成器、菜单生成器、报表生成器和项目管理器等强大的工具，使得 FoxPro 荣获当年度美国 Computer Language、Infoworld、BYTE、PC Magazine、PC Computing、Data Based Advisor、LAN Times 和 Systems Integration 等诸多杂志所评选的多项优秀成果奖。

1992 年，Fox 软件公司被 Microsoft 软件公司收购后，于年中推出了 FoxPro 2.5。Microsoft FoxPro 2.5 以其优越的性能、最快的速度而领先于任何其他微机数据库管理软件，已被认为是用户首选的微机数据库产品。它可运行在 MS-DOS、Windows、Machitosh、UNIX 操作系统环境下，并且保持了对每一级用户拥有相同的图形用户界面、工具和语言，是领先于任何其他微机数据库管理软件的优秀产品。

FoxPro 2.6 是对 FoxPro 2.5 的扩充，它使用户可以很容易地管理目录文件、提供了向导工具 Wizards、增强了与 dBASE IV 的兼容性、扩展了 FoxPro 2.5 的功能。

3．Visual FoxPro 的推出

1995 年 9 月，Microsoft 公司推出了最新的 FoxPro 版本 Visual FoxPro 3.0，它集 Wizards 技术和 Rushmore 技术于一体，人们认为它是多年来出现的在关系数据库方面最重要的产品。是继 Visual C++、Visual Basic 后又一可视化产品。目前 Visual FoxPro 的最高中文版本是 1998 年推出的 Visual FoxPro 6.0 中文版。

Visual FoxPro 是 Microsoft 公司推出的全新 PC 平台关系型数据库管理系统。它具有强大的性能、无与匹敌的速度、完整而丰富的工具、极其友好的图形用户界面、简单的数据存取方式、良好的兼容性、独一无二的跨平台特性及真正的可编译性，使系统成为目前最快、最完美的数据库系统。不但兼容早期的 dBASE 以及 FoxBASE 各种版本同时还提供了许多基于 Windows 的崭新功能。Visual FoxPro 作为具有 Windows 9x 兼容标志的应用软件，具有快速开发应用程序、面向对象和客户机/服务器的强大功能，它是多年来出现在关系数据库方面最重要的产品。随着桌面操作系统由 Windows 3.x 逐渐向 Windows 9x 升级的发展潮流，Visual FoxPro 必将成为今后数据库产品中的主流。

Microsoft Visual FoxPro 是一个 32 位的数据库开发系统，可运行于 Windows 9x 和 Windows NT 操作系统。与 FoxPro 2.5、2.6 相比，它是一个革命性的软件产品，引进了可视编程和面向对象的概念。

Visual FoxPro 既具有 Visual 系列的功能强大、直观易用、面向对象等优点，又兼具 Windows 和 FoxPro 的长处。提供了“向导”、“设计器”和“生成器”等工具，使得数据库的管理工作变得容易。

Visual FoxPro 的易用性使初学者和那些想避免涉及 FoxPro 复杂命令的人能很快用它来管理自己的数据库，制作各种报表、标签等；增添的面向对象的编程方式等新特色，使之成为应用程序开发人员的强有力工具；其兼容性使原来的广大 xBase 用户能迅速转为使用 Visual FoxPro；Visual FoxPro 还能广泛地与其他许多软件（如 Excel、Word、Lotus 1-2-3 等）共享和交换数据。正是由于其易用性、先进性和广泛性，使 Visual FoxPro 真正做到了面向各种水平的用户。

1.4.2 VisualFoxPro 的特点

1．简单、易学、易用

（1）快速完成应用任务

提供了“向导”、“生成器”和“设计器”三种工具，这三种工具都使用图形交互界面方式，使用户能够最简单而又最快地完成数据操作任务。

操作“向导”提供了用户要完成某项工作所需的详细操作步骤，在这些步骤的指导下，用户可以一步步地很简单地完成任务。例如，用户可用“表向导”来帮助建立一个数据表，用“表单向导”来建立表单，而“查询向导”将指示用户建立一个标准查询所需的完整步骤。

“生成器”也是一种具有友好界面的图形工具，它的主要功能是在用户自己的应用程序中加入一定的控制功能。例如“列表框生成器”就是一个带有标签的对话界面，利用列表框生成器，用户可以在表单中设计出一个列表框，并且可以在这种生成器中设置一个列表框的共同属性。

如果用户想突破向导和生成器的限制，想要自己对应用程序进行更复杂或更灵活的控制，可以利用另一种 Visual FoxPro 提供的方便有效的工具——“设计器”。设计器也提供了一个友好的图形应用程序开发接口，通过它用户能建立起自己的应用程序。例如，用户可以用“表单设计器”定义和生成一个表单，用“数据表设计器”定义和生成一个数据表。

（2）一致的用户界面，使用方便的工具栏

Visual FoxPro 改进了用户界面，其主窗口与许多 Microsoft 其他产品（如 Word、Excel）

更趋于一致，使得用户更容易操作，系统功能更易于发挥。Visual FoxPro 也给用户提供了使用方便的“工具栏”，工具栏里有许多按钮，它们代表着菜单里的某些选项。一般来说，用户经常执行的操作（如“打开文件”）或使用的对象（如“命令窗口”）都对应一个按钮，用户可以通过选择这些按钮方便而迅速地完成操作，而不必通过菜单选项。

另外，用户可以自己定制 Visual FoxPro 中的工具栏，增加或减少一些按钮，还可以在自己建立的应用程序中定义和实现方便用户使用的工具栏。

Visual FoxPro 支持鼠标右键激活快捷菜单，用户可更加快捷地操作屏幕。

（3）不编程而建立应用程序界面

Visual FoxPro 提供的“表单设计器”是一种功能强大的工具，用户能够不编程或使用很少的代码来实现友好的交互式应用程序界面，并可对界面进行控制。例如用户可以用表格控件很容易地建立一对多的表单：用户只需把一个数据表拖动到一个窗体上就可以了。也可以利用页格式控件来建立有标签的对话框或用户自己的生成器界面。

（4）用项目管理器统一管理工作

Visual FoxPro 提供的另一高效易用的工具是“项目管理器”，通过项目管理器，用户可以集中地管理数据、文档、类库、源代码等各种资源。例如，用户可以建立和更新数据库，设计或改变窗体和报表，定义或改变类库，生成或重新生成自己的应用程序。另外，用户也能在项目管理器中使用 Visual FoxPro 提供的简单而有效的其他工具，如向导、生成器、工具栏等。

2．功能更强大

Visual FoxPro 能通过使用快速查询（Rushmore）技术和对系统的优化，使用户最大限度地体会到快速而又功能强大的优点。

（1）真正的数据库概念

以前的 xBase 软件中称.DBF 文件为数据库，使人容易产生一个数据库就是一个二维表的错误认识。而 Visual FoxPro 废除了以前 xBase 不合理的数据库概念，采用独特的数据库容器（DataBase Container），为用户管理应用系统中的表、查询、表单、报表、程序等数据提供了方便，支持长数据库文件名和字段名，可为字段名设置新的显示标题，为字段指定默认值，设置字段级和记录级的有效性规则，设置表的插入、删除和改变记录的触发事件代码。

在 Visual FoxPro 中，原来的.DBF 文件变成了数据库中的一个表，不属于任何数据库的表称为自由表。数据库是若干个表、表之间的关系和触发程序的集合，合理地体现了关系型数据库的思想，与关系数据库理论统一起来。新的数据库把有关系的表（.DBF）封装在一起，关系清晰、合理且处理方便。

Visual FoxPro 由于使用了这种真正的数据库概念，使得它的数据库结构与 SQL 等标准结构统一，从而使数据交换和相互操作的实现更加标准、合理、方便。

（2）可视化编程技术

Visual FoxPro 用与 Visual C++、Visual Basic 同样的编程技术，这是它取名为 Visual FoxPro 的原因。可视化编程技术给人一种所见即所得的感受，在编辑屏幕表单、报表、菜单时，可以直接运行，不必来回调试，极为方便。

（3）具有面向对象编程的能力

Visual FoxPro 在支持标准 xBase 传统的面向结构的编程方式的同时，也提供了完全的面

向对象编程（OPP）能力。在 Visual FoxPro 的对象模式下，用户可以利用所有的面向对象编程特性，这些特性包括“继承”、“封装”、“多态性”以及“分类”，它们都作为用户所熟悉的 xBase 编程语言的扩展集而实现。

Visual FoxPro 提供了两种类型近 30 个基类，包括表单、工具栏、页格式等，使用这些类，用户可以建立基本的表单、工具栏或页格式，这样就可以一方面减少用户编程工作量，另一方面又加快程序开发过程。

再进一步，用户可以将自己定义的类再进行分类，这样可利用用户已有的源代码或表单。例如，用户可以将基本的表单类再进行分类而建立自己的子类，这个子类将根据用户的要求自动地在应用程序中建立起一个用户希望看到的表单，它的结构是由用户分类决定的。

Visual FoxPro 类模式能够在用户应用程序中对对象进行深入而全面的控制。例如，用户在设计时可用表单设计器对表单中的对象进行完全的控制，而类模式下当用户运行程序时可对表单中对象的表现和行为提供相同的控制。

在 Visual FoxPro 中，用户可以用“类设计器”交互式地建立一个类，或者用 DEFINE CLASS 命令来编程建立。

Visual FoxPro 使用面向对象迎合了时代的潮流，是 xBase 产品的革命。

（4）更容易处理事件

Visual FoxPro 包含一种事件模式，它能够帮助用户自动地处理事件。在这种事件模式下，用户可以获取并控制所有标准的 Windows 事件，例如鼠标的移动。通过处理这一事件，用户可以拖动和放置一个对象。用户可以用两种方法来控制事件：一种是通过“属性窗口”来可视地控制；另一种是通过 Visual FoxPro 的编程语言来控制。这两种方法都能使用户很容易地建立起完全的事件驱动应用程序而不用考虑 READ 层次及浏览窗口限制，也不用编写事件处理程序。

新增加的命令 BEGIN TRANSACTION...END TRANSACTION，提供对事件处理的支持，深度可达 5 级。

（5）新增许多功能强大的命令和函数，SQL 语句更加丰富

增加了 7 种新的字段类型：整型、货币型、日期时间型、双精度型、通用型、二进制字符型和二进制备注型。

在结构化的复合索引中可以建立 4 种类型的索引：主索引、候选索引、普通索引和惟一索引。

允许在表中使用空值 NULL，以保证与采用 SQL 标准的数据库管理系统的兼容和数据共享。

（6）最优化系统

Visual FoxPro 能够通过优化用户的系统设计来提高自身的性能。在所有的优化措施中，最有效的方法是尽可能多地增加用户的扩展内存（Extended memory）或者减少被其他应用程序（如 Windows）所占用的内存。另外提高 Visual FoxPro 性能的措施还包括加快启动速度和优化设置（SET）命令。

（7）使用快速查询技术

快速查询（Rushmore）技术是一种专用的数据查询技术，它能够迅速地从数据库中选择出一组满足用户要求的记录。使用这种技术能将数据查询所需的时间从几小时或几分钟减少

到几秒钟，这样可以极大地提高数据查询的效率。

（8）使用 32 位方式

Visual FoxPro 使用 32 位方式，其运算速度、存储能力大大提高。

3．支持客户机/服务器结构

Visual FoxPro 可作为开发强大的客户机/服务器（Client/Server）应用程序的前台。Visual FoxPro 既支持高层次的对服务器数据的浏览，又提供了对本地服务器语法的直接访问，这种直接访问给用户提供了开发灵活的客户机/服务器应用程序的坚实的基础。Visual FoxPro 提供了支持客户机/服务器结构所需的各种特性：多功能的数据词典、本地和远程视图、空值 NULL 支持、事务处理、对任何 ODBC 数据资源的访问。

（1）用数据词典定义规则

Visual FoxPro 数据库（.DBC）提供了一个数据词典，使用这个数据词典，用户可以对数据库中的每一个数据表添加规则、视窗、触发器、永久关系和连接。

在一个数据库中，用户可以定义：

① 字段级或记录级的规则，这种规则将在用户的应用程序中，对该数据表操作时起作用。

② 主索引键和候选索引键。

③ 本地和远程视图。

④ 触发器。

⑤ 数据表之间的永久关系。

⑥ 对远程数据资源的连接。

⑦ 存储进程。

⑧ 字段的默认值。

⑨ 长表名及字段名。

另外，用户可以通过“引用完整生成器”来定义插入、更新和删除规则，这样可以加强每一个保存关系的引用完整性。

Visual FoxPro 也支持数据表中的 NULL 值，这种能力极大地提高了 Visual FoxPro 同其他数据资源的兼容性和连接能力，这些数据资源包括 Microsoft Access，Visual Basic 和基于 SQL 服务器。

（2）查看远程或异种数据

用户可以用来自远程、本地或多数据表的异种数据，以便在用户的本地计算机上开发和测试一个客户机/服务器应用程序。本地数据视图使用本地计算机上的数据表而不是远程服务器上的数据表。而多表数据查看使用的是多个不同数据表中的相关数据。为了减少用户从服务器上卸载的数据量，用户可以建立带参数的视图，然后从用户的 Visual FoxPro 客户机/服务器应用程序中更新远程数据。

（3）用事务处理来控制共享访问

共享访问是指多个用户对数据的共享以及相应的一些必要的访问限制，例如为了不让某用户访问某些数据，用户可以建立起支持数据共享访问的应用程序。用户在建立应用程序时，如果使用事务处理和缓冲手段（记录级或数据表级），则可以减少编程的工作量。Visual FoxPro 内含的批处理进程和详细的对更新冲突处理的控制，可以使多用户环境中的数据更新过程得以简化。

（4）实现客户机/服务器应用程序

在客户机/服务器应用程序开发中，用户除了使用数据视图以外，还可以通过 Visual FoxPro 的 SQL 通路功能来发送当前服务器所识别的控制台命令，这样用户可以直接访问服务器。这种功能比数据视图提供了更多的对服务器的访问和控制。

Visual FoxPro 具有将用户的应用程序升档的能力。升档是指用户在本地机上建立一个应用程序后，可以基于一个后台的数据资源使应用程序运行在一个客户机/服务器环境中，这样做的好处之一就是用户可以用和本地的 Visual FoxPro 数据表结构一样的结构建立起远程的服务器数据库。不仅如此，用户在升档时可以选择哪些数据表放在服务器中而哪些表放在本地机上，这样可以既提供共享能力，又提高访问效率。

4．同其他软件的高度兼容性

Visual FoxPro 可以同其他 Microsoft 软件共享数据，例如用户可用 OLE 自动共享其他软件（如 Excel、Word）中的对象并在 Visual FoxPro 中使用这些软件。

（1）同其他软件共享数据

在 Visual FoxPro 中同其他软件共享数据是很容易的。用户可用“主元表向导”使 Excel 共享 Visual FoxPro 数据，还可以用“邮件合并向导”使 Word 共享 Visual FoxPro 数据。

（2）导入和导出数据

用户能够在 Visual FoxPro 和其他软件之间输入和输出数据，即导入和导出，输入数据是指 Visual FoxPro 利用其他软件生成的数据。导出数据是指 Visual FoxPro 生成一定格式的数据以供其他软件使用。这种导入、导出是通过不同的文件格式的转换来实现的，不同的文件格式包括文本、电子表格（Spreadsheets）和表。在 Visual FoxPro 中，用户可用“导入向导”来帮助决定使用哪一种文件格式。

（3）使用自动 OLE 控制其他软件

Visual FoxPro 提供的自动 OLE 能够加强用户应用程序的功能。用户可以通过编程来运行其他的软件。例如用户可以调用 Excel 来完成某些计算，Graph 命令将运行结果绘制成图，然后把图存放在一个 Visual FoxPro 表的通用型字段中，所有这些工作都可通过 Visual FoxPro 的编程来实现。

1.5 习题

一、选择题

1．在下列四个选项中，不属于基本关系运算的是（　　）。

A）连接　　B）投影　　C）选择　　D）排序

2．如果一个班只能有一个班长，而且一个班长不能同时担任其他班的班长，班级和班长两个实体之间的关系属于（　　）。

A）一对一联系　　B）一对二联系　　C）多对多联系　　D）一对多关系

3．Visual FoxPro 支持的数据模型是（　　）。

A）层次数据模型　B）关系数据模型 C）网状数据模型 D）树状数据模型

4．用二维表格来表示实体与实体之间联系的数据模型称为（　　）。

A）实体-联系模型 B）层次模型　　C）网状模型　　D）关系模型

5．数据库（DB）、数据库系统（DBS）、数据库管理系统（DBMS）三者之间的关系是（　　）。

A）DBS 包括 DB 和 DBMS　　B）DBMS 包括 DB 和 DBS

C）DB 包括 DBS 和 DBMS　　D）DBS 就是 DB，也就是 DBMS

6．数据库系统与文件系统的主要区别是（　　）。

A）数据库系统复杂，而文件系统简单

B）文件系统不能解决数据冗余和数据独立性问题，而数据库系统可以解决

C）文件系统只能管理程序文件，而数据库系统能够管理各种类型的文件

D）文件系统管理的数据量较少，而数据库系统可以管理庞大的数据量

7．Visual FoxPro 6.0 是一种关系型数据库管理系统，所谓关系是指（　　）。

A）各条记录中的数据彼此有一定的关系

B）一个数据库文件与另一个数据库文件之间有一定的关系

C）数据模型符合满足一定条件的二维表格式

D）数据库中各个字段之间彼此有一定的关系

8．设有关系 R1 和 R2，经过关系运算得到结果 S，则 S 是（　　）。

A）一个关系　　B）一个表单　　C）一个数据库　　D）一个数组

二、填空题

1．Visual FoxPro 6.0 是一个_____位的数据库管理系统。

2．在连接运算中，_____连接是去掉重复属性的等值连接。

3．数据模型不仅表示反映事物本身的数据，而且表示_____。

4．用二维表的形式来表示实体之间联系的数据模型称为_____；二维表中的列称为关系的_____，二维表中的行称为关系的_____。

5．在关系数据库的基本操作中，从表中取出满足条件元组的操作称为_____，把两个关系中相同属性值的元组联接到一起形成新的二维表的操作称为_____，从表中抽取属性值满足条件列的操作称为_____。

第 2 章　VisualFoxPro 简介

Visual FoxPro（简称 VFP）是目前微机上使用的最优秀的数据库管理系统之一，它采用了可视化的、面向对象的程序设计方法，大大简化了应用程序的开发过程，并提高了系统的模块性和紧凑性。本章简单介绍 Visual FoxPro 6.0（简称 VFP6.0）的基本知识。

2.1　安装 Visual FoxPro 6.0

首先介绍安装 VFP 6.0 中文版的系统要求，以及安装 VFP 6.0 中文版的方法。

2.1.1　安装 VisualFoxPro 6.0 的必要条件

可以在 Windows 95/98（中文版）或更高版本，或者在 Windows NT（R） 4.0（中文版）或更高版本中运行 VFP 6.0。下面是在 Windows 95/98（中文版）中运行 VFP 6.0 的最低系统要求：

① 一台带有 486/66MHz 处理器（或更高档处理器）的 IBM 兼容机。

② 一个鼠标。

③ 16MB 内存。

④ 便携式安装需要 15MB 的硬盘空间，用户自定义安装需要 100MB 硬盘空间，完全安装需要 240MB 硬盘空间。

默认情况下，联机文档文件保存在光盘上。为了随时查看，可将这些文件（105MB）复制到本地硬盘上，安装时选择“用户自定义安装”选项，接着选择“全部选中”。

⑤ 推荐使用 VGA 或更高分辨率（800×600 像素）的显示器。

2.1.2　安装 VisualFoxPro 6.0

可以从 CD-ROM 或网络上安装 VFP 6.0。从 CD-ROM 上安装 VFP 的步骤为：

① 把 VFP 6.0 光盘插入 CD-ROM 驱动器。

② 如果 CD-ROM 没有取消“自动插入公告”功能，插入 VFP 6.0 光盘后，将自动显示“Visual FoxPro 6.0 安装向导”，如图 2-1 所示。如果没有出现安装向导，可以在“我的电脑”或“资源管理器”中选择光盘驱动器，双击 VFP 6.0 安装光盘上的 Setup.exe，也将打开“Visual FoxPro 6.0 安装向导”。

按照屏幕上显示的提示操作，一般只需单击“下一步”或“继续”按钮。

③ 在选择安装类型时，若要进行最小化安装（15MB），或要安装包括 ActiveX 和企业文件的所有 VFP 文件（192MB），请单击“自定义安装”按钮。该选项允许只选取必须的文件。对于初学者应选择“典型安装”按钮，如图 2-2 所示。

④ 若是选择“自定义安装”，则在“自定义安装”对话框选择所需的选项，例如，可以单击“全部选中”按钮安装所有选项，如图 2-3 所示。

⑤ 单击“继续”按钮，开始安装 VFP 6.0，直到显示“Visual FoxPro 6.0 安装程序已成功安装”，单击该对话框的“确定”按钮，安装完成。

图2-1　Visual FoxPro 6.0的安装向导

图2-2　选择安装类型

⑥ 此时，将显示“安装 MSDN”对话框，如图 2-4 所示。

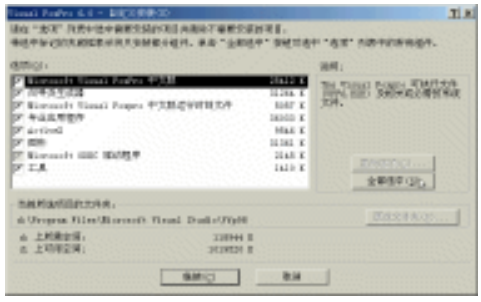


图2-3　“自定义安装”对话框

图2-4　“安装MSDN”对话框

2.1.3　安装示例和联机文档

VFP 6.0 的所有文档和示例，都包含在 Visual Studio 6.0 应用程序和组件的联机文档 MSDN Library 中。Visual Studio 6.0 MSDN Library 由两张光盘组成，插入第一张 MSDN 光盘，单击“下一步”或“继续”按钮。

在选择安装类型对话框中，选择“自定义安装”按钮，而后将打开安装选项对话框，如图 2-5 所示。

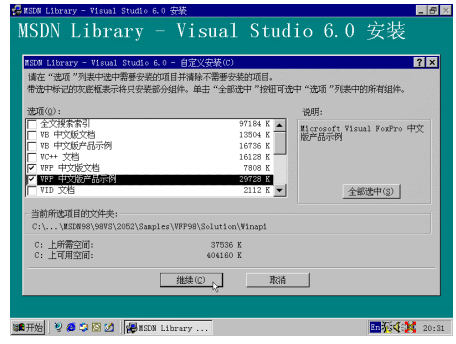


图 2-5　MSDN Library 安装选项对话框

如果要安装 VFP 6.0 示例，选中“Visual FoxPro 中文版产品示例”复选框。这些示例将被放置在公用的 MSDN 示例路径下。用户可以通过使用_SAMPLES 系统变量或执行 HOME（2）命令，以编程方式进行访问。如果在安装类型对话框中选择“典型”选项，VFP 将从 MSDN 光盘而不从硬盘访问该帮助文件。

VFP 帮助文件（包括 Foxhelp.chm）安装在硬盘下面的位置：

drive:\Program Files\Microsoft Visual Studio\Msdn98\98vs\1033

当用户在 VFP 中按〈F1〉键、在“命令”窗口输入“HELP”或使用“帮助”菜单请求帮助时，如果已安装 MSDN，则 VFP 的默认行为是调用 Msdnvs98.col。如果该文件不存在，

则将默认使用 Foxhelp.chm。如果正在利用 Msdnvs98.col 解决问题，可以重新定向 VFP 的帮助系统使其使用 Foxhelp.chm。若要设置帮助为 Foxhelp.chm，则采取以下步骤：

① 在“工具”菜单上单击“选项”命令。

② 单击“文件位置”选项卡。

③ 从所列的文件类型中选择“帮助文件”，然后单击“修改”按钮。路径不变，将 Msdnvs98.col 替换为 Foxhelp.chm，或使用对话按钮定位 Foxhelp.chm。

④ 单击“确定”按钮。如果希望将 Foxhelp.chm 用做将来的 VFP 工作期的默认帮助文件，请单击“设置为默认值”按钮。

⑤ 单击“确定”按钮。

安装完成后，将自动建立一个名称为“Microsoft Visual FoxPro”的文件夹。

2.1.4 VisualFoxPro 6.0 的启动

在成功安装 VFP 后，就可以启动 VFP。从“开始”菜单中选择“Microsoft Visual FoxPro 6.0”，如图 2-6 的左图所示。

图2-6 启动Visual FoxPro

进入 VFP 6.0 后，将显示如图 2-6 的右图所示的对话框，对话框有 5 个单选项和 1 个复选项：

① 打开组件管理器：打开新的组件管理库，管理 VFP 组件。

② 查找示例程序：将打开示例应用程序窗口，示例应用程序的目的是帮助用户学习使用 VFP。通过研究每个示例，可以看到示例是如何运行的，了解如何用代码来实现这些示例，并把示例中的一些特性应用到用户的应用程序中。

③ 创建新的应用程序：将打开“程序”窗口，创建新的应用程序。

④ 打开一个已存在的项目：将显示“打开”项目对话框。

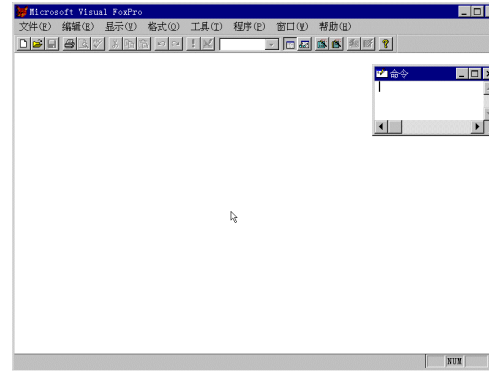


图 2-7 Visual FoxPro 的主界面窗口

⑤ 关闭此屏：关闭此对话框，回到 VFP 的主界面窗口。

如果选择了“以后不再显示此屏”复选项，在以后启动 VFP 后将直接进入 VFP 的主界面窗口。关闭本对话框后，窗口的显示将与以前版本一样，如图 2-7 所示。对于初学者应选择“关闭此屏”，然后在 VFP 的主界面窗口中操作。

2.1.5 VisualFoxPro 6.0 的退出

下列方法中的任何一种，都可以退出 VFP 6.0 系统：

- 在命令窗口键入 QUIT，再按〈Enter〉键。
- 用鼠标单击系统窗口右上角的“关闭”按钮。
- 用鼠标单击“文件”菜单中的“退出”项。
- 同时按下〈Alt〉+〈F4〉键。

2.2 Visual FoxPro 集成环境

VFP 集成环境包括主界面以及 VFP 6.0 提供的便于进行应用程序开发的各种向导、设计器和生成器。

2.2.1 VisualFoxPro 的主界面

VFP 的主窗口如图 2-7 所示，具有标准的 Windows 风格。除了在窗口的上边有标题栏、控制菜单图标、极小化按钮、极大化按钮和关闭按钮外，还包括菜单栏、工具栏、主窗口、命令窗口和状态栏。

1．菜单栏

在主窗口的最上一行是菜单栏，通过它可以完成绝大部分的操作。在默认的情况下，共有 8 个子菜单。随着用户操作的不同，子菜单和菜单项会有相应的增加与减少。

有两种方法选择菜单以及菜单命令：

- 鼠标：在主菜单中用鼠标单击菜单标题，在下拉子菜单中继续用鼠标选择菜单项。
- 访问键：按下〈Alt〉键，同时按下菜单标题中有下划线的字母，可以打开相应的子菜单。然后移动方向键，在子菜单中选择菜单项，在需要的菜单项上按〈Enter〉键。

2．工具栏

工具栏上的按钮对应于最常使用的菜单命令，所以使用工具栏可以加快某些任务的执行。

工具栏可以显示为窗口形式，只要将鼠标指针指向工具栏中按钮之外的地方，然后把工具栏“拖”（按下鼠标左键，同时移动鼠标）下来。

可以将工具栏四处拖动，放在主窗口的任何地方。将工具栏放在窗口四周的动作称为“停放（Dook）”。

将鼠标放在某个按钮上停一会，就会出现该按钮的说明文字，称为工具提示（Tooltip）。

3．命令窗口

命令窗口是一个接受输入命令的地方，就像早期的 xBASE 的环境状态。可以直接在“命令”窗口中输入 VFP 命令。VFP 命令提供的功能远远超过主菜单，这是因为主菜单只列出最常用的功能。

“命令”窗口是 VFP 的一种系统窗口。当选择菜单命令时，相应的 VFP 语句自动反映

在“命令”窗口中。初学者可以通过“命令”窗口学习许多的命令。在“命令”窗口中可以：

- 在按〈Enter〉键执行命令之前，按〈Esc〉键删除文本。
- 将光标移到以前命令行的任意位置按〈Enter〉键重新执行此命令。
- 选择要重新处理的代码块，然后按〈Enter〉键。
- 若要分割很长的命令，可以在所需位置的空格后接分号，然后按〈Enter〉键。
- 可在“命令”窗口内或向其他编辑窗口中移动文本，选择需要的文本，并将其拖动到需要的位置即可。
- 可在“命令”窗口内或向其他编辑窗口中复制文本，而不必使用“编辑”菜单的命令。选择需要的文本，按下〈Ctrl〉键，将其拖动到需要的位置即可。

2.2.2 VisualFoxPro 的向导

向导是一个交互式程序，通过用户在一系列向导屏幕上的问题回答或者选择选项，向导会根据这些回答生成文件或者执行任务，帮助用户快速完成一般性的任务。例如创建表单、编排报表的格式、建立查询、输入及升迁数据、制作图表、生成邮件合并、生成数据透视表、生成交叉表报表以及在 Web 上按 HTML 格式发布等。

VFP 6.0 中带有的向导，见表 2-1。

表 2-1　VFP6.0 中可使用的向导

向 导 名 称	功　　能
应用程序向导	创建一个 Visual FoxPro 应用程序
代码生成向导	从 Microsoft Visual Modeler（.MDL）文件中导入一个对象模型到 Visual FoxPro 中
交叉表向导	创建一个交叉表查询
数据库向导	生成一个数据库
文档向导	从项目和程序文件的代码中生成文本文件，并且编排文本文件的格式
表单向导	创建一个表单
图形向导	创建一个图形
导入向导	导入或追加数据
标签向导	创建邮件标签
本地视图向导	创建视图
邮件合并向导	创建邮件合并文件
一对多表单向导	创建一对多表单
一对多报表向导	创建一对多报表
Oracle 升迁向导	创建一个 Oracle 数据库，该数据库将尽可能多地体现原 Visual FoxPro 数据库的功能
数据透视表向导	创建数据透视表
查询向导	创建查询
远程视图向导	创建远程视图
报表向导	创建报表
逆向工程向导	导出 Visual FoxPro 类到一个 Microsoft Visual Modeler 对象模型文件（.MDL）中
示例向导	生成一个自定义向导
安装向导	基于发布树中的文件创建发布磁盘

（续）

向 导 名 称	功　　能
SQL Server 升迁向导	创建一个 SQL Server 数据库，该数据库尽可能多地体现原 Visual FoxPro 数据库的功能
表向导	创建表
Web 发布向导	在 HTML 文档中显示表或视图中的数据
WWW 搜索页向导	创建一个 Web 页，允许 Web 页的访问者从您的 Visual FoxPro 表中搜索和下载记录

可以利用向导来完成创建新任务的工作。

1．启动向导

若要启动向导，可以从“文件”菜单中选择“新建”，在“新建”对话框中选择待创建文件的类型。然后单击“向导”按钮。另外，利用“工具”菜单中的“向导”，可以直接访问大多数的向导。

如果使用的 VFP 向导中有来自数据库表中的数据，可以利用存储在数据库中的样式及字段映射，并将其反映到表单、表、标签、查询和报表中。

2．使用向导

启动向导后，要依次回答每一向导屏幕所提出的问题。在准备好进行下一个屏幕的操作时，可选择“下一步”按钮。如果操作中出现错误，或者原来的想法发生了变化，可选择“上一步”按钮来查看前一屏幕的内容，以便进行修改。选择“取消”将退出向导而不会产生任何结果。如果在使用过程中遇到困难，可按〈F1〉键取得帮助。到达最后一屏时，如果您准备退出向导，请选择“完成”按钮。

也可以选择“完成”按钮直接走到向导的最后一步，跳过中间所要输入的选项信息，而使用向导提供的默认值。

根据所用向导的类型，每个向导的最后一屏都会要求提供一个标题，并给出保存、浏览、修改或打印结果的选项。使用“预览”选项，可以在结束向导的操作前查看向导的结果。如果需要做出不同的选择来改变结果，可以返回到前边重新进行选择。对向导的结果满意后，请选择“完成”按钮。

3．修改用向导创建的项

创建好表、表单、查询或报表后，可以用相应的设计工具将其打开，并做进一步的修改。不能用向导重新打开一个用向导建立的文件，但是可以在退出向导之前，预览向导的结果并做适当的修改。

2.2.3　VisualFoxPro 的设计器

VFP 提供的各种设计器是功能强大的可视化编程工具，这些工具使得创建表、表单、数据库、查询和报表来管理数据变得轻而易举。

表 2-2 说明了为完成不同的任务所使用的设计器。

表 2-2　VFP 可使用的设计器

设计器名称	功　　能
表设计器	创建并修改数据库表、自由表、字段和索引

（续）

设计器名称	功　能
数据库设计器	创建并管理数据库，在不同的表之间查看并创建关系
查询设计器	创建并修改在本地表中运行的查询
视图设计器	创建可更新的查询，即视图；在远程数据源上运行查询
表单设计器	创建并修改表单和表单集
报表设计器	创建并修改用于显示和打印数据的报表
菜单设计器	创建并修改菜单
数据环境设计器	创建并修改表单或报表的数据环境
连接设计器	为远程视图创建连接

无论是创建一个新的任务，还是编辑一个已有的任务，系统都将自动打开相应的设计器。

（1）数据库设计器

可以显示数据库中包含的全部表、视图和关系。在激活“数据库设计器”窗口时，VFP 显示“数据库”菜单和“数据库设计器”工具栏。

每个表用一个可调整大小的窗口代表，窗口中列出表的字段和索引（如果存在）。“数据库设计器”用表之间连接索引的线段显示永久关系。

可以通过拖动表和视图来移动位置，或者使用“数据库”菜单中的“重排”命令。如果要显示或隐藏窗口中的对象，可使用“数据库”菜单中的“属性”命令。

（2）表设计器

使用“表设计器”可以创建并修改数据库表、自由表、字段和索引。“表设计器”可以实现诸如有效性检查和默认值等高级功能，如图 2-8 所示。

其中，字段选项卡在可滚动表格中显示表字段；索引选项卡则在可滚动表格中显示定义索引；表选项卡显示有关表的信息，允许指定触发器和记录级规则。

（3）查询和视图设计器

可以创建和修改查询或视图。当“查询和视图设计器”处于当前工作状态时，VFP 显示“查询”菜单和“查询设计器”工具栏或“视图设计器”工具栏。

设计器的顶部窗格显示查询或视图中的表，如图 2-9 所示。每一个表用列有表字段和索引的可调整大小的窗口代表。设计器用连接表之间字段的线条表示链接条件。可在表间拖动已索引的字段来创建链接条件。双击一个线条可显示编辑条件的“链接条件”对话框。

图2-8　表设计器

图2-9　查询和视图设计器

其中：

字段选项卡用于指定字段，SUM 或 COUNT 之类的合计函数，或其他表达式。

链接选项卡指定链接表达式，用它来匹配多个表或视图中的记录。

筛选选项卡指定选择记录的条件，比如在字段内指定值或在表之间定义临时关系的连接条件。

排序依据选项卡指定字段，SUM 或 COUNT 之类的合计函数，或用于把有相同字段值的记录合并为一组的其他表达式。

分组依据选项卡指定字段，SUM 或 COUNT 之类的合计函数，或用于把有相同字段值的记录合并为一组的其他表达式。

更新条件选项卡指定更新视图的条件（仅适用于“视图设计器”）。

杂项选项卡指定是否要对重复记录进行检索，同时是否对记录（返回记录的最大数目或最大百分比）做限制。

（4）表单设计器

可视化地创建并修改表单和表单集。一个表单集由一个或多个可作为一个整体处理的表单构成。表单和表单集是有自己的属性、事件和方法程序的对象。

当“表单设计器”窗口处于当前工作状态时，VFP 显示“表单”菜单、“表单控件”工具栏、“表单设计器”工具栏和“属性”窗口。在第 8 章中将详细介绍。

（5）报表设计器

可以创建并修改报表。在“报表设计器”窗口处于当前工作状态时，VFP 显示“报表”菜单和“报表控件”工具栏。

若要快速创建简单的报表布局，可以选择“报表”菜单中的“快速报表”命令，“快速报表”提示输入报表所需的字段和布局。

每种设计器都有一个或多个工具栏，可以很方便地使用大多数常用的功能或工具操作。例如，表单设计器就有分别用于控件、控件布局以及调色板的工具栏。

工作时，可以根据需要在屏幕上放置多个工具栏。通过把工具栏停放在屏幕的上部、底部或两边，可以定制工作环境。VFP 能够记住工具栏的位置，再次进入 VFP 时，工具栏将位于关闭时所在的位置上。

2.2.4 VisualFoxPro 的生成器

生成器是带有选项卡的对话框，用于简化对表单、复杂控件和参照完整性代码的创建和修改过程。每个生成器显示一系列选项卡，用于设置选中对象的属性。可使用生成器在数据库表之间生成控件、表单、设置控件格式和创建参照完整性。

VFP 中提供的生成器见表 2-3。

表 2-3 VFP 可使用的生成器

生成器名称	功　能
组合框生成器	在生成器对话框中通过选择选项为组合框控件设置属性
命令按钮组生成器	在生成器对话框中通过选择选项为命令按钮组控件设置属性
编辑框生成器	在生成器对话框中通过选择选项为编辑框控件设置属性
表单生成器	在生成器对话框中通过选择选项为表单添加控件和指定样式
表格生成器	在生成器对话框中通过选择选项为表格控件设置属性
列表框生成器	在生成器对话框中通过选择选项为列表框控件设置属性

（续）

生成器名称	功　能
选项按钮组生成器	在生成器对话框中通过选择选项为选项按钮组控件设置属性
文本框生成器	在生成器对话框中通过选择选项为文本框控件设置属性
自动格式生成器	对选中的相同类型的控件应用一组样式
参照完整性生成器	在数据库表间创建参照完整性
应用程序生成器	组合数据库、表单、报表为应用程序，或者从零开始创建应用程序，或者创建一个程序框架

通常在下列 5 种情况下使用生成器：

① 使用表单生成器来创建表单。

② 对表单中的控件使用相应的生成器。

③ 使用自动格式生成器设置控件的格式。

④ 对数据库中的表使用参照完整性生成器。

⑤ 使用应用程序生成器为开发的项目生成应用程序。

1．使用表单生成器

首先打开表单设计器，然后使用下列三种方法之一启动表单生成器：

- 在表单上单击鼠标右键，从弹出的快捷菜单上选择“生成器”命令。
- 在“表单”菜单中选择“快速表单”命令。
- 单击表单设计器工具栏中的“表单生成器”按钮。

在打开的“表单生成器”中，如图 2-10 所示，依次选择数据库和（或）表、字段及样式，单击“确定”按钮，即可生成所需的表单。

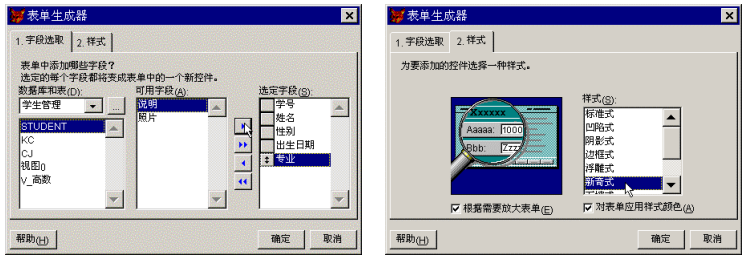


图2-10　表单生成器

2．对表单中的控件使用生成器

在表单设计器中，将所需控件添加到表单上，然后在控件上单击鼠标右键，从弹出的快捷菜单上选择“生成器”命令，即可打开与该控件相应的生成器，如图 2-11 所示。

图2-11　不同控件相应的生成器

按照需要做出选择，单击“确定”按钮，即可使对控件属性的设置生效。

另外，从“表单控件”工具栏中，选择“生成器锁定”按钮。每次向表单添加新控件，VFP 将自动打开相应的生成器。

3．使用自动格式生成器

可以对多个同类控件设置相同格式，具体步骤如下：

在表单设计器中，按下〈Shift〉键，同时选择多个控件。然后从表单设计器工具栏中选择“自动格式”按钮。打开自动格式生成器，选择每个选项卡的选项再单击“确定”按钮即可使属性设置对所有选定控件生效，如图 2-12 所示。

4．使用参照完整性生成器

参照完整性生成器帮助在数据库中设置触发器，用来控制在相关表中插入、更新或删除记录时确保参照完整性，如图 2-13 所示。打开参照完整性生成器的方法有以下三种：

图2-12　自动格式生成器

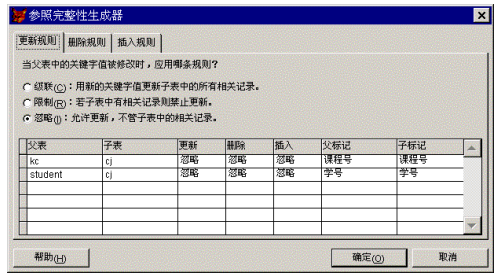


图2-13　参照完整性生成器

- 在数据库设计器中双击两个表中的关系连线，然后在“编辑关系”对话框中单击“参照完整性”按钮。
- 选择“数据库”主菜单中的“编辑参照完整性”菜单项。
- 在“数据库设计器”中单击鼠标右键，从弹出的快捷菜单中选择“编辑参照完整性”命令。

5．使用应用程序生成器

使用应用程序生成器，无需编写任何代码便可创建完整的应用程序。应用程序生成器可以帮助用户完成以下工作：

① 添加、修改或删除应用程序的组件，如表、表单或报表。

② 设置表单和报表的外观样式。

③ 加入常用的应用程序元素，如启动画面、“关于”对话框、“登录”对话框和“标准”工具栏等。

启动应用程序生成器的方法是：在“工具”菜单中选择“向导”子菜单，再选择其中的“全部”项，在打开的“向导选取”对话框中选择“应用程序生成器”，如图 2-14 所示，并按“确定”按钮。

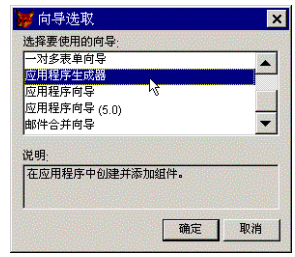


图 2-14　选择应用程序生成器

2.3　项目管理器

项目管理器主要用于帮助用户在创建数据库和表、编制查询和视图、建立表单和报表、

构造应用程序时组织和管理文件。

利用项目管理器，用户可以向项目文件中加入文件、删除文件、创建新文件、修改已有文件、运行命令文件以及与其他项目文件建立关联，并且最后将其编译成一个可独立运行的.APP 或.EXE 文件。

一个项目文件其实就是文件、数据、文档和 VFP 对象的集合，以.PJX 为扩展名保存在磁盘上。但是，值得注意的是，项目文件中保存的并非它所包含的文件，而仅是对这些文件的引用。因此，对于项目文件中的任何文件，用户既可以利用项目管理器对其修改，也可单独对其修改，而且这些文件可同时用于多个项目文件。另外，用户在开发应用程序时，既可一开始就使用项目管理器进行文件的创建和修改，也可分别创建和修改这些文件，最后利用项目管理器进行集成、调试和编译。

2.3.1 创建和打开项目

最好把应用程序中的文件都组织到项目管理器中，这样便于查找。程序开发人员可以用项目管理器把应用软件的多个文件组织成一个文件，生成一个.APP 文件或者.EXE 文件，其中.APP 文件可以用 DO 命令来执行，也可以用 VFP 专业版编译成.EXE 文件。应用程序中的所有文件如.PRG、报表格式文件和标签格式文件，都能组合在一个文件中。如果表和索引不再修改、添加，也可以组合到里面。

1．创建新项目

从“文件”菜单中选择“新建”命令，可以随时创建新项目。创建新项目的步骤为：

① 从“文件”菜单中选择“新建”，或者单击常用工具栏上的“新建”按钮，则打开“新建”对话框，如图 2-15 所示。

选择“项目”选项，然后单击“新建文件”按钮。此时将打开“创建”项目对话框。

② 在“创建”对话框中，输入新项目的名称，在“保存在”中选择保存新项目的文件夹，例如 d:\ch2。

③ 单击“保存”按钮，如图 2-16 所示。

图2-15 “新建”对话框

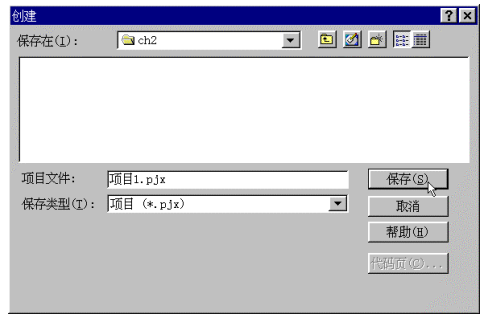


图2-16 “创建”项目对话框

2．打开已有项目

从“文件”菜单中选择“打开”命令，可以随时打开新项目。打开已有项目的步骤为：

① 从“文件”菜单中选择“打开”命令，或者单击常用工具栏上的“打开”按钮，则显示“打开”对话框，如图 2-17 所示。VFP 当前默认的文件夹为 VFP，所以显示此文件夹下的

内容。

② 选择“文件类型”为“项目”。

③ 在“打开”对话框中，输入或选择已有项目的名称。

④ 打开项目文件后将显示项目管理器窗口，如图 2-18 所示，这时就可用“项目管理器”来组织和管理文件了。

图2-17 “打开”对话框

图2-18 项目管理器窗口

2.3.2 项目管理器的操作

“项目管理器”为数据提供了一个组织良好的分层结构视图。若要处理项目中某一特定类型的文件或对象，可选择相应的选项卡。在建立表和数据库，以及创建表单、查询、视图和报表时，所要处理的主要是“数据”和“文档”选项卡中的内容。

图 2-19 “项目管理器”中的“数据”选项卡

1．查找数据文件

“数据”选项卡包含了一个项目中的所有数据：数据库、自由表、查询和视图。“项目管理器”中的“数据”选项卡，如图 2-19 所示。

数据库是表的集合，一般通过公共字段彼此关联，使用“数据库设计器”可以创建一个数据库，数据库文件的扩展名为.DBC。数据库表与自由表都是存储在以.DBF 为扩展名的文件中，前者是数据库的一部分，而后者则独立存在于任何数据库之外。查询是检查存储在表中的特定信息的一种结构化方法，利用“查询设计器”，您可以设置查询的格式，该查询将按照输入的规则从表中提取记录，查询被保存为.QPR 扩展名的文件。视图是特殊的查询，通过更改由查询返回的记录，可以用视图访问远程数据或更新数据源，视图只能存在于数据库中，它不是独立的文件。

2．查找表单和报表文件

“文档”选项卡中包含了处理数据时所用的全部文档：输入和查看数据所用的表单，以及打印表和查询结果所用的报表及标签。“项目管理器”中的“文档”选项卡，如图 2-20 所示。

表单用于显示和编辑表的内容。报表是一种文件，它告诉 VFP 如何设置查询以便从表中提取结果，以及如何将它们打印出来。标签是打印在专用纸上的带有特殊格式的报表。

其余选项卡（如“类”、“代码”及“其他”）主要用于为最终用户创建应用程序。

3．查看文件详细内容

“项目管理器”中的项是以类似于大纲的结构来组织的，可以将其展开或折叠，以便查看不同层次中的详细内容。大纲显示了项目中不同层次内的详细内容，如图 2-21 所示。

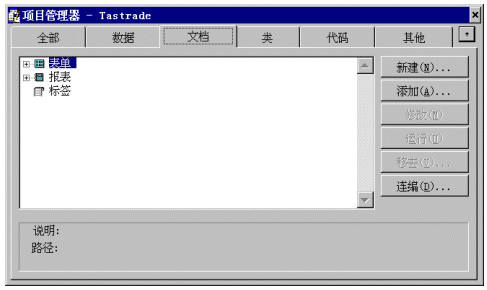


图2-20 “项目管理器”中的“文档”选项卡

图2-21 树状大纲结构

如果项目中具有一个以上某一类型的项，其类型符号旁边会出现一个“+”号。单击符号旁边的“+”号可以显示项目中该类型项的名称及该项的组件，此时“+”号变为“-”号。

例如，单击“自由表”符号旁边的“+”号，可以看到项目中自由表的名称及表中的字段名和索引名。

若要折叠已展开的列表，可单击列表旁边的“-”号。

4．添加或移去文件

要想使用“项目管理器”，必须在其中添加已有的文件或者用它来创建新的文件。例如，如果想把一些已有的扩展名为.DBF 的表添加到项目中，只需在“数据”选项卡中选择“自由表”，然后用“添加”按钮把它们添加到项目中。

在项目中加入文件的步骤为：首先选择要添加项的类型，单击“添加”按钮，在“打开”对话框中，选择要添加的文件名，最后单击“确定”按钮。

从项目中移去文件的步骤为：选定要移去的内容，单击“移去”按钮。

如果要从计算机中删除文件，单击“删除”按钮。

5．创建和修改文件

“项目管理器”简化了创建和修改文件的过程。只需选定要创建或修改的文件类型，然后选择“新建”或“修改”按钮，VFP 将显示与所选文件类型相应的设计工具。

创建文件的步骤为：首先选定要创建的文件类型，单击“新建”按钮。然后进行相应的创建操作，新创建的文件即被添加到“项目管理器”中。

对于某些类型的文件，可以利用向导来创建。

修改文件的步骤为：选定一个已有的文件，单击“修改”按钮。

例如，要修改一个表，先选定表的名称，然后单击“修改”按钮，该表便显示在“表设计器”中。

创建或添加新的文件时，可以为文件加上说明。文件被选定时，说明将显示在“项目管理器”的底部。为文件添加说明的步骤为：首先在“项目管理器”中选定文件，并从“项目”菜单中选择“编辑说明”，然后在“说明”对话框中键入对文件的说明，最后单击“确定”按钮。

6．查看表中的数据

从“项目管理器”中可以浏览项目中表的内容。浏览表的步骤为：选择“数据”选项卡，

从中选定一个表并单击“浏览”按钮。

7．项目间共享文件

通过与其他项目共享文件，你可以共享在其他项目开发上的工作成果。此文件并未复制，项目只储存了对该文件的引用。文件可同时和不同的项目连接。

在项目之间共享文件的步骤为：在 VFP 中，打开要共享文件的两个项目，在包含该文件的“项目管理器”中，选择该文件，拖动该文件到另一个的项目容器中。

2.3.3 定制“项目管理器”

可以定制可视工作区域，方法是改变“项目管理器”的外观或设置在“项目管理器”中双击运行的文件。

1．改变显示外观

“项目管理器”通常显示为一个独立的窗口。可以移动它的位置、改变它尺寸或者将它折叠起来只显示选项卡。

移动“项目管理器”：将鼠标指针指向标题栏，然后将“项目管理器”拖到屏幕上的其他位置。

改变“项目管理器”窗口的大小：将鼠标指针指向“项目管理器”窗口的顶端、底端、两边或角上，拖动鼠标即可扩大或缩小它的尺寸。

折叠“项目管理器”：单击右上角的上箭头，在折叠情况下只显示选项卡，如图 2-22 所示，这时上箭头变为下箭头。

图2-22 折叠项目管理器

可以很容易地将“项目管理器”还原为通常大小。若要还原“项目管理器”，只需单击右上角的下箭头即可。

2．拖开选项卡

折叠“项目管理器”后，可以拖开选项卡，该选项卡成为浮动状态，可根据需要重新安排它们的位置。拖开某一选项卡后，它可以在 VFP 的主窗口中独立移动。

拖开某一选项卡的步骤为：折叠“项目管理器”，然后选定一个选项卡，将它拖离“项目管理器”，如图 2-23 所示。

当选项卡处于浮动状态时，通过在选项卡中单击鼠标右键可以访问“项目”菜单中的选项，如图 2-24 所示。

3．“项目管理器”中的选项卡

如果希望选项卡始终显示在屏幕的最表层，可以单击选项卡上的图钉图标，这样，该选项卡就会一直保留在其他 VFP 窗口的上面。可以使多个选项卡都处于“顶层显示”的状态。再次单击图钉图标可以取消选项卡的“顶层显示”设置，如图 2-25 所示。

若要还原选项卡，单击选项卡上的“关闭”按钮即可。或者，将选项卡拖回到“项目管理器”中。

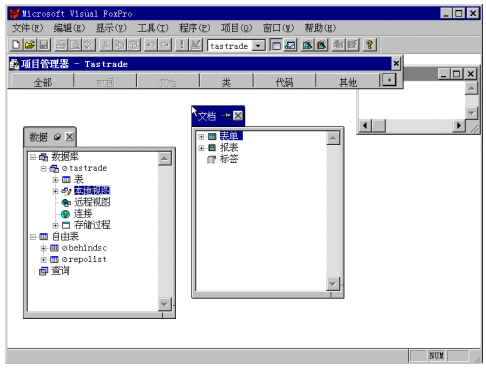


图2-23　浮动选项卡

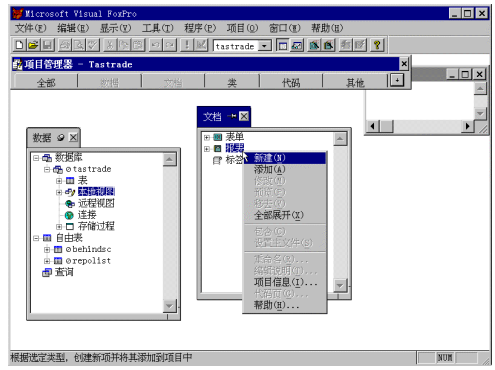


图2-24　在浮动选项卡中单击鼠标右键

还可以停放“项目管理器”，使它像工具栏一样显示在 VFP 主窗口的顶部。

若要停放“项目管理器”，将“项目管理器”拖到 VFP 主窗口的顶部即可。

停放“项目管理器”后，它就变成窗口工具栏区域的一部分。“项目管理器”处于停放状态时，不能将其展开，但是可以单击每个选项卡来进行相应的操作。对于停放的“项目管理器”，同样可以从中拖开选项卡，如图 2-25 所示。

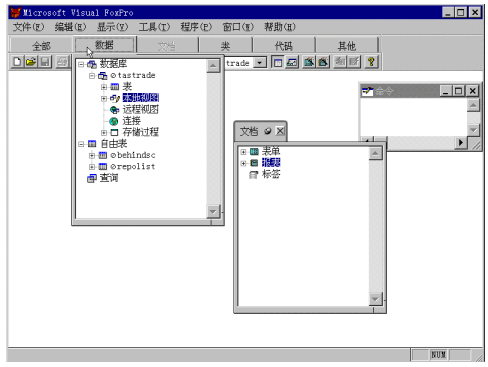


图2-25　停放“项目管理器”

2.3.4　项目管理器中的命令按钮

项目管理器中每个选项卡右方都包含几个命令按钮，这些按钮会根据选取的文件而改变，显示出可以使用的按键，无法使用的按钮显示灰色而无法选取。在使用这些命令之前，要先在左方选取需要的文件类型。表 2-4 给出项目管理器中的命令按钮及其使用说明。

表 2-4　项目管理器中的命令按钮

按　钮	说　明
新文件	创建一个新文件或对象。此按钮与项目菜单的新文件命令作用相同。新文件或对象的类型与当前选定项的类型相同。从“文件”菜单中创建的文件不会自动包含在项目中。由“项目”菜单的“新文件”命令或项目管理器上的“新文件”按钮创建的文件自动包含在项目中
添加	把已有的文件添加到项目中。此按钮与“项目”菜单的添加文件命令作用相同
修改	在合适的设计器中打开选定项。此按钮与“项目”菜单的修改文件命令作用相同

（续）

按　钮	说　明
浏览	在浏览窗口中打开一个表。此按钮与“项目”菜单的浏览文件命令作用相同，且仅当选定一个表时可用
关闭	关闭一个打开的数据库。此按钮与“项目”菜单的关闭文件命令作用相同，且仅当选定一个表时可用。如果选定的数据库已关闭，此按钮变为“打开”
打开	打开一个数据库。此按钮与“项目”菜单的打开文件命令作用相同，且仅当选定一个表时可用。如果选定的数据库已打开，此按钮变为“关闭”
移去	从项目中移去选定文件或对象。Visual FoxPro 会询问是仅从项目中移去此文件还是同时将其从磁盘中删除。此按钮与“项目”菜单的移去文件命令作用相同
连编	连编一个项目或应用程序，在专业版中，还可以连编一个可执行文件。此按钮与“项目”菜单的连编命令作用相同
预览	在打印预览方式下显示选定的报表或标签。当选定“项目管理器”中的一个报表或标签时可用。此按钮与“项目”菜单的预览文件命令作用相同
运行	执行选定的查询、表单或程序。当选定项目管理器中的一个查询、表单或程序时可用。此按钮与项目菜单的运行文件命令作用相同

2.4　配置 Visual FoxPro

安装 VFP 后，可以根据需要定制开发环境。环境设置包括主窗口标题、默认目录、项目、编辑器、调试器及表单工具选项、临时文件存储、拖放字段对应的控件和其他选项。用户既可以用交互式，也可以用编程的方法配置 VFP。甚至可以使 VFP 启动时调用用户自建的配置文件。

2.4.1　设置环境和管理临时文件

1．使用“选项”对话框

可以在“命令”窗口的程序中使用 SET 命令设置环境，也能使用下列方式交互地在“选项”对话框中设置、查看或更改环境选项。

从“工具”菜单中选择“选项”命令，打开“选项”对话框。“选项”对话框中具有一系列代表不同类别环境选项的选项卡。例如，在“显示”选项卡中，可以设置 VFP 主窗口显示方式，如图 2-26 的左图所示。在“区域”选项卡中，可以设置日期格式、货币格式等项，图 2-26 的右图所示。

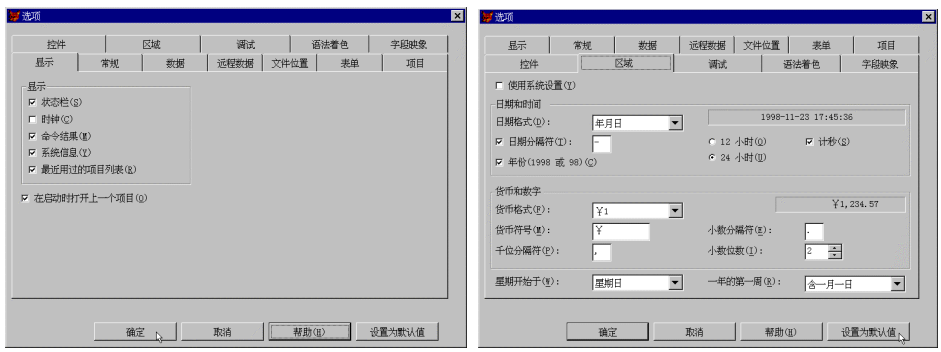


图2-26　“选项”对话框

2．保存设置

可以把在“选项”对话框中所做设置保存为在当前工作期有效或者是 VFP 的默认（永久）

设置。

把设置保存为仅在当前工作期有效的方法为：在“选项”对话框中更改设置，然后单击“确定”按钮。所作的设置一直作用到退出 VFP（或直到再次更改它们）为止。

把当前设置保存为默认设置的方法为：在“选项”对话框中更改设置，然后单击“设置为默认值”按钮，再单击“确定”按钮。系统将把设置存储在 Windows 注册表中。

通过执行 SET 命令或在启动 VFP 时指定一个配置文件，可以忽略默认设置。

3．管理临时文件

VFP 在许多操作中产生临时文件。例如，编辑、索引、排序时，都要产生临时文件。文本编辑期间也会产生正在编辑的文件的临时副本。

除非为临时文件指定其他位置，否则 VFP 会在 Windows 保存临时文件的目录中创建临时文件。指定临时文件位置的步骤为：

① 从“工具”菜单中，选择“选项”命令，然后选择“文件位置”选项卡。

② 输入临时文件的位置，如图 2-27 所示。若要永久保存所做的更改，单击“设置为默认值”按钮。

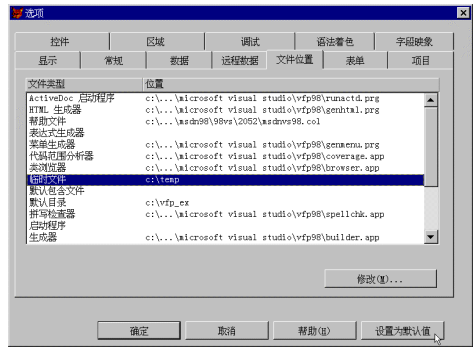


图2-27　输入临时文件的位置

2.4.2　配置 VisualFoxPro 工具栏

VFP 可定制的工具栏，见表 2-5。

表 2-5　VFP 可定制的工具栏

工　具	相关的工具栏	命　令
数据库设计器	数据库	CREATE DATABASE
表单设计器	表单控件、表单设计器、调色板、布局	CREATE FORM
打印预览	打印预览	
查询设计器	查询设计器	CREATE QUERY
报表设计器	报表控件、报表设计器、调色板、布局	CREATE REPORT

1．激活工具栏及使工具栏不活动

默认情况只有“常用”工具栏可见。当使用一个 VFP 设计器工具时，将激活相关的工具栏。当然，还可以在任何需要时激活一个工具栏。

- 若要激活一个工具栏，可以运行相应的设计器工具，或者从“显示”菜单选择“工具栏”，在“工具栏”对话框中选择希望激活的工具栏，如图 2-28 所示。
- 若要使一个工具栏不活动，可以关闭相应工具，或者从“显示”菜单选择“工具栏”，在“工具栏”对话框清除欲使之不活动的工具栏。

2．定制现有工具栏

创建自定义工具栏最简单的方法就是修改 VFP 提供的工具栏。可以通过添加或移去按钮修改现有工具栏，创建包含现有工具栏按钮的新工具栏。也可以使用代码创建一个自定义工具栏类来定义自定义工具栏。

（1）修改现有 VFP 工具栏

可以从一个现有工具栏中移去一个按钮或者从一个工具栏向另一个工具栏复制按钮。修改 VFP 工具栏的步骤为：首先，从“显示”菜单中选择“工具栏”，在工具栏对话框中选定希望定制的工具栏并单击“定制”按钮，打开该工具栏和“定制工具栏”对话框。

可以在“定制工具栏”对话框中选择适当的类别，然后把所需按钮拖到工具栏上，如图 2-29 所示，即向工具栏添加按钮；也可从希望定制的工具栏中将某些按钮拖离工具栏而移去它们。

图2-28 “工具栏”对话框

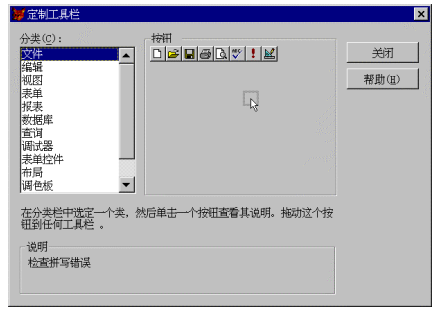


图2-29 把按钮拖到工具栏

最后，单击“关闭”按钮，结束工具栏的定制工作。

在更改了 VFP 工具栏之后，还可以把它恢复到原来的配置。方法是在“工具栏”对话框中选择工具栏，然后单击“重置”按钮。

（2）从现有工具栏创建新工具栏

可以创建由来自其他工具栏的按钮组成的全新工具栏。创建 VFP 新工具栏的步骤为：从“显示”菜单中选择“工具栏”命令，在工具栏对话框中单击“新建”按钮，在打开的“新工具栏”对话框中命名工具栏，如图 2-30 所示，并单击“确定”按钮。

图 2-30 “新工具栏”对话框

然后，在“定制工具栏”对话框中选择一个类别，把所需的按钮拖到新建的工具栏上。可以用鼠标把工具栏上的按钮拖动到所需位置来重排它们。

最后，在“定制工具栏”对话框中单击“关闭”按钮，结束工具栏的创建工作。不能重置自己创建的工具栏上的按钮。

若要删除创建的工具栏，从“显示”菜单选择“工具栏”命令，选定欲删除的工具栏，

单击“删除”按钮，再单击“确定”按钮，确认删除。不能删除 VFP 提供的工具栏。

2.4.3 设置编辑器选项

在“编辑属性”对话框中，可以配置 VFP 编辑器使之按用户希望的方式显示文本。如可以设置字体与文本对齐，还可以设置缩进、换行、自动备份文件等其他特性，使编辑器更易使用。

若要显示“编辑属性”对话框，首先需要打开一个编辑器窗口。下列方法之一可以打开编辑器窗口：

- 在“项目管理器”中选择一个程序或文本文件，然后单击“新建”按钮。或者双击现有程序或文本文件的名称。
- 在“命令”窗口中输入 MODIFY COMMAND，MODIFY FILE 或 MODIFY MEMO。
- 从“文件”菜单选择“新建”命令，然后指定文件类型为“程序”或“文本文件”；或者选择“打开”命令，然后选择程序或文本文件名称。
- 在“表单设计器”中双击一个表单或控件。

在编辑器窗口的任意位置上单击鼠标右键弹出快捷菜单，然后选择“属性”命令，将打开“编辑属性”对话框，如图 2-31 所示。

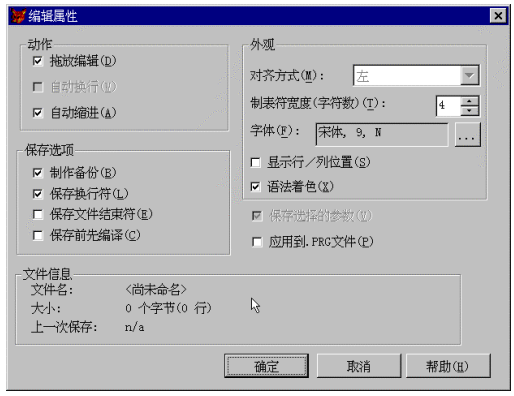


图2-31 “编辑属性”对话框

默认时，在“编辑属性”对话框中所作设置仅用于当前编辑窗口。例如，如果更改字体，当前窗口全部文本的字体都更改。如果打开另外一个编辑窗口或关闭当前窗口再重新打开一次，仍将使用原来的默认字体。

如果愿意，可以永久地保存用户的设置，或者说它们将用于所有相似类型的文件。如果选择对相似文件类型应用选项，在打开具有同样扩展名的文件时（例如，所有.PRG 文件或所有表单设计器的方法程序代码），VFP 将使用该设置。

若要永久保存编辑器选项，则在“编辑属性”对话框中，选择“保存选择的参数”复选框，单击“确定”按钮。

若要对相似文件应用编辑器选项，则在“编辑属性”对话框中，选择“应用到.PRG 文件”或“应用到.TXT 文件”复选框，然后单击“确定”按钮。

还能够在编辑器中设置关键字、注释以及其他程序元素专用的颜色和字体。

在编辑窗口中单击鼠标右键，从快捷菜单中选择“字体”，可以直接显示“字体”对话框。

2.4.4 恢复 VisualFoxPro 环境

如果希望关闭所有操作返回 VFP 启动时的状态，在命令窗口或在退出 VFP 之前最后调用的程序中，按下列顺序运行如下命令：

```
CLEAR ALL
CLOSE ALL
CLEAR PROGRAM
```

说明：CLEAR ALL 从内存中移去所有对象，按顺序关闭所有私有数据工作期以及其中的临时表。

在 CLEAR ALL 正确执行后，CLOSE ALL（关闭）VFP 默认数据工作期，即数据工作期 1 中的所有数据库、表以及临时表。

CLEAR PROGRAM 清除最近执行程序的程序缓冲区。CLEAR PROGRAM 迫使 VFP 从磁盘而不是从程序缓冲区中读取文件。

在事务过程中清理：如果事务正在执行过程中，应在执行 CLEAR ALL，CLOSE ALL 以及 CLEAR PROGRAM 之前对每一层事务使用 END TRANSACTION 命令。

在缓冲式更新过程中清理：如果在缓冲式更新的过程中，应在执行 CLEAR ALL，CLOSE ALL 以及 CLEAR PROGRAM 之前，对每一个有缓冲式更新的临时表使用 TABLEUPDATE( ) 或 TABLEREVERT( )函数。

2.5 使用帮助和联机文档

使用 VFP 帮助系统，可以快速查询到有关 VFP 设计工具和程序语言的信息。

2.5.1 获得帮助

如果对某个窗口或对话框的含义不理解，只要按〈F1〉键，就可以显示出关于该窗口或对话框的上下文相关的帮助信息。

选择“帮助”菜单中的“Microsoft Visual FoxPro 帮助主题”命令，可以得到 VFP 联机帮助的内容概述。若要查找有关特定术语或主题的帮助信息，请选择“帮助”菜单中的“索引”选项卡。

2.5.2 联机文档

从任何一个对话框中单击“帮助”按钮、按〈F1〉键，或者从“开始”菜单“程序”中的“Microsoft Developer Network”中单击“MSDN Library Visual Studio 6.0”，都将打开联机文档，如图 2-32 所示。

在 VFP 的联机文档中带有非常详细的帮助内容，如安装指南、用户指南、开发指南、语言参考，在语言参考中有 VFP 全部的语句和函数，读者应该学会通过联机文档学习 VFP，所以在本教材中为了节省篇幅，有些语句和函数没有给出语法和说明，读者应该使用联机文档

自己查看。

图2-32　联机文档

MSDN Library 是一个分为三个窗格的帮助窗口。顶端的窗格包含有工具栏，左侧的窗格包含有各种定位方法，右侧的窗格则显示主题内容，此窗格拥有完整的浏览器功能。定位窗格包含有“目录”、“索引”、“搜索”及“书签”选项卡。单击目录、索引或书签列表中的主题，即可浏览 Library 中的各种信息。“搜索”选项卡可用于查找出现在任何主题中的所有单词或短语。单击主题中带有下划线的词，即可查阅与该主题有关的其他内容。

- 单击有下划线的彩色字，可链接到另一个主题、网页、其他主题的列表或是某个应用程序。
- 先单击出现在主题开始处的带有下划线的“请参阅”一词，然后再单击要浏览的主题，即可选择含有相关内容的其他主题。
- 选中某个词或短语使其突出显示，然后按下〈F1〉键，即可查看“索引”中是否有包含该词或短语的主题。
- 搜索主题时可使用布尔操作符来优化搜索。
- 如果正在主题窗格中浏览网页上的内容，可使用工具栏上的“停止”和“刷新”按钮来中断下载或刷新网页的内容。

2.5.3　获得示例

为了演示其程序设计技术，VFP 提供了一系列有关应用程序、数据库和文件的示例。有关其详细内容，请从 VFP 的“帮助”菜单中选择“示例应用程序”命令。

2.6　使用 Visual FoxPro

本节介绍使用 VFP 的一些基本概念。

2.6.1　VisualFoxPro 的工作方式

VFP 的工作方式分为交互方式与程序方式两种。

1．交互方式

交互方式是通过人机对话来执行各项操作。在 VFP 中，有两种交互方式，即命令方式和

可视化操作方式。

- 命令方式即通过命令窗口输入合法的 VFP 命令来完成各种操作，例如，在命令窗口输入命令：

```
DIR
```

按〈Enter〉键后，系统将在 VFP 主窗口显示当前目录下，所有数据表文件（.DBF）的列表。

- 可视化操作方式则是利用 VFP 集成环境提供的各种工具，如菜单、工具栏、设计器、生成器、向导等，来完成各项操作，这种方法非常直观，简单易学。

2．最简单的操作命令

（1）输出命令

?命令计算并在 VFP 主窗口中显示各表达式的值。命令格式为：

```
? [〈表达式列表〉]
```

说明：若表达式多于一项，各表达式间要用逗号隔开，显示时各表达式值间各空一格，如图 2-33 所示。

图2-33　非格式输出

（2）清屏命令

CLEAR 用来清除 VFP 主窗口中的任何输出内容。命令格式为：

```
CLEAR
```

3．程序方式

通过把 VFP 的合法命令组织、编写成命令文件（程序），或是利用 VFP 提供的各种程序生成工具，如表单设计器、菜单设计器、报表设计器等来设计程序。然后执行程序，来完成特定的操作任务。

2.6.2　VisualFoxPro 的语法规则

VFP 的命令由命令名和一些命令短语组成。只要符合 VFP 的语法规则，就可以用命令名和短语组合成多种命令，所以掌握语法规则十分重要。

在进一步说明之前，我们对一些符号作出以下约定：

① [ ] —— 任选项约定符，表示其中内容可选可不选。

②〈 〉—— 必选项约定符，表示其中内容由用户输入，必须选择。

③ { | } —— 选择项约定符，表示其中多项内容选择其一。

1．命令格式

（1）命令的一般格式

VFP 命令的一般格式为：

〈命令名〉 [〈短语 1〉] [〈短语 2〉] ... [〈短语 *n*〉] （*n*≥0）

其中命令名表示命令的功能，短语给出操作的对象和条件。

（2）记录操作命令的一般格式

对记录操作命令的一般格式为：

〈命令名〉 [〈范围〉] [FIELDS 〈字段名表〉] [{FOR | WHILE}〈条件〉]

2．命令短语

在 VFP 命令中，一般包含如下命令短语（选项）：

（1）范围选项

对数据记录不止一条的表进行操作时就需要指明记录范围，用来确定执行该命令涉及的记录，VFP 命令的范围有 4 种表示方法，其意义见表 2-6。

表 2-6 范围选项

选 项	说 明
ALL	对全部记录进行操作
NEXT〈*n*〉	只对包括当前记录在内的以下连续的 *n* 个记录进行操作
RECORD〈*n*〉	只对第 *n* 号记录进行操作
REST	只对当前记录起到文件尾的所有记录进行操作

默认范围选项时一般为 ALL，个别命令的默认范围是当前一条记录，以后如不作特别说明默认范围为 ALL。

（2）条件选项

命令的条件选项是指对选择范围内的数据进行“ 筛选 ”，用以筛选出用户需要操作的数据记录，其中的〈条件〉为逻辑或关系表达式，用以指定选择记录的条件。

FOR 选项表示对〈范围〉内所有满足〈条件〉的记录执行该命令，有些默认范围为当前记录的语句，使用 FOR 选项后范围扩充为 ALL；WHILE 选项表示对〈范围〉内满足〈条件〉的记录逐条执行命令，一旦遇到不满足者，即停止执行，如果当前记录就使〈条件〉为“ 假 ”，则相当与操作没有进行。

若一条命令中同时有 FOR 与 WHILE 子句则后者优先处理。

（3）字段选项

VFP 命令的字段选项是从表的所有字段中“ 过滤 ”出需要操作的字段，字段列表中的字段数如果有两个以上时，用“ , ”分隔。

3．命令的书写规则

① 每个命令必须以一个命令名开始。

② 命令格式中各短语顺序可调换，例如下面的两条命令是等效的：

DISPLAY ALL FOR 年龄 < 30 FIELDS 姓名, 工作日期
DISPLAY FIELDS 姓名,工作日期 ALL FOR 年龄 < 30

③ 命令行中各个词之间至少应以一个空格隔开。若两个词之间已有引号等界限符，则空格可以省略，但要注意：逻辑值中的小圆点与字母之间不能有空格。

④ 命令中的英文字母可以用大写、小写或大小混写。

⑤ 命令中的单词可以用其前 4 个或 4 个以上字符缩写表示，如上面的命令可以写成：

DISP FIEL 姓名, 工作日期 ALL FOR 年龄 < 30

⑥ 一条命令的长度可达 8192 个字符，若一行写不下，可在适当位置使用续行符“ ; ”并回车，然后在下一行继续键入该命令的剩余部分。

2.7 习题

一、选择题

1．退出 VFP 的方法是（　　）。

A）从“ 文件 ”菜单中选择“ 退出 ”菜单项

B）用鼠标左键单击关闭窗口按钮

C）在命令窗口键入 QUIT 命令，然后按〈Enter〉键

D）以上方法都可以

2．显示与隐藏命令窗口的操作方法是（　　）。

A）用鼠标单击“ 常用 ”工具栏上的“ 命令窗口 ”按钮

B）通过“ 窗口 ”菜单下的“ 命令窗口 ”项来切换

C）直接按〈Ctrl+F2〉或〈Ctrl+F4〉组合键

D）以上方法都可以

3．下述关于工具栏的叙述，错误的是（　　）。

A）可以创建用户自己的工具栏　　B）可以修改系统提供的工具栏

C）可以删除用户创建的工具栏　　D）可以删除系统提供的工具栏

4．在“ 选项 ”对话框的“ 文件位置 ”选项卡中可以设置（　　）。

A）表单的默认大小　　B）默认目录

C）日期和时间的显示格式　　D）程序代码的颜色

5. “ 项目管理器 ”的“ 数据 ”选项卡用于显示和管理（　　）。

A）数据库、自由表和查询　　B）数据库、视图和查询

C）数据库、自由表、查询和视图　　D）数据库、表单和查询

6. “ 项目管理器 ”的“ 文档 ”选项卡用于显示和管理（　　）。

A）表单、报表和查询　　B）数据库、表单和查询

C）查询、报表和视图　　D）表单、报表和标签

二、填空题

1．安装完 VFP 之后，系统自动使用默认值来设置环境，要定制自己的系统环境应选择_____菜单下的_____菜单项。

2．在“选项”对话框中，要设置日期和时间的显示格式，应当选择_____选项卡。

3．扩展名为.PRG 的程序文件在“项目管理器”中的_____选项卡中显示和管理。

4．“项目管理器”中的‘移去’按钮有两个功能：一是把文件_____，二是_____文件。

第 3 章　数据及其运算

数据是计算机程序处理的对象，也是运算产生的结果，所以我们首先应该认识 VFP 能处理哪些数据，掌握各种形式数据的表示方法。可以从各种不同的角度对数据进行分类。

从数据的类型来分，数据可分为数值型数据、字符型数据、逻辑型数据等。

从数据的存储方式来分，数据可分为常量和变量。

从数据的处理层次上分，数据又可分为常量、变量、函数和表达式。常量和变量是数据处理的基本对象，而表达式和函数则体现了计算机语言对数据处理的能力。

3.1　数据的类型

数据类型是数据的基本属性。对数据进行操作的时候，只有同类型的数据才能进行操作，若对不同类型的数据进行操作，将被系统判为语法出错。

VFP 的数据类型分为两大类，即基本数据类型和只可用于字段的数据类型。

3.1.1　基本的数据类型

VFP 的基本数据类型既可用于字段变量，又可用于常量、内存变量、表达式，包括数值型、字符型、货币型、日期型、日期时间型、逻辑型等。

1．数值型数据（Numeric）

数值型数据用来表示数量，它由数字 0～9、一个符号（+或−）和一个小数点（.）组成。

计算机储存数值型数据使用 8 个字节，数值型数据取值范围：$-0.9\ 999\ 999\ 999\times10^{19}$～$0.9\ 999\ 999\ 999\times10^{20}$，一般含有数字的数据均可表示为数值型，但像电话号码、邮政编码等可表示为字符型数据。

2．货币型数据（Currency）

货币型数据用来表示货币值，其书写格式与数值型类似，但不能使用浮点法表示。货币型数据最多保留四位小数，多余小数采用四舍五入法截取。取值范围：

−922 337 203 685 477.5807～922 337 203 685 477.5807

货币型数据也使用 8 个字节存放。一般和货币有关的数据可表示为货币型，如工资、单价等。

3．字符型数据（Character）

字符型数据由字母（汉字）、数字、空格等任意 ASCII 码字符组成，每个字符占一个字节。

4．日期型数据（Date）

日期型数据用以保存不带时间的日期值，存储格式为“yyyymmdd”，其中 yyyy 为年，占 4 位，mm 为月，占 2 位，dd 为日，占 2 位。日期型数据的表示有多种格式，最常用的格式为 mm/dd/ yyyy。日期型数据取值的范围是：

公元 0001 年 1 月 1 日～公元 9999 年 12 月 31 日

日期型数据通常表示和日期有关的数据，如工作日期、入学时间等可采用此类型。

5．日期时间型数据（DateTime）

用以表示日期和时间值。日期时间型数据的存储格式为“ yyyymmddhhmmss ”，其中 yyyy 为年，占 4 位，mm 为月，占 2 位，dd 为日，占 2 位，hh 为时间中的小时，占 2 位，mm 为时间中的分钟，占 2 位，ss 为时间中的秒，占 2 位。

日期时间型数据中可以只包含一个日期或者只包含一个时间值，默认日期值时，系统自动加上 1999 年 12 月 31 日，省略时间值时，则自动加上午夜零点。

6．逻辑型数据（Logical）

逻辑型用于存储只有两个值的数据。存入的值只有真（.T.）和假（.F.）两种状态，占 1 个字节。

3.1.2 数据表中字段的数据类型

以下数据类型只能被用于数据表中的字段：

1．双精度型（Double）

用于取代数值型，以便能提供更高的数值精度。双精度型只能用于数据表中字段的定义，它采用固定存储长度的浮点数形式。与数值型不同，双精度型数据的小数点的位置是由输入的数据值来决定的。双精度型数值的取值范围是：

+/−4.94065645841247E−324～+/−8.9884656743115E307

每个双精度型数据占 8 个字节。

2．浮点型（Float）

只能用于数据表中字段的定义，包含此类型是为了提供兼容性，浮点型在功能上与数值型等价。

3．通用型（General）

用于存储 OLE 对象，只能用于数据表中字段的定义。该字段包含了对 OLE 对象的引用，而 OLE 对象的具体内容可以是一个电子表格、一个字处理器的文本、图片等，是由其他应用软件建立的。

4．整型（Integer）

用于存储无小数部分的数值，只能用于数据表中字段的定义。在数据表中，整型字段占用 4 个字节，取值范围是：−2147483647～2147483647

整型以二进制形式存储，不像数值型那样需要转换成 ASCII 字符存储。

5．备注型（Memo）

备注型用于字符型数据块的存储，只能用于数据表中字段的定义。在数据表中，备注型字段占用 10 个字节，并用这 10 个字节来引用备注的实际内容。实际备注内容的多少只受内存可用空间的限制。

备注型字段的实际内容变化很大，不能直接将备注内容存在数据表（.DBF）文件中。系统将备注内容存放在一个相对独立的文件中，该文件的扩展名为.DBT。

由于没有备注型的变量，所以对备注型字段的处理，需转换成字符型变量，然后使用字符型函数进行处理。

6．字符型（二进制）

用于存储任意不经过代码页修改而维护的字符数据，只能用于数据表中字段的定义。

7．备注型（二进制）

用于存储任意不经过代码页修改而维护的备注型数据，只能用于数据表中字段的定义。

3.2 常量与变量

在程序的运行过程中，我们把需要处理的数据存放在内存储器中，称始终保持不变的数据为“常量”，称存放可变数据的存储器单元为“变量”，其中的数据称为变量的值。

3.2.1 常量

常量是一个具体的数据项，在整个操作过程中其值保持不变。VFP 定义了如下类型的常量：

1．数值型常量

数值型常量即常数，用来表示一个数量的大小，如 2.134。数值型常量可以表示为定点形式也可表示为浮点形式，定点形式如 98，3.14159，−0.76。浮点形式如 8.67E2、−4.51E−2，分别表示 8.67×10^{2} 和 -4.51×10^{-2}，其中 E（E 为半角字符，以后不作特别说明时均表示要求使用半角符号，全角符号值只允许出现在字符型数据中）不区分大、小写。

2．字符型常量

字符型常量即字符串，凡是使用 VFP 允许的定界符（单引号、双引号或方括号）引导的符号可视为字符串。定界符必须成对出现，字符中已包含某一定界符时，可采用其他定界符引导，如"AAA"、数据库应用'、[Visual FoxPro]、"单价：'245.78"，定界符仅仅起到说明数据类型的作用，不是数据的一部分，如果定界符之间不出现任何字符（包括空格）称为“空串”，如""。

3．逻辑型常量

逻辑型常量只有“逻辑真”和“逻辑假”两个值，凡是可由两种情况表示的数据均可采用逻辑常量，如男与女、及格与不及格等。

逻辑常量使用“.”作为定界符，用.T.、.t.、.Y.、.y.表示逻辑真，用.F.、.f.、.N.、.n.表示逻辑假。

4．日期型常量

日期型常量必须用一对花括号“{”和“}”作为定界符，花括号中包含用分隔符“/”分隔的年、月、日三部分内容，其格式分为严格格式和传统格式两种：

（1）传统格式为{mm/dd/yy}，系统默认的格式为美国日期格式“月/日/年”，其中月、日、年各为两位数字。

传统格式的日期型常量要受到命令语句 SET DATE TO 和 SET CENTURY 设置的影响。即不同的设置，VFP 会对同一个日期型常量做出不同的解释，如{10/08/02}可以被解释为 2002 年 10 月 8 日、2102 年 8 月 10 日、2010 年 8 月 2 日等。

（2）严格格式为{^yyyy-mm-dd}，其中，花括号中第一个字符必须是脱字符“^”，年份必须是 4 位，年月日的顺序不能颠倒或默认。

用严格格式书写的日期常量可以表示一个确切的日期，不受命令语句 SET DATE TO 和 SET CENTURY 设置的影响。

5．日期时间型常量

日期时间型常量包括日期和时间两部分的内容，表示形式为{〈日期〉,〈时间〉}，日期部分与日期型常量相似，时间格式为“hh[:mm[:ss]][a|p]”，其中 hh 表示时（系统默认 12）、mm 表示分（系统默认 0）、ss 表示秒（系统默认 0）、a 表示上午（系统默认）、p 表示下午。时间也可以使用 24 小时制。

日期时间型常量也有传统与严格两种格式。如严格格式的日期时间型常量{^1999-10-01 10: 00:00am}，其中 am 表示上午。空的日期时间型常量值表示为{:}。

6．货币型常量

货币型常量的书写格式与数值型常量类似，但要加上一个前置符$，如$123.456。

3.2.2 变量

VFP 有三种形式的变量：内存变量、数组变量和字段变量。内存变量是存放单个数据的内存单元，数组变量是存放多个数据的内存单元组，而字段变量则是存放在数据表中的数据项。本节讨论的变量仅指内存变量。

1．变量的命名

每个变量都有一个名称，叫做变量名，VFP 通过相应的变量名来使用变量。变量名的命名规则是：

① 以字母、数字及下划线组成，中文 VFP 可以使用汉字作变量名。

② 以字母或下划线开始，中文 VFP 可以汉字开始。

③ 长度为：1~128 个字符，每个汉字占 2 个字符。

④ 不能使用 VFP 的保留字。

如果当前数据表中有同名的字段变量，则访问内存变量时，必须在变量名前加上前缀“M.”或“M->”（减号、大于号），否则系统将访问同名的字段变量。

2．变量的赋值

在 VFP 中，变量必须定义以后才能被使用。但是向内存变量赋值无需事先定义，变量的定义和赋值同时完成。赋值命令的格式有两种。

命令格式 1：

〈内存变量名〉=〈表达式〉

命令格式 2：

STORE〈表达式〉TO〈内存变量表〉

说明：

① 首先计算〈表达式〉，然后将值赋给内存变量。

②〈内存变量表〉表示用逗号分隔的多个内存变量。格式 1 一次仅给一个变量赋值，格式 2 一次可以给多个变量赋值。

【例 3-1】给内存变量 *x*，*y*，*z* 赋值。

```
x = {^2002-10-31}          && x 的值为 2002 年 10 月 31 日，类型为 D
STORE 1/2 TO y, z          && y、z 的值都为 0.5
```

在命令窗口依次键入上面的命令并按〈Enter〉键，屏幕显示如图 3-1 所示。

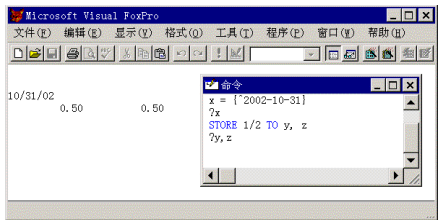


图3-1　给变量赋值

3．变量的类型

变量的类型是指其存放的数据的值。在 VFP 中，有 6 种类型的内存变量。

（1）数值型（N）

数值型存放数值型数据，当数值的位数大于或等于数值型数据的最大宽度 20 位时，则用浮点形式表示。如 12345678901234567890 表示为 1.2345678901234E+19。

（2）字符型（C）

字符型又称字符串变量，用于存放字符型数据。

（3）逻辑型（L）

逻辑型用于存放逻辑型数据，只能存放真（.T.、.t.、.Y.、.y.）或假（.F.、.f.、.N.、.n.）两种逻辑值。

（4）日期型（D）

日期型用于存放日期。

（5）日期时间型（T）

日期时间型同时存放日期和时间。

（6）货币型（Y）

货币型用于存放货币型数据。

（7）对象型（O）

对象型用于存放对象型数据。

建立变量时不必指定变量的类型，在内存变量中存放什么类型的数据，该变量就具有什么类型。可以通过赋值命令随时建立、修改变量的类型和值。

4．变量的作用域

命令窗口定义的变量在本次 VFP 运行期间都可以使用，直到使用 CLEAR MEMORY 命令或 RELEASE 命令将其清除。但如果是在程序中定义的变量，情况有所不同。一般来说，变量的作用域包括定义它的程序以及该程序所调用的子程序范围。也就是说，在某个过程代码中定义的变量只能在该过程以及该过程所调用的过程中使用。

【例 3-2】在表单的 Init 事件代码中写下命令：

```
k = 0
THIS.Activate
```

在同一表单的 Activate 事件代码中写下命令：

```
MESSAGEBOX（"变量 k 的值为："+ALLT（STR（k））,48,"在过程 Activate 中"）
```

运行该表单，依次出现如图 3-2 所示的提示。

图3-2　变量的作用域

由于表单的 Init 事件先于 Activate 事件被激发，因此当表单开始运行，变量 k 在 Init 事件代码中被定义为 0，此值被该事件代码调用的过程如下：首先使用 Activate 事件代码中的函数 MESSAGEBOX（），如图 3-1 的左图所示；接下来 Activate 事件重新被激发，出现“程序错误”提示信息“找不到变量'k'”如图 3-1 的右图所示，说明 Init 事件过程中定义的变量不能在 Activate 事件过程中使用。

在 VFP 中，可以使用 LOCAL，PRIVATE 和 PUBLIC 命令强制规定变量的作用范围。

① 用 LOCAL 创建的变量只能在创建它们的过程中使用和修改，不能被更高层或更低层的过程访问，因此被称为局部变量。

② PRIVATE 创建的变量称为私有变量。它用于定义当前过程中的变量，并将以前过程中定义的同名变量保存起来，在当前过程中使用私有变量而不影响这些同名变量的原始值。系统默认定义的都属于私有变量。私有变量可以被当前过程及所调用的过程使用，如上述例子。

③ PUBLIC 用于定义全局变量。在本次 VFP 运行期间，所有过程及程序都可以使用这些全局变量。在命令窗口中定义的变量都属于全局变量。

在例 3-2 中，如果我们修改表单的 Init 事件代码为：

```
PUBLIC k
k=0
THIS.Activate
```

则在表单的运行中不会出现错误，将会出现两次图 3-2 左图中的提示框。

说明：关于变量作用域的详细讨论，将在第 7 章中给出。为了避免使用全局变量，可以使用一种特殊的变量——属性。

5．变量的释放

当程序结束，或在程序的剩余部分不再使用某些变量，可以将这些变量从内存中释放掉。从内存中删除或释放变量的命令是：

```
RELEASE〈内存变量表〉
```

这里，〈内存变量表〉中的各个变量用逗号分隔。

还可以使用 CLEAR MEMORY 命令清除所有的内存变量。

6．变量的显示

显示内存变量的命令格式有两种：

```
LIST MEMORY [LIKE〈通配符〉][TO PRINTER [PROMPT]|[TO FILE〈文件名〉]
DISPLAY MEMORY [LIKE〈通配符〉][TO PRINTER][PROMPT]|[TO FILE〈文件名〉]
```

说明：

① 使用 LIKE **可以筛选出需要的变量，缺省该选项，系统默认为全体变量。**

② 通配符包括“ * **”和“** ? **”。** *__代表多个字符，__?__代表一个字符，如__*__、__A*__、__?__、__?B?__分别代表所有变量、变量名以__ A __开头的变量、变量名是__ 1 __个字符的变量、变量名是__ 3 __个字符中间为__ B __的变量。**

③ 选项 TO PRINT **可将显示内容输出到打印机，**PROMPT **显示打印提示窗口；选项** TO FILE **〈文件名〉则将显示内容保存到文本文件（扩展名为**.TXT**）。**

3.3 表达式与运算符

VFP **表达式是指用运算符将常量、变量、字段、函数按** VFP **的语法规则连接起来的式子。作为特例，单个的常量、变量、字段和函数均为最简单的表达式。**

每个表达式都有确定的值，按照值的数据类型，把表达式分为算术表达式、字符表达式、日期表达式、逻辑表达式、关系表达式和名表达式。

运算符用来处理同种类型的数据。对于 VFP **的数据类型，有如下一些类型的运算符：算术运算符、字符运算符、日期时间运算符、逻辑运算符和关系运算符。**

3.3.1 算术运算符与算术表达式

算术表达式也称数值型表达式，由算术运算符、数值型常量、变量、函数和圆括号组成，其运算结果为一数值。例如，50 * 2 +（70 – 6）/ 8 **的运算结果为** 108.00**。**

算术表达式的格式为：

〈**数值** 1〉〈**算术运算符** 1〉〈**数值** 2〉[〈**算术运算符** 2〉〈**数值** 3〉…]

VFP **提供的算术运算符，见表** 3-1**。在这** 6 **个算术运算符中，除取负“**–**”是单目运算符外，其他均为双目运算符。加（**+**）、减（**–**）、乘（*****）、除（**/**）、取负（**–**）、乘方（**^**或****）运算的含义与数学中的概念基本相同。**

表 3-1　**算术运算符**

运　算　符	名　　称	说　　明
+	加	同数学中的加法
-	减	同数学中的减法
*	乘	同数学中的乘法
/	除	同数学中的除法
^ 或 **	乘方	同数学中的乘方，如 4^3 表示 4^3
%	求余	12 % 5 表示 12 除以 5 所得的余数

算术运算符的优先权依次为：

（ ） → ^或 ** → *和/ → % → +和–**

算术表达式与数学中的表达式写法有所区别，在书写表达式时应当特别注意：

① 每个符号占 1 **格，所有符号都必须一个一个并排写在同一横线上，不能在右上角或右**

下角写方次或下标。例如，2^3 要写成 2^3，x_1+x_2 要写成 x1+x2。

② 原来在数学表达式中省略的内容必须重新写上。例如，2x 要写成 2 * x。

③ 所有括号都用小括号（ ），括号必须配对。例如，3[x+2（y+z）]必须写成 3 *（x+2*（y+z））。

④ 要把数学表达式中的有些符号改成 VFP 中可以表示的符号。例如，要把 2πr 改为 2 * pi * r。

3.3.2 字符串运算符与字符串表达式

一个字符串表达式由字符串常量、字符串变量、字符串函数和字符串运算符组成。它可以是一个简单的字符串常量，也可以是若干个字符串常量或字符串变量的组合。字符串表达式的值为字符串。VFP 提供的字符运算符有两个（其运算级别相同），见表 3-2。

表 3-2 字符运算符

运 算 符	名 称	说 明
+	连接	将字符型数据进行连接
-	空格移位连接	两字符型数据连接时，将前一数据尾部的空格移到后面数据的尾部

字符串表达式的格式为：

〈字符串 1〉〈字符串运算符 1〉〈字符串 2〉[〈字符串运算符 2〉〈字符串 3〉…]

【例 3-3】下面是一些字符串表达式的示例。

```
"ABC123" + "666xyz"                  && 连接后结果为"ABC123666xyz"
"计算机" + "世界"                    && 连接后结果为"计算机世界"
"123   45" + "abcd   " + "   xyz   "  && 连接后结果为"123   45abcdxyz   "
"ABC   " – " DEFG"                   && 连接后结果为"ABCDEFG   "
```

使用?命令，可以得到表达式的计算结果，如图 3-3 所示。

图3-3 字符串表达式的连接

在字符串中嵌入引号，只需将字符串用另一种引号括起来即可。例如：

```
QM = ""
cString = cString + QM + ALLTRIM（THIS.Edit1.Value） + QM + ','
```

3.3.3 日期时间运算符与日期时间表达式

日期型表达式由算术运算符（“ + ”、“ – ”）、算术表达式、日期型常量、日期型变量和函

数组成。日期型数据是一种特殊的数值型数据，它们之间只能进行加“+”、减“-”运算，主要有以下三种情况：

（1）两个日期型数据相减

两个日期型数据可以相减，结果是一个数值型数据（两个日期相差的天数）。例如：

```
{^2002/12/19} - {^2002/11/16}          && 结果为数值型数据 33
```

（2）日期型数据加数值型数据

一个表示天数的数值型数据可加到日期型数据中，其结果仍然为一个日期型数据（向后推算日期）。例如：

```
{^2002/11/16} + 33                     && 结果为日期型数据{^2002/12/19}
```

（3）日期型数据减数值型数据

一个表示天数的数值型数据可从日期型数据中减掉它，其结果仍然为一日期型数据（向前推算日期）。例如：

```
{^2002/12/19} - 33          && 结果为日期型数据{^2002/11/16}
```

VFP 将无效的日期处理成空日期。

3.3.4 关系运算符与关系表达式

关系表达式是指用关系运算符将两个表达式连接起来的式子（例如 $a + b > 0$），关系运算符又称比较运算符，用来对两个表达式的值进行比较，比较的结果是一个逻辑值（.T. 或 .F.），这个结果就是关系表达式的值。

VFP 提供的关系运算符有 8 种，见表 3-3。

表 3-3 关系运算符

运 算 符	名 称	例 子	说 明
<	小于	3 < 4	值为 .T.
<=	小于或等于	4 <= 3	值为 .F.
>	大于	0 > 1	值为 .F.
>=	大于或等于	"aa" >= "ab"	值为 .F.
=	等于		
<>、#、!=	不等于		
$	包含于	"Fox" $ "FoxPro"	值为 .T.
==	等同于		

说明：

① 关系运算符两边值或表达式的类型应一致。

② 数学不等式 $a \leq x \leq b$，在 VFP 中不能写成 $a <= x <= b$。

③ 字符型数据按其 ASCII 码值进行比较。在比较两个字符串时，首先比较两个字符串的第一个字符，其中 ASCII 码值较大的字符所在的字符串大。如果第一个字符相同，则比较第二个，…，依此类推。

④ == 表示“等同于”，用于精确匹配。例如，使用条件 UPPER（NAME） = "SMITH" 进行查找，将找出 SMITHSON、SMITHERS 和 SMITH 的记录，而用==（等同于）可得到精确匹配 SMITH 的记录。

3.3.5 逻辑运算符与逻辑表达式

对于较为复杂的条件，必须使用逻辑表达式。逻辑表达式是指用逻辑运算符连接若干关系表达式或逻辑值而成的式子。如不等式 $a \leq x \leq b$ 可以表示为 $a <= x$ AND $x <= b$。逻辑表达式的值也是一个逻辑值。VFP 提供的逻辑运算符有以下 3 种，见表 3-4。

表 3-4 逻辑运算符

运 算 符	名 称	例 子	说 明
AND	与	（4 > 5） AND （3 < 4）	值为.F.，两个表达式的值均为真，结果才为真，否则为假
OR	或	（4 > 5） OR （3 < 4）	值为.T.，两个表达式中只要有一个值为真，结果就为真，只有两个表达式的值均为假，结果才为假
NOT	非	NOT （1 > 0）	值为.F.，由真变假或由假变真，进行取“反”操作

逻辑运算的运算规则，见表 3-5。

表 3-5 逻辑运算真值表

a	*b*	*a* AND *b*	*a* OR *b*	NOT *a*
.T.	.T.	.T.	.T.	.F.
.T.	.F.	.F.	.T.	.F.
.F.	.T.	.F.	.T.	.T.
.F.	.F.	.F.	.F.	.T.

说明：

① 运算符的优先顺序：在一个表达式中进行多种操作时，VFP 会按一定的顺序进行求值，称这个顺序为运算符的优先顺序。运算符的优先顺序，见表 3-6。表中同级运算按照它们从左到右出现的顺序进行计算。可以用括号改变优先顺序，强令表达式的某些部分优先运行。括号内的运算总是优先于括号外的运算，在括号之内，运算符的优先顺序不变。

表 3-6 运算符的优先顺序

优 先 顺 序	运算符类型	运 算 符
1	算术运算符	^（指数运算）
2		-（负数）
3		*、/（乘法和除法）
4		%（求模运算）
5		+、-（加法和减法）
6	字符串运算符	+、-（字符串连接）
7	关系运算符	=、<>、<、>、<=、>=、$、==
8	逻辑运算符	NOT
9		AND
10		OR

② 在早期的版本中，逻辑运算符的两边必须使用点号，如.AND.、.OR.、.NOT.，在 VFP 中，两者可以通用。

【例 3-4】设变量 x = 3，y = −2，a = 6.5，b = −7.2，求表达式

x + y > a + b AND NOT y < b

的值。

解：① 先作算术运算：　1　>　−0.7 AND NOT y < b

② 再作关系运算：　.T.AND NOT .F.

③ 作逻辑非运算：　.T.AND　.T.

④ 最后得：　.T.

3.3.6 类与对象运算符

类与对象运算符专门用于实现面向对象的程序设计。有以下两种：

. —— 点运算符，确定对象与类的关系，以及属性、事件和方法与其对象的从属关系。

:: —— 作用域运算符，用于在子类中调用父类的方法。

3.3.7 名表达式

在 VFP 中，许多命令和函数需要提供一个名。可在 VFP 中使用的名有：表/.DBF 的文件名、表/.DBF 的别名、表/.DBF 的字段名、索引文件名、文件名、内存变量和数组名、窗口名、菜单名、表单名、对象名、属性名等。

在 VFP 中定义一个名时，需要遵循以下原则：

① 名中只能使用字母（汉字）或下划线开始；

② 名中只能使用字母（汉字）、数字和下划线字符；

③ 不能使用 VFP 的保留字；

④ 名的长度可以为 1～128 个字符，但自由表中的字段名、索引标记名最多为 10 个字符。文件名按操作系统的规定。

名不是变量或字段，但是可以定义一个名表达式，以代替同名的变量或字段的值。名表达式为 VFP 的命令和函数提供了灵活性。将名存放到变量或数组元素中，就可以在命令或函数中用变量来代替该名，只要将存放一个名的变量或数组元素用一对括号括起来。如：

```
STORE "CITY" TO a
REPLACE （a） WITH "Beijing"
```

字段名 CITY 被存放在变量 *a* 中，在使用 REPLACE 命令时，名表达式（*a*）将用字段名代替变量，这种方法称为间接引用。

3.4 函数

对于用户来说，程序设计语言中的函数与数学上的函数没有什么区别，使用函数要有参数（自变量），可以从函数得到一个返回的值（因变量）。而从程序设计的角度来看，函数是子程序的一种，它能完成一种特定的运算。

3.4.1 函数的分类

VFP 的函数有两种，一种是用户自定义的函数，一种是系统函数。自定义函数由用户根据需要自行编写，系统函数则是由 VFP 提供的内部函数，用户可以随时调用。

VFP 提供的系统函数大约有 380 多个，主要分为数值函数、字符处理函数、表和数据库函数、日期时间函数、类型转换函数、测试函数、菜单函数、窗口函数、数组函数、SQL 查询函数、位运算函数、对象特征函数、文件管理函数以及系统调用函数等 14 类。通过查阅“帮助”中的“语言参考”可以了解到函数参数的类型、函数返回值的类型以及函数的使用方法。

3.4.2 常用函数

VFP 提供了大量的系统函数供编程人员使用，下面列出常用的一些函数。

1.数学函数

常用的数学函数，见表 3-7。

表 3-7 常用数值函数

函 数 名	功 能	例子与结果	
ABS（<N>）	N 的绝对值	ABS（3），ABS（−7.8）	3，7.8
SQRT（<N>）	N 的平方根	SQRT（2）	1.41
EXP（<N>）	e^N 的值	EXP（1），EXP（−2）	2.72，0.14
INT（<N>）	N 的整数部分	INT（3.6），INT（−2.14）	3，−2
LOG（<N>）	N 的自然对数	LOG（10），LOG（2.7183）	2.30，1.0000
LOG10（<N>）	N 的常用对数	LOG10（10），LOG10（2.7183）	1.00，0.4343
COS（<N>），SIN（<N>）	N 弧度的余弦、正弦值	COS（90/180*PI（））， SIN（90/180*PI（））	0.00，1.00
ACOS（<N>），ASIN（<N>）	N 的反余弦、反正弦值	ACOS（1）/PI（）*180，ASIN（1）/PI（）*180	0.0000，90.0000
TAN （<N>）	N 弧度的正切值	TAN（45/180*PI（））	1.00
ATAN （<N>）	N 的反正切值	ATAN（1）/PI（）*180	45.0000
MAX（<*1>,<*2> [,…]）	系列值较大者	MAX（2,3, −5,0），MAX（"A","C","B"）	3，C
MIN（<*1>,<*2> [,…]）	系列值较小者	MIN（"男","女"），MIN（{^2002-01-24,},{2001-10-01}）	男，10/01/01
SING（<N>）	N 的符号	SIGN（−3），SIGN（0），SIGN（6.53）	−1，0，1
PI（）	圆周率	PI（）	3.14
FLOOR（<N>）	不大于 N 的最大整数	FLOOR（−3.45），FLOOR（0.7），FLOOR（2.8）	−4，0，2
CEILING（<N>）	不小于 N 的最小整数	CEILING（−3.45），CEILING（0.7），CEILING（2.8）	−3，1，3
MOD（<N1>,<N2>）	N1 和 N2 相除后的余数	MOD（5,3），MOD（−10,3）	2，2
ROUND（<N1>,<N2>）	N1 保留 N2 位小数	ROUND（12.647,2），ROUND（12.647, −1）	12.65，10
RAND()	（0，1）的随机数	RAND()	

说明：

① MAX 和 MIN 函数中的参数可以是同种类型的多个参数。

② ROUND 函数按四舍五入保留指定位数的小数。

③ MOD 函数如果被除数与除数的符号相同时返回值是两数相除的余数 如果符号不同，

返回值是相除的余数加上除数，符号与除数相同。

2．字符串函数

常用的字符串函数，见表 3-8。

表 3-8　常用字符函数

函　数　名	功　　能	例子与结果	
SUBSTR（<C1>,<N1>[,<N2>]）	取 C1 中从 N1 开始长度为 N2 的子串，省略 N2 取到最后	SUBSTR（"ABC",2,1）	B
LEFT（<C>,<N>）	从 C 左取长度为 N 的子串	LEFT（"ABC",2）	AB
RIGHT（<C>,<N>）	从 C 右取长度为 N 的子串	RIGHT（"ABC",2）	BC
LEN（<C>）	求 C 的长度	LEN（"ABC"）,LEN（"函数"）	3,4
AT（<C1>,<C2>[,<N>]）	从 N 位置开始求 C2 在 C1 中第一次出现的位置，省略 N 则从 C1 开始起	AT（"ABC","B"）,AT（"ABAB","B",3）	2,4
ATC（<C1>,<C2>[,<N>]）	与 AT 函数类似，但不区分大小写	ATC（"ABC",b"）, AT （"ABC",b"）	2,0
LTRIM（<C>）	删除 C 的左端空格	"ab"+LTRIM（"??cd"）	abcd
RTRIM（<C>）	删除 C 的右端空格	RTRIM （"ab??"）+ "cd"	abcd
ALLTRIM（<C>）	删除 C 的左、右两端空格	[a]+ALLTRIM（"?b?"）+[c]	abc
SPACE（<N>）	生成 N 个空格	'a'+SPACE（2）+'b'	a??b
UPPER（<C>）	把 C 转换成大写	UPPER（"aBc"）	ABC
LOWER（<C>）	把 C 转换成小写	LOWER（"aBc"）	abc
OCCURE （<C1>,<C2>）	C1 在 C2 中出现的次数	OCCURE （"abcabcd","c"）	2
STUFF （<C1>,<N1>,<N2>,<C2>）	从 C1 的 N1 开始删除 N2 个字符后插入 C2	STUFF（"aBc",2,1,"b"）	abc
CHRTRAIN （<C1>,<C2>,<C3>）	以 C3 替换在 C1 中出现的 C2	CHRTRAIN（"ABA","A","C"）	CBC
LIKE（<C1>,<C2>）	比较 C1 与 C2 是否匹配，如果匹配返回.T.，否则返回.F.，C1 可以使用通配符＊或？	LIKE （ "AB*","abc" ） ,LIKE （"Abc","abc"）	.T.,.F.

注：?表示空格。

3．日期函数

常用的日期函数，见表 3-9。

表 3-9　日期函数

函 数 格 式	说　　明	例子与结果	
DATE（）	系统当前日期	DATE（）	
TIME（）	系统当前时间	TIME（）	
DATETIME（）	系统当前日期和时间	DATETIME（）	
DOW（表达式）	取日期表达式的星期号（1 为星期天）	DOW（{^2002-11-09}）	7
YEAR（表达式）	取日期表达式的年份值	YEAR（{^2002-11-09}）	2002
MONTH（表达式）	取日期表达式的月份值	MONTH（{^2002-11-09 08:25:37 P}）	11
DAY（表达式）	取日期表达式在月份中的天数值	DAY（{^2002-11-09 }）	9
HOUR（表达式）	取时间表达式中的小时数	HOUR（{^2002-11-09 08:25:37 P}）	20
MINUTE（表达式）	取时间表达式中的分钟数	MINUTE（{^2002-11-09 08:25:37 P}）	25
SEC（表达式）	取时间表达式中的秒数	SEC（{^2002-11-09 08:25:37 P}）	37

4．类型转换函数

常用的类型转换函数，见表 3-10。

表 3-10　类型转换函数

函　数　名	功　　能	例子与结果	
VAL（<C>）	C 型数据转换成 N 型数据	VAL（"23.7"），VAL（"2+3"）	23.7，2
STR（<N1>[,<N2>[,<N3>]]）	N1 转换成小数位数为 N3 长度为 N2 的 C 型数据	"π="+STR（3.141593,7,3）	π=?? 3.142
CHR（<N>）	ASCII 码值为 N 对应的字符	CHR（65），CHR（98）	A，b
ASC（<C>）	C 第一个字符的 ASCII 码值	ASC（"bcd"），ASC（"0"）	98，46
CTOD（<C>）	C 型数据转换成 D 型数据	CTOD（"2002/10/12"）	2002/10/12
CTOT（<C>）	C 型数据转换成 T 型数据	CTOT（"11/7/02,13"）	11/07/02 01:00:00 PM
DTOC（D[,1]）	D 型数据转换成 C 型数据	DTOC（{^2002-11-27}），DTOC（{^2002-11-27},1）	11/27/02，20021127
TTOC（T[,1]）	T 型数据转换成 C 型数据	TTOC（{^2002-11-27,8:23}）	11/27/0208:23:00 AM

说明：

① VAL 函数转换字符时，仅转换符合数值常量格式的部分。

② STR 函数的 N2 包括小数点和负号。如果 N3 过大，则首先保证整数部分，再考虑小数部分。

③ DTOC、TTOC 函数中的 1 参数用来强制要求转换结果为“ yyyymmdd ”型式。

5．测试函数

常用的测试函数，见表 3-11。

表 3-11　测试函数

函　数　名	功　　能
BETWEEN（表达式 1，表达式 2，表达式 3）	判断表达式 1 的值是否大于等于表达式 2 的值并且小于等于表达式 3 的值
ISNULL（表达式）	判断表达式的运算结果是否等于 NULL
EMPTY（表达式）	判断表达式的运算结果是否为“ 空 ”
VARTYPE（表达式）	以一个大写字母的形式返回表达式的类型
EOF（工作区号 | 表别名）	测试指定表文件中的记录指针是否指向文件尾
BOF（工作区号 | 表别名）	测试指定表文件中的记录指针是否指向文件首
RECNO（工作区号 | 表别名）	返回指定表文件中当前记录的记录号
RECCOUNT（工作区号 | 表别名）	返回指定表文件中的记录个数
DELETED（工作区号 | 表别名）	测试指定表文件中当前记录是否有删除标记

3.5　习题

一、选择题

1．将内存变量定义为全局变量的 VFP 命令是（　　）。

A）LOCAL　　B）PRIVATE　　C）PUBLIC　　D）GLOBAL

2．下列函数中函数值为字符型的是（　　）。

A）DATE（）　　B）TIME（）　　C）YEAR（）　　D）DATETIME（）

3．在下面的数据类型中默认值为.F.的是（　　）。

A）数值型　　B）字符型　　C）逻辑型　　D）日期型

4．设有下列赋值语句（　　）。

```
X = {^2003-10-01 08:00:00 AM}
Y = .F.
Z = 123.45
M = $123.45
N = '123.45'
```

依次执行上述命令后，内存变量 X、Y、Z、M、N 的数据类型分别是（　　）。

A）D、L、N、Y、C　　B）D、L、N、M、C

C）T、L、N、M、C　　D）T、L、N、Y、C

5．下列日期型常量中，正确表示的是（　　）。

A）{"2003-01-01"}-10　　B）{^2003-01-01}

C）{2003-01-01}　　D）{[2003-01-01]}

6．下列表达式中，不正确表示的是（　　）。

A）{^2003-01-01 10:10:10 AM}-7　　B）{^2003-01-01} – DATE（）

C）{^2003-01-01} + DATE（）　　D）[^2003-01-01] + 1000

7．下列表达式值为逻辑真的是（　　）。

A）EMPTY（.NULL）　　B）LIKE（'acd','ac?'）

C）AT（'a','123abc'）　　D）EMPTY（SPACE（2））

8．设 C = 5 < 6，VARTYPE（C）的值是（　　）。

A）L　　B）C　　C）N　　D）D

9．数学式子 sin25° 写成 VFP 表达式是（　　）。

A）SIN25　　B）SIN（25）　　C）SIN（25°）　　D）SIN（25*PI（）/180）

10．如果 x 是一个正实数，对 x 的第 3 位小数四舍五入的表达式是（　　）。

A）0.01 * INT（x + 0.005）　　B）0.01 * INT（100 *　（x + 0.005））

C）0.01 * INT（100 *（x + 0.05））　　C）0.01 * INT（x + 0.05）

11．下列哪个符号不能作为 VFP 中的变量名（　　）？

A）ABCDEFG　　B）P000000　　C）89TWDDFF　　D）xyz

12．下列符号哪一个是 VFP 中的合法变量名（　　）？

A）AB7　　B）7AB　　C）IF　　D）A[B]7

13．下列函数值为数值的是（　　）。

A）BOF　　B）CTOD（"01/02/03"）

C）AT（"计算机","全国计算机等级考试"）　　D）SUBSTR（DTOC（DATE（）,7））

二、填空题

1．LEFT（"123456789",LEN（"数据库"））的计算结果是_____。

2．表达式 VAL（SUBS（"奔腾 586",5,1））* LEN（"Visual FoxPro"）的值是_____。

3．ROUND（123.4567,3）的值是_____。

4．$\sqrt{s(s-a)(s-b)(s-c)}$ 的 VFP 表达式是_____。

5．VFP 表达式 a /（b + c /（d + e / SQRT（f））的数学表达式是_____。

6.设 A = 7,B = 3,C = 4,则表达式 A % 3 +B^3 / C 和 A / 2 * 3 / 2 的值分别是_____和_____。

7．在没有特别声明的情况下，VFP 6.0 程序中变量的作用域是_____。

第 4 章　数据表与数据库

本章介绍 VFP 数据库的基本操作，包括建立和使用表来存储数据、建立和管理由表组成的数据库以及使用索引和数据完整性等方面的内容。

4.1　创建新表

表是处理数据和建立关系型数据库及应用程序的基本单元。表的操作包括：创建新表、处理当前存储在表中的信息、定制已有的表。可以使用索引对记录排序及快速处理。

4.1.1　表的结构

1. 表与关系

表 4-1 是一个描述学生基本情况的“二维”表格，是一个典型的“关系”。在 VFP 中，表中的每一列都是一个字段，第一行中的每一项是相应字段的字段名；第一行以下的每一行都是一条记录；表格的表头可以看作 VFP 的表名。

表 4-1　学生情况表

学　号	姓　名	性　别	出生日期	专　业	说　明	照　片
1999220203	李富强	男	08/03/1980	计算机应用		
1999220212	冯见岳	男	06/18/1979	计算机应用		
1999160208	罗海燕	女	12/06/1980	国际贸易		
1999180105	张丽萍	女	01/08/1980	会计		
1999180102	刘刚	男	07/13/1980	会计		
1999160211	赵江山	男	05/16/1981	国际贸易		
1999160221	许海霞	女	02/12/1979	国际贸易		
1999220115	王春雷	男	10/20/1979	计算机应用		
1999220207	李家富	男	03/13/1980	计算机应用		
1999220215	张仙见	女	09/21/1979	计算机应用		
…	…	…	…	…	…	…

VFP 建立表结构的过程和我们建立表格的过程类似：

① 首先建立一个表格名（表名）。

② 然后建立栏目（字段名、字段类型）。

③ 设计表格宽度（字段长度）。

所不同的是表格需要事先知道需要多少行（记录），而 VFP 的表不需要事先知道，它可以根据记录数自动调整。

2．表的结构设计

表的结构设计包括定义表名以及定义字段的属性。字段的属性指：字段名、字段类型、字段长度（数值型还需要定义小数位数）。

3．表名

表在计算机内以文件形式出现，其类型为“.DBF”，表名按文件名要求。

4．字段名

字段名即表的栏目名或关系的属性名，可以用来引用该列数据。要求：

① 汉字或字母开头，由汉字、字母、数字和下划线组成。

② 自由表字段名长度不能超过 10 个字符，数据库表字段名最长可使用 128 个字符，如“学号”、“Name”、“工资 1”、“X_1”等是符合要求的，而“说 明”、“1-1”等不能作为字段名使用。

③ 不能使用保留字。

④ 同一表中字段名不允许重复。

⑤ 字段名最好能简要说明该字段的意义，如“姓名栏”可以用“姓名”、“name”作为字段名。

5．字段类型和宽度

VFP 的字段类型和宽度见表 4-2。

表 4-2　字段类型和宽度

类　型	代　号	说　明	字 段 宽 度
字符型	C	汉字或字符	最多 254 个字符，汉字占 2 个字符
数值型	N	整数或小数	最多 20 位，小数点和正负号各占一位
货币型	Y	保留 4 位小数	8 个字节
日期型	D	格式为 MM/DD/YY	8 个字节
日期时间型	T	日期和时间	8 个字节
逻辑型	L	逻辑值“真”或“假”	1 个字节
浮点型	F	整数或小数，同数值型	
整型	I	存放整数	4 个字节
双精度型	B	存放精度较高的数值	8 个字节
备注型	M	接收字符型数据，存放在文件名与表明相同的“.DPT”文件中	4 个字节
通用型	G	存放图形、声音等 OLE 对象（对象链接与嵌入），与备注型存放位置相同	4 个字节
字符型（二进制）		同前述“字符型”，但是当代码页更改时字符值不变	同“字符型”
备注型（二进制）		同前述“备注型”，但是当代码页更改时备注不变	同“备注型”

说明：

① 字符型、数值型、浮动型三种字段根据数据的实际需要设定合适的宽度，备注型和通用型数据与其他数据并不存放在一起，而是存放在与表同名的.FTP 文件中。字符型字段的宽度应按数据最大宽度计。

② 虽然有些数据由数字构成，但它们却可以定义成字符型，如电话号码、邮政编码、身份证号码等。

③ 凡是只有两种状态的数据可以定义成逻辑型，如男女、是否团员、是否及格等。

由此可以设计表格 4-1 的结构，见表 4-3。

表 4-3　学生情况表结构

字　段　名	字 段 类 型	宽　　度
学号	字符型	10
姓名	字符型	6
性别	逻辑型	1
出生日期	日期型	8
专业	字符型	20
说明	备注型	4
照片	通用型	4

表名为：student

上述结构可以表示为：学生（学号（C,10**），姓名（**C,6**），性别（**L**），出生日期（**D**），专业（**C,20**），说明（**M**），照片（**G**））。**

4.1.2　使用表设计器

利用“ 表向导 ”，可以创建新表，但是，在表向导中包含的内建表中的字段几乎都不符合国情，所以“ 表向导 ”实用价值不大，一般使用“ 表设计器 ”。

1．创建新表

创建新表的操作步骤为：

① 从“ 文件 ”菜单中选择“ 新建 ”命令，或者单击常用工具栏中的“ 新建 ”按钮，打开“ 新建 ”对话框，如图 4-1 **所示。**

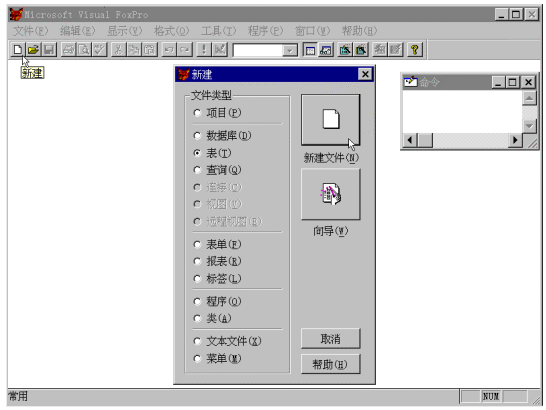


图4-1　“ 新建 ”对话框

② 在“ 新建 ”对话框中选择“ 表 ”，然后单击“ 新建文件 ”。

③ 在“ 创建 ”对话框中，输入表的名称并单击“ 保存 ”按钮，如图 4-2 **所示。**

④ 在“ 表设计器 ”中，选择“ 字段 ”选项卡，在“ 字段名 ”区域键入字段的名称；在“ 类型 ”列中，选择列表中的某一字段类型；在“ 宽度 ”列中，设置以字符为单位的列宽；如果“ 字段类型 ”是“ 数值型 ”或“ 浮点型 ”，还要设置“ 小数位 ”框中的小数点位数。

如果希望为字段添加索引，则在“ 索引 ”列中选择一种排序方式，如果想让字段能接受

null 值，选中“ NULL ”。

一个字段定义完后，单击下一个字段名处，输入另一组字段定义，一直把所有字段都定义完，如图 4-3 所示。

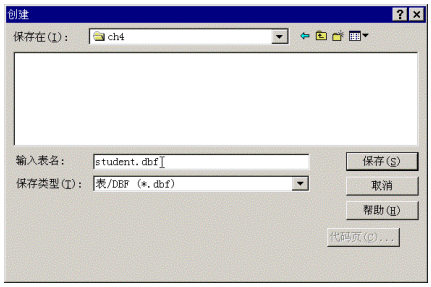


图4-2 “ 创建 ”表对话框

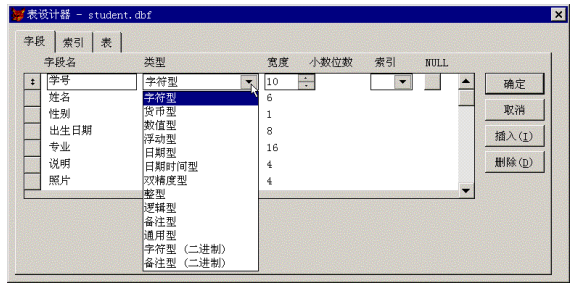


图4-3 “ 表设计器 ”对话框

如果单击“ 插入 ”按钮，则在已选定字段前插入一个新字段。

如果单击“ 删除 ”按钮：则从表中删除选定字段。

当鼠标指针指向字段名左端的方块时，将变为上下双向箭头，拖动上下箭头可以改变字段的顺序。注意，在输入过程中，不能按回车键，回车则表示整个“ 创建 ”过程的结束。定义好各个字段后单击“ 确定 ”按钮，就完成了表结构的定义工作。这时会出现一个确认对话框，显示“ 现在输入数据记录吗？ ”，若需要马上输入记录则选择“ 是 ”，不输入记录则选择“ 否 ”。

2．在表中添加记录

若要在表中添加记录：

① 从“ 文件 ”菜单中选择“ 打开 ”命令，或者单击常用工具栏上的“ 打开 ”按钮。

② 在“ 打开 ”对话框中，如图 4-4 所示，选择“ 文件类型 ”为“ 表（*.dbf）”，选择表所在的文件夹，找到表文件后，双击要打开的表。

③ 在“ 显示 ”菜单中，选择“ 浏览 ”命令，如图 4-5 所示，这时将显示打开的表。

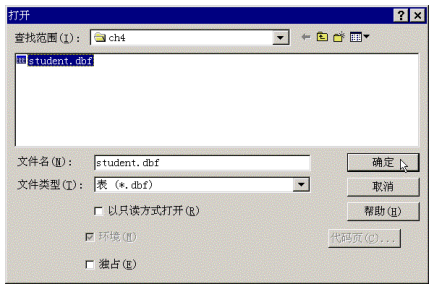


图4-4 “ 打开 ”对话框

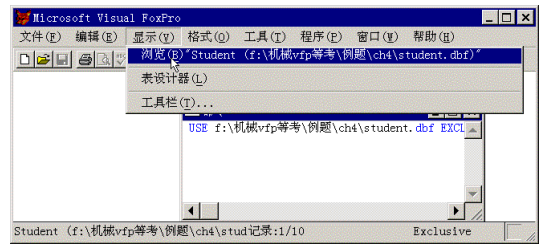


图4-5 “ 显示 ”菜单

④ 再次从“ 显示 ”菜单中选择“ 追加方式 ”，如图 4-6 所示。

⑤ 在“ 浏览 ”（图4-6左）或“ 编辑 ”（图4-6右）窗口中输入新的记录。

4.1.3 使用命令

在命令窗口中或是在代码中使用命令，是专业人员所习惯的，也更能发挥 VFP 的强大功能。

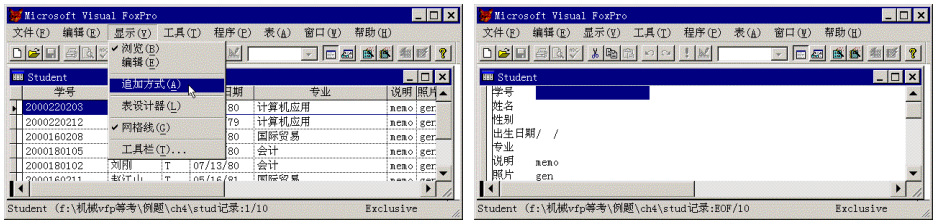


图4-6 “显示”菜单

1．创建新表

使用CREATE〈新表文件名〉命令也可以打开“表设计器”，创建一个新的表文件结构。使用CREATE ？命令则可以在其他目录中创建一个新的表文件结构。

使用下述命令可以不使用“表设计器”，直接创建表的结构：

CREATE TABLE 〈新表文件名〉(〈字段名 1〉〈类型〉(〈长度〉)
[,〈字段名 2〉〈类型〉(〈长度〉) …])

【例 4-1】在命令窗口输入以下命令：

CREATE TABLE Student1（学号 c（10），姓名 c（6），性别 l，出生日期 d（8），专业 c（20），说明 m，照片 g）

可以建立包含学号、姓名、性别、出生日期、专业、说明、照片等字段的一个新的数据表 Student1.dbf。

2．打开与关闭表

打开与关闭表都是使用 USE 命令，其格式为：

USE [〈表文件名〉]

说明：

① 使用参数〈表文件名〉可以打开一个已经存在的数据表（名称为〈表文件名〉）。

【例 4-2】在命令窗口输入以下命令：

USE Student

即可打开数据表 Student。

② 使用不带参数的 USE 命令可以关闭已打开的数据表。

3．添加记录

使用 APPEND 命令可以向打开的数据表中添加记录。在“显示”菜单中，可以选择“编辑”或“浏览”方式输入新的记录，如图 4-7 所示。

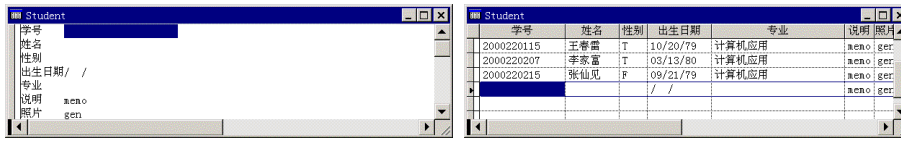


图4-7 在“编辑”或“浏览”窗口中添加新的记录

使用 APPEND BLANK 命令可以在打开的数据表中添加一个空白记录。

4．复制表

使用 COPY TO 命令可以将当前数据表中指定范围内所有符合条件的记录复制到新的表文件中，新文件结构仅包含指定的字段。其命令格式为：

COPY TO〈新文件名〉[〈范围〉] [{FOR | WHILE}〈条件〉] [FIELDS〈字段名表〉]

说明：

①〈新文件名〉可省略扩展名.DBF，〈条件〉为逻辑表达式。

② 若缺省 FOR|WHILE 〈条件〉，则〈范围〉默认为 ALL，即将所有记录复制到新表中。

③ 若缺省 FIELDS 〈字段名表〉，则新文件保留全部字段。

④ 若所有可选项都缺省，则原样复制表文件。

⑤ 若原文件带有备注型字段，则其相伴的备注文件（扩展名为.DBT）也同时自动被复制。

4.2 表的基本操作

表的基本操作包括查看记录、编辑记录、增加记录、删除记录以及记录指针的定位。

4.2.1 使用“浏览”窗口

使用“浏览”窗口可以完成所有对表中记录的基本操作。

1.“浏览”窗口

“浏览”窗口中显示的内容是由一系列可以滚动的行和列组成的。若要浏览一个表，可以从“文件”菜单中选择“打开”命令，选定想要查看的表名，然后从“显示”菜单中选择“浏览”，如图 4-8 所示。

为方便输入，可以把“浏览”窗口设置为“编辑”方式。在“编辑”方式下，列名显示在窗口的左边，如图 4-9 所示。若要将“浏览”窗口改为编辑方式，只需从“显示”菜单中选择“编辑”项即可。

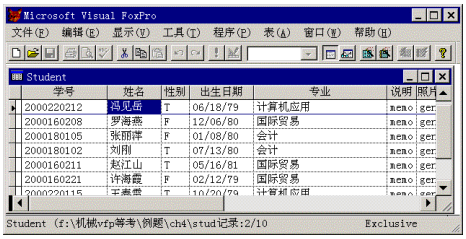


图4-8 浏览方式

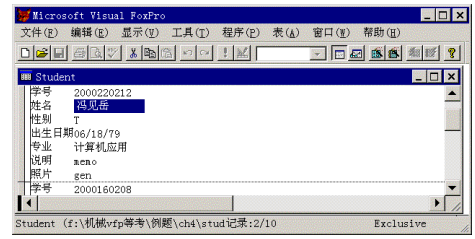


图4-9 编辑方式

2.“表”菜单

一旦打开“浏览”窗口，系统菜单中将增加一个“表”菜单，如图 4-10 所示。“表”菜单中包括所有对表的基本操作，选择相应的子菜单项，按照屏幕提示，即可完成各项操作。

3．转到记录

在任何一种方式下，使用滚动条可以来回移动表的记录指针，显示表中不同的字段和记录。也可以用箭头键和〈Tab〉键移动。若要查看不同的记录：

① 从“表”菜单中选择“转到记录”命令。

② 在子菜单中选择“第一个”、“最后一个”、“下一个”、“前一个”或“记录号”命令。

③ 如果选择了“记录号”命令，在“转到记录”对话框中输入待查看记录的编号，然后单击“确定”按钮。

图4-10 “表”菜单

4．编辑字段

若要改变“字符型”字段、“数值型”字段、“逻辑型”字段、“日期型”字段或“日期时间型”字段中的信息，可以把光标设在字段中并编辑信息，或者选定整个字段并键入新的信息。

若要编辑“备注型”字段，可在“浏览”窗口中双击该字段或按下〈Ctrl〉+〈PgDn〉键。这时会打开一个“编辑”窗口，其中显示了“备注型”字段的内容。

“通用型”字段包含一个嵌入或链接的 OLE 对象。通过双击“浏览”窗口中的“通用型”字段可以打开通用型窗口；从“编辑”菜单中选择“粘贴”、“选择性粘贴”或“插入对象”命令，可以编辑这个对象，如直接编辑文档（如 Microsoft Word 文档或 Microsoft Excel 工作表），或者双击对象打开其父类应用程序（如 Microsoft 画笔对象）。

5．添加新记录

若想在表中快速加入新记录，可以将“浏览”和“编辑”窗口设置为“追加方式”。在“追加方式”中，文件底部显示了一组空字段，可以在其中填入内容来建立新记录。设置为追加方式的方法为从“显示”菜单选择“追加方式”命令。

在新记录中填充字段，用〈Tab〉键可以在字段间进行切换。每完成一条记录，在文件的底端就会又出现一条新记录。

6．删除记录

若要从表中删除记录，单击记录左边的小方框，标记待删除的记录，如图 4-11 所示。

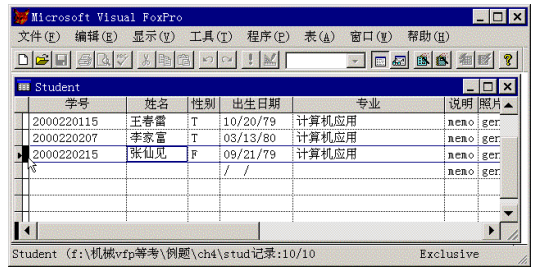


图4-11 标有删除标记的记录

标记记录并不等于删除记录，属于“逻辑删除”。要想真正地删除记录（物理删除），应从“表”菜单中选择“彻底删除”。当出现提示“是否从表中移去已删除的记录？”，单击“是”按钮即可。这个过程将删除所有标记过的记录，并重新构造表中余下的记录。删除记录时将关闭浏览窗口，若要继续工作，要重新打开浏览窗口。

若要有选择地删除一组记录，可从“表”菜单中选择“删除记录”，则打开“删除”对话框，输入删除条件，选择删除记录的范围，如图 4-12 所示。

如果待删除记录能够描述出来，可以建立一个描述表达式。选择 FOR 后面的“...”按钮，激活“表达式生成器”对话框，如图 4-13 所示。例如，使用表达式 YEAR（DATE（））–YEAR（出生日期）>=28 可选定年龄 28 岁以上的记录，并为它们加上删除标记。

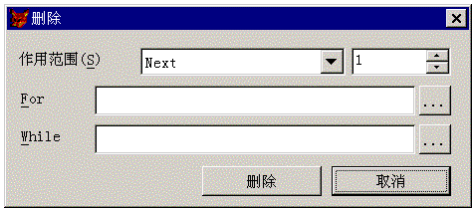


图4-12 “删除”对话框

图4-13 “表达式生成器”对话框

4.2.2 定制“浏览”窗口

要按照不同的需求定制“浏览”窗口，可以重新安排列的位置、改变列的宽度、显示或隐藏表格线或者把“浏览”窗口分为两个窗格。

1．重新安排列

可以重新安排“浏览”窗口中的列，使它们按照需要的顺序进行排列，但这并不影响表的实际结构。若要在“浏览”窗口中重新安排列，将列标头拖到新的位置。或者，从“表”菜单中选择“移动字段”，然后用上下箭头键移动列，最后按〈Enter〉键。

2．拆分“浏览”窗口

（1）拆分窗口

通过拆分“浏览”窗口，可以很方便地查看同一表中的两个不同区域，或者同时在“浏览”和“编辑”方式下查看同一记录。若要拆分“浏览”窗口，将鼠标指针指向窗口左下角的拆分条。向右方拖动拆分条，将“浏览”窗口分成两个窗格，如图 4-14 所示。或者，从“表”菜单中选择“调整分区大小”，用左右光标键移动拆分条，按〈Enter〉键结束。

图4-14 拆分“浏览”窗口

（2）调整拆分窗格的大小

将指针指向拆分条，向左或向右拖动拆分条，改变窗格的相对大小。或者，从“表”菜

单中选择“调整分区大小”命令，按左箭头或右箭头键移动拆分条，按〈Enter〉键结束。

默认情况下，“浏览”窗口的两个窗格是相互链接的，即在一个窗格中选择了不同的记录，这种选择会反映到另一个窗格中。取消“表”菜单中“链接分区”的选中状态，可以中断两个窗格之间的联系，使它们的功能相对独立。这时，滚动某一个窗格时，不会影响到另一个窗格中的显示内容。

3．改变显示时的列宽

在列标头中，将鼠标指针指向两个字段之间的结合点，拖动鼠标调整列的宽度，如图 4-15 所示。或者，先选定一个字段，然后从“表”菜单中选择“调整字段大小”，并用左、右箭头键来调整列宽，最后按〈Enter〉键。这种尺寸调整不会影响到字段的长度或表的结构。如果想改变字段的实际长度，应使用“表设计器”修改表的结构。

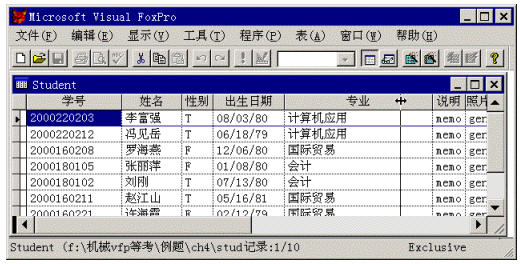


图4-15　改变显示时的列宽

4．打开或关闭网格线

也可以隐藏“浏览”窗口中的网格线。若要显示或隐藏网格线，从“显示”菜单中选择“网格线”命令即可。

4.2.3　使用命令

直接在命令窗口或者在程序中使用 VFP 命令，不仅可以完成对表中数据的各项基本操作，而且更加灵活。

1．打开浏览窗口

使用 BROWSE 命令可以打开浏览窗口，其格式为：

BROWSE [FIELDS 〈字段名表〉] [LOCK 〈表达式〉][FREEZE 〈字段名〉] ...

说明：BROWSE 命令的可选参数很多，下面只列出常用的几种。

① FIELDS〈字段名表〉：指定显示在浏览窗口中的字段。

② LOCK〈表达式〉：指定浏览窗口左分区中可见的字段个数。

③ FREEZE〈字段名〉：在浏览窗口中指定只修改一个字段（光标冻结），其他字段可显示但不能修改。

④ NOAPPEND：禁止在浏览窗口中通过交互方式增加记录。

⑤ NOMODIFY：禁止在浏览窗口中修改，只能浏览。

⑥ NOMENU：从系统菜单中删除“表”子菜单，防止调用该子菜单。

2．查看记录

还可以在 VFP 主窗口或用户自定义窗口（如表单）中显示当前表中的记录。显示命令有

两个，其格式分别是：

LIST [〈范围〉] [FIELDS〈字段名表〉] [{FOR | WHILE}〈条件〉] [OFF][TO PRINT]
DISPLAY [〈范围〉] [FIELDS〈字段名表〉] [{FOR | WHILE}〈条件〉] [OFF][TO PRINT]

说明:

① LIST 称为连续显示命令，〈范围〉项缺省时，系统默认为 ALL，其功能是连续显示满足条件的所有记录直到结束。

② DISPLAY 称为分页显示命令，若无 FOR|WHILE 短语，〈范围〉缺省时，系统默认为当前记录，若满足条件的记录较多，则显示满屏时暂停，待按任意键后继续显示后面的内容直到结束。

③ 若有 FIELDS 子句则显示指定字段内容，否则显示所有字段内容（不包括备注型字段）。

④ 若有 OFF 短语，不显示记录号;否则显示每个记录的记录号。

⑤ 若有 TO PRINT 短语，则显示内容送往屏幕的同时还从打印机输出。

3．编辑

使用命令 EDIT，可以打开“编辑”窗口，编辑已打开的数据表。其格式为：

EDIT [〈范围〉] [FIELDS〈字段名表〉] [{FOR | WHILE}〈条件〉]

说明：〈范围〉项缺省时，系统默认为 ALL，在满足条件的记录范围内可前后翻页。

4．记录定位

记录定位是指根据需要将记录指针移到指定记录，使之成为当前记录，以便对之操作。

文件的首记录又称为 TOP，尾记录称为 BOTTOM。用 USE 命令打开数据库时，指针总是指向第一条记录（TOP）。

可以在命令窗口或程序中使用命令来移动记录指针。移动记录指针的命令分为绝对移动（GO）和相对移动（SKIP）两种，其格式如下：

（1）绝对移动

绝对移动记录指针的命令格式为：

GO { BOTTOM | TOP |〈记录号〉}

其中 BOTTOM 表示末记录，TOP 表示首记录，〈记录号〉可以是数值表达式，按四舍五入取整数，但是必须保证其值为正数且位于有效的记录数范围之内。

（2）相对移动

相对移动记录指针的格式为：

SKIP {n | −n }

其中 n 为数值表达式，四舍五入取整数。若是正数，向记录号增加的方向移动，若是负数，向记录号减少的方向移动。

5．使用批替换命令

批替换命令 REPLACE 可对字段内容成批自动地进行修改（替换），而不必在编辑状态下逐条修改。批替换命令的语法格式为：

REPLACE [〈范围〉] 〈字段名 1〉 WITH 〈表达式 1〉
[,〈字段名 2〉 WITH 〈表达式 2〉...] [{FOR | WHILE} 〈条件〉]

说明：

① 若无〈范围〉项，则只对当前记录操作。

② 对指定范围内满足条件的各记录，以〈表达式 1〉的值替换〈字段名 1〉的内容，〈表达式 2〉的值替换〈字段名 2〉的内容… （备注型字段除外）。

【例 4-3】要将表 Student.dbf 的所有学号字段中的“1999”改为“2000”，可用下面的命令：

```
USE Student
REPLACE ALL 学号 WITH "2000" + RIGHT（学号,6）
```

6．在表中添加新记录

（1）使用 APPEND 命令

APPEND 命令可以在表的尾部添加新的记录，其格式为：

```
APPEND [BLANK]
```

说明：如果缺省 BLANK 选项，系统按“追加方式”打开“浏览”或“编辑”窗口，为数据表添加新记录。如果有 BLANK 选项，则在表的尾部添加一个空记录而不打开“浏览”或“编辑”窗口。

若要从其他表中追加记录，可以使用如下格式的命令：

```
APPEND FROM〈表文件名〉FOR〈逻辑表达式〉
```

（2）使用 INSERT 命令

INSERT 命令可以在表的任何位置插入一条新的记录，其格式为：

```
INSERT [BLANK] [BEFORE]
```

说明：若无 BLANK 选项，系统打开“浏览”或“编辑”窗口，在当前记录后（无 BEFORE）或在当前记录前（有 BEFORE）为数据表插入新记录；若有 BLANK 短语，则插入一条空记录而不打开“浏览”或“编辑”窗口。

7．删除记录

（1）逻辑删除记录

逻辑删除记录命令可以对数据表中指定范围内满足条件的记录加注删除标记，其格式为：

```
DELETE [〈范围〉] [FOR 〈条件〉]
```

说明：

① 本命令属逻辑删除命令，加注删除标记后记录仍能被操作（修改、复制、显示等）。

② 撤销删除标记命令可以恢复数据表中指定范围内满足条件的被加注删除标记的记录。

（2）恢复逻辑删除的记录

使用撤销标记命令，可以恢复逻辑删除的记录，其格式为：

```
RECALL [〈范围〉] [FOR 〈条件〉]
```

说明：RECALL **是** DELETE **的逆操作，作用是取消删除标记，恢复成正常记录。**

（3）物理删除加注删除标记的记录

可以将数据表中所有具有删除标记的记录正式从表文件中删掉。其格式为：

```
PACK
```

说明：PACK **为物理删除命令，一旦执行，无法用** RECALL **恢复。**

（4）直接删除所有记录

直接删除所有记录命令可以一次删除数据表中的全部记录，但保留表结构。其格式为：

```
ZAP
```

说明：本命令等价于 DELETE ALL **与** PACK **连用，但速度更快。属于物理删除命令，一旦执行，无法恢复。**

4.3 修改表结构

建立表之后，还可以修改表的结构和属性。例如可能要添加或删除字段，更改字段的名称、宽度、数据类型，改变默认值或规则或添加注释、标题等。

可以打开“表设计器”修改表的结构，也可以使用 ALTER TABLE **命令以编程方式来更改表的结构。当然，在修改表结构前，必须独占地访问该表。**

4.3.1 使用表设计器

利用“表设计器”，可以改变已有表的结构，如增加或删除字段、设置字段的数据类型及宽度、查看表的内容以及设置索引来排序表的内容。如果正在进行修改的表是数据库的一部分，那么还可以得到附加的与数据库有关的字段和表的属性。

1．打开“表设计器”

在“文件”菜单中选择“打开”命令，选定要打开的表。然后从“显示”菜单中选择“表设计器”命令，则打开“表设计器”对话框。表的结构将显示在“表设计器”中。

在命令窗口可以使用如下命令打开表设计器：

```
MODIFY STRUCTURE
```

2．修改表的结构

若要在表中增加字段，在“表设计器”中单击“插入”按钮。在“字段名”列中，键入新的字段名。在“类型”列中，选择字段的数据类型。在“宽度”列中，设置或输入字段宽度。如果使用的数据类型为“数值型”或“浮点型”，还需要设置“小数位”列的小数位数。如果想让表接受“null**”值，请选中“**NULL**”列。**

若要删除表中的字段，选定该字段，并单击“删除”按钮。

改变字段的数据类型及宽度的操作：选定该字段，在“类型”列中，选择字段的数据类型。在宽度列中，设置或输入字段宽度。

最后，单击“确定”按钮，在弹出的对话框中选择“是”，改变表的结构。

4.3.2 以编程方式修改表结构

可以在程序中使用 ALTER TABLE 命令直接修改数据表的结构。ALTER TABLE 命令提供了扩展子句，能够添加或去掉字段，创建或去掉主关键字、惟一关键字或外部关键字标识，并重新命名已有的字段。有些子句仅适用于与数据库相关联的表。

ALTER TABLE 命令的语法格式较长，第 6 章中将给出完整的格式。下面仅给出几个简单的例子。

1．给表添加字段

使用 ALTER TABLE 命令的 ADD [COLUMN] 子句。例如，可以使用以下命令把备注型字段“地址”添加到 Student 表中，并允许该字段有 null 值：

```
ALTER TABLE Student ADD COLUMN 地址 m NULL
```

2．重新命名表字段

使用 ALTER TABLE 命令的 RENAME COLUMN 子句。例如，可以使用以下命令对 Student 表的“地址”字段重新命名：

```
ALTER TABLE Student RENAME COLUMN 地址 TO 家庭住址
```

3．从表中删除字段

使用 ALTER TABLE 命令的 DROP [COLUMN]子句。例如，可以使用以下命令从 Student 表中删掉“家庭住址”字段：

```
ALTER TABLE Student DROP COLUMN 家庭住址
```

4.4 定制表

可以在表中设置一个过滤器来定制自己的表，有选择地显示某些记录。还可以通过设置字段过滤器，对表中的某些字段的访问进行限制，这样可以选择显示哪些字段。

4.4.1 筛选表

如果只想查看某一类型的记录，可以通过设置过滤器对“浏览”窗口中显示的记录进行限制。在某些情况下，例如只想查看销售额高于某一数值的商品，或者在某段时间内雇用的职员，筛选就显得非常有用。

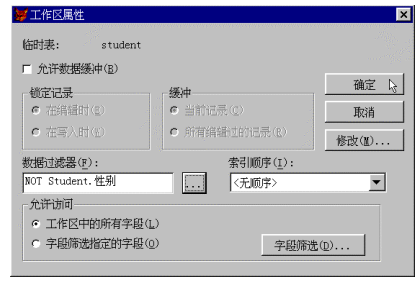


图4-16 “工作区属性”对话框

1．使用“浏览”窗口

若要在表中建立过滤器，首先打开表，然后浏览要筛选的表。从“表”菜单中选择“属性”命令。在“工作区属性”对话框中的“数据过滤器”框内输入筛选表达式，如图 4-16 所示。或者，选择“数据过滤器”框后面的对话按钮，在“表达式生成器”中创建一个表达式来选择要查看的记录，然后单击“确定”按钮。浏览表时则只显示经筛选表达式筛选过的

记录。例如，在“数据筛选”对话框中输入显示所有女同学记录的表达式。

2．使用命令

可用 SET FILTER 命令筛选数据。若要指定一个暂时的条件，使表中只有满足该条件的记录才能访问时，这个命令特别有用。SET FILTER 命令的语法格式为：

SET FILTER TO [〈逻辑表达式〉]

【例 4-4】只显示所有女同学记录的筛选命令为：

SET FILTER TO NOT Student.性别

如果要关闭当前表的筛选条件，可以执行不带表达式的 SET FILTER TO 命令。

4.4.2 限制对字段的访问

在表单中浏览或使用表时，若想只显示某些字段，可以设置字段筛选来限制对某些字段的访问。选出要显示的字段后，剩下的字段就不可访问了。

1．使用“浏览”窗口

从“表”菜单中选择“属性”命令，在“工作区属性”对话框的“允许访问”框内，选中“字段筛选指定的字段”，然后单击“字段筛选”按钮，打开“字段选择器”对话框。

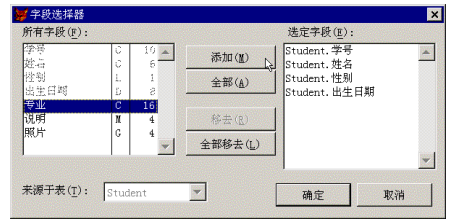


图4-17 “字段筛选器”对话框

在“字段选择器”对话框中，将所需字段移入“选定字段”栏，如图 4-17 所示。然后单击“确定”按钮。浏览表时，只有在字段筛选中的“选定字段”才被显示出来。

2．使用命令

可以使用如下格式的命令限制对字段的访问：

SET FIELDS TO {ALL | 〈字段名表〉}

其中〈字段名表〉是希望访问的字段名称列表，各字段之间用“,”分开。ALL 选项将取消所有的限制，而显示所有的字段。

4.5 数据表的索引

建立索引的最直接的理由是为了排序。在建立数据表时，记录一般是随机输入，其排列顺序无规律。如果要按照某些字段值的顺序排列，就要对数据表进行排序操作或者建立索引。

4.5.1 基本概念

1．索引与索引表达式

数据表的索引是按指定的索引表达式对数据表建立的一个文件——索引文件。索引文件是一个记录号的列表，它指向待处理的记录，并确定了记录的处理顺序，即按新顺序存储着数据表所对应的记录号。

索引表达式可以是表中的字段或字段的组合，前者又称为索引字段。

索引并不改变表中所存储数据的顺序，它只改变了 VFP 读取每条记录的顺序。

可以利用索引对数据表中的数据进行排序，以便加速检索数据的速度。可以用索引快速显示、查询或者打印记录。还可以选择记录、控制重复字段值的输入并支持表间的关系操作。索引对于数据库内表之间创建关联也很重要。

2．VFP 中的索引

VFP 中的索引分为主索引、候选索引、惟一索引和普通索引 4 种。这些索引控制着在表字段和记录中禁止（前两种）或允许（后两种）重复值。

（1）主索引

主索引绝对不允许在指定的字段或表达式（索引关键字）中出现重复的值。主索引只能在数据库表中创建，主要用于在永久关系中的主表或被引用表中建立完整参照系。

主索引可以确保字段中输入值的惟一性并决定了处理记录的顺序。可以为数据库中的每一个表建立一个主索引。建立主索引的索引关键字可以看作主关键字，由于一个表只能有一个主关键字，所以一个表只能创建一个主索引。如果某个表已经有了一个主索引，可以为它添加候选索引。

（2）候选索引

像主索引一样要求索引关键字值的惟一性并决定了处理记录的顺序。因为候选索引禁止重复值，所以它们在表中有资格被选作主索引，即主索引的“候选项”。

建立候选索引的索引关键字可以看作是候选关键字，在数据库表和自由表中可为每个表建立多个候选索引。

在定义“一对多”或“一对一”永久关系中的“一”方时，既可以使用候选索引，也可以使用主索引。

（3）惟一索引

惟一索引允许在索引关键字中出现重复的值，但只存储索引文件中重复值的第一次出现。在这个意义下，“惟一”指的是索引文件中入口值是惟一的。提供惟一索引是为了保持同早期版本的兼容性。

（4）普通索引

普通索引可以决定记录的处理顺序，但是允许字段中出现重复值。在一个表中可以加入多个普通索引。在“一对多”永久关系的“多”方，可以使用普通索引。

4.5.2 建立索引

1. 使用表设计器

使用表设计器建立索引的步骤如下：

① 从“文件”菜单中选择“打开”命令，选定要打开的表。

② 从“显示”菜单中选择“表设计器”命令，表的结构将显示在“表设计器”中。

③ 在“表设计器”中，选择“索引”选项卡，如图 4-18 所示。

④ 输入索引名和索引类型。

⑤ 在“表达式”一栏中，键入作为排序依据的索引表达式。索引表达式可以是一个表中的字段，或者是一个作为排序依据的表达式。可以通过选择该方框右边的按钮，用“表达式

生成器”来建立索引表达式。

图4-18 “表设计器”中的“索引”选项卡

⑥ 若想有选择地输出记录，可在“筛选”框中输入筛选表达式，或者选择该框后面的按钮来建立表达式。

⑦ 单击“确定”按钮完成。

2．索引文件

在上述步骤中，创建表的第一个索引关键字时，VFP 将自动创建一个基本名与表文件名相同、扩展名为.CDX 的索引文件。

.CDX 文件称为结构复合压缩索引文件，它是在 VFP 数据库中最普通也最重要的一种索引文件。结构一词是指 VFP 把该文件当作表的固有部分来处理。结构复合压缩索引文件具有以下特点：

① 在打开表时自动打开。

② 在同一索引文件中能包含多个排序方案，或索引关键字。

③ 在添加、更改或删除记录时自动维护。

一个 VFP 表若有与之相关联的索引文件，则它通常是一个结构复合压缩索引文件。

VFP 还提供另外两种类型的索引文件，即非结构的.CDX 文件和单关键字的.IDX 文件。一般只使用结构复合压缩索引，而另外两种非结构索引多半是为了与以前版本兼容，在新的应用中不再使用。如果是临时使用，不希望系统自动维护索引，或是使用后就删除索引文件，则可以使用后两种非结构索引。

3．索引命令

在 VFP 中，一般情况下可以在表设计器中建立索引，特别是主索引和候选索引是在设计数据库时确定好的。但有时需要在程序中临时建立一些普通索引或惟一索引，所以仍然需要使用命令方式建立索引。索引命令的语法格式为：

```
INDEX ON〈索引表达式〉{TO〈IDX 文件名〉| TAG〈索引名〉[OF〈CDX 文件名〉]};
  [FOR〈表达式〉];
  [{ASCENDING | DESCENDING}];
  [{UNIQUE | CANDIDATE}];
  [ADDITIVE]
```

说明：

① TO〈IDX 文件名〉：建立一个单关键字的.IDX 文件，该项是为了与以前版本兼容，一般只是在建立一些临时索引时才使用。

② OF〈CDX 文件名〉:可以建立一个包含多个索引的复合索引文件名,扩展名也是.CDX。

③ FOR〈表达式〉:给出索引过滤条件,只索引满足条件的记录,该选项一般不使用。

④ ASCENDING | DESCENDING:选择建立升序或降序索引,默认升序。

⑤ UNIQUE | CANDIDATE:建立惟一索引或候选索引。

⑥ ADDITIVE:与建立索引本身无关,说明现在建立索引时是否关闭以前的索引,默认是关闭已经使用的索引,使新建立的索引成为当前索引。

【例 4-5】可以使用以下命令为数据表 Student 创建普通索引:

```
USE Student
INDEX ON 总学分 TAG 学分
```

4. 删除索引

如果某个索引不再使用了则可以删除它,删除索引的办法是在表设计器中使用“索引”选项卡,选中需要删除的索引,单击“删除”命令按钮即可删除。也可以使用命令方式删除。删除索引的命令格式为:

```
DELETE TAG 〈索引名 1〉[OF 〈CDX 文件名 1〉][,〈索引名 2〉[OF 〈CDX 文件名 2〉]] ...
```

或

```
DELETE TAG ALL [OF 〈CDX 文件名〉]
```

说明:前者可以删除索引文件中指定的索引,后者则删除全部索引。

4.5.3 使用索引排序

通过建立和使用索引,可以提高完成某些重复性任务的工作效率,例如对表中的记录排序,以及在表之间建立联系等。根据所建索引类型的不同,可以完成不同的任务,如表 4-5 所示。

表 4-5 按不同任务建立索引的类型

任 务	使 用
排序记录,以便提高显示、查询或打印的速度	使用普通索引、候选索引或主索引
在字段中控制重复值的输入并对记录排序	对数据库表使用主索引或候选索引,对自由表使用候选索引

可以用字段名或其他索引表达式对记录排序。索引将对表达式进行计算,以此确定记录出现的顺序,然后存储一个按此顺序处理表中记录的指针列表。

1. 使用对话框

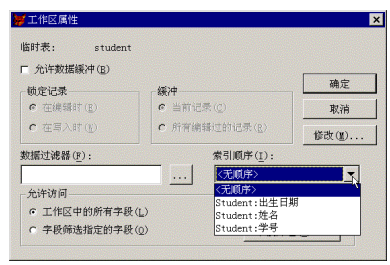
建好表的索引后,便可以用它来为记录排序。若要用索引对记录排序,操作步骤如下:

① 从“文件”菜单中选择“打开”命令,选择已建好索引的表。

② 单击“浏览”按钮。

③ 从“表”菜单中选择“属性”命令。

④ 在“工作区属性”对话框中,选择要用的索引,如图 4-19 所示。



⑤ 单击“确定”按钮。这时，显示在“浏览”窗口中的表将按照索引指定的顺序排列记录。选定索引后，通过运行查询或报表，还可对它们的输出结果进行排序。

2．使用命令

尽管结构索引在打开表时都能够自动打开，但需要使用某个特定索引项进行查询或需要记录按某个特定索引项的顺序显示时，还必须设置当前索引。设置当前索引的格式为：

SET ORDER TO [[TAG] 〈索引名〉[OF 〈CDX 文件名〉][{ASCENDING | DESCENDING}]]

说明：

① TAG〈索引名〉OF〈CDX 文件名〉：指定 CDX 索引文件中的一个索引作为主控索引。其中〈索引名〉为按照某个索引表达式建立的索引的标识名。

② {ASCENDING | DESCENDING}：不管索引是按升序或降序建立的，在使用时都可以用 ASCENDING 指定升序或 DESCENDING 指定降序。

【例 4-6】下述命令可以将学生表按年龄（升序）排序：

```
USE student
INDEX ON 出生日期 TAG 出生日期
SET ORDER TO 出生日期 ASCE
BROW
```

4.5.4 查找记录

除了使用 GO 和 SKIP 命令移动记录指针外，还可以使用 VFP 的查找记录命令来定位记录指针。查找命令有三个，即 FIND、SEEK 和 LOCATE，前 2 个需要使用索引，而后一个可以在无索引的表中进行查找。

1．字符查找命令（FIND）

查找关键字与所给字符串相匹配的第一个记录，若找到，指针指向该记录；否则指向文件尾，给出信息“没找到”。语法格式为：

FIND〈字符串〉|〈数值〉

说明：

① FIND 只能查找字符串或常数，而且表必须按相应字段索引。

② 查找的字符串无需加引号，若按字符型内存变量查找，必须使用宏代换函数&。

③ 本命令只能找出符合条件的第一个记录，若要继续查找其他符合条件的记录，可使用 SKIP 命令。

④ 使用本命令时，若是找到了符合条件的首记录，则置函数 FOUND（）的值为.T.；否则置函数 FOUND（）的值为.F.。

【例 4-7】下述命令在学生表中查找第一个姓李的同学，并显示该同学的信息：

```
USE student
INDEX ON 姓名 TAG 姓名
SET ORDER TO TAG 姓名
FIND 李
DISP
```

2．表达式查找命令（SEEK）

查找关键字与所给字符串相匹配的第一个记录，若找到，指针指向该记录；否则指向文件尾，给出信息“没找到”。语法格式为：

SEEK〈表达式〉

说明：

① 只能找出符合条件的第一条记录。

② 本命令可查找字符、数值、日期和逻辑型索引关键字。

③ 若〈表达式〉为字符串，则须用界限符括起来（’’，" "，[ ]）；若按字符型内存变量查找，不必使用宏代换&函数。

④ 使用本命令时，若是找到了符合条件的首记录，则置函数 FOUND（）的值为.T.；否则置其值为.F.。

【例 4-8】下述命令在学生表中查找第一个出生日期为 1980 年 1 月 8 日的同学，并显示该同学的信息：

```
USE student
INDEX ON 出生日期 TAG 出生日期
SET ORDER TO TAG 出生日期
SEEK CTOD（"01/08/80"）
DISP
```

3．顺序查询命令（LOCATE）

查找当前数据表中满足条件的第一条记录。语法格式为：

LOCATE [〈范围〉][FOR 〈条件〉]

说明：

①〈范围〉项缺省时，系统默认为 ALL。

② 若找到满足条件的首记录，则指针指向该记录，否则指向范围尾或文件尾。

③ 若缺省所有可选项，则记录指针指向 1 号记录。若想继续找，可以利用下面的继续查找命令（CONTINUE）。

4．继续查找命令（CONTINUE）

使最后一次 LOCATE 命令继续往下搜索，指针指向满足条件的下一条记录。命令格式为：

CONTINUE

说明：

① 使用本命令前，必须使用过 LOCTAE 命令。

② 此命令可反复使用，直到超出〈范围〉或文件尾。

【例 4-9】下述命令在学生表中查找姓李的同学，并显示该同学的信息：

```
CLEAR
USE student
LOCATE FOR 姓名 = "李"
```

```
DISP
CONTINUE
DISP
```

4.6 使用多个表

前面各节所介绍的都是对当前表进行的操作，似乎默认了在同一时刻只能使用一个表，其实不然。VFP **允许在应用程序中同时打开多个表。**

在下面的示例中还需要使用两个数据表，成绩表 cj.dbf **和课程表** kc.dbf**，其中内容见表** 4-6**、表** 4-7**。**

表 4-6 **课程表** kc.dbf

课 程 号	课 程 名	学 分
111101	高等数学（上）	5
111102	高等数学（下）	5
411206	电工学	3
141203	理论力学‘	3
321202	机械设计	4
411201	数据结构	4
631206	经济法	3
631208	管理学原理	2
621201	会计学	4
611555	财务管理	2
151101	大学英语（1）	4
151102	大学英语（2）	4
151103	大学英语（3）	4
…	…	…

结构为 kc**（课程号（**C,6**），课程号（**C,16**），学分（**N,2**））**，以字段“课程号”建立了主索引：课程号。

表 4-7 **成绩表** cj.dbf

学 号	课 程 号	成 绩	补 考 成 绩
2000314110	111101	78	0
2000314110	141203	88	0
2000220203	111102	78	0
2000220203	141203	74	0
…	…	…	…

结构为 cj**（学号（**C,10**），课程号（**C,6**），成绩（**N,3**），补考成绩（**N,3**））**，表中按“学号”建立普通索引。

4.6.1 工作区

若要使用多个表，就要使用多个工作区。一个工作区是一个编号区域，用它来标识一个已打开的表，每个工作区中只能打开一个表。VFP 可以在 32767 个工作区中打开和操作表。

工作区除了可以用它的编号表示外，还可以用在工作区中打开的表的名称、别名来标识。表别名是一个名称，它可以引用在工作区中打开的表。

1．指定工作区

如果没有指定工作区，系统默认总是在第 1 个工作区中工作，在第 1 个工作区中打开和关闭表。使用 SELECT 命令可以将指定的工作区设为当前工作区，其语法格式为：

```
SELECT〈工作区号〉|〈表别名〉
```

说明：

①〈工作区号〉的取值范围为 0～32767。如果取值为 0，则激活尚未使用的工作区中编号最小的那一个。

②〈表别名〉是打开表的别名，用来指定包含打开表的工作区。也可以用从 A 到 J 中的一个字符作为〈表别名〉来激活前 10 个工作区中的一个。

2．在不同的工作区中打开和关闭表

可以使用 USE 命令在不同的工作区中打开或关闭表。

（1）在当前工作区打开和关闭表

当执行不带表名的 USE 命令，并且在当前所选工作区中有打开的表文件时，则关闭该表。例如，可以使用以下代码打开 cj 表，显示“浏览”窗口，然后关闭此表：

```
USE cj
BROWSE
USE
```

（2）在最低可用工作区中打开表

可以在 USE 命令 IN 子句后面加工作区 0。例如，如果工作区 1 到 10 中都有打开的表，可以使用以下命令，在工作区 11 中打开 cj 表：

```
USE cj IN 0
```

说明：在一个工作区中，不能同时打开多个表。

（3）在指定工作区中关闭表

使用 USE 命令的 IN 子句，指出想要关闭的表所在的工作区。

当在同一工作区中打开其他表，或者发出有 IN 子句的 USE 命令并指定当前工作区时，可以自动关闭已打开的表。下面的代码首先打开 cj 表，然后显示它，最后通过发出 USE IN 命令和 cj 表别名关闭 cj 表：

```
USE cj
BROWSE
USE IN cj
```

（4）关闭所有工作区中打开的表

使用命令 CLOSE ALL 可以关闭所有工作区中已打开的表，并将 1 号工作区置为当前工作区。

3．使用表别名

表别名是 VFP 用来指定在一个工作区中打开的表的名称。

（1）默认表别名

打开一个表时，VFP 自动使用文件名作为默认表别名。例如，如果用下面的命令在 0 号工作区打开文件 cj.dbf，则自动为表指定默认别名 cj：

```
SELECT 0
USE cj
```

然后，可以使用别名 cj 在命令或函数中标识该表。

（2）创建用户自定义别名

在打开表时，使用包含 ALIAS〈表别名〉子句的 USE 命令可以为它指定用户自定义的表别名。例如，可以使用以下命令，在 0 号工作区中打开文件 cj.dbf，并为它指定一个别名“成绩”：

```
SELECT 0
USE cj ALIAS 成绩
```

然后必须使用别名成绩引用打开的表。

别名最多可以包括 254 个字母、数字或下划线，但首字符必须是汉字、字母或下划线。如果所提供的别名包含不支持的字符，则 VFP 会自动创建一个别名。

（3）使用 Visual FoxPro 指定的别名

如果使用包含 AGAIN 子句的 USE 命令同时在多个工作区中打开同一个表，并且在每个工作区中打开该表时没有指定别名，或者别名发生冲突时，VFP 将自动指定表的别名。

在前 10 个工作区中指定的默认别名是工作区字母 A 到 J，在工作区 11 到 32767 中指定的别名是 W11 到 W32767。可以像使用任何默认别名或用户自定义别名那样，使用这些 VFP 指定的别名来引用在一个工作区中打开的表。

4．引用其他工作区中打开的表

在表别名后加上点号分隔符“.”或“->”操作符，然后再接字段名，可以引用其他工作区中的字段。例如，可以使用以下代码，在一个工作区中访问其他工作区中打开的 Cj 表的“课程号”字段：cj.课程号

如果要引用的表是用别名打开的，则也可以使用别名。例如，如果 cj 表是使用别名“成绩”打开的，那么，可以使用以下代码引用表中的课程号字段：成绩.课程号

可以在一个表所在的工作区之外，使用表名或表别名来明确标识该表。

5．使用数据工作期

“数据工作区”窗口是 VFP 提供的一个管理工作区的工具。使用“数据工作区”窗口，可以查看在一个 VFP 工作期中已打开表的列表，还可以在工作区中打开表、关闭表。

打开“数据工作期”窗口：从“窗口”菜单选择“数据工作期”菜单项，或者使用 SET 命令。当在“命令”窗口中键入 SET 时，VFP 打开“数据工作期”窗口，并显示在当前数据

工作期中打开表的工作区别名。

在工作区中打开表：在“数据工作期”窗口中，单击“打开”按钮。

在工作区中关闭表：在“数据工作期”窗口中选定要关闭的表别名，然后单击“关闭”按钮。

4.6.2 设置表间的临时关系

在建立表间的临时关系（关联）后，会使得一个表（从表）的记录指针自动随另一个表（主表）的记录指针移动。这样，便允许当在关系中“一”方（或主表）选择一个记录时，会自动去访问表关系中“多”方（或从表）的相关记录。例如，可以关联 student 表和 cj 表，此后当把 student 表的记录指针移到一个特定学生时，cj 表的记录指针也移到有相同的学号的记录上去。

可以使用“数据工作期”窗口或使用 SET RELATION 命令建立两个表之间的关系。

1．使用“数据工作期”窗口

通过“数据工作期”窗口可以创建表间的临时关系。若要临时关联表，在“数据工作期”窗口中选定要关联的主表，单击“关系”按钮，然后选择从表，在弹出的“设置索引顺序”对话框中选择索引（如“学号”），创建关联。

2．使用 SET RELATION 命令。

SET RELATION 命令可以建立两表之间的关系，其中一个是在当前选定工作区中打开的表，而另一个则是在其他工作区中打开的表。通常这两个表具有相同字段，而且用来建立关系的表达式常常就是从表主控索引的索引表达式。

【例 4-10】学生可以有许多相关联的成绩记录。如果在两个表共同拥有的字段之间创建关系，就能很容易地看到任何一个学生的所有成绩记录。下面的代码中，在创建 student 表中的“学号”字段和 cj 表中的“学号”索引标识之间的关联时，使用了两个表都有的字段“学号”。

```
USE student IN 1                    && 在 1 号工作区中打开 student 表（主表）
USE cj IN 2                         && 在 2 号工作区中打开 cj 表（从表）
SELECT cj                           && 选定从表工作区
SET ORDER TO TAG 学号               && 使用索引标识“学号”指定从表的顺序
SELECT student                      && 选定主表工作区
SET RELATION TO 学号 INTO cj        && 创建主表与从表的主控索引之间的关联
SELECT cj
BROWSE NOWAIT
SELECT student
BROWSE NOWAIT
```

在“命令窗口”依次执行上述命令，将打开两个“浏览”窗口，移动主表的记录指针会改变在从表中显示的数据集合，如图 4-20 所示。

【例 4-11】设有一个单科（高等数学（上））成绩表 d_cj.dbf（学号（C,10），成绩（N,3）），试用 d_cj.dbf 中的成绩来修改 cj.dbf 中的相应成绩。相应的命令如下：

```
USE kc
```

```
LOCATE FOR 课程名 = "高等数学（上）"
no = 课程号
SELECT 2
USE d_cj
INDEX ON 学号 TAG 学号
SELECT 1
USE cj1
SET RELATION TO 学号 INTO b
REPL ALL 成绩 WITH b->成绩 FOR 学号 = b->学号 AND 课程号 = no
```

注意：有的时候需要将“多表”设为主表，“一表”设为从表，如本节习题中上机题 2 **中的（1)、(3)、(4）和（5)。**

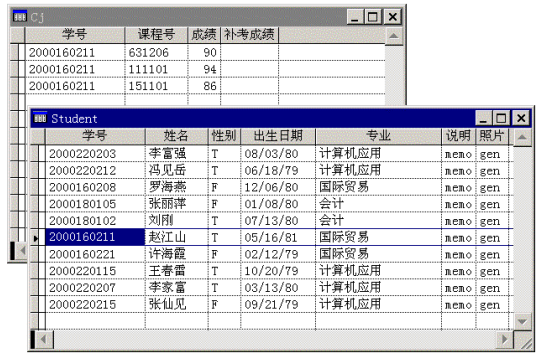


图4-20　**设置表间的临时关系**

3．关联单个表中的记录

可以在单个表中创建记录间的关系，即自引用关系。若需要的所有信息都存储在单个表中，这种关系很有用。

【例 4-12**】如果遍历** student **表中的班级，随着记录指针从一个班级到另一个班级的移动，每个班级的学生自动更改。**

若要创建自引用关系，可以两次打开同一个表，在一个工作区中打开一个表，并使用 USE AGAIN **命令在另外工作区中再次打开此表，然后使用索引来关联记录。例如，可以使用以下代码，根据“专业”字段对** student **表进行排序，然后创建索引标识** zy**，并以此建立并浏览一个自引用关联：**

```
SELECT 0
USE student ALIAS Student
SELECT 0
USE student AGAIN ALIAS Student_a
INDEX ON 专业 TAG zy
SET ORDER TO zy
SELECT Student
SET RELATION TO 专业 INTO Student_a ADDITIVE
SELECT Student_a
```

```
BROWSE NOWAIT
SELECT Student
BROWSE NOWAIT
```

在“命令窗口”依次执行上述命令，在“数据工作期”窗口中浏览表 Student **和** Student_a**，当在** Student **“浏览”窗口中移动记录指针时，会自动刷新** Student_a **“浏览”窗口，并在其中显示隶属于选定专业的学生，如图** 4-21 **所示。**

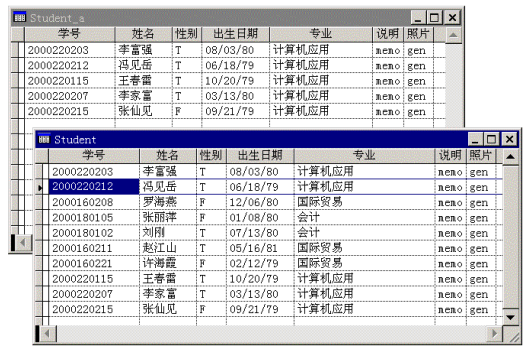


图4-21　**自引用关系**

4.7　Visual FoxPro **的数据库**

数据库提供了如下的工作环境：存储一系列的表，在表间建立关系，设置属性和数据有效性规则使相关联的表协同工作。数据库文件保存为带.DBC **扩展名的文件。数据库可以单独使用，也可以将它们合并成一个项目，用“项目管理器”进行管理。数据库必须在打开后才能访问它内部的表。**

4.7.1　**数据库表与自由表**

在 VFP **中，有两种状态的表：数据库表（与数据库相关联的表），自由表（与数据库无关联的表）。相比之下，数据库表具有如下的优点：**

① **长表名和表中的长字段名。**
② **表中字段的标题和注释。**
③ **默认值、输入掩码和表中字段格式化。**
④ **表字段的默认控件类。**
⑤ **字段级规则和记录级规则。**
⑥ **支持参照完整性的主关键字索引和表间关系。**
⑦ INSERT、UPDATE **或** DELETE **事件的触发器。**

通过把表放入数据库中，可以减少冗余数据的存储，保护数据的完整性；可以控制字段怎样显示或键入到字段中的值；还可以添加视图并连接到一个数据库中，用来更新记录或扩充访问远程数据的能力。

4.7.2 创建数据库

要想把数据并入数据库中，必须先建立一个新的数据库，然后加入需要处理的表，并定义它们之间的关系。VFP 提供两种方法建立数据库，即使用“数据库设计器”或使用命令。

1．打开数据库设计器

打开数据库设计器的方法有两种：在项目管理器中打开和通过菜单命令打开。下面介绍通过菜单命令打开数据库设计器的步骤：

① 从“文件”菜单中选择“新建”命令。在“新建”对话框中选择“数据库”单选钮，然后单击“新建文件”按钮，打开“创建”对话框。

② 在“创建”对话框中输入新数据库名（默认的数据库名为“数据 1”），单击“保存”按钮，即创建了一个新数据库（空），并同时打开“数据库设计器”窗口。此时，“数据库设计器”工具栏将变为有效，单击鼠标右键将弹出相关的快捷菜单，如图 4-22 所示。

图4-22 “数据库设计器”窗口

数据库建立后，自动产生同名但类型不同的三个文件.DBC（数据库文件）、.DCT（数据库备注文件）、.DCX（数据库索引文件）。后两个文件依附于数据库文件，但不可缺少，数据库备份时一定要同时备份其他两个文件，否则备份后的数据库将不能使用。

2．使用数据库建立命令

在命令窗口键入 CREATE DATABASE 命令，正确选择保存位置和数据库名，单击“保存”按钮将创建一个新数据库（空）。

如果数据库文件已经存在，并且设置 SET SAFETY ON 时，会出现是否覆盖提示，否则直接覆盖。以后新建文件都有类似的情况。

4.7.3 在数据库中加入表

建立数据库后的第一步是向数据库中添加表，可以选定目前不属于任何数据库的表。因为一个表在同一时间内只能属于一个数据库，所以必须将表先从旧的数据库中移去后才能将它用于新的数据库中。

1．向数据库中添加表

从“数据库”菜单或“数据库设计器”工具栏或在数据库设计器窗口上单击鼠标右键，从弹出快捷菜单中选择“添加表”命令。在“打开”对话框中选定一个表，然后单击“确定”按钮，即可将表添加到数据库中。

还可以使用命令方式将自由表添加到当前的数据库中，命令格式是：

ADD TABLE {〈表文件名〉| ?} [NAME〈长文件名〉]

说明：

①〈表文件名〉是指要添加到数据库中去的自由表表文件名，使用?号则显示“ 打开 ”对话框以便从中选择添加到数据库中去的自由表。

② NAME 〈长文件名〉则为表指定一个长表名，最多可以有 128 个字符。

2．从数据库中移去表

当数据库不再需要某个表或其他数据库需要使用此表时，可以从该数据库中移去此表。若要从数据库中移去表：选定要移去的表，从“ 数据库 ”菜单中选择“ 移去 ”，或者从“ 数据库设计器 ”工具栏中单击“ 移去表 ”按钮。在对话框中单击“ 移去 ”按钮。

也可以使用命令方式将表移出当前的数据库，命令格式是：

REMOVE TABLE {〈表文件名〉| ？}[DELETE]

说明：DELETE 选项不仅把表移出数据库，还将其从磁盘上删除。

4.7.4 打开数据库

数据库使用之前必须先打开，打开数据库的方法可以使用菜单方式或命令方式。

1．菜单方式

在 VFP 的“ 文件 ”菜单中选择“ 打开 ”命令，在弹出的“ 打开 ”对话框中选择“ 文件类型 ”为数据库（*.DBC），选取需要打开的数据库文件后，单击“ 确定 ”按钮，如图 4-1 所示。

2．命令方式

打开数据库命令的格式为：

```
OPEN DATABASE [〈数据库文件名〉｜?]
  [EXCLUSIVE｜SHARED]
  [NOUPDATE]
  [VALIDATE]
```

说明：

① 不指定文件名而使用问号时，显示“ 打开 ”对话框。

② EXCLUSIVE｜SHARED：数据库以独占或共享方式打开。如果无该项，打开方式由设置语句控制，即 SET EXCLUSIVE ON（默认）｜OFF，系统默认为“ 独占 ”方式。

③ NOUPDATE：数据库以只读方式打开，等效于在“ 打开 ”对话框中选中“ 以只读方式打开 ”复选框。系统默认为读写方式。

④ VALIDATE：检查数据库中引用的对象是否合法，如检查数据库中的表和索引是否可用等。

3．同时打开多个数据库

VFP 允许同时打开多个数据库，但只有一个数据库是当前数据库，当前数据库可在打开的数据库之间选择，选择方法可以使用 VFP“ 常用 ”工具栏的“ 数据库 ”下拉列表框中选择，如图 4-23 所示。

图4-23　同时打开多个数据库

也可以使用命令 SET DATABASE TO [〈数据库名〉]选择当前数据库。

注意：虽然数据库已经打开，但要使用数据库中的表，仍然需要打开表。

4.7.5　关联表

通过链接不同表的索引，“数据库设计器”可以很方便地建立表之间的关系。因为这种在数据库中建立的关系被作为数据库的一部分而保存起来，所以称为永久关系。每当在“查询设计器”或“视图设计器”中使用表，或者在创建表单时所用的“数据环境设计器”中使用表时，这些永久关系将作为表间的默认链接。

1．准备关联

表之间创建关系之前，想要关联的表要有一些公共的字段和索引。这样的字段称为主关键字字段和外部关键字字段。主关键字字段标识了表中的特定记录。外部关键字字段标识了存于数据库里其他表中的相关记录。需要对主关键字字段做一个主索引，对外部关键字字段做普通索引。

要决定哪个表需要这些字段，可考虑使用记录号怎样关联数据。例如，一个学生可能有多个成绩。因此，学生（student）表应包含主记录，成绩（cj）表包含相关记录。

为准备两个关联表中的主表，需要在主表中添加主关键字字段，如“学号”。这是因为学生表中一条记录与成绩表的多个记录关联。

要在两个表之间提供公共字段，需要在带有关联记录的表添加外部关键字字段，如成绩表。外部关键字字段必须以相同的数据类型匹配主关键字字段，而且一般用相同的名称。且以主关键字字段和外部关键字字段创建的索引必须带有相同的表达式，如图 4-24 所示。

2．创建和编辑关系

定义完关键字段和索引后，即可创建关系。如果表中还没有索引，需要在“表设计器”中打开表，并且向表添加索引。

若要在表间建立关系，将一个表的索引拖到另一个表的相匹配的索引上。设置完关系之后，在数据库设计器中可看到一条线连接两表，如图 4-25 所示。

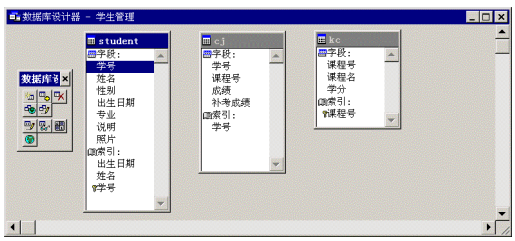


图4-24　数据库中的表

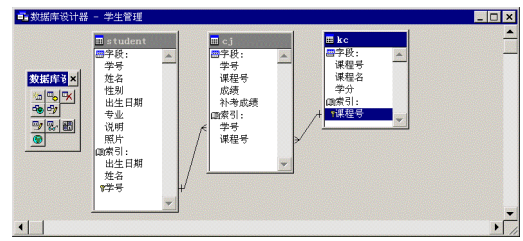


图4-25　关系线表示了两个表之间的连接

只有在“数据库属性”对话框中的“关联”选项打开时，才能看到这些表示关系的连线。可以打开“数据库属性”对话框，方法是从“数据库设计器”的快捷菜单中选择“属性”。

也可编辑关系，若要编辑表间的关系，双击表间的关系线，再选择“编辑关系”对话框中的设置即可，如图 4-26 所示。

图4-26 “编辑关系”对话框

所建关系的类型是由子表中所用索引的类型决定的。例如，如果子表的索引是主索引或候选索引，则关系是一对一的；对于惟一索引和普通索引，将会是一个一对多的关系。

3．永久关系

上面用索引在数据库中建立了表间的永久关系。永久关系是存储在数据库文件中的关系，并在“查询设计器”和“视图设计器”中，自动作为默认连接条件使用的数据库表间关系。永久关系在“数据库设计器”中显示为表索引间的连接线，当在数据环境中打开这些表时，永久关系也作为默认关系显示。与 SET RELATION 命令设置的临时关系不同，永久关系在每次使用表时不需要重新创建。但是，由于永久关系并不控制表中记录指针间的关系，因此在开发 VFP 应用程序时，不仅需用永久关系，也需使用临时的 SET RELATION 关系。

4.7.6 定义字段显示

将表添加入数据库后，便可以立即获得许多在自由表中得不到的属性。这些属性被作为数据库的一部分保存起来，并且一直为表所拥有，直到表从这个数据库中移去为止。通过设置数据库表的字段属性，可以为字段设置标题、为字段输入注释、为字段设置默认值、设置字段的输入掩码和显示格式、设置字段的控件类和库、设置有效性规则对输入字段的数据加以限制等。

1．设置字段标题

通过在表中给字段建立标题，可以显示在“浏览”窗口显示的字段的说明性标签或表单。按照以下步骤可以给字段指定一个标题：

① 在“数据库设计器”中选定表，单击该表或者从数据库工具栏中选择“修改表”。

② 选定需要指定标题的字段。

③ 在“标题”框中，键入为字段选定的标题。例如，某字段名为“学号”，当使用“学生证编号”作为标题显示该字段时，浏览窗口中原来的“学号”字段名被替换为“学生证编号”，如图 4-27 所示。由于可以用自己命名的标题来取代原来的字段名，这为显示表单中的表提供了很大的灵活性。

④ 单击“确定”按钮。

2．为字段输入注释

在建立好表的结构以后，可能还想输入一些注释，来提醒自己或其他人表中的字段代表什么意思。在“表设计器”中的“字段注释”框内输入信息，即可对每一个字段进行注释。按照以下步骤可以为一个字段作注释：

① 在“表设计器”中，选定字段。

② 在“字段注释”框中键入注释内容，如图 4-27 所示。

③ 单击“确定”按钮。

4.7.7 控制字段数据输入

可使表中数据输入更容易一些，方法是提供字段的默认值，定义输入到字段的有效性规则。

1．设置默认字段值

要想在创建新记录时自动输入字段值，可以在“表设计器”中用字段属性为该字段设置默认值。例如，如果买主大部分来自一个特殊地区，可把该地区名称设置为地区字段的默认值。按照以下步骤，可以设置字段的默认值：

① 在“数据库设计器”中选定表。

② 从“数据库”菜单中选择“修改”。

③ 在“表设计器”中选定要赋予默认值的字段。

④ 在‘默认值’框中键入要显示在所有新记录中的字段值（字符型字段应用引号括起来）。例如，可以使 Student 表中“学号”字段的所有新记录都有一个默认值为“2000000000”，如图 4-27 所示。

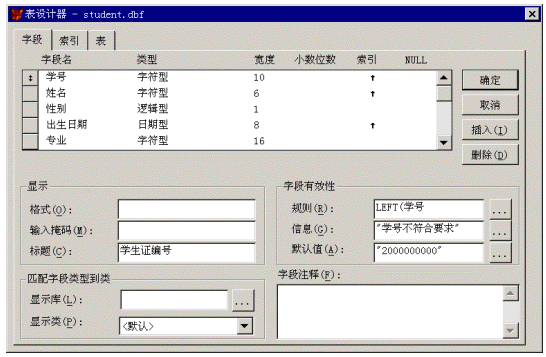


图4-27　基于标题的列名

⑤ 单击“确定”按钮。

2．设置有效性规则和有效性说明

如果在定义表的结构时输入字段的有效性规则，那么可以控制输入该字段的数据类型。例如，可以限制某字段可接受的输入范围，如分数在 0～100 之间。按照以下步骤可以为字段设置有效性规则和有效性说明：

① 在“表设计器”中打开表。

② 在“表设计器”中选定要建立规则的字段名。

③ 在“规则”方框旁边选择对话按钮。

④ 在“表达式生成器”中设置有效性表达式，并单击“确定”按钮。例如，限制“学号”字段的前 4 位只能为"2000"，并且输入的学号必须满 10 位：

SUBSTR（学号,1,4） = "2000" AND LEN（TRIM（学号）） = 10

建立有效性规则时，必须创建一个有效的 VFP 表达式，其中要考虑到这样一些问题：字段的长度、字段可能为空或者包含了已设置好的值等。表达式也可以包含结果为真或假的函数。

⑤ 在“信息”框中，键入用引号括起的错误信息，例如，显示“学号不符合要求”，如图 4-27 所示。

⑥ 单击“确定”按钮。

如果输入的信息不能满足有效性规则，在“有效性说明”中设定的信息便会显示出来，如图 4-28 所示。

图4-28　屏幕上显示有效性说明

4.7.8　控制记录的数据输入

不但可以给表中的字段赋与数据库的属性，而且可以为整个表或表中的记录赋与属性。在“表设计器”中，通过“表”选项卡可以访问这些属性。

1．设置表的有效性规则

在向表中输入记录时，要想比较两个以上的字段，或查看记录是否满足一定的条件，可以为表设置有效性规则。按照以下步骤可以设置有效性规则：

① 选定表，在“数据库”菜单中选择“修改”。将打开“表设计器”对话框。

② 在“表设计器”中选择“表”选项卡。

③ 在“规则”框中，输入一个有效的 VFP 表达式定义规则，或者单击“…”按钮使用“表达式生成器”来生成一个表达式规则，如图 4-29 所示。例如，Student 表中，学生的年龄必须小于 30 岁，则可以在“表”选项卡的“有效性规则”框中键入下述表达式：

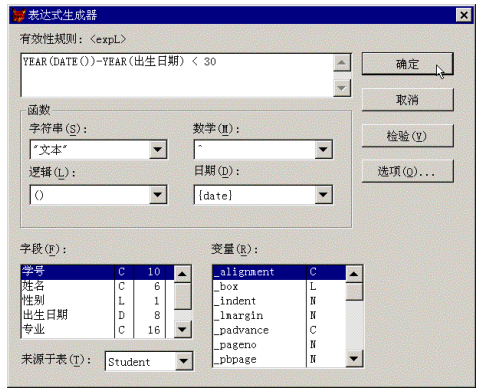


图4-29　“表达式生成器”对话框

YEAR（DATE（））-YEAR（出生日期） < 30

④ 在“信息”框中输入提示信息。例如，“信息”文字可以是："新生年龄必须小于 30 岁"。当有效性规则未被满足时，将会显示该信息。

⑤ 单击“确定”按钮。

⑥ 在“表设计器”中单击“确定”按钮。

2．设置触发器

触发器是一个在输入、删除或更新表中的记录时被激活的表达式。通常，触发器一般需要输入一个程序或存储过程，在表被修改时，它们被激活。

4.7.9 管理数据库记录

建立关系后，也可设置管理数据库关联记录的规则。这些规则控制参照完整性。例如，如果添加一个成绩记录，可能想在成绩表中自动添加关于该学生的基本信息。为了帮助设置规则，控制如何在关系表中插入、更新或删除记录，可使用“参照完整性设计器”。若要使用“参照完整性生成器”，可以完成以下步骤：

① 在“数据库设计器”中建立两表之间的关系，或者双击关系线来编辑关系。

② 在“编辑关系”对话框中选择“参照完整性”按钮，如图 4-30 所示。

图4-30 在“编辑关系”对话框中的参照完整性按钮

③ 在“参照完整性生成器”中选择更新、删除或插入记录时所遵循的若干规则，如图 4-31 所示。

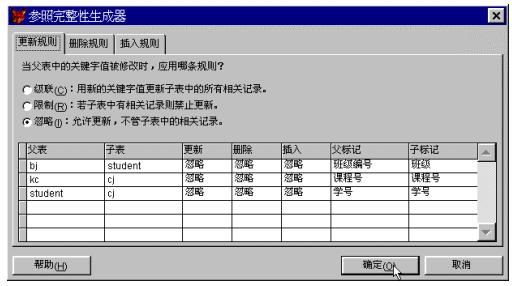


图4-31 “参照完整性生成器”对话框

④ 单击“确定”按钮，然后单击“是”按钮保存所做的修改，生成“参照完整性”代码，并退出参照完整性生成器。

4.7.10 为数据库添加备注

若要使用数据库的说明，可添加注释。若要添加数据库的备注，只需从“数据库”菜单中选择“属性”命令，再在“注释”框键入备注内容即可。

4.8 习题

一、选择题

1．在 Visual FoxPro 的命令窗口中键入 CREATE DATA 命令以后，屏幕会出现一个创建对话框，要想完成同样的工作，还可以采取如下步骤（　　）。

A）单击“文件”菜单中的“新建”按钮，然后在新建对话框中选定“数据库”单选钮，再单击“新建文件”命令按钮

B）单击“文件”菜单中的“新建”按钮，然后在新建对话框中选定“数据库”单选钮，再单击“向导”命令按钮

C）单击“文件”菜单中的“新建”按钮，然后在新建对话框中选定“表”单选钮，再单击“新建文件”命令按钮

D）单击“文件”菜单中的“新建”按钮，然后在新建对话框中选定“表”单选钮，再单击“向导”命令按钮

2．扩展名为 DBC 的文件是（　　）。

A）表单文件　　B）数据库表文件　　C）数据库文件　　D）项目文件

3．下面有关索引的描述正确的是（　　）。

A）建立索引以后，原来的数据库表文件中记录的物理顺序将被改变

B）索引与数据库表存储在一个文件中

C）创建索引是创建一个指向数据库表文件记录的指针构成的文件

D）使用索引并不能加快对表的查询操作

4．若所建立索引的字段值不允许重复，并且一个表中只能创建一个，它应该是（　　）。

A）主索引　　B）惟一索引　　C）候选索引　　D）普通索引

5．参照完整性的规则不包括（　　）。

A）更新规则　　B）删除规则　　C）插入规则　　D）检索规则

6．一数据库名为 student，要想打开该数据库，应使用命令（　　）。

A）OPEN student　　B）OPEN DATA student

C）USE DATA student　　D）USE student

7．要为当前表所有职工增加 100 元工资，应该使用命令（　　）。

A）CHANGE 工资 WITH 工资+100

B）REPLACE 工资 WITH 工资+100

C）CHANGE ALL 工资 WITH 工资+100

D）REPLACE ALL 工资 WITH 工资+100

8．以下关于自由表的叙述，正确的是（　　）。

A）全部是用以前版本的 FoxPro（FoxBASE）建立的表

B）可以用 Visual FoxPro 建立，但是不能把它添加到数据库中

C）自由表可以添加到数据库中，数据库表也可以从数据库中移出成为自由表

D）自由表可以添加到数据库中，但数据库表不可以从数据库中移出成为自由表

9．Visual FoxPro 数据库文件是（　　）。

A）存放用户数据的文件　　B）管理数据库对象的系统文件

C）存放用户数据和系统数据的文件　　D）前三种说法都对

二、填空题

1．自由表的扩展名是_____。

2．同一个表的多个索引可以创建在一个索引文件中，索引文件名与相关的表同名，索引文件的扩展名是_____，这种索引称为_____。

3．Visual FoxPro 的主索引和候选索引可以保证数据的_____完整性。

4．实现表之间临时联系的命令是_____。

5．在定义字段有效性规则时，在规则框中输入的表达式类型是_____。

三、上机题

1．首先完成以下基本操作题：

（1）创建一个新的项目“客户管理”。

（2）在新建立的项目“客户管理”中创建数据库“订货管理”。

（3）在“订货管理”数据库中建立表 order_list，表中数据见表 4-8。

表 4-8　order_list 表

客户号	订单号	订购日期	总金额
C10001	OR-01C	2001-10-10	4000
A00112	OR-22A	2001-10-27	5500
B20001	OR-02B	2002-2-13	10500
C10001	OR-03C	2002-1-13	4890
C10001	OR-04C	2002-2-12	12500
A00112	OR-21A	2002-3-11	30000
B21001	OR-11B	2001-5-13	45000
C10001	OR-12C	2001-10-10	3210
B21001	OR-13B	2001-5-5	3900
B21001	OR-23B	2001-7-8	4390
B20001	OR-31B	2002-2-10	39650
C10001	OR-32C	2001-8-9	7000
A00112	OR-33A	2001-9-10	8900
A00112	OR-41A	2002-4-1	8590
C10001	OR-44C	2001-12-10	4790
B21001	OR-37B	2002-3-25	4450

其结构描述为 order_list（客户号（C,6），订单号（C,6），订购日期（D），总金额（F,15.2））。

（4）为 order_list 表创建一个主索引，索引名和索引表达式均是订单号。

（5）在“订货管理”数据库中建立表 order_detail，表中数据见表 4-9。

表 4-9　order_detail 表

订单号	器件号	器件名	单价	数量
OR-01C	P1001	CPU P4 1.4G	1050	2

（续）

订 单 号	器 件 号	器 件 名	单 价	数 量
OR-01C	D1101	3D 显示卡	500	3
OR-02B	P1001	CPU P4 1.4G	1100	3
OR-03C	S4911	声卡	350	3
OR-03C	E0032	E 盘（闪存）	280	10
OR-03C	P1001	CPU P4 1.4G	1090	5
OR-03C	P1005	CPU P4 1.5G	1400	1
OR-04C	E0032	E 盘（闪存）	290	5
OR-04C	M0256	内存	350	4
OR-11B	P1001	CPU P4 1.4G	1040	3
OR-12C	E0032	E 盘（闪存）	275	20
OR-12C	P1005	CPU P4 1.5G	1390	2
OR-12C	M0256	内存	330	4
OR-13B	P1001	CPU P4 1.4G	1095	1
OR-21A	S4911	声卡	390	2
OR-21A	P1005	CPU P4 1.5G	1350	1
OR-22A	M0256	内存	400	4
OR-23B	P1001	CPU P4 1.4G	1020	7
OR-23B	S4911	声卡	400	2
OR-23B	D1101	3D 显示卡	540	2
OR-23B	E0032	E 盘（闪存）	290	5
OR-23B	M0256	内存	395	5
OR-31B	P1005	CPU P4 1.5G	1320	2
OR-32C	P1001	CPU P4 1.4G	1030	5
OR-33A	E0032	E 盘（闪存）	295	2
OR-33A	M0256	内存	405	6
OR-37B	D1101	3D 显示卡	600	1
OR-41A	M0256	内存	380	10
OR-41A	P1001	CPU P4 1.4G	1100	4
OR-44C	S4911	声卡	385	3
OR-44C	E0032	E 盘（闪存）	296	2
OR-44C	P1005	CPU P4 1.5G	1300	2

其结构描述为 order_detail（订单号（C,6），器件号（C,6），器件名（C,16），单价（F,10.2），数量（I））。

（6）为新建立的 order_detail 表建立一个普通索引，索引名和索引表达式均是“订单号”。

（7）为表 order_detail 的“单价”字段定义默认值为 NULL。

（8）为表 order_detail 的“单价”字段定义约束规则“单价 > 0”，违背规则时的提示信息是“单价必须大于零”。

（9）建立表 order_list 和表 order_detail 间的永久联系（通过“订单号”字段）。

（10）为以上建立的联系设置参照完整性约束：更新规则为“限制”，删除规则为“级联”，插入规则为“限制”。

（11）关闭“订货管理”数据库，然后建立自由表 customer，表的内容见表 4-10。

表 4-10　customer 表

客户号	客户名	地址	电话
C10001	三益贸易公司	平安大道 100 号	66661234
C10005	比特电子工程公司	中关村南路 100 号	62221234
B20001	萨特高科技集团	上地信息产业园	87654321
C20111	一得信息技术公司	航天城甲 6 号	89012345
B21001	爱心生物工程公司	生命科技园 1 号	66889900
A00112	四环科技发展公司	北四环路 211 号	62221234

其结构描述为 customer（客户号（C,6），客户名（C,16），地址（C,20），电话（C,14））。

（12）打开“订货管理”数据库，并将表 customer 添加到该数据库中。

（13）为 customer 表创建一个主索引，索引名和索引表达式均是“客户号”。

2．在完成基本操作题的基础上，完成以下应用题：

（1）列出客户名为“三益贸易公司”的订购单明细（order_detail）记录（将结果先按“订单号”升序排列，同一订单的再按“单价”降序排列），并将结果存储到 results1 表中（表结构与 order_detail 表结构相同）。

（2）列出目前有订购单的客户信息（即有对应的 order_list 记录的 customer 表中的记录），同时要求按客户号升序排序，并将结果存储到 results2 表中（表结构与 customer 表结构相同）。

（3）列出所有订购单的订单号、订购日期、器件号、器件名和总金额（按订单号升序），并将结果存储到 results3 表中（其中订单号、订购日期、总金额取自 order_list 表，器件号、器件名取自 order_detail 表）。

（4）按总金额降序列出所有客户的客户号、客户名及其订单号和总金额，并将结果存储到 results4 表中（其中客户号、客户名取自 customer 表，订单号、总金额取自 order_list 表）。

（5）为 order_detail 表增加一个新字段：新单价（类型与原来的单价字段相同），然后根据 order_list 表中的“订购日期”字段的值确定 order_detail 表的“新单价”字段的值，原则是：订购日期为 2001 年的“新单价”字段的值为原单价的 90%，订购日期为 2002 年的“新单价”字段的值为原单价的 110%（在修改操作过程中不要改变 order_detail 表记录的顺序）。

第 5 章　查询与视图

数据的检索是应用程序处理数据的重要任务之一，上一章介绍的对表中数据的索引、查找等命令都是早期 xBase 语言中最常用的。在 VFP 中，“查询”与“视图”是为方便检索数据而提供的工具，在需要迅速获得所需要的数据时，可以使用“查询”或“视图”来检索存储在表中的信息。查询与视图的目的都是为了从数据中快速获得所需要的结果。

查询与视图的本质都是 SQL 语言中的 SELECT 查询语句，前者通过对话框、设计器创建，后者则使用命令建立。

5.1　创建查询

在 VFP 中，查询是一个扩展名为.QPR 的文件——查询文件。在很多情况下都需要建立查询，例如为报表组织信息、即时回答问题或者查看数据中的相关子集。无论目的是什么，建立查询的基本过程是相同的。

5.1.1　启动“查询设计器”

1．启动“查询设计器”

从“项目管理器”或“文件”菜单中，都可以启动“查询设计器”。启动“查询设计器”的步骤如下：

① 从“文件”菜单中选择“新建”命令，或者单击常用工具栏上的“新建”按钮。

② 在“新建”对话框中，选中“查询”单选钮，然后单击“新建文件”按钮。

③ 在创建新查询时，系统打开“添加表或视图”对话框，提示从当前数据库或自由表中选择表或视图，如图 5-1 所示。

依次选择所需要的表或视图，单击“添加”按钮，最后单击“关闭”按钮，VFP 将显示“查询设计器”窗口，如图 5-2 所示。

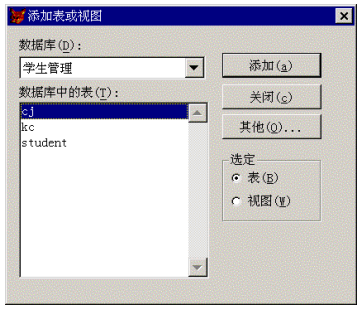


图5-1　添加表或视图

图5-2　“查询设计器”窗口

2．添加和移去表

若要添加和移去表：

① 选择当前表，再选择“查询设计器”工具栏上的“移去表”按钮。

② 从“查询设计器”工具栏上选择“添加表”按钮，再选择想要添加的表或视图。

5.1.2 定义结果

打开“查询设计器”，并选择了包含想要信息的表或视图后，就可定义查询的输出结果。首先选择所需的字段，可以设置选择字段的显示顺序和设置过滤器筛选要显示的记录，用以定义输出结果。

1．选择所需字段

在运行查询之前，必须选择表或视图，并选择要包括在查询结果中的字段。在某些情况下，可能会使用表或视图中的所有字段。但在另一些情况下，也许只想使查询与选定的部分字段相关，比如要加到报表中的字段。

使用“查询设计器”下半部窗格中的“字段”选项卡来选定需要包含在查询结果中的字段或字段表达式，如图 5-3 所示选择字段的方法是先选定字段名，然后单击“添加”按钮。或者，将字段名拖到“选定字段”框中。

图5-3 添加字段

说明：

① 选择输出全部字段：可使用名字或通配符选择全部字段。

如果使用名字选择字段，查询中要包含完整的字段名。此时若向表中添加字段后，再运行查询，则输出结果不包含新字段名。

如果使用通配符，通配符可以包含当前查询的表中的全部字段。如果创建查询后，表结构改变了，新字段也会出现在查询结果中。

若要在查询中一次添加所有可用的字段，可以单击“全部添加”按钮，按名字添加字段。或者，将表顶部的*号（通配符）拖到“选定字段”框中。

② 显示字段的别名：可使查询结果易于阅读和理解，方法是在输出结果字段添加说明标题。例如，您可在结果列的顶部显示“平均成绩”来代替字段名或表达式 AVG（成绩）。

若要给字段添加别名，在“函数和表达式”框中，键入字段名，接着键入“AS”和别名，例如 AVG（成绩） AS 平均成绩。选择“添加”在“选定字段”框中放置带有别名的字段。

2．设置输出字段的次序

在“字段”选项卡中，字段的出现顺序决定了查询输出中信息列的顺序。

若要改变查询输出的列顺序，只需上、下拖动位于字段名左侧的移动框即可。

如果想改变信息行的排序次序，请使用“排序依据”选项卡。

3．选定所需的记录

选定想要查找的记录是决定查询结果的关键。用“查询设计器”中的“筛选”选项卡，可以构造一个带有 WHERE 子句的选择语句，用来决定想要搜索并检索的记录。

可能需要查找一个特定的数据子集，并将其包含在报表或其他输出中。例如，所有考试成绩及格的学生。若只想查看所需的记录，可以输入一个值或值的范围与记录进行比较。使用“筛选”选项卡可以确定用于选择记录的字段、选择比较准则以及输入与该字段进行比较的示例值。指定过滤器的步骤如下：

① 从“字段名”列表中选定用于选择记录的字段。通用字段和备注字段不能用于过滤器中。

② 从“条件”列表中选择比较的类型。

③ 在“实例”文本框中，输入比较条件，如图 5-4 所示。

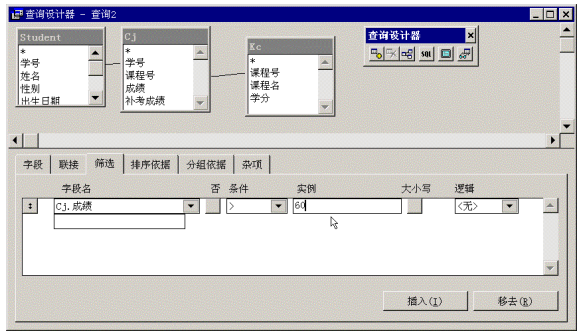


图5-4　在“筛选”选项卡中定义查询结果的条件

在输入比较条件时要注意：

- 仅当字符串与查询的表中字段名相同时，用引号括起字符串。否则，无需用引号将字符串引起来。
- 日期也不必用花括号括起来。
- 逻辑值的前后必须使用句点号，如（.T.)。
- 如果输入查询中表的字段名，VFP 就将它识别为一个字段。
- 在搜索字符型数据时，如果想忽略大小写匹配，请选择“大小写”下面的按钮。

④ 若想对逻辑操作符的含义取反，请选择“否”下面的按钮。若要更进一步搜索，可在过滤器选项卡中添加更多的筛选项。如果查询中使用了多个表或视图，按选取的联接类型扩充所选择的记录。

5.1.3　排序与分组

定义查询输出后，可组织出现在结果中的记录，方法是对输出字段排序和分组。也可筛选出现在结果中的分组。

1．排序查询结果

排序决定了查询输出结果中记录或行的先后顺序。例如，按考试成绩和学号对记录排序。利用“排序依据”选项卡设置查询的排序次序，排序次序决定了查询输出中记录或行的排列

顺序。首先，从“选定字段”框中选定要使用的字段，并把它们移到“排序条件”框中，然后根据查询结果中所需的顺序排列这些字段。

① 设置排序条件：在“选定字段”框中选定字段名，单击“添加”按钮，如图 5-5 所示。

图5-5 “排序依据”选项卡

② 排序顺序：字段在“排序条件”框中的次序决定了查询结果排序时的重要性次序，第一个字段决定了主排序次序。例如，在“排序条件”框中的第一个字段是成绩，第二个字段为学号，查询结果将首先以成绩进行排序，如果成绩表中有一个以上的记录具有同样的成绩字段值，这些记录则再以学号进行排序。

为了调整排序字段，可在“排序条件”框中，将字段左侧的按钮拖到相应的位置上。通过设置“排序选项”区域中的按钮，可以确定升序或降序的排序次序。在“筛选”选项卡的“选定字段”框中，每一个排序字段都带有一个上箭头或下箭头，该箭头表示按此字段排序时，是升序排序还是降序排序。

③ 移去排序条件：选定一个或多个想要移去的字段，单击“移去”按钮。

2．分组查询结果

所谓分组就是将一组类似的记录压缩成一个结果记录，这样就可以完成基于一组记录的计算。例如，若想得到某一学生的所有课程的平均成绩，不用单独查看所有的记录，可以把所有记录合成一个记录，来获得所有成绩的平均值。首先在“字段”选项卡中，把表达式 AVG（成绩）添加到查询输出中，如图 5-3 所示，然后利用“分组依据”选项卡，根据学号分组，输出结果显示了每个学生的平均成绩。

若要控制记录的分组，可使用“查询设计器”中的“分组依据”选项卡。分组在与某些累计函数联合使用时效果最好，诸如 SUM、COUNT、AVG 等。设置分组选项的步骤如下：

① 在“字段”选项卡中，在“函数和表达式”框中键入表达式。或者，选择要使用“表达式生成器”的对话框，在“函数和表达式”框中键入表达式。

② 选择“添加”按钮，在“选定字段”框中放置表达式。

③ 在“分组依据”选项卡中，加入分组结果依据的表达式。也可以在已分组的结果上设置选定条件，如图 5-6 所示。

3．选择分组

若要对已进行过分组或压缩的记录而不是对单个记录设置过滤器，可在“分组依据”选项卡中选定“满足条件”按钮。可使用字段名、字段名中的函数，或者“字段名”框中的另外的表达式。可以使用按学号显示平均成绩的查询，进一步利用“满足条件”按钮，限制查

询的结果为那些平均分数大于 85 分的学生。为分组设置条件的步骤如下：

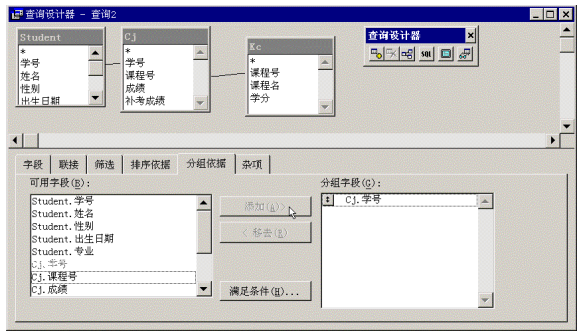


图5-6　设置“分组依据”选项

① 在“分组依据”选项卡上，选择“满足条件”按钮。
② 在“满足条件”对话框中，选定一个函数，并在“字段名”域中选定字段名。
③ 单击“确定”按钮。

5.1.4　输出查询

1．定向输出查询结果

可以把查询结果输出到不同的目的地。如果没有选定输出目的地，查询结果将显示在“浏览”窗口中。从“查询”菜单中选择“查询去向”，或在“查询设计器”工具栏中选择“查询去向”按钮，此时将显示一个“查询去向”对话框，可以在其中选择将查询结果送往何处。选择查询结果去向的步骤如下：

① 从“查询设计器”工具栏或“查询”菜单中选择“查询去向”。
② 在“查询去向”对话框中选择输出去向，并填写所需的其他选项，如图 5-7 所示。

图5-7　选择查询去向

表 5-1 列出查询输出去向的说明。

表 5-1　查询输出的去向

选　项	说　明
浏览	在“浏览”窗口中显示查询结果
临时表	将查询结果存储在一个命名的临时只读表中
表	使查询结果保存为一个命名的表

（续）

选　项	说　明
图形	使查询结果可用于 Microsoft Graph（Graph 是包含在 Visual FoxPro 中的一个独立的应用程序）
屏幕	在 Visual FoxPro 主窗口或当前活动输出窗口中显示查询结果
报表	将输出送到一个报表文件（.FRX）
标签	将输出送到一个标签文件（.LBX）

许多选项都有一些可以影响输出结果的附加选择。例如，“报表”选项可以打开报表文件，并在打印之前定制报表，也可以选用“报表向导”帮助自己创建报表。

2．保存查询

选择文件菜单中的“保存”命令，或是单击工具栏中的“保存”按钮，都可以将“查询”保存为扩展名.QPR 的查询文件中。首次保存文件将打开“另存为”对话框，以选择保存的文件名和位置。

3．运行查询

在完成了查询设计并指定了输出目的地后，可以用“运行”按钮启动该查询。VFP 执行用“查询设计器”产生的 SELECT-SQL 语句，并把输出结果送到指定的目的地。若尚未选定输出目的地，结果将显示在“浏览”窗口中。上述操作所得的查询显示如图 5-8 所示。

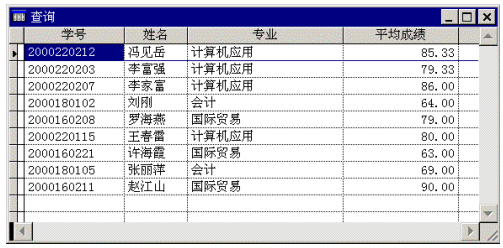


图5-8　浏览查询的结果

若要运行查询，首先在“项目管理器”中选定查询的名称，然后单击“运行”按钮。

如果想使结果输出到不同的目的文件，可将结果定向输出到表单、表、报表或其他目的文件。如果想了解 SQL 语句如何，可查看生成的 SQL 语句。

还可以在“命令”窗口或程序中运行生成的查询文件，运行查询文件的命令格式如下：

DO〈查询文件名〉.QPR

其中，扩展名.QPR 不能缺省。

5.1.5　查询的 SQL 语句

如果要确认查询的定义是否正确，可查看使用“查询设计器”生成的 SQL 语句。也可添加注释来说明查询的目的，添加的注释将出现在 SQL 窗口里。

1．查看 SQL 语句

在建立查询时，从“查询”菜单中选择“查看 SQL”，或从工具栏上选择“SQL”按钮，可以查看查询生成的 SQL 语句。SQL 语句显示在一个只读窗口中，可以复制此窗口中的文本，并将其粘贴到“命令”窗口或加入到程序中。

复制到“命令”窗口中的 SQL 语句可以立即执行，加入到程序中的 SQL 语句则在程序

执行的过程中生效。

还可以将 SQL 语句粘贴到报表或一些控件的数据源中。

2．SQL 语句分析

按照上述步骤生成查询的 SQL 语句为：

```
SELECT Student.学号, Student.姓名, Student.专业,;
   AVG（Cj.成绩） as 平均成绩;                          && 这是由第一步产生的
FROM  学生管理!student INNER JOIN 学生管理!cj;
      INNER JOIN 学生管理!kc ;
   ON  Kc.课程号 = Cj.课程号 ;                          && 这是数据库的连接关系
   ON  Student.学号 = Cj.学号;
 WHERE Cj.成绩 > 60;                                  && 筛选条件
 GROUP BY Cj.学号;                                    && 分组依据
 ORDER BY Student.姓名, 4 DESC                        && 排序依据
```

5.2 定制查询

利用“查询设计器”中其他可用的选项，很容易定制进一步查询。可使用过滤器扩充或缩小搜索，也可以添加表达式计算字段中的数据。

5.2.1 精确搜索

可能需要对查询所返回的结果做更多的控制，例如，查找满足多个条件的记录，某课程不及格的学生，或者搜索满足两个条件之一的记录。这时，就需要在“筛选”选项卡中加进更多的语句。如果在“筛选”选项卡中连续输入选择条件表达式，那么这些表达式自动以逻辑“与”（AND）的方式组合起来，如果想使待查找的记录满足二个以上条件中的任意一个时，可以使用“添加‘或’”按钮在这些表达式中间插入逻辑“或”（OR）操作符。

1．缩小搜索

如果想使查询检索同时满足一个以上条件的记录，只需在“筛选”选项卡中的不同行上列出这些条件，这一系列条件自动以“与”（AND）的方式组合起来，因此只有满足所有这些条件的记录才会被检索到。例如，假设搜索成绩及格的女同学，则可以在不同的行上输入两个搜索条件。

若要设置“与”（AND）条件，可以在“筛选”选项卡中键入筛选条件，在“逻辑”列中选择“AND”，如图 5-9 所示。

2．扩充搜索

若要使查询检索到的记录满足一系列选定条件中的任意一个时，可以在这些选择条件中间插入“或”（OR）操作符将这些条件组合起来。

若要设置“或”（OR）条件，可以选择一个筛选条件，再在“逻辑”列中选择“OR”。

3．组合条件

可以把“与”（AND）和“或”（OR）条件组合起来以选择特定的记录集。

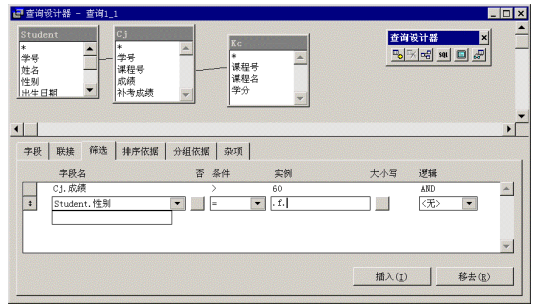


图5-9　组合两个过滤器

4．在查询中删除重复记录

重复记录是指其所有字段值均相同的记录。如果想把查询结果中的重复记录去掉，只需选中“杂项”选项卡中的“无重复记录”框。否则，应确认“无重复记录”框已被清除，如图 5-10 所示。如果选中了“无重复记录”框，在 SELECT 命令的 SELECT 部分中，字段前会加上 DISTINCT。

图5-10　“杂项”选项卡

5．查询一定数目或一定百分比的极值记录

可使查询返回包含指定数目或指定百分比的特定字段的记录。例如，查询可显示含 10 个指定字段最大值或最小值的记录，或者显示含有 10%的指定字段最大值或最小值的记录。

利用“杂项”选项卡的顶端设置，可设置一定数目或一定百分比的记录。若要设置是否选取最大值或最小值，可设置查询的排序顺序。降序可查看最大值记录，升序可查看最小值记录。检索一定数目或一定百分比的极值记录的步骤如下：

① 在“排序依据”选项卡中，选择要检索其极值的字段，接着选取“降序”显示最大值或“升序”显示最小值。如果还要按其他字段排序，可按列表顺序将其放在极值字段的后面。

② 在“杂项”选项卡中，在“记录个数”框中，键入想要检索的最大值或最小值的数目。若要显示百分比，请选中“百分比”复选框。

③ 如果不希望数目或百分比中含有重复的记录，请选中“无重复记录”复选框。

5.2.2　在查询输出中添加表达式

使用“字段”选项卡底部的方框，可以在查询输出中加入函数和表达式。

1．在结果中添加表达式

可以显示列表来查看可用的函数，或者直接向框中键入表达式。如果希望字段名中包含表达式，可以添加别名。

可直接在对话框中键入一个表达式或使用“字段”选项卡中的“表达式生成器”。在查询输出中添加表达式的步骤如下：

① 在“字段”选项卡的“函数和表达式”框中键入表达式。或者选择对话按钮使用“表达式生成器”，再在“表达式”框中键入一个表达式。例如，要使查询结果包括别名为“最高分数”的“成绩”，则键入 MAX（成绩） AS 最高分数

② 选择“添加”按钮，在“选定字段”框中键入表达式。计算机将忽略 null 值。

2．用表达式筛选

不同于简单搜索与一个或多个字段相匹配的记录，使用一个表达式可以组合两个字段，或基于一个字段执行某计算并且搜索匹配该组合或计算字段的记录。

可直接在示例框中键入表达式。如需帮助，可使用“表达式生成器”，“表达式生成器”可从字段选项卡的“函数和表达式”框旁边的对话按钮中得到。

5.3 创建视图

在应用程序中，若要创建自定义并且可更新的数据集合，可以使用视图。视图兼有表和查询的特点：与查询相类似的地方是，可以用来从一个或多个相关联的表中提取有用信息；与表相类似的地方是，可以用来更新其中的信息，并将更新结果永久保存在磁盘上。可以用视图使数据暂时从数据库中分离成为自由数据，以便在主系统之外收集和修改数据。

由于视图和查询有很多类似处，创建视图与创建查询的步骤相似：选择要包含在视图中的表和字段，指定用来联接表的连接条件，指定过滤器选择指定的记录。

与查询不同的是，视图可选择如何将在视图中所做的数据修改传给原始文件，或建立视图的基表。另外，视图是数据库中的一个特有功能，只有在包含视图的数据库打开时，才能使用视图。

可以创建两种类型的视图——本地视图和远程视图。远程视图使用远程 SQL 语法从远程 ODBC 数据源表中选择信息，本地视图使用 VFP SQL 语法从视图或表中选择信息。本书只介绍本地视图的创建与使用。

可以使用视图设计器或 CREATE SQL VIEW 命令创建本地视图。

5.3.1 启动“视图设计器”

本地表包括本地 VFP 表、任何使用.DBF 格式的表和存储在本地服务器上的表。若要使用“视图设计器”来创建本地表的视图，首先应创建或打开一个数据库。

1．使用菜单启动“视图设计器”

选择“文件”菜单中的“新建”命令，或是单击工具栏中的“新建”按钮，打开“新建”对话框。选择“视图”选项，并单击“新建文件”按钮，打开“添加表或视图”对话框（如图 5-1 所示）。

如果对话框中的“视图”选项不可用，说明还没有打开数据库。

在“添加表或视图”对话框中，选定想使用的表或视图，再单击“添加”按钮，将表或视图添加到视图中。

最后，单击“关闭”按钮即可打开“视图设计器”，如图 5-11 所示。

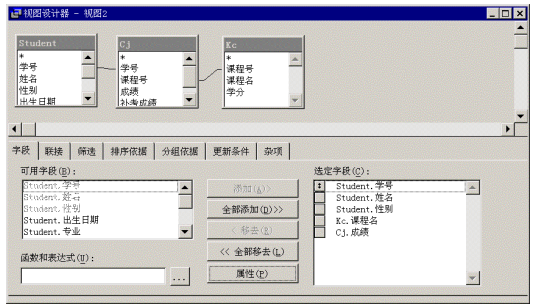


图5-11　视图设计器

2．在项目管理器中启动“视图设计器”

展开“项目管理器”中数据库名称旁边的加号“+”，“数据”选项卡上将显示出数据库中的所有组件。启动“视图设计器”的步骤如下：

① 从“项目管理器”中选定一个数据库。

② 单击“数据库”符号旁的加号“+”。

③ 在“数据库”下，选定“本地视图”，然后单击“新建”按钮。

④ 打开“新建本地视图”对话框，选择“新建视图”按钮。

⑤ 在“添加表或视图”对话框中，选定想使用的表或视图，再单击“添加”按钮，如图 5-1 所示。

⑥ 选择视图中想要的视图后，单击“关闭”按钮。出现“视图设计器”对话框，显示选定的表或视图，如图 5-11 所示。

3．使用命令启动“视图设计器”

打开一个数据库后，在命令窗口输入以下命令也可以启动“视图设计器”：

```
CREATE VIEW
```

5.3.2　视图设计器

查询设计器与视图设计器的使用方法几乎完全一样。主要有如下几点不同：

① 由于视图是可以更新数据的，因此视图设计器多了一个“更新条件”选项卡，它可以控制数据更新的条件。

② 查询设计器的结果是将查询保存为查询文件（.QPR 扩展名）；而视图设计器的结果不是文件，只是一个保存在数据库中的视图定义，包括视图中的表名、字段名以及它们的属性设置。

③ 视图设计器中没有“查询去处”的问题。

5.3.3　使用“视图设计器”修改视图

如果要修改视图，首先打开包含该视图的数据库，在命令窗口输入以下命令可以启动“视

图设计器”：

MODIFY VIEW〈视图名〉

其中，〈视图名〉即待修改的视图。此时，视图设计器中包含该视图。

5.4 定制视图

可以在视图中包含表达式、设置提示输入值，也可以设置高级选项来协调与服务器交换数据的方式，可用以下方法来定制视图。

5.4.1 控制字段显示和数据输入

因为视图是数据库的一部分，可利用数据库提供的表中字段的一些相同属性。例如，可分配标题，输入注释，或设置控制数据输入的有效性规则。控制字段显示和数据输入的步骤如下：

① 在“视图设计器”中创建或修改视图。

② 在“字段”选项卡中，单击“属性”按钮，打开“视图字段属性”对话框，如图 5-12 所示。

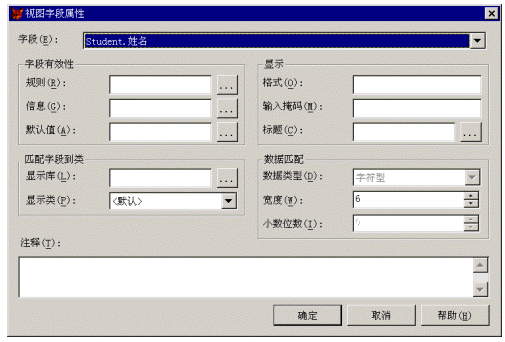


图5-12 “视图字段属性”对话框

③ 在“视图字段属性”对话框中选定字段，然后可以输入有效性规则、显示内容及字段类型设置。有关字段有效性规则、显示和映射的内容，与处理表相似。

5.4.2 参数提示

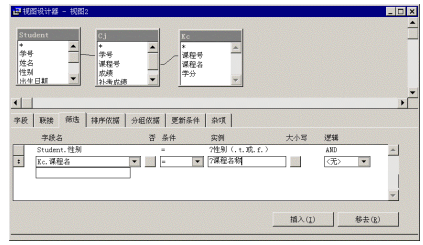


图 5-13 作为视图过滤器的一部分所键入的参数值

可设置视图对完成查询所输入的值进行提示。例如，假设要创建查询，寻找指定班级的学生。要做到这项任务，需要在班级字段中定义一个过滤器并且指定一个参数作为过滤器的实例。参数名可以是任意字母、数字和单引号的组合。对视图设置参数的步骤如下：

① 在“视图设计器”中，添加新过滤器或从“筛选”选项卡中选择存在的过滤器。

② 在“实例”框中，键入一个问号“?”和参数名，如图 5-13 所示。

当使用视图时，将显示一个信息框提示输入作为包含在过滤器中的值，如图 5-14 所示。

图5-14　显示一个信息框

5.4.3　控制更新方法

若要控制关键字段的信息怎样在服务器上更新，选择使用更新中的选项。当记录中的关键字更新时，这些选项决定发送到服务器或源表中的更新语句使用什么 SQL 命令。

1．设置关键字段

当在“视图设计器”中首次打开一个表时，“更新条件”选项卡会显示表中哪些字段被定义为关键字段。VFP 用这些关键字段来惟一地标识那些已在本地修改过的远程表中的更新记录。若要设置关键字段，在“更新条件”选项卡中，单击字段名旁边的“关键”列，如图 5-15 所示。

图5-15　在“更新条件”选项卡中设置关键字段

如果已经改变了关键字段，而又想把它们恢复到源表中的初始设置，选择“重置关键字”。VFP 会检查远程表并利用这些表中的关键字段。

2．更新指定字段

可以指定任一给定表中仅有某些字段允许更新。若使表中的任何字段是可更新的，在表中必须有已定义的关键字段。如果字段未标注为可更新的，用户可以在表单中或浏览窗口中修改这些字段，但修改的值不会送到远程表中。若要使字段为可更新的，可在“更新条件”选项卡中，单击字段名旁边的可更新列（笔形），如图 5-15 所示。

3．更新所有字段

如果想使表中的所有字段可更新，可以将表中的所有字段设置成可更新的。

若要使所有字段可更新，在“更新条件”选项卡中，选择“全部更新”。

若要使用“全部更新”，在表中必须有已定义的关键字段。“全部更新”不影响关键字段。

4．控制如何检查更新冲突

如果在一个多用户环境中工作，服务器上的数据也可以被别的用户访问，也许别的用户也在试图更新远程服务器上的记录，为了让 VFP 检查用视图操作的数据在更新之前是否被别的用户修改过，可使用“更新条件”选项卡上的选项。在“更新条件”选项卡中，“SQL WHERE 子句包括”框中的选项可以帮助管理遇到多用户访问同一数据时应如何更新记录。在允许更新之前，VFP 先检查远程数据源表中的指定字段，看看它们在记录被提取到视图中后有没有改变，如果数据源中的这些记录被修改，就不允许更新操作。

在如图 5-16 所示的“更新条件”选项卡中设置 SQL WHERE 子句，这些选项决定哪些字段包含在 UPDATE 或 DELETE 语句的 WHERE 子句中，VFP 正是利用这些语句将在视图中修改或删除的记录发送到远程数据源或源表中，WHERE 子句就是用来检查自从提取记录用于视图中后，服务器上的数据是否已被改变，如表 5-2 所示。

表 5-2　使更新失败选择的 SQL WHERE 选项

选　项	说　明
关键字段	当源表中的关键字段被改变时，使更新失败
关键字和可更新字段	当远程表中任何标记为可更新的字段被改变时，使更新失败
关键字和已修改字段	当在本地改变的任一字段在源表中已被改变时，使更新失败
关键字和时间戳	当远程表上记录的时间戳在首次检索之后被改变时，使更新失败（仅当远程表有时间戳列时有效）

5．向表发送更新数据

在“视图设计器”中，“更新条件”选项卡可以控制把对远程数据的修改（更新、删除、插入）回送到远程数据源中的方式，也可以打开和关闭对表中指定字段的更新，并设置适合服务器的 SQL 更新方法。

如果想使在表的本地版本上的修改能回送到源表中，就需要设置“发送 SQL 更新”选项，必须至少设置一个关键字段来使用这个选项。如果选择的表中有一个主关键字段并且已在字段选项卡选中，则“视图设计器”自动使用表中的该主关键字段作为视图的关键字段。

若要允许源表的更新，可在“更新条件”选项卡中，设置“发送 SQL 更新”选项。

6．保存视图

选择文件菜单中的“保存”项，或是单击工具栏中的“保存”按钮，都可以将“视图”保存在数据库中。首次保存文件将打开“保存”对话框，以选择保存的视图名称。如将上述步骤创建的视图保存为“视图 0”，如图 5-16 所示。

图5-16　保存视图

5.5　使用视图

视图是操作表的一种手段，通过视图可以查询表，也可以更新表。

5.5.1 视图处理

视图建立之后，不但可以用它来显示和更新数据，而且还可以通过调整它的属性来提高性能。处理视图类似于处理表，具有以下一些功能特点：

- 使用 USE 命令并指定视图名来打开一个视图。
- 使用 USE 命令关闭视图。
- 在“浏览”窗口中，显示视图记录。
- 在“数据工作期”窗口中显示打开的视图。
- 在文本框、表格控件、表单或报表中使用视图作为数据源。

5.5.2 视图使用举例

1．使用命令

可以借助 VFP 语言来使用视图。下面的代码在浏览窗口中显示“视图 0”：

```
OPEN DATABASE 学生管理
USE 视图 0
BROWSE
```

在两次弹出的“视图参数”对话框中依次输入“.t.”和“高等数学”，如图 5-17 所示。在“浏览”窗口中将只看到所有男生的“高等数学”课程的记录内容，如图 5-18 所示。

图5-17　显示一个信息框

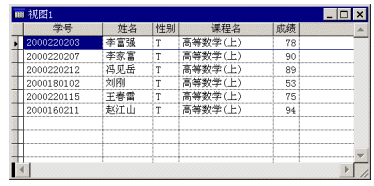


图5-18　“浏览”窗口

可以在浏览窗口中对视图的指定字段（如成绩）进行更新。

2．通过“项目管理器”浏览视图

首先创建一个项目，并将“学生管理”数据库添加到项目中。在“项目管理器”中，选择数据库，可以看到视图和表一样，都包含在数据库中，如图 5-19 所示。然后选择视图，单击“浏览”按钮，即可在“浏览”窗口中显示视图，并可对视图的指定字段进行操作。

图 5-19　项目管理器

一个视图在使用时，将作为临时表在自己的工作区中打开。如果此视图基于本地表，则在 VFP 的另一个工作区中可同时打开基表。视图的基表是指由 SELECT – SQL 语句访问的表，此语句在创建视图时包含在 CREATE SQL VIEW 命令中。在前面的示例中，使用“视图 0”的同时，表 Student、cj、kc 也自动打开。

5.6 习题

一、选择题

1．下面关于查询描述正确的是（　　）。

A）不能根据自由表建立查询　　B）只能根据自由表建立查询

C）只能根据数据库表建立查询　　D）可以根据数据库表和自由表建立查询

2．下列关于视图的正确描述是（　　）。

A）可以根据自由表建立视图　　B）可以根据查询建立视图

C）可以根据数据库表建立视图　　D）可以根据数据库表和自由表建立视图

3．视图设计器中含有、但查询设计器中却没有的选项卡是（　　）。

A）筛选　　B）排序依据　　C）分组依据　　D）更新条件

4．下面关于查询描述正确的是（　　）。

A）可以使用 CREATE VIEW 打开查询设计器

B）使用查询设计器可以生成所有的 SQL 查询语句

C）使用查询设计器生成的 SQL 语句存盘后将放在扩展名为 qpr 的文件中

D）使用 DO 命令执行查询时，可以不带扩展名

5．查询设计器包括的选项卡有（　　）。

A）字段、筛选、排序依据　　B）字段、条件、分组依据

C）条件、排序依据、分组依据　　D）条件、筛选、杂项

二、填空题

1．查询设计器的筛选选项卡用来指定查询的_____。

2．通过视图，不仅可以查询数据库表，还可以_____数据库表。

三、上机题

1．建立一个查询，查询客户名为“三益贸易公司”的订购单明细（order_detail）记录（将结果先按“订单号”升序排列，同一订单的再按“单价”降序排列），将结果存储到 results5_1 表中（表结构与 order_detail 表结构相同）。

2．建立一个查询，查询目前有订购单的客户信息（即有对应的 order_list 记录的 customer 表中的记录），同时要求按客户号升序排序，将结果存储到 results5_2 表中（表结构与 customer 表结构相同）。

3．建立一个查询，查询所有订购单的订单号、订购日期、器件号、器件名和总金额（按订单号升序），并将结果存储到 results5_3 表中（其中订单号、订购日期、总金额取自 order_list 表，器件号、器件名取自 order_detail 表）。

4．建立一个查询，按总金额降序列出所有客户的客户号、客户名及其订单号和总金额，并将结果存储到 results5_4 表中（其中客户号、客户名取自 customer 表，订单号、总金额取自 order_list 表）。

5．对表 order_detail 建立查询，把“订单号”尾部字母相同并且订货相同（“器件号”相同）的订单合并为一张订单，新的“订单号”就取原来的尾部字母，“单价”取最低价，“数量”取合计；结果先按新的“订单号”升序排序，再按“器件号”升序排序；最终记录的处

理结果保存在表 results5_5 中。

6．打开数据库学生管理.dbc，使用视图设计器创建一个名为 sview5_6 的视图，该视图的 SELECT 语句完成查询：选课门数是 2 门以上（不包括 2 门）的每个学生的学号、姓名、平均成绩、最低分和选课门数，并按“平均成绩”降序排序。

7．完成以下基本操作题：

（1）创建一个新的项目“salary_p”。

（2）在新建立的项目“salary_p”中创建数据库“salary_db”。

（3）在“salary_db”数据库中建立表 salary，表中数据见表 5-3。

表 5-3　工资表 salary.dbf

部门号	雇员号	姓　名	工　资	补　贴	奖　励	医疗统筹	失业保险	银行账号
01	0101	王万程	2580	300	200	50	10	20020101
01	0102	王旭	2500	300	200	50	10	20020102
01	0103	汪涌涛	3000	300	200	50	10	20020103
01	0104	李迎新	2700	300	200	50	10	20020104
02	0201	李现峰	2150	300	300	50	10	20020201
02	0202	李北红	2350	300	200	50	10	20020202
02	0203	刘永	2500	300	400	50	10	20020203
02	0204	庄喜盈	2100	300	200	50	10	20020204
02	0205	杨志刚	3000	300	380	50	10	20020205
03	0301	杨昆	2050	300	200	50	10	20020301
03	0302	张启训	2350	300	500	50	10	20020302
03	0303	张翠芳	2600	300	300	50	10	20020303
04	0401	陈亚峰	2600	350	700	50	10	20020401
04	0402	陈涛	2150	400	500	50	10	20020402
04	0403	史国强	3000	500	600	50	10	20020403
04	0404	杜旭辉	2800	450	500	50	10	20020404
05	0501	王春丽	2100	300	250	50	10	20020501
05	0502	李丽	2350	300	200	50	10	20020502
05	0503	刘刚	2200	300	300	50	10	20020503
01	0105	冯见越	2320	300	200	50	10	20020105
02	0206	罗海燕	2200	300	200	50	10	20020206
01	0106	张立平	2150	300	300	50	10	20020106
05	0504	周九龙	2900	300	350	50	10	20020504
01	0107	周振兴	2120	300	200	50	10	20020107
01	0108	胡永萱	2100	300	200	50	10	20020108
01	0109	姜黎萍	2250	300	200	50	10	20020109
01	0110	梁栋	2200	300	200	50	10	20020110
03	0304	崔文涛	3000	300	800	50	10	20020304

其结构描述为 salary （部门号（C,2），雇员号（C,4），姓名（C,8），工资（N,4），补贴

（N,3），奖励（N,3），医疗统筹（N,3），失业保险（N,2），银行账号（C,10））

（4）在“ salary_db ”数据库中建立表 dept，表中数据见表 5-4。

表 5-4　部门表 dept.dbf

部　门　号	部　门　名
01	制造部
02	销售部
03	项目部
04	采购部
05	人事部

其结构描述为 dept（部门号（C,2），部门名（C,20））

（5）在 salary_db 数据库中为 dept 表创建一个主索引（升序），索引名和索引表达式均是“ 部门号 ”；为 salarys 表创建一个普通索引（升序），索引名和索引表达式均是“ 部门号 ”，再创建一个主索引（升序）、索引名和索引表达式均是“ 雇员号 ”。

（6）通过“ 部门号 ”字段建立 salarys 表和 dept 表间的永久联系。

（7）为以上建立的联系设置参照完整性约束：更新规则为“ 限制 ”，删除规则为“ 级联 ”，插入规则为“ 限制 ”。

8．在 salary_db 数据库中，使用视图设计器创建一个名称为 sview5_8 的视图，该视图的 SELECT 语句查询 salarys 表（雇员工资表）的部门号、雇员号、姓名、工资、补贴、奖励、失业保险、医疗统筹和实发工资，其中实发工资由工资、补贴和奖励三项相加，然后再减去失业保险和医疗统筹得出，结果按“ 部门号 ”降序排序。

第 6 章　关系数据库标准语言 SQL

查询与视图的本质都是一条 SQL 语句，但是查询与视图设计器只能生成简单的 SQL 语句，即只能完成简单的查询操作，对于复杂的查询就力不从心了。掌握 SQL 语法可以更加灵活地建立查询和视图。

6.1　SQL 简介

SQL 是 Structured Query Language 的缩写，即结构化查询语言。它是关系数据库的标准语言，来源于 20 世纪 70 年代 IBM 的一个被称为 SEQUEL(Structured English Query Language)的研究项目。20 世纪 80 年代，SQL 由 ANSI 进行了标准化，它包含了定义和操作数据的指令。由于它具有功能丰富、使用方式灵活、语言简洁易学等突出特点，在计算机界深受广大用户欢迎，许多数据库生产厂家都相继推出各自支持的 SQL 标准。1989 年 4 月，ISO 提出了具有完整性特征的 SQL，并将其定为国际标准，推荐它为标准关系数据库语言。1990 年，我国也颁布了《信息处理系统数据库语言 SQL》，将其定为中国国家标准。

6.1.1　SQL 语言的主要特点

SQL 语言的主要特点有：

① 一体化语言。SQL 提供了一系列完整的数据定义、数据查询、数据操纵和数据控制等方面的功能。用 SQL 可以实现数据库生命周期中的全部活动，包括简单地定义数据库和表的结构，实现表中数据的录入、修改、删除、查询和维护，数据库重构、数据库安全性控制等一系列操作要求。

② 高度非过程化。SQL 和其他数据操作语言不同，SQL 是一种非过程性语言，它不必一步步地告诉计算机“如何”去做，用户只需说明做什么操作，而不用说明怎样做，不必了解数据存储的格式及 SQL 命令的内部，就可以方便地对关系数据库进行操作。

③ 语言简洁。虽然 SQL 的功能很强大，但语法却很简单，只有为数不多的几条命令。表 6-1 给出了分类的命令动词，从该表可知，它的词汇很少。初学者经过短期的学习就可以使用 SQL 进行数据库的存取等操作，因此，易学易用是它的最大特点。

表 6-1　SQL 命令动词

SQL 功能	命 令 动 词
数据查询	SELECT
数据定义	CREATE、DROP、ALTER
数据操纵	INSERT、UPDATE、DELETE
数据控制	GRANT、REVOKE

④ 统一的语法结构对待不同的工作方式。SQL 语言可以直接在 VFP 的命令窗口以人机交互的方式使用，也可以嵌入到程序设计中以程序方式使用，比如，SQL 语言写在.PRG 文件中也能运行。在书写的时候，如果语句太长，可以用“;”号换行。现在很多数据库应用开

发工具都将 SQL 语言直接融入到自身的语言之中，使用起来更方便，VFP 就是如此。这些使用方式为用户提供了灵活的选择余地。此外，尽管 SQL 的使用方式不同，但 SQL 语言的语法基本是一致的。

6.1.2 SQL 语句的执行

SQL 语句可以在命令窗口中执行，也可以作为查询或视图（的内容）被使用，还可以在程序文件中被执行。

6.2 查询功能

数据库中最常见的操作是数据查询，也是 SQL 的核心。

6.2.1 SQL 语法

SQL 给出了简单而又丰富的查询语句形式，SQL 的查询命令也称作 SELECT 命令，它的基本形式由 SELECT-FROM-WHERE 查询块组成，多个查询块可以嵌套执行。SELECT-SQL 的语法格式如下：

```
SELECT [ALL | DISTINCT] [TOP〈表达式〉]
  [〈别名〉]〈Select 表达式〉[AS〈列名〉][，[〈别名〉]〈Select 表达式〉[AS〈列名〉] ...]
FROM [〈数据库名〉!]〈表名〉[[AS] Local_Alias]
  [[INNER | LEFT [OUTER] | RIGHT [OUTER] | FULL [OUTER]
    JOIN [〈数据库名〉!]〈表名〉[[AS] Local_Alias][ON〈联接条件〉]]
  [INTO〈查询结果〉| TO FILE〈文件名〉[ADDITIVE]
    | TO PRINTER [PROMPT] | TO SCREEN]
  [PREFERENCE PreferenceName]
  [NOCONSOLE]
  [PLAIN]
  [NOWAIT]
[WHERE〈联接条件 1〉[AND〈联接条件 2〉...][AND | OR〈筛选条件〉...]]
  [GROUP BY〈组表达式〉[,〈组表达式〉...]]
  [HAVING〈筛选条件〉]
  [UNION [ALL]〈SELECT 命令〉]
  [ORDER BY〈关键字表达式〉[ASC | DESC] [,〈关键字表达式〉[ASC | DESC] ...]]
```

说明：SELECT-SQL 命令的格式包括三个基本子句 SELECT 子句、FROM 子句、WHERE 子句，还包括操作子句 ORDER 子句、GROUP 子句、UNION 子句以及其他一些选项。

1．SELECT 子句

SELECT 子句用来指定查询结果中的数据。其中：

选项 ALL 表示选出的记录中包括重复记录，这是缺省值；DISTINCT 则表示选出的记录中不包括重复记录。

选项 TOP〈表达式〉表示在符合条件的记录中选取指定数量或百分比（〈表达式〉）的记录。

选项[〈别名〉] 〈Select 表达式〉 [AS 〈列名〉]中的别名是字段所在的表名；Select 表达式可以是字段名或字段表达式；列名用于指定输出时使用的列标题，可以不同于字段名。

〈Select 表达式〉用一个*号来表示时，指定所有的字段。

2．FROM 子句

用于指定查询的表与联接类型。其中：

JOIN 关键字用于联接其左右两个〈表名〉所指定的表。

INNER | LEFT[OUTER] | RIGHT[OUTER] | FULL[OUTER]选项指定两表联接时的联接类型，联接类型有 4 种，见表 6-2。其中的 OUTER 选项表示外部联接，即允许满足联接条件的记录，又允许不满足联接条件的记录。若省略 OUTER 选项，效果不变。

表 6-2　联接类型

联 接 类 型	意　义
Inner Join（内部联接）	只有满足联接条件的记录包含在结果中
Left Outer Join （左联接）	左表某记录与右表所有记录比较字段值，若有满足联接条件的，则产生一个真实值记录；若都有满足，则产生一个含.NULL 值的记录。直至右表所有记录都比较完
Right Outer Join（右联接）	右表某记录与左表所有记录比较字段值，若有满足联接条件的，则产生一个真实值记录，若都不满足，则产生一个含.NULL 值的记录。直至右表所有记录都比较完
Full Join （完全联接）	先按右联接比较字段值，再按左联接比较字段值。不列入重复记录

ON 选项用于指定联接条件。FORCE 选项表示严格按指定的联接条件来联接表，避免 VFP 因进行联接优化而降低查询速度。

INTO 与 TO 选项用于指定查询结果的输出去向，默认查询结果显示在浏览窗口中。INTO 选项中的〈查询结果〉有 3 种，见表 6-3。

表 6-3　查询结果

目　标	输 出 形 式
ARRAY<数组>	查询结果输出到数组
CURSOR <临时表>	查询结果输出到临时表
TABLE | DBF <表名>	查询结果输出到表

TO FILE 选项表示输出到指定的文本文件，并取代原文件内容。ADDITIVE 表示只添加新数据，不清除原文件的内容。TO PRINTER 选项表示输出到打印机，PROMPT 表示打印前先显示打印确认框。TO SCREEN 表示输出到屏幕。

PLAIN 选项表示输出时省略字段名。NOWAIT 选项显示浏览窗口后程序继续往下执行。

3．WHERE 子句

用来指定查询的条件。其中的〈联接条件〉指定一个字段，该字段连接 FROM 子句中的表。如果查询中包含不止一个表，就应该为第一个表后的每一个表指定连接条件。

4．其他子句和选项

GROUP BY 子句：对记录按〈组表达式〉值分组，常用于分组统计。

HAVING 子句：当含有 GROUP BY 子句时，HAVING 子句可用作记录查询的限制条件；无 GROUP BY 子句时，HAVING 子句的作用如同 WHERE 子句。

UNION 子句：可以用 UNION 子句嵌入另一个 SELECT-SQL 命令，使这两个命令的查询

结果合并输出，但输出字段的类型和宽度必须一致。UNION 子句默认组合结果中排除重复行，使用 ALL 则允许包含重复行。

ORDER BY 子句：指定查询结果中记录按〈关键字表达式〉排序，默认升序。选项 ASC 表示升序，DESE 表示降序。

SELECT 查询命令的使用非常灵活，用它可以构造各种各样的查询。本章将通过大量的实例来介绍 SELECT 命令的使用。

6.2.2 简单查询

简单查询只含有基本子句，可有简单的查询条件。

【例 6-1】下述代码可以查询学生表中的所有字段：

```
SELECT * FROM student
```

代码执行结果如图 6-1 所示。

图6-1　学生表中的所有字段

【例 6-2】下述代码可以从学生表中检索所有专业名称：

```
SELECT 专业 FROM student
```

代码执行结果如图 6-2 所示。

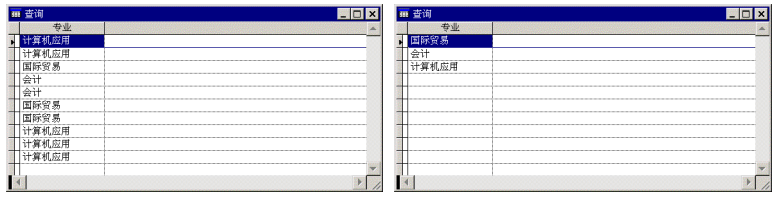


图6-2　所有专业名称

可以看到在结果中有重复值，如果要去掉重复值只需指定 DISTINCT 短语：

```
SELECT DISTINCT 专业 FROM student
```

【例 6-3】下述代码可以检索成绩大于 85 分的学生号：

```
SELECT 学号 FROM cj WHERE 成绩 > 85
```

代码执行结果如图 6-3 所示。

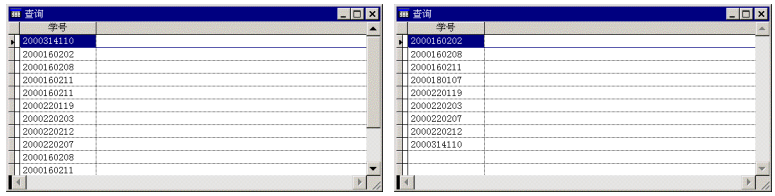


图6-3　成绩大于85分的学生号

去掉重复值，可以得到至少有一门课程成绩大于 85 的学号：

```
SELECT DISTINCT 学号 FROM cj WHERE 成绩 > 85
```

说明：在这个例子中，显然只对惟一的学生号感兴趣，而在【例 6-1】中也许对所有记录都感兴趣，因此不管是否有重复值。决定是否消除重复值是有一定实际意义的，也是 SQL 的重要一面。再回到【例 6-3】，假设对所有学生的平均成绩感兴趣，那么指定 DISTINCT 显然是错误的，至于什么时候用，视具体情况而定。

【例 6-4】下述代码可以查询哪些学生至少有一门课程成绩大于 85：

这里所要求检索的信息分别出自 student（姓名字段）和 cj（成绩字段）两个表，这样的检索肯定是基于多个表的，此类查询一般用联接查询来实现。

```
SELECT DISTINCT 姓名 FROM student, cj WHERE 成绩 > 85 and student.学号 = cj.学号
```

代码执行结果如图 6-4 所示。

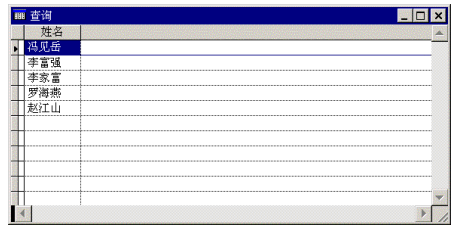


图6-4　查询哪些学生至少有一门课程成绩大于85

说明：

① 如果在检索命令的 FROM 之后有两个表，那么这两个表之间肯定有一种联系。如 student 表和 cj 表之间都有学号这个字段，否则无法构成检索表达式。

② “student.学号 = cj.学号”是联接条件。

③ 当 FROM 之后的多个关系中含有相同的字段名时，这时必须用表别名前缀直接指明字段所属的表，如 student.学号，“.”前面是表别名，后面是字段名。

④ 本例还可以用联接短语来实现：

```
SELECT DISTINCT 姓名 FROM student JOIN cj ON student.学号 = cj.学号;
    WHERE 成绩 > 85
```

参见 6.2.6 超联接查询。

6.2.3　几个特殊运算符

在 SQL 语句中，WHERE 子句后面的联接条件除了使用 VFP 语言中的关系表达式以及逻

辑表达式外，还使用几个特殊运算符：

① [NOT] IN：表示[不]在…之中。

② [NOT] BETWEEN…AND…：表示[不]在…之间。

③ [NOT] LIKE：表示[不]与…匹配。

下面以几个实例来说明。

【例 6-5】在 student 表中检索所有国际贸易专业的学生信息，不要其他的学生信息。

这是一个字符串匹配的查询，LIKE 运算符专门对字符型数据进行字符串比较。

```
SELECT * FROM student WHERE 专业 LIKE "国际贸易"
```

代码执行结果如图 6-5 所示。

图6-5　国际贸易专业的学生信息与非国际贸易专业的学生信息

说明：可以使用 NOT 运算符来设计否定条件，如下述代码所有不是国际贸易专业的学生信息：

```
SELECT * FROM student WHERE NOT（专业 LIKE "国际贸易"）
```

LIKE 运算符提供两种字符串匹配方式，一种是使用下划线符号“ _ ”匹配一个任意字符，另一种是使用百分号“ % ”匹配 0 个或多个任意字符。

【例 6-6】在“ student ”表中检索所有姓李的学生信息。

```
SELECT * FROM student WHERE 姓名 LIKE "李%"
```

代码执行结果如图 6-6 所示。

【例 6-7】在“ student ”表中检索所有姓刘或姓张的学生信息。

这是一个字符串包含的查询，可以使用 IN 运算符：

```
SELECT * FROM student WHERE 姓名 IN（"张","刘"）
```

代码执行结果如图 6-7 所示。

图6-6　所有姓李的学生信息

图6-7　姓张或姓刘的学生信息

说明：

① IN 运算符，格式为 IN（常量 1，常量 2，…）。含义为查找和常量相等的值。

② 上式改为 VFP 条件为：

```
SELECT * FROM student WHERE 姓名 = "张" OR 姓名 = "刘"
```

【例 6-8】检索所有成绩在 80 和 90 之间的学生信息。

这个查询的条件是值在某一范围之间，可以使用 BETWEEN 运算符：

```
SELECT DISTINCT 姓名 FROM student,cj ;
    WHERE （成绩 BETWEEN 80 AND 90） AND （student.学号 = cj.学号）
```

代码执行结果如图 6-8 所示。

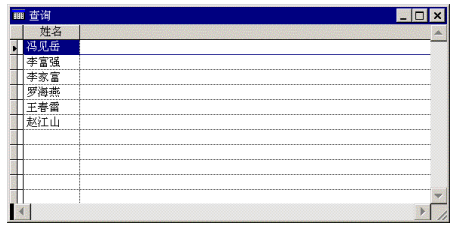


图6-8　成绩在80和90之间的学生

说明：上式如用 VFP 条件为

```
SELECT DISTINCT 姓名 FROM student,cj ;
    WHERE 成绩 >= 80 AND 成绩 <= 90 AND （student.学号 = cj.学号）
```

6.2.4　嵌套查询

在前面的例子中，WHERE 子句的〈联接条件〉是一个简单条件，有时，〈联接条件〉本身涉及多个表，或由查询得到，这时就需要使用 SQL 的嵌套查询功能。

嵌套查询一般具有以下两种形式：

```
〈表达式〉〈比较运算符〉[ANY | ALL | SOME]（〈子查询〉）
[NOT] EXISTS （〈子查询〉）
```

说明：

① 其中的〈比较运算符〉除了在第 3 章介绍的关系运算符之外，还有前面提到的特殊运算符。

② ANY、ALL、SOME 是量词，其中 ANY 和 SOME 是同义词，在进行比较运算时只要子查询中有一条记录为真，则结果为真；而 ALL 则要求子查询中的所有记录都为真，结果才为真。

③ EXISTS 是谓词，用来检查子查询中是否有结果返回（是否为空）。NOT EXISTS 表示是空的结果集。

【例 6-9】检索哪些专业至少有一个学生的成绩等于 90 分。

这个例子要求查询“ student ”表中的专业信息，而查询条件是“ cj ”表的成绩字段值，为此可以使用如下的嵌套查询：

```
SELECT DISTINCT 专业 FROM student WHERE 学号 IN ;
    （SELECT 学号 FROM cj WHERE 成绩 = 90）
```

代码执行结果如图 6-9 **所示。**

说明：命令中含有两个 SELECT-FROM-WHERE **查询块，即内层查询块和外层查询块，内层查询块检索到的学号值为** 2000160211、2000220207、2000220212**，对应于** student **表中的专业名为国际贸易和计算机应用。**

【例 6-10**】找出与“ 李富强 ”专业相同的学生。**

使用如下的嵌套查询：

```
SELECT * FROM student WHERE 专业 = ;
    （SELECT 专业 FROM student WHERE 姓名 in （"李富强"））
```

代码执行结果如图 6-10 **所示。**

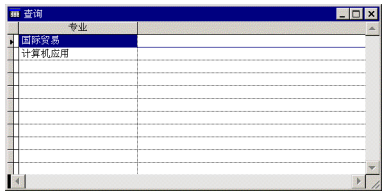


图6-9　**至少有一个学生的成绩等于**90**分的专业**

图6-10　**与“ 李富强 ”专业相同的学生**

【例 6-11**】检索哪些学生的所有成绩都大于等于** 80 **分。**

可以使用如下的嵌套查询：

```
SELECT 姓名 FROM student WHERE 学号 NOT IN ;
    （SELECT 学号 FROM cj WHERE 成绩 <= 80）
```

代码执行结果如图 6-11 **的左图所示。**

说明：student **表中学生“ 张仙见 ”在** cj **表中没有相应的记录，但上述代码将该学生也检索出来，这显然是错误的。可以要求排除那些没有成绩记录的学生，如图** 6-11 **的右图所示，代码改为：**

```
SELECT 姓名 FROM student WHERE 学号 NOT IN ;
        （SELECT 学号 FROM cj WHERE 成绩 <= 80） ;
  AND 学号 IN （SELECT 学号 FROM cj）
```

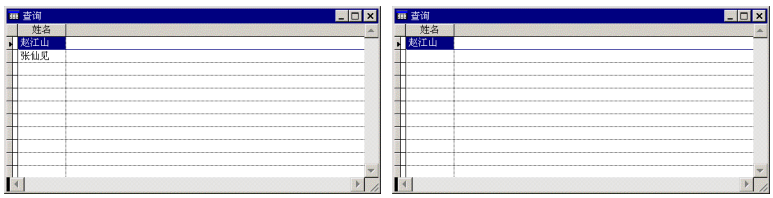


图6-11　**所有课程的成绩都大于**80**分的学生**

【例 6-12**】在** cj **表中检索选修** 111101 **号课的学生中成绩比选修** 141203 **号课的最低成绩**

要高的学生的学号和成绩。

此查询可以先求出选修 141203 号课的所有学生成绩（结果是 74，69，90，89），然后选出 111101 号课成绩中高于 141203 号课中某一个成绩的那些学生。要解决此题就要用到谓词 ANY 和 SOME，语句代码如下。代码执行结果如图 6-12 所示。

```
SELECT 学号, 成绩 FROM cj WHERE 课程号 = "111101" AND 成绩 > ANY ;
   (SELECT 成绩 FROM cj WHERE 课程号= "141203")
```

说明：

① 由于谓词 ANY 与 SOME 等价，所以与此命令等价的代码为：

```
SELECT 学号, 成绩 FROM cj WHERE 课程号 = "111101" AND 成绩 > SOME ;
   (SELECT 成绩 FROM cj WHERE 课程号= "141203")
```

② 本例可以进一步改为在 cj 表中检索选修高等数学的学生中成绩比选修理论力学课的最低成绩要高的学生的学号和成绩：

```
SELECT 学号, 成绩 FROM cj, kc WHERE 课程名= "高等数学" and cj.课程号 = kc.课程号 ;
  AND 成绩 > ANY (SELECT 成绩 FROM cj,kc ;
     WHERE 课程名= "理论力学" and cj.课程号 = kc.课程号)
```

【例 6-13】求选修 111101 号课学生中成绩比选修 141203 号课的任何学生的成绩都要高的学生的学号和成绩。

此查询可以先找出选修 141203 号课的所有学生成绩，然后再求出 111101 号课成绩高于 141203 号课所有成绩的那些学生，即比 141203 号课最高成绩还要高的学生。

```
SELECT 学号, 成绩 FROM cj WHERE 课程号 = "111101" AND 成绩 > ALL ;
   (SELECT 成绩 FROM cj WHERE 课程号= "141203")
```

代码执行结果如图 6-13 所示。

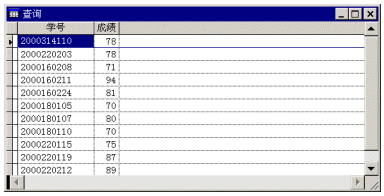


图6-12 比选修141203号课的最低成绩要高

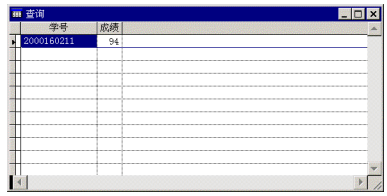


图6-13 比选修141203号课的任何成绩要高

6.2.5 分组、排序及系统函数的使用

1．用于计算的系统函数

SQL 不仅具有一般的检索数据的功能，而且还有计算检索的功能。SQL 用于计算检索的系统函数有 COUNT（计数）、SUM（求和）、AVG（求平均）、MAX（最大值）、MIN（最小值）。

【例 6-14】下述代码计算出高等数学课程的平均成绩、最高成绩和最低成绩。

```
SELECT 课程名,AVG（成绩） AS "平均成绩", MAX（成绩） AS "最高成绩", ;
```

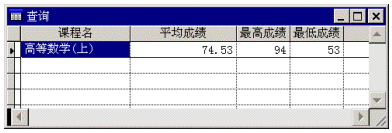
MIN（成绩） AS "最低成绩";
FROM cj, kc WHERE 课程名 = "高等数学" and cj.课程号 = kc.课程号

代码执行结果如图 6-14 所示。

【例 6-15】下述代码计算出 student 表中专业的数目。

SELECT COUNT（DISTINCT 专业） FROM student

代码执行结果如图 6-15 所示。



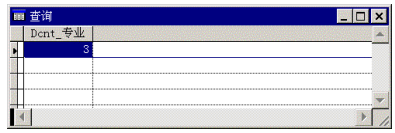


图6-14 高等数学的平均、最高和最低成绩图　　6-15 专业的数目

说明：计算专业数目应排除相同的项，因此使用 DISTINCT 选项。若无 DISTINCT 选项，将对记录个数进行计数：

SELECT COUNT（*） FROM student

【例 6-16】下述代码找出高等数学成绩大于平均成绩的学号等信息。

SELECT * FROM cj WHERE 课程号 = "111101" AND 成绩 > ;
（SELECT AVG（成绩） FROM cj WHERE 课程号 = "111101"）

代码执行结果如图 6-16 的左图所示。

图6-16 高等数学成绩大于平均成绩

说明：如图 6-16 的右图所示的查询结果来源于下述代码：

SELECT 姓名,课程名,成绩 FROM student,cj,kc ;
WHERE student.学号 = cj.学号 ;
AND kc.课程号 = cj.课程号 ;
AND 课程名 = "高等数学" ;
AND 成绩 >;
（SELECT AVG（成绩） FROM cj, kc WHERE kc.课程号 = cj.课程号 AND 课程名 = "高等数学"）

2．排序

SQL 中排序操作使用 ORDER BY 子句，其具体格式为：

ORDER BY〈关键字表达式 1〉[ASC | DESC] [,〈关键字表达式 2〉[ASC | DESC] ...]

其中，ASC 为升序，DESC 为降序。

【例 6-17】下述代码在 student 表中按出生日期字段升序（年龄降序）检索出全部学生信息：

```
SELECT * FROM student ORDER BY 出生日期
```

代码执行结果如图 6-17 所示。

说明：如果要对出生日期进行降序排列，只要加上 DESC：

```
SELECT * FROM student ORDER BY 出生日期 DESC
```

【例 6-18】下述代码在 cj 表中，按学号升序、成绩降序排列检索出成绩信息：

```
SELECT * FROM cj ORDER BY 学号 ASC, 成绩 DESC
```

代码执行结果如图 6-18 所示。

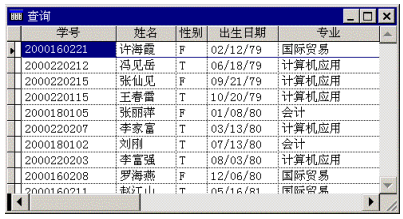


图6-17　按出生日期字段升序

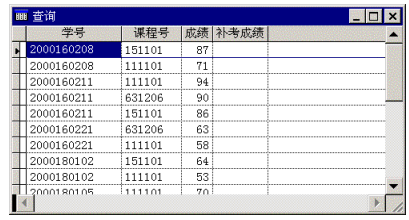


图6-18　按学号升序、成绩降序

3．分组

SQL 提供的系统函数可以对满足条件的记录进行各种运算，这些函数还可以从一组值中计算出一个汇总信息，通常和 GROUP BY 分组子句配合使用，完成一些特定功能的查询。

【例 6-19】下述代码在 cj 中按课程号分组并汇总成绩信息，检索出课程名和成绩信息，成绩字段以平均成绩显示。

```
SELECT 课程名, AVG（成绩） AS 平均成绩 FROM kc,cj ;
  GROUP BY cj.课程号 WHERE cj.课程号 = kc.课程号
```

代码执行结果如图 6-19 所示。

【例 6-20】下述代码列出各门课的平均成绩、最高成绩、最低成绩。

```
SELECT 课程名, AVG（成绩） AS 平均成绩, MAX（成绩） AS 最高分, MIN（成绩） AS 最低分 ;
 FROM kc,cj   GROUP BY cj.课程号 WHERE cj.课程号 = kc.课程号
```

代码执行结果如图 6-20 所示。

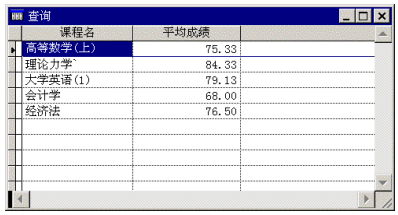


图6-19　各课程的平均成绩

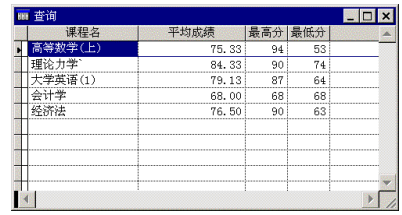


图6-20　各门课的平均成绩、最高成绩、最低成绩

【例 6-21】下述代码检索出最少选修了三门课程的学生姓名。

```
SELECT 姓名 FROM student WHERE 学号 IN ;
   (SELECT 学号 FROM cj GROUP BY 学号 HAVING COUNT(*) >=3)
```

代码执行结果如图 6-21 所示。

说明：以上查询中分别用到了 HAVING 子句和 WHERE 子句。它们的区别在于：WHERE 子句是用来指定表中各行所应满足的条件，而 HAVING 子句是用来指定每一分组所满足的条件，只有满足 HAVING 条件的那些组才能在结果中被显示。在上例中，先在选课表中按每个学生进行分组，然后在每个分组中检测其记录个数是否大于等于 3，如果满足条件，则该组的学号即为所求；再根据学生表找出其对应姓名。

【例 6-22】下述代码检索出高等数学成绩最高的学生的学号、姓名和成绩。

```
SELECT A.学号, 姓名, 成绩 FROM student A, cj B, kc cj WHERE ;
  A.学号 = B.学号 AND B.课程号 = cj.课程号 AND cj.课程名 = "高等数学" ;
  AND 成绩 = (SELECT MAX(成绩) FROM cj, kc WHERE cj.课程号 = kc.课程号;
  AND kc.课程名 = "高等数学")
```

代码执行结果如图 6-22 所示。

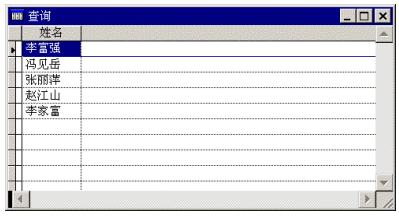


图6-21 最少选修了三门课程的学生

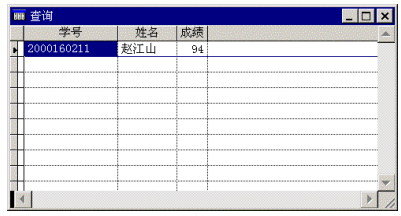


图6-22 高等数学成绩最高的同学

说明：以上查询中需要用表名作前缀，有时表名很长，用起来很麻烦，SQL 允许在 FROM 短语中为表名定义别名，格式为：

〈关系名〉〈别名〉

例如，上例中使用 FROM student A, cj B, kc C，以后便可以用 A、B 和 C 分别代表 student 表、cj 表和 kc 表，但是在嵌套的 SQL 子句中不能使用外层定义的别名。

6.2.6 超联接查询

SQL 中 FROM 子句后的联接称为超联接，超联接有四种形式，其格式为：

```
FROM〈表名〉[[INNER | LEFT [OUTER] | RIGHT [OUTER] | FULL [OUTER]
  JOIN [〈数据库名〉!]〈表名〉[[AS] Local_Alias][ON〈联接条件〉]]
```

其中：OUTER 关键字可被省略，包含 OUTER 强调这是一个外连接（outer join）。

下面我们分别以几个实例来说明这四种超联接的含义及区别。

1．内部联接

使用 INNER JOIN 形式的联接称为内部联接，INNER JOIN 等价于 JOIN。INNER JOIN

与普通联接相同：只有满足条件的记录才出现在查询结果中。

【例 6-23】下述代码将 cj 表和 kc 表按内部形式联接，包含学号、课程名、成绩字段。

```
SELECT 学号, 课程名, 成绩 FROM cj A INNER JOIN kc B ON A.课程号 = B.课程号 ;
  order by 学号
```

代码执行结果如图 6-23 所示。

说明：上述联接与下述 WHERE 条件等价：

```
SELECT A.学号, B.课程名, A.成绩 FROM cj A, kc B WHERE A.课程号 = B.课程号 ;
  ORDER BY 学号
```

2．左联接

LEFT [OUTER] JOIN 称为左联接，在查询结果中包含 JOIN 左侧表中的所有记录，以及 JOIN 右侧表中匹配的记录。

【例 6-24】下述代码将 cj 表和 kc 表左联接，包含学号、课程名、成绩字段。

```
SELECT 学号, 课程名, 成绩 FROM cj A LEFT JOIN kc B ON A.课程号 = B.课程号 ;
  ORDER by 学号
```

代码执行结果如图 6-24 所示。

查询

学号	课程名	成绩
2000160208	高等数学(上)	71
2000160208	大学英语(1)	87
2000160211	经济法	90
2000160211	高等数学(上)	94
2000160211	大学英语(1)	86
2000160221	高等数学(上)	58
2000160221	经济法	63
2000180102	高等数学(上)	53
2000180102	大学英语(1)	64
2000180105	大学英语(1)	69

图6-23　内部联接

查询

学号	课程名	成绩
2000160208	高等数学(上)	71
2000160208	大学英语(1)	87
2000160211	经济法	90
2000160211	高等数学(上)	94
2000160211	大学英语(1)	86
2000160221	高等数学(上)	58
2000160221	经济法	63
2000180102	高等数学(上)	53
2000180102	大学英语(1)	64

图6-24　左联接

从以上查询结果中可以看到，首先以左边表即 A 表中的第一条记录为准，在 B 表中查询，找到了，则显示，找不到相应的字段以 NULL 显示，本例中有相应的值。以下记录也是按照这种方法进行查询的。

3．右联接

RIGHT [OUTER] JOIN 称为右联接，在查询结果中包含 JOIN 右侧表中的所有记录，以及 JOIN 左侧表中匹配的记录。

【例 6-25】下述代码将 cj 表和 kc 表右联接，包含学号、课程名、成绩字段。

```
SELECT 学号, 课程名, 成绩 FROM cj A RIGHT JOIN kc B ON A.课程号 = B.课程号
```

代码执行结果如图 6-25 所示。

以上查询过程如下：首先以右表为准，即 B 表（kc 表），其第一条记录的课程号字段值为 111101，然后在左表（A 表）中检索与 111101 相等的课程号记录，找到了 9 条，其学号、成绩值分别如图 6-25 所示；第二条记录的课程号字段值为 111102，在左表（A 表）中没有检索到与 111102 相等的课程号记录，因此，学号字段以 NULL 显示。B 表中的以下记录均按此联接。

4．完全联接

FULL [OUTER] JOIN 称为完全联接，在查询结果中包含 JOIN 两侧所有的匹配记录和不匹配的记录。

【例 6-26】下述代码将 cj 表和 kc 表完全联接，包含学号、课程名、成绩字段。

```
SELECT 学号, 课程名, 成绩 FROM cj A FULL JOIN kc B ON A.课程号 = B.课程号
```

代码执行结果如图 6-26 所示。

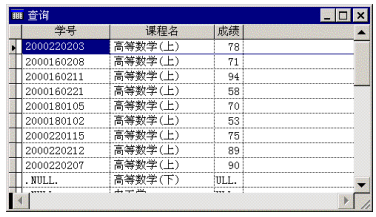


图6-25　右联接

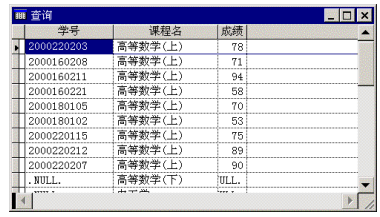


图6-26　完全联接

以上查询的过程是首先以右表为准，和右联接过程相同，然后再以左表为准，和左联接相同，联接完成后，在生成的记录中，将重复记录删除即可。

6.2.7　集合的并运算

使用 UNION 子句可以进行集合的并运算，即可以将两个 SELECT 语句的查询结果合并成一个查询结果。当然，要求进行并运算的两个查询结果具有相同的字段个数，并且对应字段的值要具有相同的数据类型和取值范围。

UNION 子句的语法格式为：

```
〈Selcct 命令 1〉 UNION [ALL] 〈Selcct 命令 2〉
```

说明：

① 可以使用多个 UNION 子句，ALL 选项防止删除合并结果中重复的行（记录）。

② 不能使用 UNION 来组合子查询。

③ 只有最后的〈Selcct 命令〉中可以包含 ORDER BY 子句，而且必须按编号指出排序的列（它将影响整个结果）。

【例 6-27】下述代码找出高等数学成绩大于平均成绩的学生和大学英语成绩大于平均成绩的学生信息。

```
SELECT 姓名,课程名,成绩 FROM student,cj,kc ;
  WHERE student.学号= cj.学号 AND kc.课程号= cj.课程号 AND 课程名= "高等数学" ;
  AND 成绩 > ;
  （SELECT AVG（成绩） FROM cj, kc WHERE kc.课程号= cj.课程号 AND 课程名= "高等数学"） ;
UNION SELECT 姓名,课程名,成绩 FROM student,cj,kc ;
  WHERE student.学号= cj.学号 AND kc.课程号= cj.课程号 AND 课程名= "大学英语" ;
  AND 成绩 > ;
  （SELECT AVG（成绩） FROM cj, kc WHERE kc.课程号= cj.课程号 AND 课程名= "大学英语"） ;
  ORDER BY 2
```

代码执行结果如图 6-27 所示。

6.2.8 查询输出去向及几个特殊选项

1．显示部分结果

有时只需要满足条件的前几个记录，这时使用 TOP 〈数值表达式〉 [PERCENT]格式的子句非常有用，在前面的 SQL 语法格式中，已经说明了它的含义，下面以实例来说明具体应用。

【例 6-28】下述代码在 student 中，显示年龄最大的三位学生的信息。

```
SELECT * TOP 3 FROM student ORDER BY 出生日期
```

代码执行结果如图 6-28 所示。

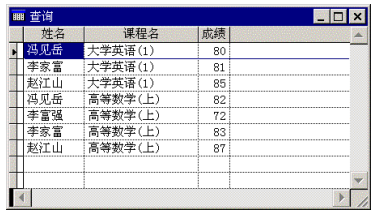


图6-27 并运算

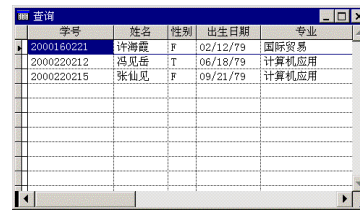


图6-28 年龄最大的三位学生

【例 6-29】下述代码在 student 表中，显示年龄最小的 40%学生的信息。

```
SELECT * TOP 40 PERCENT FROM student ORDER BY 出生日期 DESC
```

代码执行结果如图 6-29 所示。

2．查询去向

FROM 子句中的 INTO 与 TO 选项用于指定查询结果的输出去向，默认查询结果显示在浏览窗口中。INTO 选项中的〈查询结果〉有 3 种：ARRAY〈数组〉、CURSOR〈临时表〉和 TABLE | DBF〈表名〉；TO 选项也有 3 种：文件、屏幕和打印机。

下面以实例来具体说明 INTO 选项的使用。

【例 6-30】下述代码将表 student 按专业升序排列，生成永久表 student2.dbf。

```
SELECT * FROM student INTO DBF student2 ORDER BY 出生日期
```

执行代码，然后打开表 student2，浏览结果如图 6-30 所示。

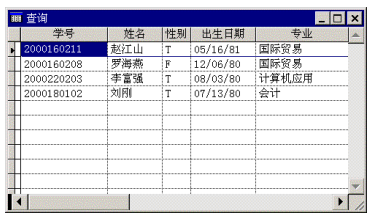


图6-29 年龄最小的40%学生

图6-30 按专业升序排列的表

【例 6-31】下述代码在 cj 表中检索出成绩的最大值并将结果保存到数组 a 中。

```
SELECT max（成绩） FROM cj INTO ARRAY a
```

在表中检索出一个最高分 94，将此值赋予 a，使用命令：

```
?a（1）
```

可以查看到数组首元素 a[1]的值为 94。

【例 6-32】下述代码列出各门课的平均成绩、最高成绩、最低成绩。并将查询结果保存到临时表 temp 中。

```
SELECT 课程名, AVG（成绩） AS 平均成绩, MAX（成绩） AS 最高分, MIN（成绩） AS 最低分 ;
 FROM kc,cj  GROUP BY cj.课程号 WHERE cj.课程号 = kc.课程号 INTO CURSOR temp
```

执行代码，然后在“显示”菜单中选择浏览“TEMP”，结果如图 6-31 所示。

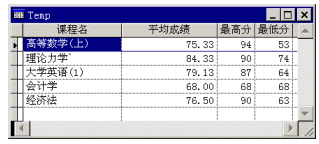


图 6-31 临时表 temp

注意，临时表只能暂时保存数据，一但关闭 VFP 就会随之消失。

至此，我们已经介绍了 SQL-SELECT 查询中的常用语句，掌握 SQL-SELECT 不仅对学好 VFP 至关重要，也是以后使用其他数据库或开发数据库应用程序的基础。由于篇幅有限，我们只介绍到这里，希望读者多下一些功夫将查询弄明白，牢固掌握。

6.3 操作功能

SQL 语言的操作功能包括对表中数据的增加、删除和更新操作。

6.3.1 插入

1．SQL 语法

在一个表的尾部追加数据时，要用到插入功能，SQL 的插入命令包括以下 3 种格式：

```
INSERT INTO〈表名〉[（〈字段名 1〉[，〈字段名 2〉，…]）]
    VALUES （〈表达式 1〉[，〈表达式 2〉…]）
```

和

```
INSERT INTO〈表名〉FROM ARRAY〈数组名〉
INSERT INTO〈表名〉FROM MEMVAR
```

说明：

① 第 1 种格式在指定的表的表尾添加一条新记录，其值为 VALUES 后面的表达式的值。当需要插入表中所有字段的数据时，表名后面的字段名可以缺省，但插入数据的格式必须与表的结构完全吻合；若只需要插入表中某些字段的数据，就需要列出插入数据的字段，当然相应表达式的数据位置会与之对应。

② 第 2 种格式也是在指定的表的表尾添加一条新记录，新记录的值是指定的数组中各元素的数据。数组中各元素与表中各字段顺序对应。如果数组中元素的数据类型与其对应的字段类型不一致，则新记录对应的字段为空值；如果表中字段个数大于数组元素的个数，则多

出的字段为空值。

③ 第 3 种格式也是在指定的表的表尾添加一条新记录，新记录的值是指定的内存变量的值。添加的新记录的值是与指定表各字段名同名的内存变量的值，如果同名的内存变量不存在，则相应的字段为空。

2．示例

【例 6-33】下述代码在 student 表中插入一条新记录。

```
INSERT INTO student（学号，姓名，性别，出生日期，专业） ;
  VALUES（'2000180201', '张强', .T., CTOD（'04/01/80'）, '会计'）
```

【例 6-34】下述代码在 cj 表中插入一条新记录。

```
INSERT INTO cj（学号，课程号，成绩） VALUES（'2000180201','141203', 77）
```

【例 6-35】先定义数组 A（7），A 中各元素的值见表 6-4。

表 6-4 数组 A( )中元素的值

A（1）	A（2）	A（3）	A（4）	A（5）	A（6）	A（7）
'2000180123'	'王刚'	.T.	CTOD（'05/06/81'）	会计		

在学生表中插入一条记录：

```
INSERT INTO student FROM ARRAY A
```

【例 6-36】如果定义了内存变量：学号 = '2000180123'，姓名 = "李丽"，性别 = .F.，出生日期 = CTOD（"03/23/80"）。下述代码在 student 表中添加一条记录：

```
INSERT INTO student FROM MEMVAR
```

新记录中除了学号、姓名、性别、出生日期字段外，其他 3 个字段值均为空值。

6.3.2 删除

用 SQL 语言删除记录的命令格式如下：

```
DELETE FROM [〈数据库!〉]〈表名〉
  [WHERE〈条件表达式 1〉[AND | OR〈条件表达式 2〉...]]
```

说明：

① [〈数据库!〉]〈表名〉：指定加删除标记的表名及该表所在的数据库名，用“!”分割表名和数据库名，数据库名为可选项。

② WHERE 选项：指明只对满足条件的记录加删除标记。

③ 上述删除只是加删除标记并没有从物理上删除，只有执行了 PACK 命令，有删除标记的记录才能真正从物理上删除。置了删除标记的记录可以用 RECALL 命令取消删除标记。

【例 6-37】下述代码将表 student 中的“李丽”逻辑删除。

```
DELETE FROM student WHERE 姓名 = "李丽"
```

6.3.3 更新

更新是指对存储在表中的记录进行修改，SQL **更新命令的语法格式为：**

```
UPDATE [〈数据库〉! ]〈表名〉
  SET〈列名 1〉=〈表达式 1〉[,〈列名 2〉=〈表达式 2〉...]
  [WHERE 〈条件表达式 1〉[AND | OR〈 条件表达式 2〉...]]
```

说明：

① [**〈数据库〉**!]**〈表名〉：指定要更新数据的记录所在的表名及该表所在的数据库名。**

② SET **〈列名〉**=**〈表达式〉：指定被更新的字段及该字段的新值。如果省略** WHERE **子句，则该字段每一行都用同样的值更新。**

③ WHERE **〈条件表达式〉：指明将要更新数据的记录。即更新表中符合条件表达式的记录。**

【例 6-38**】下述代码将** cj **表中所有高等数学（上）的考试成绩降低** 2%**。**

```
UPDATE cj SET 成绩 = 成绩*.98 ;
  WHERE 课程号 = （SELECT DIST 课程号 FROM kc WHERE 课程名 = "高等数学（上）"）
```

6.4 定义功能

SQL **不但可以支持数据查询、数据操作功能，而且还支持数据定义功能。定义功能包含数据库的定义、表的定义、视图的定义、存储过程的定义、规则的定义和索引的定义等若干部分，本节主要介绍** SQL **支持的表定义功能和视图定义功能。**

6.4.1 表结构的定义

1.SQL 语法

表结构的定义是指创建一个含有指定字段的表。SQL **语句中表定义命令的语法格式如下：**

```
CREATE TABLE | DBF 〈表名 1〉[NAME 〈长表名〉][FREE]
(〈字段名 1〉〈类型〉[（〈字段宽度〉[,〈小数位数〉]）]
  [NULL | NOT NULL]
  [CHECK 〈逻辑表达式 1〉[ERROR 〈字符型文本信息 1〉]]
  [DEFAULT 〈表达式 1〉]
  [PRIMARY KEY | UNIQUE]
  [REFERENCES 〈表名 2〉[TAG 〈标识名 1〉]]
  [NOCPTRANS]
 [,〈字段名 2〉…]
  [,PRIMARY KEY 〈表达式 2〉 TAG 〈标识名 2〉
  |,UNIQUE 〈表达式 3〉 TAG 〈标识 3〉]
  [,FOREIGN KEY 〈表达式 4〉 TAG 〈标识名 4〉[NODUP]
       REFERENCES 〈表名 3〉 [TAG 〈标识名 5〉]]
  [,CHECK 〈逻辑表达式 2〉[ERROR 〈字符型文本信息 2〉]])
```

| FROM ARRAY 〈数组名〉

说明：用 CREATE TABLE 命令可以完成第 4 章中介绍的“ 表设计器 ”具有的所有操作。其中的选项简单说明如下：

① TABLE 和 DBF 选项等价，都是建立表文件。

②〈表名〉：为新建表指定表名。

③ NAME〈长表名〉：为新建表指定一个长表名。只有打开数据库，在数据库中创建表时，才能指定一个长表名。长表名可以包含 128 个字符。

④ FREE：建立的表是自由表，不加入到打开的数据库中。当没有打开数据库时，建立的表是自由表。

⑤〈字段名 1〉〈类型〉 [（〈字段宽度〉[,〈小数位数〉]）]：指定字段名、字段类型、字段宽度及小数位数。字段类型可以用一个字符表示，其含义如表 4-2 所示。

⑥ NULL：允许该字段值为空；NOT NULL：该字段值不能为空。缺省值为 NOT NULL。

⑦ CHECK〈逻辑表达式〉：指定该字段的合法值及该字段值的约束条件。

⑧ ERROR〈字符型文本信息〉指定在浏览或编辑窗口中该字段输入的值不符合 CHECK 子句的合法值时，VFP 显示的错误信息。

⑨ DEFAULT〈表达式〉：为该字段指定一个缺省值，表达式的数据类型与该字段的数据类型要一致。即每添加一条记录时，该字段自动取该缺省值。

⑩ PRIMARY KEY：为该字段创建一个主索引，索引标识名与字段名相同。主索引字段值必须惟一。UNIQUE：为该字段创建一个候选索引，索引标识名与字段名相同。

⑪ REFERENCES 〈表名〉 [TAG 〈标识名〉]：指定建立持久关系的父表，同时以该字段为索引关键字建立外索引，用该字段名作为索引标识名。表名为父表表名，标识名为父表中的索引标识名。如果省略索引标识名，则用父表的主索引关键字建立关系，否则不能省略。如果指定了索引标识名，则在父表中存在的索引标识字段上建立关系。父表不能是自由表。

⑫ CHECK 〈逻辑表达式〉 [ERROR 〈字符型文本信息〉]：由逻辑表达式指定表的合法值。不合法时，显示由字符型文本信息指定的错误信息。该信息只有在浏览或编辑窗口中修改数据时显示。

⑬ FROM ARRAY 〈数组名〉：由数组创建表结构。数组名指定的数组包含表的每一个字段的字段名、字段类型、字段宽度及小数位数。

2．示例

用 SQL 命令来建立的数据表，可以与用“ 表设计器 ”建立的表作对比。

【例 6-39】下述代码建立商品表，该表不属于任何数据库。

```
CREATE TABLE 商品 FREE;
   (货号 C（6）,品名 C（8）,进口 L（1）,单价 N（7,2）, 数量 N（2）,开单日期 D（8）,生
   产单位 C（16））
```

下面例题中的数据表“ order_detail ”、“ order_list ”和“ customer ”，其表结构分别参见第 4 章习题中的数据库“ 订货管理.dbc ”。

【例 6-40】假设已经建立了“ 订货管理 ”数据库，下述代码在订货管理.dbc 中建立

order_detail 表，创建该表的 SQL 语言命令为：

```
OPEN DATABASE 订货管理
CREATE TABLE order_detail;
   （订单号 C（6） NOT NULL, 器件号 C（6），器件名 C（16），单价 F（10,2）;
  NULL CHECK 单价>0 ERROR "单价必须大于 0", 数量 I（4） DEFAULT 1）
```

说明：在“订货管理”中建立了 order_detail 表后，通过浏览窗口向 order_detail 表输入数据或修改数据时，单价字段的值必须大于 0，否则显示“单价必须大于 0”信息，并等待输入合法的值；添加新记录时，数量字段自动填入 1。

【例 6-41】下述代码在“订货管理”中建立 order_list 表：

```
OPEN DATABASE 订货管理
CREATE TABLE order_list （客户号 C（6），订单号 C（6） PRIMARY KEY, ;
  订购日期 D, 总金额 F（15,2））
```

说明：由于订单号字段建立了主索引，订单号字段不能输入重复值。

【例 6-42】下述代码在“订货管理”中建立 customer 表：

```
OPEN DATABASE 订货管理
CREATE TABLE customer （客户号 C（6） REFERENCES order_list, ;
    客户名 C（16），地址 C（20），电话 C（14））
```

6.4.2 表的删除

随着数据库应用的变化，往往有些表连同它的数据不再需要了，这时可以删除这些表，以节省存储空间。删除表使用的 SQL 命令格式如下：

```
DROP TABLE〈表名〉
```

说明：DROP TABLE 直接从磁盘上删除表名所对应的 DBF 文件。如果表名是数据库中的表并且相应的数据库是当前数据库，则从数据库中删除表；否则虽然从磁盘上删除了 DBF 文件，但是记录在数据库 DBC 文件中的信息却没有删除，此后会出现错误提示。所以要删除数据库中的表时，最好应使数据库是当前打开的数据库，在数据库中进行操作。

【例 6-43】下述代码在“订货管理”中删除 customer 表：

```
OPEN DATABASE
DROP TABLE customer
```

由于上表属于数据库表，所以先打开其所在数据库，然后再删除此表。

【例 6-44】下述代码删除商品表：

```
DROP TABLE 商品
```

6.4.3 表结构的修改

用户使用数据时，随着应用要求的改变，往往需要对原来的表格结构进行修改，修改表结构的 SQL 命令是 ALTER TABLE，该命令格式有三种。

1．第 1 种格式

第 1 种格式的 ALTER TABLE 命令可以为指定的表添加字段或修改已有的字段。其格式为：

```
ALTER TABLE〈表名 1〉ADD | ALTER [COLUMN]
    〈字段名 1〉〈字段类型〉[（〈长度〉[，〈小数位数〉]）][NULL | NOT NULL]
  [CHECK〈逻辑表达式 1〉[ERROR〈字符型文本信息〉]][DEFAULT〈表达式 1〉]
  [PRIMARY KEY | UNIQUE][REFERENCES〈表名 2〉[TAG〈标识名 1〉]]
  [NOCPTRANS]
```

说明：

①〈表名 1〉：指明被修改表的表名。

② ADD [COLUMN]：该子句指出新增加列的字段名及它们的数据类型等信息。在 ADD 子句中使用 CHECK、PRIMARY KEY、UNIQUE 任选项时需要删除所有数据，否则违反有效性规则，命令不被执行。

③ ALTER [COLUMN]：该子句指出要修改列的字段名以及它们的数据类型等信息。在 ALTER 子句中使用 CHECK 任选项时，需要被修改字段的已有数据满足 CHECK 规则；使用 PRIMARY KEY、UNIQUE 任选项时，需要被修改字段的已有数据满足惟一性，不能有重复值。

从命令格式可以看出，该格式可以修改字段的类型、宽度、有效性规则、错误信息、默认值，定义主关键字和联系等；但是不能修改字段名，不能删除字段，也不能删除已经定义的规则等。

【例 6-45】下述代码在 student 表中，增加籍贯字段，字段类型为 C 型，长度为 10。

```
ALTER TABLE student ADD 籍贯 C（10）
```

若将籍贯字段宽度修改为 12，合法值为北京市的或上海市的，则 SQL 代码为：

```
ALTER TABLE student ALTER 籍贯 C（12）；
  CHECK LEFT（籍贯,6） ="北京市" OR LEFT（籍贯,6）=" 上海市"
```

执行该命令前需要将学生表中所有记录的籍贯字段改为开头三个汉字为北京市或上海市，以满足 CHECK 规则。

2．第 2 种格式

第 2 种格式的 ALTER TABLE 命令主要用于修改指定表中指定字段的 DEFAULT 和 CHECK 约束规则，不影响原有表的数据。其格式为：

```
ALTER TABLE〈表名〉ALTER [COLUMN]〈字段名〉[NULL | NOT NULL]
  [SET DEFAULT〈表达式〉]
  [SET CHECK〈逻辑表达式〉[ERROR〈字符型文本信息〉]]
  [DROP DEFAULT][DROP CHECK]
```

说明：

①〈表名〉：指明被修改表的表名。

② ALTER [COLUMN] 〈字段名〉：指出要修改列的字段名。

③ NULL| NOT NULL：指定该字段可以为空或不能为空。

④ SET DEFAULT〈表达式〉：重新设置该字段的缺省值。

⑤ SET CHECK〈逻辑表达式〉[ERROR 〈字符型文本信息〉]：重新设置该字段的合法值，要求该字段的原有数据满足合法值。

⑥ DROP DEFAULT：删除缺省值。

⑦ DROP CHECK：删除该字段的合法值限定。

【例 6-46】下述代码删除 student 表中籍贯字段的合法值约束，设置默认值为北京市。

```
ALTER TABLE student ALTER 籍贯 DROP CHECK
ALTER TABLE student SET DEFALUT LEFT（籍贯, 6） = "北京市"
```

3．第 3 种格式

第 3 种格式的 ALTER TABLE 命令可以删除指定表中的指定字段、修改字段名、修改指定表的完整性规则，包括添加或删除主索引、外索引、候选索引及表的合法值限定。其格式为：

```
ALTER TABLE〈表名〉[DROP[COLUMN]〈字段名 1〉]
  [SET CHECK〈逻辑表达式 1〉[ERROR〈字符型文本信息〉]]
  [DROP CHECK]
  [ADD PRIMARY KEY〈表达式 1〉TAG〈标识名 1〉[FOR〈逻辑表达式 2〉]]
  [DROP PRIMARY KEY ]
  [ADD UNIQUE〈表达式 2〉[TAG〈标识名 2〉[FOR〈逻辑表达式 3〉]]]
  [DROP UNIQUE TAG〈标识名 3〉]
  [ADD FOREIGN KEY [〈表达式 3〉] TAG〈标识名 4〉[FOR〈逻辑表达式 4〉]
    REFERENCES 表名 2 [TAG〈标识名 4〉]]
  [DROP FOREIGN KEY TAG〈标识名 5〉[SAVE]]
  [RENAME COLUMN〈字段名 2〉TO〈字段名 3〉]
  [NOVALIDATE]
```

说明：

① DROP[COLUMN]〈字段名〉：从指定表中删除指定的字段。

② SET CHECK〈逻辑表达式〉[ERROR 〈字符型文本信息〉]：为该表指定合法值及错误提示信息。DROP CHECK：删除该表的合法值限定。

③ ADD PRIMARY KEY〈表达式〉 TAG 〈标识名〉：为该表建立主索引，一个表只能有一个主索引。DROP PRIMARY KEY：删除该表的主索引。

④ ADD UNIQUE〈表达式〉[TAG〈标识名〉]：为该表建立候选索引，一个表可以有多个候选索引。DROP UIQUE TAG 〈标识名〉：删除该表的候选索引。

⑤ ADD FOREIGN KEY：为该表建立外（非主）索引，与指定的父表建立关系，一个表可以有多个外索引。

⑥ DROP FOREIGN KEY TAG：删除外索引，取消与父表的关系，SAVE 子句将保存该索引。

⑦ RENME COLUMN〈字段名 2〉TO〈字段名 3〉：修改字段名，字段名 2 指定要修改的字段名，字段名 3 指定新的字段名。

⑧ NOVALIDATE：修改表结构时，允许违反该表的数据完整性规则，默认值为禁止违反数据完整性规则。

注意：修改自由表时，不能使用 DEFAULT，FOREIGN KEY，PRIMARY KEY，REFERENCES 或 SET 子句。

【例 6-47】下述代码删除 student 表中籍贯字段，修改专业为专业名称字段。

```
ALTER TABLE student DROP 籍贯 RENAME COLUMN 专业 TO 专业名称
```

说明：如果在被删除字段上建立了索引，要先将索引删除，再删除该字段。

【例 6-48】下述代码在 student 表的学号字段上建立候选索引。

```
ALTER TABLE student ADD UNIQUE 学号 TAG 学号
```

【例 6-49】下述代码在 student 表中添加一个家庭住址字段，删除说明字段。

```
ALTER TABLE 学院 ADD 家庭住址 M
ALTER TABLE 学院 DROP 说明
```

【例 6-50】使用 SQL 语言重做【例 4-11】。

```
* 将两个表 cj 与 d_cj 左联接为临时表
SELECT a.*,b.学号 AS b 学号,b.成绩 AS b 成绩 FROM ;
  cj a LEFT JOIN d_cj b ON a.学号 = b.学号 INTO DBF ls
* 对临时表中的成绩进行更新
UPDATE ls SET 成绩 = b 成绩 WHERE 学号=b 学号
* 删除临时表中多余的字段
ALTER TABLE ls DROP b 学号
ALTER TABLE ls DROP b 成绩
* 删除表 cj
DROP TABLE cj
* 重新生成表 cj
SELECT * FROM ls INTO DBF cj
```

6.4.4 视图

在第 5 章中已经介绍了如何使用“视图设计器”来创建视图，本节介绍如何利用 SQL 语言建立视图。

1．视图的定义

视图是根据对表的查询定义的，定义视图的 SQL 命令格式如下：

```
CREATE SQL VIEW〈视图名〉[（〈字段名 1〉[,〈字段名 2〉]...）] AS〈查询语句〉
```

其中，〈查询语句〉可以任意的 SELECT 查询语句，它说明并限制了视图中的数据。如果没有为视图指定字段名，视图中的字段名将与〈查询语句〉中指定的字段名相同。

【例 6-51】下述代码在例 6-16 中的查询的基础上定义一个视图，它包含高等数学成绩大于平均成绩的学生姓名、课程名和成绩等信息。

```
OPEN DATABASE 学生管理
```

```
CREATE SQL VIEW v_高数 AS ;
  SELECT 姓名,课程名,成绩 FROM student,cj,kc ;
  WHERE student.学号 = cj.学号 ;
  AND kc.课程号 = cj.课程号 ;
  AND 课程名 = "高等数学" ;
  AND 成绩 >;
  （SELECT AVG（成绩） FROM cj, kc WHERE kc.课程号 = cj.课程号 AND 课程名 = "高等
  数学"）
```

执行上述代码，然后查询 v_高数：

```
SELECT * FROM v_高数
```

查询结果如图 6-32 所示。

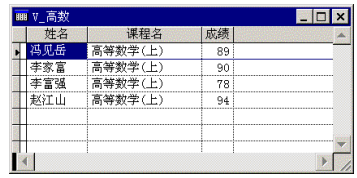


图 6-32　视图查询结果

说明：视图必须建立在数据库中。

上述命令与 VFP 中创建视图的命令 CREAT VIEW 等价。

2．视图的删除

由于视图是从表中派生出来的，所以不存在修改结构的问题，但是可以删除视图。使用 SQL 语言删除视图的格式如下：

```
DROP VIEW〈视图名〉
```

【例 6-53】下述代码删除视图 v_高数。

```
OPEN DATABASE 学生管理
DROP VIEW v_高数
```

说明：上述命令与 VFP 中删除视图的命令 DELETE VIEW 〈视图名〉等价。

6.5　习题

一、选择题

1．下列对于 SQL 语言所具有功能的说法，错误的是（　　）。

A）数据查询　　B）数据定义　　C）数据操纵　　D）以上都不对

2．下面有关 HAVING 子句描述错误的是（　　）。

A）HAVING 子句必须与 GROUP BY 子句同时使用，不能单独使用

B）使用 HAVING 子句的同时不能使用 WHERE 子句

C）使用 HAVING 子句的同时可以使用 WHERE 子句

D）使用 HAVING 子句的作用是限定分组的条件

3．下列不属于数据定义功能的 SQL 语句是（　　）。

A）CREATE TABLE　　B）CREATE CURSOR

C）UPDATE　　D）ALTER TABLE

下面各题使用当前盘当前目录下的数据库 db_stock，其中有数据表 stock.dbf，该数据库表的内容见表 6-5。

表 6-5　股票表 stock.dbf

股票代码	股票名称	单　价	交易所
600600	青岛啤酒	7.48	上海
600600	方正科技	15.20	上海
600600	广电电子	10.40	上海
600600	兴业房产	6.76	上海
600600	二纺机	9.96	上海
600600	轻工机械	14.59	上海
000001	深发展	7.48	深圳
000002	深万科	6.50	深圳

4．以 stock 表为依据，执行如下 SQL 查询语句后结果是（　　）。

```
SELECT * FROM stock INTO DBF stock ORDER BY 单价
```

A）系统会提示出错信息

B）会生成一个按“单价”升序排序的表文件，将原来的 stock.dbf 文件覆盖

C）会生成一个按“单价”降序排序的表文件，将原来的 stock.dbf 文件覆盖

D）不会生成排序文件，只在屏幕上显示一个按“单价”升序排序的结果

5．有如下 SQL 语句

```
SELECT * FROM stock WHERE 单价 BETWEEN 6.76 AND 15.20
```

与该语句等价的是（　　）。

A）SELECT * FROM stock WHERE 单价 <= 15.20 .AND. 单价 >= 6.76

B）SELECT * FROM stock WHERE 单价 < 15.20 .AND. 单价 > 6.76

C）SELECT * FROM stock WHERE 单价 >= 15.20 .AND. 单价 <= 6.76

D）SELECT * FROM stock WHERE 单价 > 15.20 .AND. 单价 < 6.76

6．有如下 SQL 语句：

```
SELECT max（单价）INTO ARRAY a FROM stock
```

执行该语句后（　　）。

A）a[1]的内容为 15.20　　B）a[1]的内容为 6

C）a[0]的内容为 15.20　　D）a[0]的内容为 6

7．有如下 SQL 语句：

```
SELECT 股票代码, AVG（单价） as 均价 FROM stock;
   GROUP BY 交易所 INTO DBF temp
```

执行该语句后 temp 表中第二条记录的“均价”字段的内容是（　　）。

A）7.48　　B）9.99　　C）6.73　　D）15.20

8．执行如下 SQL 语句后：

```
SELECT DISTINCT 单价 FROM stock;
```

WHERE 单价 = （SELECT min（单价） FROM stock） INTO DBF stock_x

表 stock_x 中的记录个数是（ ）

A）1 B）2 C）3 D）4

9．求每个交易所的平均单价的 SQL 语句是（ ）。

A）SELECT 交易所, avg（单价）FROM stock GROUP BY 单价

B）SELECT 交易所, avg（单价）FROM stock ORDER BY 单价

C）SELECT 交易所, avg（单价）FROM stock ORDER BY 交易所

D）SELECT 交易所, avg（单价）FROM stock GROUP BY 交易所

10．将 stock 表的股票名称字段的宽度由 8 改为 10，应用 SQL 语句（ ）。

A）ALLTER TABLE stock 股票名称 WITH c（10 ）

B）ALLTER TABLE stock 股票名称 c（10 ）

C）ALLTER TABLE stock ALLTER 股票名称 c（10 ）

D）ALLTER stock 股票名称 c（10 ）

11．在当前盘当前目录下删除表 stock 的命令是（ ）。

A）DROP stock B）DELETE TABLE stock

B）DROP TABLE stock D）DELETE stock

12．有如下 SQL 语句：

CREATE VIEW stock_view AS SELECT * FROM stock WHERE 交易所 = "深圳"

执行该语句后产生的视图包含的记录个数是（ ）。

A）1 B）2 C）3 D）4

13．有如下 SQL 语句：

CREATE VIEW view_stock AS SELECT 股票名称 AS 名称, 单价 FROM stock

执行该语句后产生的视图包含的字段名是（ ）。

A）股票名称、单价 B）名称、单价

C）名称、单价、交易所 D）股票名称、单价、交易所

14．下面有关对视图的描述正确的是（ ）。

A）可以使用 MODIFY STRUCTURE 命令修改视图的结构

B）视图不能删除，否则影响原来的数据文件

C）视图是对表的复制产生的

D）使用 SQL 对视图进行查询时必须事先打开该视图所在的数据库

二、填空题

1．使用 SQL 语句将一条新的记录插入“课程”表 kc.dbf：

INSERT_____kc（课程号, 课程名, 学分）_____（"431231", "自动控制原理", 3）

2．使用 SQL 语句求“李富强”的总分：

SELECT_____（成绩） FROM cj;

WHERE 学号 IN （SELECT 学号 FROM_____WHERE 姓名 = "李富强"）

3．使用 SQL 语句完成操作：将所有高等数学的成绩提高 3%

_____ cj SET 成绩 = 成绩 * 1.03_____课程号 IN;

（SELECT 课程号_____ kc WHERE 课程名="高等数学（上）"）

4．如将第 2 题的查询结果存入到永久表中应使用_____短语。

5．在 SQL 命令中用于求和与计算平均值的函数为_____和_____。

三、上机题

（1～5 题在数据表 stock.dbf 中进行操作）

1．从表中检索出单价在 7.48 和 6.50 之间的股票信息。

2．从表中检索出单价为 14.59 和 15.20 的股票信息。

3．找出所有交易所不是上海的股票信息。

4．在 stock 表的基础上定义一个视图，它包含股票名称和交易所两个字段。

5．删除视图 v_sal。

（6～10 题在订货管理库中进行操作）

6．列出总金额大于所有订购单总金额平均值的订购单（order_list）清单（按客户号升序排列），并将结果存储到 results6_2 表中（表结构与 order_list 表结构相同）。

7．列出客户名为“三益贸易公司”的订购单明细（order_detail）记录（将结果先按“订单号”升序排列，同一订单的再按“单价”降序排列）,并将结果存储到 results6_3 表中（表结构与 order_detail 表结构相同）。

8．将 customer1 表中的全部记录追加到 customer 表中，然后用 SQL SELECT 语句完成查询：列出目前有订购单的客户信息（即有对应的 order_list 记录的 customer 表中的记录），同时要求按客户号升序排序，并将结果存储到 results6_4 表中（表结构与 customer 表结构相同）。

9．将 order_detail1 表中的全部记录追加到 order_detail 表中，然后用 SQLSELECT 语句完成查询：列出所有订购单的订单号、订购日期、器件号、器件名和总金额（按订单号升序），并将结果存储到 results6_5 表中（其中订单号、订购日期、总金额取自 order_list 表，器件号、器件名取自 order_detail 表）。

10．将 order_list1 表中的全部记录追加到 order_list 表中，然后用 SQL SELECT 语句完成查询：按总金额降序列出所有客户的客户号、客户名及其订单号和总金额，并将结果存储到 results6_6 表中（其中客户号、客户名取自 customer 表，订单号、总金额取自 order_list 表）。

第 7 章　程序设计基础

前面各章介绍的操作都属于交互方式，即通过可视化操作或是在命令窗口直接输入命令来创建、维护和使用数据库。可视化操作方式不用记忆命令，但是操作繁琐、效率低；而直接在命令窗口输入命令比菜单选择快捷，但要记忆命令，尤其是对于重复进行的操作需要反复输入命令，对于不熟悉 VFP 命令的用户来说更是难以完成的。为此，VFP 提供了批命令工作方式，即程序方式。程序方式是指预先将多条命令按一定逻辑结构组织成一个命令序列：程序，运行程序，系统会自动有序地执行该程序中的有关命令。程序方式的运行效率远远高于交互方式，也只有程序方式才能充分体现计算机进行数据处理的高速度和自动化特点。

VFP 支持两种类型的编程，一种是早期 xBase 和 FoxPro 语言所支持的过程编程方式，另一种是面向对象的编程方式。过程方式就是用结构化编程语言来编写程序，可以把一个复杂的程序分成较小的过程，进行单独的调试；面向对象的编程方式则是将编程工作主要集中在描述的对象上，程序由事件驱动。

VFP 既包含原来的 xBase 和 FoxPro 语言命令，可以进行过程方式编程，同时也提供了许多新命令，使之具有初步的面向对象的编程能力。由于过程方式编程比较容易，同时也是面向对象编程方式的基础，所以本章首先介绍过程编程方式。

7.1　程序文件

VFP 程序是包含一系列命令的文本文件，因此程序文件又称为命令文件。

可以用任何文本编辑器或字处理软件来建立一个程序文件，当然，最方便的方法还是使用 VFP 本身提供的编辑器。

7.1.1　程序文件的建立

在 VFP 中，可以通过以下方式创建程序文件。

1．命令方式

创建一个程序文件的最简单方法是在命令窗口中使用命令 MODIFY COMMAND。其命令格式为：

MODIFY COMMAND [〈文件名〉]

图 7-1　程序编辑器

说明：

① 若省略文件名，系统打开一个编辑器窗口，其标题栏中是默认的文件名：程序 1，如图 7-1 所示；若省略扩展名，则默认为.PRG。

② 若命令中的文件名是已经存在的程序文件，将打开该程序文件。

③ 文件名中如果包含通配符“*”或“?”，所有与之相匹配的程序文件都将被打开。

④ 关闭编辑器有两种方式：若不存盘退出，可按〈Esc〉键或〈Ctrl〉+〈Q〉；若存盘退

出，可按〈Ctrl〉+〈S〉键。

2．菜单方式

从“文件”菜单中选择“新建”命令，或者单击常用工具栏上的“新建”按钮，打开“新建”对话框。在对话框中选择“程序”，然后选择“新建文件”，即可打开编辑器窗口，如图 7-1 所示。其标题栏中是默认的标题：程序 1。

从“文件”菜单中选择“保存”或“另存为”命令，可以将程序文件存盘。

3．项目管理器方式

在项目管理器的“代码”选项卡中选择“程序”项，然后单击“新建”按钮，如图 7-2 所示，也可以打开编辑器窗口，编写新的程序。

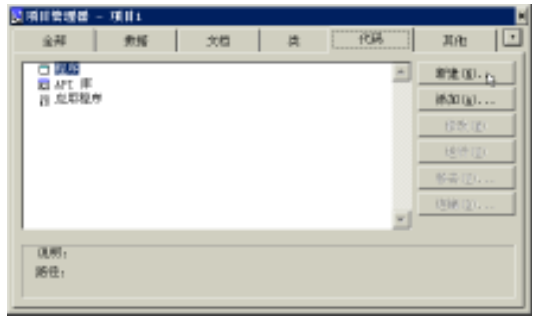


图7-2 “代码”选项卡

7.1.2 程序文件的运行

1．命令方式

VFP 中运行程序的命令是 DO 命令。DO 命令可以调用并执行指定的程序文件，其命令格式为：

DO〈程序文件名〉

说明：

① 如果用户运行的程序不带扩展名，则 VFP 按下列顺序寻找并执行程序：.EXE、.APP、.FXP、.PRG，因此在运行程序文件时，最好加上扩展名。

② 执行 DO 命令时，系统将到默认的路径上查找程序文件，如果文件不在当前路径，则应在程序名前加上路径。

③ 一个程序文件中可以包含其他的 DO 命令。DO 命令允许最多 128 级嵌套。

④ 在命令窗口执行命令：

DO ?

将打开“运行”对话框，如图 7-3 所示。选择想要运行的程序文件，单击“运行”按钮，即可运行程序。

2．菜单方式

从“程序”菜单中选择“运行”命令，将打开“运行”对话框。选择想要运行的程序文件，单击“运行”按钮，即可运行程序。

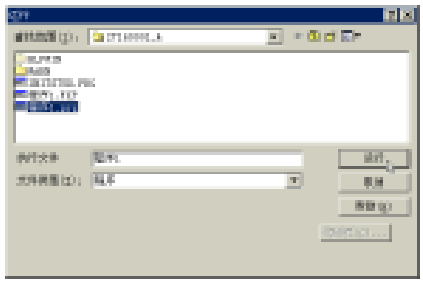


图7-3 “运行”对话框

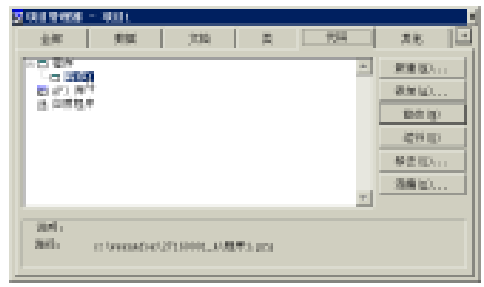


图7-4 在“项目管理器”中运行程序

说明：单击常用工具栏上的“运行”按钮，将运行当前编辑器中的程序文件。

3．项目管理器方式

在项目管理器的“代码”选项卡中展开“程序”项，选择项目中想要运行的程序文件，如图 7-4 所示。然后单击“运行”按钮，也可以运行程序文件。

7.1.3 程序文件的修改

程序保存后可以随时修改，可以按以下方式打开想要修改的程序文件。

1．命令方式

在命令窗口输入命令 MODIFY COMMAND [**〈文件名〉**]**，即可在编辑器中打开想要修改的程序文件。**

或者，在命令窗口输入命令 MODIFY COMMAND ?**，在“打开”对话框中选择想要修改的程序，单击“打开”按钮，即可在编辑器中打开想要修改的程序文件，如图 7-5 所示。**

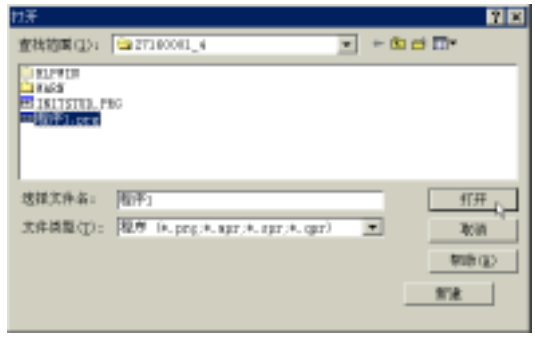


图7-5 “打开”对话框

2．菜单方式

从“文件”菜单中选择“打开”命令，或者单击常用工具栏上的“打开”按钮，打开“打开”对话框，选择想要修改的程序，单击“打开”按钮，即可在编辑器中打开想要修改的程序文件。

3．项目管理器方式

在项目管理器的“代码”选项卡中展开“程序”项，选择项目中想要修改的程序文件，然后单击“修改”按钮，即可在编辑器中打开想要修改的程序文件。

7.1.4 程序中常用的操作命令

1．结束程序运行的命令

当系统运行一个程序，执行完最后一条语句后，将自动结束本程序的运行而返回。VFP**还提供了几条专用的结束程序运行的命令。**

（1）返回调用程序命令

其格式为：

```
RETURN
```

说明：系统执行到该语句时，结束本程序的运行，返回到上级程序调用处的下一条语句继续执行；若无上级程序，则返回到命令窗口或 Windows **系统。**

（2）结束程序命令

其格式为：

```
CANCLE
```

说明：系统执行到该语句时，结束本程序的运行，清除所有的私有变量，关闭所有已打开的文件，直接返回到命令窗口或 Windows **系统。**

（3）退出系统命令

其格式为：

```
QUIT
```

说明：系统执行到该语句时，结束本程序的运行，清除所有的私有变量，关闭所有已打开的文件，退出 VFP **系统，返回** Windows **系统。**

以上语句可以放在程序的任何地方，也可以在一个程序中多次出现。

2．非格式化输出命令

程序运行的结果需要输出，显示在屏幕上、打印到纸上或输出到其他的设备。非格式化输出命令，可以依次计算并在 VFP **主窗口中显示各表达式的值。第** 2 **章中曾经介绍过的最简单的输出命令就是非格式化输出命令，其命令格式为：**

```
? | ?? [〈表达式列表〉]
```

说明：

① 若表达式多于一项，各表达式间要用逗号隔开，显示时各表达式值间各空一格。

② 若选用??，则从 VFP **主窗口当前光标处开始显示表达式的值；若选用?，则从** VFP **主窗口当前光标处的下一行第一列开始显示，如图** 7-6 **所示。**

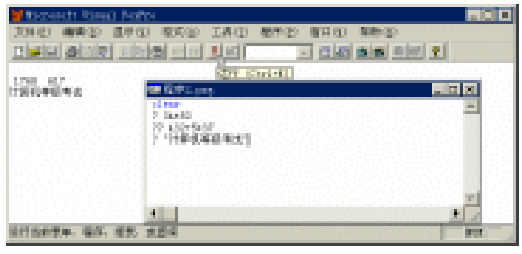


图7-6　**非格式输出**

③ 若在?中省略〈表达式列表〉，则输出一个空行。

④ 程序中的 CLEAR 命令用来清除 VFP 主窗口中的任何输出内容。

3．等待命令 WAIT

命令格式为：

WAIT [〈提示信息〉] [TO〈内存变量〉]

说明：

① 从屏幕当前光标处开始显示〈提示信息〉，同时暂停程序运行，等待用户从键盘输入一个字符，并将其存入指定的内存变量，接着继续运行程序。

② 本语句只能输入单个字符。

③〈提示信息〉可以是字符型表达式，若是字符串，要用引号括住。

④ 若缺省〈提示信息〉项，则屏幕显示“ 按任意键继续... ”。

⑤ 若缺省 TO〈内存变量〉项，则输入的字符不保存。

4．字符输入命令 ACCEPT

命令格式为：

ACCEPT [〈提示信息〉] TO 〈内存变量〉

说明：

① 从屏幕当前光标处开始显示〈提示信息〉，同时暂停程序运行，等待用户从键盘输入一字符串，并将其存入指定的内存变量，接着继续运行程序。

② 用户从键盘上输入字符串时，不必用引号括住；字符串输完后，按回车键表示结束。

③〈提示信息〉可以是字符型表达式，若是字符串，要用引号括住；若缺省该项则不显示任何提示信息。

5．任意类型数据输入命令 INPUT

命令格式为：

INPUT [〈提示信息〉] TO〈内存变量〉

功能：从屏幕当前光标处开始显示〈提示信息〉，同时暂停程序运行，等待用户从键盘输入一个数据，并将其存入指定的内存变量，接着继续运行程序。

说明：

① 用户由键盘上输入的数据可以是常数，也可以是表达式。但是不能不输入任何内容直接按〈Enter〉键。

② 输入的字符型常数，必须用引号括住；逻辑型常数，必须用圆点定界（.T.或.F.）；日期型常量，必须使用 CTOD（）函数转换，或用大括号括住（{^2002-10-1}）；数值型常数，可以直接输入数值。数据输完后，必须按〈Enter〉键表示结束。

③〈提示信息〉可以是字符型表达式，若是字符串，要用引号括住；若缺省该项则不显示任何提示信息。

④ 内存变量的类型依用户从键盘上输入的数据类型而定，可以是字符型、数值型、日期型或逻辑型。

【例 7-1】编写程序，由键盘输入一个圆半径，计算机自动计算并输出圆的面积。

```
CLEAR
INPUT "请输入圆的半径:" TO q
AREA = 3.14 * q * q
? "圆的面积＝",AREA
RETURN
```

运行该程序，若输入的圆半径为 12，则屏幕显示如图 7-7 所示。

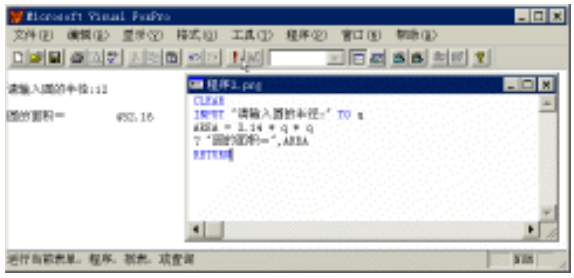


图7-7　计算圆的面积

6．内存变量赋值命令 STORE| =

赋值命令用来建立并初始化一个或多个内存变量。有两种格式的赋值命令：

```
STORE 〈表达式〉 TO 〈内存变量表〉
〈内存变量〉=〈表达式〉
```

说明：

① 内存变量表可以是多个内存变量，内存变量之间用逗号隔开。

② 命令执行时，首先计算表达式的值，然后将值赋予〈内存变量表〉中列出的各个内存变量。

③ 表达式的值应该是确定的。

④ 内存变量的类型由表达式的类型而定。

⑤ STORE 可以同时给多个变量赋值，而“ = ”只能给一个变量（或数组）赋值。

7．其他辅助命令

（1）注释命令

用于在程序中插入注释内容，以提高程序的可读性。其格式有三种形式：

```
*〈注释内容〉
NOTE〈注释内容〉
&&〈注释内容〉
```

说明：

① 以*、&&或 NOTE 开头的语句可以出现在程序的任何地方，称为注释行；&&还可以用于在命令语句的尾部加注释信息。

② 长注释行可分行书写，但需要在分行处加上续行符“ ;”。

程序运行时遇到注释行，系统将跳过它们直接执行其后的命令。

（2）清屏命令

用于清除 VFP 主窗口内容，并将光标置于 VFP 主窗口的左上角。其语法格式为：

CLEAR

说明：编程时通常把 CLEAR 用在程序的开始处。

8．程序书写格式

程序中的每个命令行都以回车键结尾，一行只能包含一条命令。若一条命令在一行写不下，也可分行书写，但需在分行处加上续行符“；”。

7.1.5 使用对话框

对话框是用户与应用程序之间交换信息的最佳途径之一。使用对话框函数可以在 VFP 的内部对话框中显示信息，等待用户单击按钮，然后返回一个整数以标明用户单击了哪个按钮。这种方法具有操作简单及快速的特点。其语法格式为：

[〈变量名〉] = MESSAGEBOX（〈信息内容〉[,〈对话框类型〉[,〈对话框标题〉]]）

说明：

①〈信息内容〉指定在对话框中出现的文本。〈信息内容〉中使用硬回车符（CHR（13））可使文本换行。对话框的高度和宽度随着〈信息内容〉的增加而增加，最多可有 1024 个字符。

②〈对话框类型〉指定对话框中出现的按钮和图标，一般有三个参数。其取值和含义分别见表 7-1、表 7-2、表 7-3。

表 7-1 参数 1——出现按钮

值	说　明	值	说　明
0	“确定”按钮	3	“是”、“否”和“取消”按钮
1	“确定”和“取消”按钮	4	“是”和“否”按钮
2	“终止”、“重试”和“忽略”按钮	5	“重试”和“取消”按钮

表 7-2 参数 2——图标类型

值	说　明	值	说　明
16	停止图标	48	感叹号“！”图标
32	问号“？”图标	64	信息图标

表 7-3 参数 3——默认按钮

值	说　明
0	指定默认按钮为第一按钮
256	指定默认按钮为第二按钮
512	指定默认按钮为第三按钮

上述三种参数值可以相加以达到所需要的样式。

③〈对话框标题〉指定对话框的标题。若缺省此项，系统给出默认的标题 Microsoft VFP。

下述代码将显示如图 7-8 所示的对话框。

msg = MESSAGEBOX （"请确认输入的数据是否正确！", 3 + 48 + 0, "数据检查"）

图7-8　信息对话框

④ MESSAGEBOX( )返回的值指明了在对话框中选择哪一个按钮，表 7-4 列出了函数的返回值及其说明。

表 7-4　函数的返回值

返　回　值	选 定 按 钮	返　回　值	选 定 按 钮
1	确定	5	忽略
2	取消	6	是
3	终止	7	否
4	重试		

⑤ 如果省略了某些可选项，必须加入相应的逗号分隔符。

⑥ 省略〈变量名〉，将忽略返回值。

在程序运行的过程中，有时需要显示一些简单的信息如警告或错误等，此时可以利用“ 信息对话框 ”来显示这些内容。当用户接收到信息后，可以单击按钮来关闭对话框，并返回单击的按钮值。

【例 7-2】在【例 7-1】中使用信息对话框来显示计算结果，如图 7-9 所示。

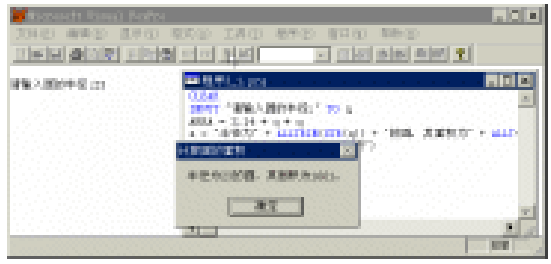


图7-9　使用信息对话框

只需改写程序代码为：

```
CLEAR
INPUT "请输入圆的半径:" TO q
AREA = 3.14 * q * q
a = "半径为" + ALLTRIM（STR（q））  + "的圆，其面积为" + ALLT（STR（AREA））  + "。"
= MESSAGEBOX（a,0,"计算圆的面积"）
RETURN
```

7.2 顺序结构

7.2.1 结构化程序设计

程序结构是指程序中命令或语句执行的流程结构。结构化程序是指设计的程序必须由三种基本控制结构组成，即顺序结构、选择结构与循环结构。结构化程序便于阅读和修改，提高了程序的可读性和可维护性。

需要说明的是，基本结构之间是可以互相嵌套的，在一个基本结构中又可以包含一个或者多个基本结构，例如，在循环结构中可以包含顺序结构，也可以包含选择结构。

7.2.2 顺序结构及其流程图

顺序结构是最简单的程序结构，它按命令在程序中出现的先后次序依次执行。前面的【例 7-1】、【例 7-2】都是顺序结构程序的例子。

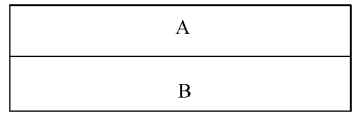


图 7-10 顺序结构

如图 7-10 所示是用N-S流程图表示的顺序结构，A和B两个框内的操作是顺序执行的。

7.2.3 顺序结构程序设计

【例 7-3】“鸡兔同笼”问题。鸡有 2 只脚，兔有 4 只脚，如果已知鸡和兔的总头数为 h，总脚数为 f 。问笼中鸡和兔各有多少只？

分析：设笼中有鸡 x 只，兔 y 只，由条件可得方程组：$\begin{cases} x+y=h \\ 2x+4y=f \end{cases}$

解方程组得：$\begin{cases} x=\dfrac{4h-f}{2} \\ y=\dfrac{f-2h}{2} \end{cases}$

编写程序如下：

```
CLEAR
INPUT "请输入笼中鸡和兔的总头数:" TO h
INPUT "  鸡和兔的总脚数:" TO f
x =  （ 4 * h - f ）  / 2
y =  （ f - 2 * h ）  / 2
? "  则笼中鸡有" +ALLT（STR（x））+"只，"
? "    兔有" +ALLT（STR（y））+"只。"
RETURN
```

运行该程序，假定输入笼中鸡和兔的总头数、总脚数分别为 18 和 44，则屏幕显示如图 7-11 所示。

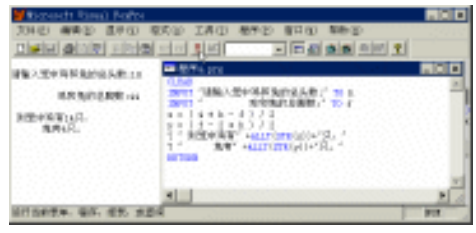


图 7-11 鸡兔同笼

【例 7-4】试分析以下程序的功能。

```
CLEAR
USE zgqk
ACCEPT "请输入姓名:" TO XM
INPUT "请输入基本工资:" TO GZ
LOCATE ALL FOR 姓名=XM
REPLACE 基本工资 WITH GZ
DISPLAY
USE
RETURN
```

该程序通过键盘输入某人姓名及基本工资，然后在数据表 zgqk.dbf 中按姓名查找到相应记录，用所输入的工资数值（GZ）替换该记录的基本工资字段值，最后显示该记录内容。

7.3 选择结构

计算机最重要的特点之一就是具有逻辑判断能力，它可以根据给定的不同逻辑条件，转向执行不同的程序方向，这些不同的转向就构成了分支结构，或称选择结构。

在 VFP 中，实现分支结构的语句有：

- IF … ELSE … ENDIF。
- DO CASE … ENDCASE。

这些语句又称为条件语句，条件语句的功能就是根据表达式的值有条件地执行一组语句。

顺序结构程序直观、简单，一般可以直接编写。对于选择结构等较复杂的程序设计，通常是先拟定解决问题的步骤，再画出流程图，最后根据流程图编写程序。

7.3.1 单条件选择语句 IF

1．单条件选择结构

单条件选择结构是最常用的双分支选择结构，其特点是：所给定的选择条件（条件表达式）的值如果为真，则执行 a_1 块；如果为假则执行 a_2 块。其一般形式如图 7-12 所示。

如果 条件			{该（选择）条件成立吗？}
	真	a_1 块	{条件成立时所执行的操作块，它一般为非空块}
	假	a_2 块	{条件不成立时所执行的操作块，它可为空块}

图7-12 单条件选择结构的流程图

说明：

① 这里的 a_1 块或 a_2 块可以是空操作块（简称空块，也就是不作任何处理的操作块）。当然，如果 a_1、a_2 操作块同时为空块的话，就失去了选择的意义。

② 为了养成良好的程序设计风格和习惯，如果必须设立空分支时，应该把它设在选择条件为假的相应分支（即 a_2 块）中。

③ 实现单条件选择结构的语句是 If 语句。

2．语法格式

单条件选择语句语法格式为：

```
IF〈条件〉
  [〈语句列 1〉]
[ELSE
  [〈语句列 2〉]
ENDIF
```

说明：

① IF、ELSE、ENDIF 必须各占一行。每一个 IF 都必须有一个 ENDIF 与之对应，即 IF 和 ENDIF 必须成对出现。

② ELSE 子句是可选的。

③〈条件〉可以是条件表达式或逻辑常量，根据〈条件〉的逻辑值，进行判断。

④ 如果〈条件〉为真（.T.），就执行〈语句列 1〉。如果〈条件〉为假（.F.），若有 ELSE 子句，则程序会执行 ELSE 部分的〈语句列 2〉；若无 ELSE 子句，则程序会直接转到 ENDIF 之后的语句继续执行。

⑤〈语句列 1〉和〈语句列 2〉中还可以包含 IF 语句，称为 IF 语句的嵌套。要注意，每次嵌套中的 IF 语句必须与 ENDIF 成对出现。

【例 7-5】求函数值。输入 x，计算 y 的值，其中：

$$y=\begin{cases} 4x & (x=0) \\ 15-2x & (x<0) \end{cases}$$

分析：该题是数学中的一个分段函数，它表示当 $x\geq 0$ 时，用公式 $y=4x$ 来计算 y 的值，当 $x<0$ 时，用公式 $y=15-2x$ 来计算 y 的值。在选择条件时，我们既可以选择 $x\geq 0$ 作为条件，也可以选择 $x<0$ 作为条件。在这里，我们选 $x\geq 0$ 作为选择条件。这时，当 $x\geq 0$ 为真时，执行 $y=4x$；为假时，执行 $y=15-2x$。

根据以上分析，画出流程图，如图 7-13 所示。

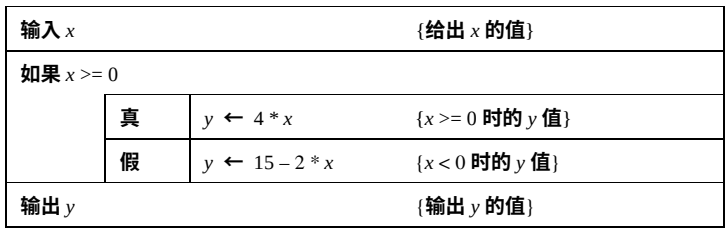


图7-13　计算y值的流程图

编写程序如下：

```
CLEAR
INPUT "请输入自变量 x 的值:" TO x
IF x >= 0
   y = 4 * x
ELSE
```

```
    y = 15 - 2 * x
ENDIF
? "函数值 y = ", y
```

运行该程序，假定输入自变量 x 的值为-37，则屏幕显示如图 7-14 所示。

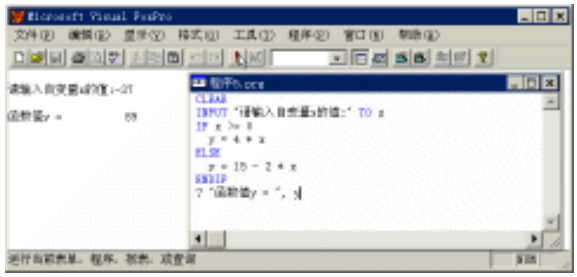


图7-14　求函数值

3．使用 IIF 函数

还可以使用 IIf 函数来实现一些比较简单的选择结构。IIf 函数的语法结构为：

```
IIf（〈条件〉,〈真部分〉,〈假部分〉）
```

说明：

①〈条件〉可以是条件表达式或逻辑常量。

②〈真部分〉是当条件为真时函数返回的值，可以是任何表达式。

③〈假部分〉是当条件为假时函数返回的值，可以是任何表达式。

④ 语句 y = IIf（〈条件〉,〈真部分〉,〈假部分〉）相当于：

```
IF 〈条件〉
   y =〈真部分〉
ELSE
   y =〈假部分〉
ENDIF
```

【例 7-6】和【例 7-5】中的程序代码可以改为：

```
CLEAR
INPUT "请输入自变量 x 的值:" TO x
y = IIF（x >= 0, 4 * x, 15 - 2 * x）
? "函数值 y = ", y
```

4．IF 语句的嵌套

如果在 IF 语句中操作块 a_1 块（语句列 1）或 a_2 块（语句列 2）本身又是一个 IF 语句，则称为 IF 语句的嵌套。

【例 7-7】铁路托运行李，从甲地到乙地，规定每张客票托运费计算方法是：行李重量不超过 50kg 时，每公斤 0.25 元；超过 50kg 而不超过 100kg 时，其超过部分每公斤 0.35 元；超过 100kg 时，其超过部分每公斤 0.45 元。编写程序，输入行李重量，计算并输出托运的费用。

分析：设行李重量为 w 公斤，应付运费为 x 元，则运费公式为：

$$x=\begin{cases}0.25\times w & (w=50)\\ 0.25\times 50+0.35\times(w-50) & (50<w=100)\\ 0.25\times 50+0.35\times 50+0.45\times(w-100) & (w>100)\end{cases}$$

根据以上分析，画出流程图如图 7-15 所示。

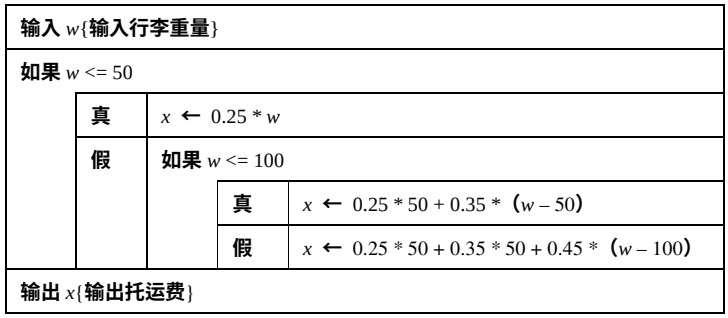


图7-15　计算托运费的流程图

编写程序代码如下：

```
CLEAR
INPUT "请输入行李重量:" TO w
IF w <= 50
   x = 0.25 * w
ELSE
   IF w <= 100
x = 0.25 * 50 + 0.35 *  （w - 50）
   ELSE
x = 0.25 * 50 + 0.35 * 50 + 0.45 *  （w - 100）
   ENDIF
ENDIF
? "托运费  x = ", x
```

运行该程序，假定输入行李重量为 73kg，则屏幕显示如图 7-16 所示。

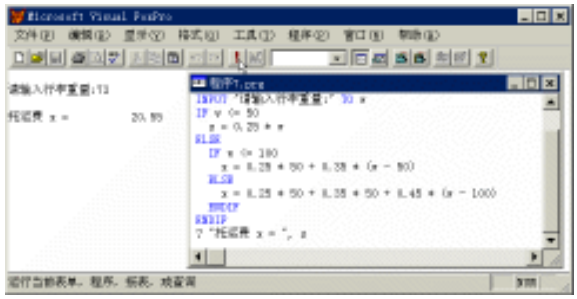


图7-16　计算托运费

【例 7-8】某百货公司为了促销，采用购物打折扣的优惠办法，每位顾客一次购物：

① 在 1000 元以上者，按九五折优惠。
② 在 2000 元以上者，按九折优惠。
③ 在 3000 元以上者，按八五折优惠。
④ 在 5000 元以上者，按八折优惠。

编写程序，输入购物款数，计算并输出优惠价。

分析：设购物款数为 x 元，优惠价为 y 元，优惠付款公式为：

$$y=\begin{cases} x & (x<1000) \\ 0.95x & (1000\le x<2000) \\ 0.9x & (2000\le x<3000) \\ 0.85x & (3000\le x<5000) \\ 0.8x & (x\ge 5000) \end{cases}$$

根据以上分析，画出流程图如图 7-17 所示。

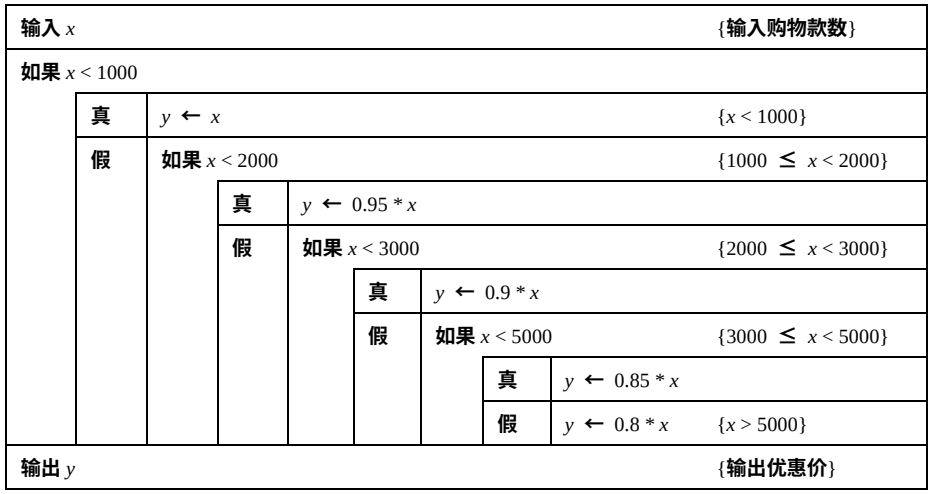


图7-17　计算优惠价的流程图

编写程序代码如下：

```
CLEAR
INPUT "请输入所购商品总金额:" TO x
IF x < 1000
   y = x
ELSE
   IF x < 2000
y = 0.95 * x
   ELSE
IF x < 3000
   y = 0.9 * x
ELSE
   IF x < 5000
```

```
        y = 0.85 * x
          ELSE
        y = 0.8 * x
          ENDIF
        ENDIF
      ENDIF
ENDIF
? "优惠价为：", y
```

运行该程序，假定输入所购商品总金额为 1875，则屏幕显示如图 7-18 所示。

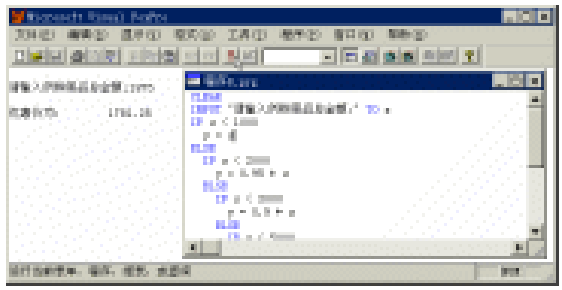


图7-18　计算优惠价

7.3.2　多分支条件选择语句 DO CASE

虽然使用嵌套的办法可以利用 IF 语句实现多分支选择，但是，用 IF 语句编写的程序会比较长，程序的可读性会明显降低。为此 VFP 提供了多分支条件选择语句——DO CASE 语句，来实现多分支选择结构。

1．多分支条件选择结构

多分支选择结构的根本特点是：从多个分支结构中，选择第一个条件为真的路线作为执行的路线。即所给定的选择条件 1 为真时，执行 a_1 块；如果为假则继续检查下一个条件。如果条件都不为真，就执行其他操作块（a_{n+1} 块），如果没有其他操作块，则不作任何操作就结束选择，如图 7-19 所示。

情形			
	条件 1	a_1 块	{条件 1 成立时所执行的操作块}
	条件 2	a_2 块	{条件 2 成立时所执行的操作块}
	…	…	
	条件 n	a_n 块	{条件 n 成立时所执行的操作块}
	其他	a_{n+1} 块	{条件都不成立时所执行的操作块}

图7-19　多条件多分支选择结构的流程图

2．多分支条件选择语句 DO CASE 的语法格式

DO CASE 语句的语法格式为：

```
DO CASE
   CASE 〈条件 1〉
      [〈语句列 1〉]
   [CASE 〈条件 2〉
      [〈语句列 2〉]
    ...
   [OTHERWISE
      [〈其他语句列〉]
ENDCASE
```

说明：

① DO CASE、CASE、OTHERWISE 和 ENDCASE 必须各占一行。每个 DO CASE 必须有一个 ENDCASE 与之对应，即 DO CASE 和 ENDCASE 必须成对出现。

②〈条件 1〉可以是条件表达式或逻辑常量。

③ 在执行 DO CASE 语句时，依次判断各〈条件〉是否满足。若〈条件 1〉的值为真（.T.），就执行相应的〈语句列 1〉，直到遇到下一个 CASE、OTHERWISE 或 ENDCASE。

④ 相应的〈语句列 1〉执行后不再判断其他〈条件〉，直接转向 ENDCASE 后面的语句。因此，在一个 DO CASE 结构中，最多只能执行一个 CASE 子句。

⑤ 如果没有一个条件为真，就执行 OTHERWISE 后面的〈其他语句列〉，直到 ENDCASE。如果没有 OTHERWISE，则不作任何操作就转向 ENDCASE 后面的语句。

⑥ 语句列中可以嵌套各种控制结构的命令语句。

【例 7-9】在【例 7-8】中使用 DO CASE 语句来计算优惠价，只需将程序代码改为：

```
CLEAR
INPUT "请输入所购商品总金额:" TO x
DO CASE
    CASE x < 1000
   y = x
    CASE x < 2000
   y = 0.95 * x
    CASE x < 3000
   y = 0.9 * x
    CASE x < 5000
   y = 0.85 * x
    OTHERWISE
   y = 0.8 * x
ENDCASE
? "优惠价为：", y
```

说明：程序运行结果与【例 7-8】相同，但是代码却清晰多了。

7.4 循环结构

在编程中常常遇到这样的情况：某一类问题的计算和处理方法完全一样，只是要求重复计算多次，而每次使用的数据都按照一定的规律在改变。例如，需要对一个班 30 名学生的成绩进行检查，将不及格者打印出来。类似这样的问题，如果使用顺序结构和选择结构语句来编程，代码将会很长，效率很低。此时，就要用到循环结构。

程序设计中的循环结构（简称循环）是指在程序中，从某处开始有规律地反复执行某一操作块（或程序块）的现象。被重复执行的该操作块（或程序块）称为循环体，循环体的执行与否及次数多少视循环类型与条件而定。当然，无论何种类型的循环结构，其共同的特点是必须确保循环体的重复执行能被终止（即非无限循环）。

7.4.1 循环结构语句

按照循环体执行的方式和条件，循环结构分为当型、直到型与步长型。由于“直到型”循环是“当型”循环派生出来的循环形式，二者可以互换，所以在 VFP 中只提供了“当型”、“步长型”和“表扫描型”三种循环语句：

- DO WHILE ... ENDDO（当型循环）。
- FOR ... ENDFOR（步长型循环）。
- SCAN ... ENDSCAN（表扫描型循环）。

7.4.2 当型循环命令 DO WHILE

想要在某一条件满足时执行循环，可以使用当型循环（DO WHILE）结构。当型循环的 N-S 流程图如图 7-20 所示。

图7-20 当型循环的流程图

当型循环的语法格式为：

```
DO WHILE 〈条件〉
  [〈命令列〉]
  [EXIT]
  [LOOP]
ENDDO
```

说明：

①〈条件〉可以是条件表达式或逻辑常量。根据〈条件〉的逻辑值进行判断，如果〈条件〉的值为.T.，则执行 DO WHILE 和 ENDDO 之间的循环体。如果〈条件〉的值为.F.，则结束循环，转去执行 ENDDO 之后的命令。

每执行一遍循环体，程序自动返回到 DO WHILE 语句，判断一次〈条件〉。

②〈命令列〉是指定〈条件〉为真时执行的那组 VFP 命令，即循环体。

③ EXIT 是无条件结束循环命令，使程序跳出 DO WHILE…ENDDO 循环，转去执行 ENDDO 后的第一条命令。EXIT 只能在循环结构中使用，但是可以放在 DO WHILE…ENDDO 中任何地方。

④ LOOP 将控制直接转回到 DO WHILE 语句，而不执行 LOOP 和 ENDDO 之间的命令。因此 LOOP 称为无条件循环命令，只能在循环结构中使用。

⑤ DO WHILE、ENDDO 必须各占一行。每一个 DO WHILE 都必须有一个 ENDDO 与其对应，即 DO WHILE 和 ENDDO 必须成对出现。

【例 7-10】求 1 + 2 + 3 + … + 100 的值。

分析：可以采用累加的方法，用变量 *s* 来存放累加的和（开始为 0），用变量 *n* 来存放“加数”（加到 *s* 中的数）。这里 *n* 又称为计数器，从 1 开始到 100 为止。

根据以上分析画出流程图，如图 7-21 所示。

s ← 0		{累加器赋初值}
n ← 1		{计数器赋初值}
当 *n* <= 100		{当计数器的值不超过 100 时}
	s ← *s* + *n*	{累加}
	n ← *n* + 1	{计数器增 1}
输出 *s*		{输出累加和}

图7-21　累加流程图

编写程序代码如下：

```
CLEAR
s = 0
n = 1
DO WHILE n <=100
    s = s + n
    n = n + 1
ENDDO
? "1 + 2 + 3 + ... + 100 =", s
```

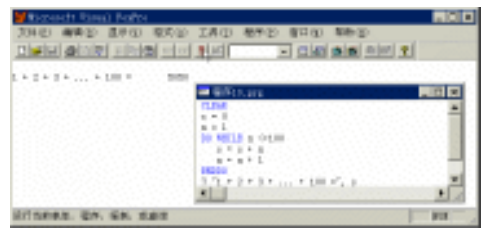


图7-22　计算累加和

运行该程序，屏幕显示如图 7-22 所示。

【例 7-11】非负整数 *n* 的阶乘定义如下：

$$n!=\begin{cases}1 & n=0\\ 1\times 2\times\Lambda\times n & n>0\end{cases}$$

输入整数 *n*，求阶乘 *n*!。

分析：可以采用累乘的方法，用变量 t 来存放累乘的积（开始为 1），用变量 *i* 来存放“乘数”。*i* 从 1 开始到 *n* 为止。

根据以上分析画出流程图，如图 7-23 所示。

输入 *n*		
t ← 1		{累加器赋初值}
i ← 1		{计数器赋初值}
当 *i* <= *n*		{当计数器的值不超过 100 时}
	t ← *t* × I	{累加}
	i ← *i*+ 1	{计数器增 1}
输出 *t*		{输出累加和}

图7-23　累加流程图

编写程序代码如下：

```
CLEAR
INPUT "请输入一个整数:" TO n
t = 1
i = 1
DO WHILE i <= n
    t = t * i
    i = i + 1
ENDDO
? Str（n） + "! = " + ALLT（Str（t,20,0））
```

运行该程序，假定输入的整数为 15，则屏幕显示如图 7-24 所示。
另外，为了防止数据溢出，限制输入的整数不超过 20。为此，改写程序代码为：

```
CLEAR
DO WHILE .T.
  INPUT "请输入一个整数:" TO n
  IF n < 0 OR n > 20
    MESSAGEBOX（"请输入不超过 20 的非负整数!"）
    LOOP
  ELSE
    EXIT
  ENDIF
ENDDO
t = 1
i = 1
DO WHILE i <= n
    t = t * i
    i = i + 1
ENDDO
? Str（n） + "! = " + ALLT（Str（t,20,0））
```

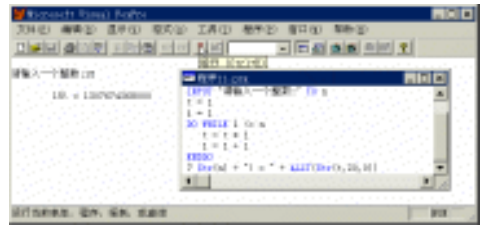


图7-24　计算阶乘

【例 7-12】输入两个正整数，求它们的最大公约数。

分析：求最大公约数可以用“ 辗转相除法 ”，方法如下：

① 以大数 *m* 作被除数，小数 *n* 做除数，相除后余数为 *r*。

② 若 *r* ≠ 0，则 *m* ← *n*，*n* ← *r*，继续相除得到新的 *r*。若仍有 *r* ≠ 0，则重复此过程，直到 *r* = 0 为止。

③ 最后的 *n* 就是最大公约数。

根据此分析画出流程图如图 7-25 **所示。**

图7-25　**辗转相除法**

编写程序代码为：

```
CLEAR
DO WHILE .T.
   INPUT "请输入第一个整数:" TO s
   INPUT "请输入第二个整数:" TO t
   IF s * t = 0
      MESSAGEBOX（"两数都不能为 0!"）
      LOOP
   ELSE
      EXIT
   ENDIF
ENDDO
IF s < t
    m = t
    n = s
ELSE
    n = t
    m = s
ENDIF
r = m % n
DO WHILE r != 0
    m = n
    n = r
    r = m % n
```

```
ENDDO
? ALLT（Str（s））  + " 与 " + ALLT（Str（t））  + "的公约数为：" + ALLT（Str（n））
```

运行该程序，输入两个整数 23456，34，屏幕显示如图 7-26 所示。

图7-26　求最大公约数

【例 7-13】输入一个正整数，利用当型循环判断是否为素数。

分析：所谓“ 素数 ”是指除了 1 和该数本身，不能被任何整数整除的数。判断一个自然数 n（$n \geq 3$）是否为素数，只要依次用 $2 \sim \sqrt{n}$ 作除数去除 n，若 n 不能被其中任何一个数整除，则 n 即为素数。根据上述分析，画出流程图如图 7-27 所示。

<table>
<tr><td colspan="3">输入 n</td></tr>
<tr><td colspan="3">$i \leftarrow 2$，$s \leftarrow 0$</td></tr>
<tr><td colspan="3">当　$i <= \sqrt{n}$　and $s = 0$</td></tr>
<tr><td></td><td colspan="2">如果 n 被 i 整除</td></tr>
<tr><td></td><td>真</td><td>$s \leftarrow 1$</td></tr>
<tr><td></td><td>假</td><td>$i \leftarrow i + 1$</td></tr>
<tr><td colspan="3">如果 $s = 0$</td></tr>
<tr><td></td><td>真</td><td>输出：n 是素数</td></tr>
<tr><td></td><td>假</td><td>输出：n 不是素数</td></tr>
</table>

图7-27　判断素数的流程图

编写程序代码为：

```
CLEAR
INPUT "请输入一个整数:" TO n
s = 0
i = 2
DO WHILE   i <= SQRT（n）  AND s = 0
    IF n  %  i = 0
       s = 1
    ELSE
       i = i + 1
    ENDIF
ENDDO
```

```
IF s = 0
   a = '是一个素数'
ELSE
   a = '不是素数'
ENDIF
= MESSAGEBOX（ALLT（STR（n））  + a, 64 + 0 + 0, "信息"）
```

运行程序，输入整数 1234567，屏幕显示如图 7-28 所示。

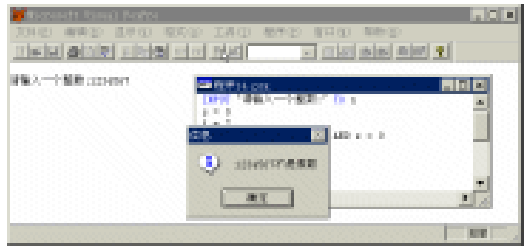


图7-28　判断素数

当型循环结构的根本特点是：当所给定循环条件为真时，就反复执行其循环体；当该条件为假时，则终止执行其循环体，而去执行其后继命令。显然，若它的循环初始条件已是假时，则并不执行其循环体，故它的循环体执行次数最少可为零。

使用当型循环结构可以事先并不清楚循环的次数，但应知道什么时候结束循环的执行。

为使程序最终能退出 DO WHILE 命令引起的循环，在没有使用 EXIT 的情况下，在每次程序的循环过程中必须修改程序给出的循环条件。否则程序将永远退不出循环，这种情况称作无限循环或死循环。在程序中要避免出现无限循环。

7.4.3　步长型循环命令 FOR

若事先知道循环次数，则可以使用步长型循环（FOR...ENDFOR）结构。步长型循环可以根据给定的次数重复执行循环体。其语法格式为：

```
FOR 〈内存变量〉=〈初值〉 TO 〈终值〉 [STEP 〈步长值〉]
    [〈命令列〉]
    [EXIT]
    [LOOP]
ENDFOR | NEXT
```

说明：

①〈内存变量〉是一个作为计数器的内存变量或数组元素，在 FOR...ENDFOR 执行之前该变量可以不存在。〈初值〉是计数器的初值，〈终值〉是计数器的终值，〈步长值〉是计数器值的增长或减少量。如果〈步长值〉是负数，则计数器被减小。如果省略 STEP 子句，则默认〈步长值〉是 1。〈初值〉、〈终值〉和〈步长值〉均为数值型表达式。

②〈命令列〉指定要执行的一个或多个 VFP 命令。

③ EXIT 跳出 FOR...ENDFOR 循环，转去执行 ENDFOR 后面的命令。可把 EXIT 放在 FOR...ENDFOR 中任何地方。

④ LOOP 将控制直接转回到 FOR 子句，而不执行 LOOP 和 ENDFOR 之间的命令。

⑤ FOR、ENDFOR | NEXT 必须各占一行。FOR 和 ENDFOR | NEXT 必须成对出现。

循环的执行过程是：开始时首先把〈初值〉、〈终值〉和〈步长值〉读入，然后〈内存变量〉的值与〈终值〉比较，如果〈内存变量〉的值在〈初值〉与〈终值〉范围内，则执行 FOR 与 ENDFOR 之间的命令，然后〈内存变量〉按〈步长值〉增加或减小，重新比较，直到〈内存变量〉的值不在〈初值〉与〈终值〉范围内，结束循环，转去执行 ENDFOR 后面的第一条命令。

如果在 FOR...ENDFOR 之间改变〈内存变量〉的值，将影响循环执行的次数。

〈命令列〉中可以嵌套控制结构的命令语句（IF、DO CASE、DO WHILE、FOR、SCAN）。

“步长型”循环结构是当型循环结构衍生出来的一种特殊变型。很容易将例 7-13 改为“步长型”循环结构。

【例 7-14】利用“步长型”循环判断素数的程序。

编写程序代码：

```
CLEAR
INPUT "请输入一个整数:" TO n
s = 0
i = 2
FOR i = 2 TO SQRT（n）
    IF n % i = 0
      s = 1
      EXIT
    ENDIF
ENDFOR
IF s = 0
    a = '是一个素数'
ELSE
    a = '不是素数'
ENDIF
= MESSAGEBOX（ALLT（STR（n））  + a, 64 + 0 + 0, "信息"）
```

程序的运行结果与【例 7-13】完全相同。

【例 7-15】求 1! + 2! + 3! + … + 20!的值。

分析：结合【例 7-10】和【例 7-11】，采用循环嵌套的方法，见流程图 7-29。

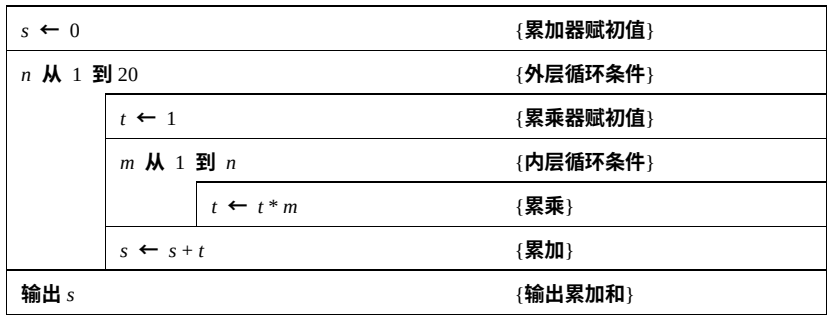


图7-29　流程图

编写程序代码为：

```
CLEAR
s = 0
FOR n = 1 TO 20
    t = 1
    FOR m = 1 TO n
        t = t * m
        ENDFOR
        s = s + t
ENDFOR
= MESSAGEBOX（"1!+2!+...+20!= " + ALLT（STR（s,20））, 64 + 0 + 0, "信息"）
```

图7-30　求阶乘之和

程序运行后显示如图 7-30 **所示。**

说明：

① 在使用循环嵌套时要注意内外循环不能交叉：

```
FOR X                FOR X
  FOR Y                FOR Y
  M                    M
  ENDFOR               ENDFOR
ENDFOR               ENDFOR
```

正确　　错误

② 内外循环的循环变量不能同名。

③ 上例可以不使用双重循环，请自行分析如图 7-31 **所示的流程图，并写出代码。**

s ← 0		{**累加器赋初值**}
t ← 1		{**累乘器赋初值**}
n **从** 1 **到** 20		{**外层循环条件**}
	t ← *t* * *m*	{**累乘**}
	s ← *s* + *t*	{**累加**}
输出 *s*		{**输出累加和**}

图7-31　流程图

7.5　数组

在许多场合，使用数组可以缩短和简化程序，因为可以利用下标值设计一个循环，高效地处理多种情况。

7.5.1　数组的概念

1．数组与数组元素

数组是用一个统一的名称表示的、顺序排列的一组变量。数组中的变量称为数组元素，用数字（下标）来标识它们，因此数组元素又称为下标变量。

可以用数组名及下标惟一地识别一个数组的元素，比如 A（5）表示名称为 A 的数组中顺序号（下标）为 5 的那个数组元素（变量）。

说明：

① 数组的命名与简单变量的命名规则相同。

② 下标必须用括号括起来，不能把数组元素 A（5）写成 A5，后者是简单变量。

③ 下标可以是常数、变量或表达式。下标还可以是下标变量（数组元素），如 B（A（4）），若 A（4） = 6，则 B（A（4）） 就是 B（6）。

④ 下标必须是整数，否则将被自动取整（舍去小数部分）。如 N（2.6）将被视为 N（2）。

2．数组的维数

如果一个数组的元素只有一个下标，则称这个数组为一维数组。例如，数组 S 有 30 个元素 S（1）、S（2）、S（3）、…、S（30），依次保存 30 个学生的一门功课的成绩，则 S 为一维数组。一维数组中的各个元素又称为单下标变量。

一维数组中的下标又称为索引（Index）。

如果有 30 个学生，每个学生有 5 门功课的成绩，见表 7-5。

表 7-5　学生成绩表

	语　文	数　学	外　语	物　理	化　学
学生 1	85	60	55	78	88
学生 2	69	74	80	76	79
学生 3	77	86	72	80	95
…	…	…	…	…	…
学生 30	88	90	75	88	82

这些成绩可以用有两个下标的数组来表示，如第 i 个学生第 j 门课的成绩可以用 S（i, j）表示。其中 i 表示学生号，称为行下标（i = 1，2，…，30）；j 表示课程号，称为列下标（j = 1，2，3，4，5）。有两个下标的数组称为二维数组，其中的数组元素称为双下标变量。在 VFP 中允许定义一维或二维数组。

VFP 对数组的大小和数据类型不作任何限制，甚至同一数组中的数组元素可以具有不同的数据类型，而对数组大小的惟一限制就是可用内存空间的大小。

7.5.2　声明与使用数组

1．数组的声明

数组在使用前必须先声明。声明的语法格式为：

{DIMENSION | DECLEAR} 〈数组名〉(〈行数〉[, 〈列数〉])

如：DIMENSION A（6,3）表示创建一个名为 A、具有 6 行 3 列的私有数组，只能在命令所在的过程及其所调用的过程中使用。

说明：

① 全局变量数组在整个 VFP 工作期中可以被任何程序访问，声明全局数组的格式为：

PUBLIC 〈数组名〉(〈行数〉[, 〈列数〉])

② 局部变量数组只能在创建它们的过程或函数中使用和更改，不能被高层或低层的程序访问，声明局部数组的格式为：

LOCAL 〈数组名〉(〈行数〉[,〈列数〉])

2．数组的赋值

数组在声明之后，每个元素被默认地赋予.F.值。

可以单独为某一个数组元素赋值。如

A（1,2）= 11　或　STORE 11 TO A（1,2）

都是将数值 11 赋予 A 数组的第一行第二列元素。也可以用一个命令为一个数组的所有元素赋相同的值。如

A = 13　或　STORE 13 TO A

都是将数值 13 赋予 A 数组的每一个元素。

【例 7-16】随机产生 10 个两位整数，找出其最大值、最小值和平均值。

分析：问题可以分为两部分，一个是产生 10 个随机整数，一个是对这 10 个整数求最大、最小以及平均值。为此，我们需要使用数组。

根据以上分析画出流程图，图 7-32 产生随机整数，图 7-33 实现查找和计算。

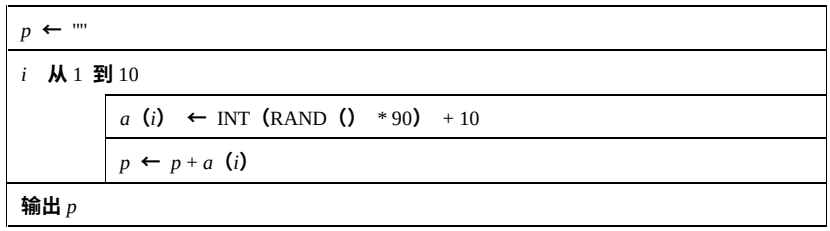


图7-32　生成随机整数流程图

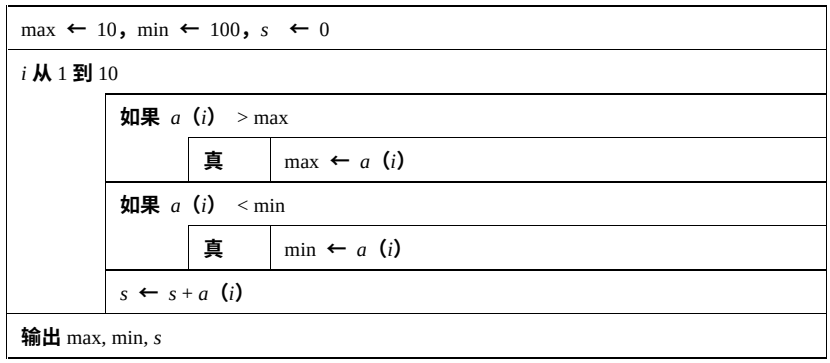


图7-33　求最大、最小、平均值流程图

编写程序代码为：

```
CLEAR
DIMENSION a（10）
```

```
FOR i = 1 TO 10
   a（i） = INT（RAND（） * 90） + 10
   ? a（i）
ENDFOR
min = 100
max = 10
s = 0
FOR i = 1 TO 10
   IF a（i） > max
      max = a（i）
   ENDIF
   IF a（i） < min
      min = a（i）
   ENDIF
   s = s + a（i）
ENDFOR
? "最大值 = " + STR（max,3）
? "最小值 = " + STR（min,3）
? "平均值 = " + STR（s / 14,6,2）
```

图7-34 求最大值、最小值和平均值

程序的运行结果如图 7-34 **所示。**

说明：如果要求产生的随机整数互不相同，可以将代码改为：

```
CLEAR
DIMENSION a（10）
FOR i = 1 TO 10
   yes = 1
   DO WHILE yes = 1
    x = INT（RAND（） * 90） + 10
    yes = 0
    FOR j = 1 TO i - 1
      IF x = a（j）
         yes = 1                    && 如与前面的元素相同，则返回到 Do 循环
         EXIT
      ENDIF
   ENDFOR
ENDDO
a（i） = x
? a（i）
ENDFOR
min = 100
max = 10
s = 0
FOR i = 1 TO 10
```

```
    IF a（i） > max
      max = a（i）
    ENDIF
    IF a（i） < min
      min = a（i）
    ENDIF
    s = s + a（i）
  ENDFOR
  ? "最大值 = " + STR（max,3）
  ? "最小值 = " + STR（min,3）
  ? "平均值 = " + STR（s / 14,6,2）
```

其中变量 *x* 用来存放刚产生的随机整数，变量 yes 用来作为标志，如果 *x* 与已放入数组中的某个随机整数相重，则 yes 为 1，否则为 0。当 yes 为 0 时将退出第二层循环 DO...ENDFOR，即可把随机数放入数组之中，即 *a*（*i*）= *x*。

3．重新定义数组的维数

重新执行 DIMENSION 命令可以改变数组的维数和大小。数组的大小可以增加或减少，一维数组可以转换为二维数组，二维数组可以降低为一维数组。

如果数组中元素的数目增加了，就将原数组中所有元素的内容复制到重新调整过的数组中，增加的数组元素初始化为“假”（.F.）。

【例 7-17】斐波那契（Fibonacci）数列问题。

Fibonacci 数列问题起源于一个古典的有关兔子繁殖的问题：假设在第 1 个月时有一对小兔子，第 2 个月时成为大兔子，第 3 个月时成为老兔子，并生出一对小兔子（一对老，一对小）。

第 4 个月时老兔子又生出一对小兔子，上个月的小兔子变成大兔子（一对老，一对大，一对小）。第 5 个月时上个月的大兔子成为老兔子，上个月的小兔子变成大兔子，两对老兔子生出两对小兔子（两对老，一对中，两对小）……。

这样，各月的兔子对数为 1，1，2，3，5，8，…。

这个数列就是 Fibonacci 数列。其中第 *n* 项的计算公式为：

Fib（*n*） = Fib（*n*-1）+ Fib（*n*-2）

分析：使用数组使得问题的解决非常简单，根据计算公式，很容易画出流程图如图 7-35 所示。

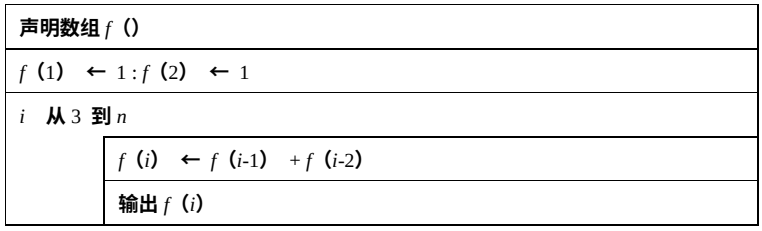


图7-35 Fibonacci数列问题流程图

编写代码如下：

```
CLEAR
DIMENSION F（1,2）
F（1,1） = "Fib（1）"
F（1,2） = 1
INPUT "请输入一个整数：" TO n
DIME F（n,2）
F（2,1） = "Fib（2）"
F（2,2） = 1
FOR I = 3 TO n
  F（i,1） = "Fib（" + ALLT（STR（i）） + "）"
  F（i,2） = F（i-1,2） + F（i-2,2）
ENDFOR
FOR I = 1 TO n
  ? F（i,1）,F（i,2）
ENDFOR
```

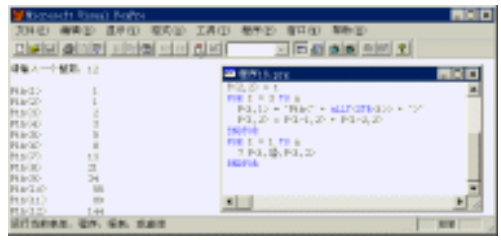


图7-36 求Fibonacci数列

程序运行结果，如图 7-36 所示。

4．数组变量的释放

使用 RELEASE 命令可以从内存中释放变量和数组。其语法是：

```
RELEASE {〈变量列表〉|〈数组名列表〉}
```

其中各变量或数组名用逗号分隔。

5．二维数组表示为一维数组

假如我们建立了一个二维数组，其下标也可以使用一维数组表示法来表示。如

```
DIMENSION A（6, 3）
FOR I=1 TO 18
   A（I） = I
ENDFOR
```

这样一来可以使代码更为简单。下面的公式可以将二维数组表示法转换成一维数组表示法：

序号（一维数组）=（行数 – 1）*列数 + 列数

或使用 AELEMENT( )函数也能取得一维数组表示法的元素位置，即

序号（一维数组）= AELEMENT（数组名，行数，列数）

【例 7-18】设有一个 5×5 的方阵，其中元素是由计算机随机生成的小于 100 的整数。求出：① 主对角线上元素之和；② 方阵中最大的元素。

分析：方阵中的元素可以用一个二维的数组表示。赋值时转换成一维数组，计算时则利用二维数组的性质。

编写代码如下：

```
CLEAR
DIMENSION a（5,5）
```

```
FOR i = 1 TO 25
   yes = 1
   DO WHILE yes = 1
      x = INT（RAND（）  * 100）
      yes = 0
      FOR j = 1 TO i - 1
         IF x = a（j）
            yes = 1                    && 如与前面的元素相同，则返回到 Do 循环
            EXIT
         ENDIF
      ENDFOR
   ENDDO
      a（i）  = x
ENDFOR
max = 0
FOR i = 1 TO 5
   f = ""
   FOR j = 1 TO 5
      IF max < a（i,j）
         max = a（i,j）
         p = i
         q = j
      ENDIF
      f = f + STR（a（i,j）,4）
   ENDFOR
   ? f
ENDFOR
s = 0
FOR i = 1 TO 5
      s = s + a（i,i）
ENDFOR
? "对角线元素之和为：", s
? "各元素中最大者是：", max
```

图7-37　矩阵计算

程序运行结果，如图 7-37 所示。

7.5.3　数组数据的处理

1．处理数组元素的函数

数组提供了一种快速排序数据的方法。如果数据保存在数组中，就可以很方便地对其进行检索、排序或其他各种操作。可以使用如下函数来处理数组元素：

（1）ASORT 函数

ASORT 函数按升序或降序对数组中的元素排序。其语法格式为：

ASORT（〈数组名〉[,〈元素号〉[,〈元素个数〉[,〈排序方向〉]]]）

说明：

① 返值数值型数据，如果排序成功，返回 1**，否则返回**-1**；**

②〈元素号〉是指开始排序的元素编号，如果忽略，则默认从数组的第一个元素开始排序。如果数组是一维的，函数从〈元素号〉开始排序。如果数组是二维的，则参数〈元素号〉既决定从第几行开始排序，又决定以每行中的第几列元素为排序依据。

③〈元素个数〉指定一维数组中参与排序的元素个数，或二维数组中参与排序的行数。如果〈元素个数〉为-1 **或忽略此参数，从包含起始元素的行到数组的最后一行都参与排序。**

④〈排序方向〉指定数组元素的排序方向（升序或降序）。默认情况下，数组元素按升序排序；如果取值为零或忽略此参数，数组中元素将按升序排序。如果取值为 1 **或任意非零值，数组元素按降序排序。**

（2**）**ASCAN **函数**

ASCAN **函数在数组中搜索与一个表达式具有相同数据和数据类型的元素。其语法格式为：**

ASCAN（〈数组名〉,〈表达式〉[,〈元素号〉[,〈元素个数〉]]）

说明：

① 返值数值型数据，如果找到匹配的元素，返回元素编号，否则返回 0**。**

②〈元素号〉是指开始搜索的元素编号，如果忽略，则默认搜索整个数组。

③〈元素个数〉是指要搜索的元素个数，如果忽略，则默认搜索到最后一个元素。

（3**）**ADEL **函数**

ADEL **函数从一个数组中删除一个、一行或一列元素。其语法格式为：**

ADEL（〈数组名〉,〈元素号〉[, 2]）

说明：

① 返值数值型数据，如果成功地删除了一个元素、行或列，则返回 1**。**

②〈元素号〉指定从数组中删除第几个元素、第几行或第几列。如果要从数组中删除一列，必须包含可选参数 2**。**

③ 从数组中删除一个、一行或一列元素，但并不改变数组的大小；后续元素、行或列元素向数组的起始方向移动，并把最后一个元素、行或列设置为“假”(.F.**)。**

（4**）**AINS **函数**

AINS **函数往一维数组中插入一个元素，或者向二维数组中插入一行或一列元素。其语法格式为：**

AINS（〈数组名〉,〈位置号〉[, 2]）

说明：

① 返值数值型数据，如果元素、行或列成功地插入到数组中，则返回 1**。**

②〈位置号〉指定在数组的第几个元素，或第几行（或列）处插入新元素。若要往一维数组中插入新元素时，新元素插到元素〈位置号〉前。若要往二维数组中插入行时，新行正好插到参数〈位置号〉指定的行前。

③ 参数 2 **表示往二维数组中插入一列。新列正好插到参数〈位置号〉指定的列前。**

④ 往数组中插入一个元素、一行或一列，并不改变数组大小。后续的元素、行或列将依次往数组末尾移一步，并丢弃数组中最后一个元素、一行或一列的数据。新插入的元素、行或列初始化为“假”(.F.)。

（5）ALEN 函数

ALEN 函数返回数组中元素、行或列的数目。其语法格式为：

ALEN（〈数组名〉[,〈返回类别〉]）

说明：

① 返值数值型数据。

② 如果参数仅包含数组名，则函数返回元素的数目。

③〈返回类别〉取值为 0 时，函数返回数组元素数目；取值为 1 时，函数返回数组的行数；取值为 2 时，函数返回数组的列数，此时如果数组是一维数组，则返回 0（没有列）。

【例 7-19】由计算机随机生成 10 个互不相同的数，然后将这些数按由小到大的顺序显示出来。

分析：这是一个“排序”问题，使用排序函数可以轻而易举地对数组元素进行排序。

编写代码如下：

```
CLEAR
DIMENSION  a（10）
p = ""
FOR i = 1 TO 10
  yes = 1
  DO WHILE yes = 1
    x = INT（RAND（）  * 100）
    yes = 0
    FOR j = 1 TO i - 1
      IF x = a（j）
        yes = 1                    && 若与前面的元素相同，则返回到 Do 循环
        EXIT
    ENDIF
  ENDFOR
ENDDO
a（i）  = x
p = p + STR（a（i），4）
ENDFOR
? "排序前的数组："
? p
ASORT（a）
p=""
FOR i = 1 TO 10
  p = p + STR（a（i），4）
ENDFOR
? "排序后的数组："
```

```
? p
```

程序运行结果，如图 7-38 所示。

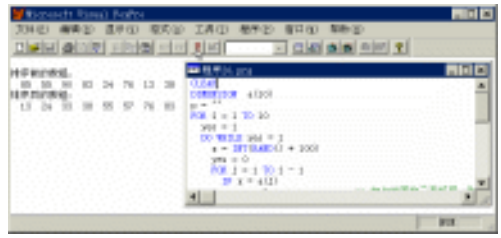


图7-38　排序

2．与数据表记录进行数据交换的命令

用于数组与数据表记录之间进行数据交换的命令有以下几个：

① SCATTER：将数据从当前记录复制到数组中去。

② GATHER：用来自数组的数据替换当前表中的数据。

③ COPY TO ARRAY：从当前表向一个数组复制数据。

④ APPEND FROM ARRAY：用来自数组的数据给当前表追加新记录。

【例 7-20】使用 SCATTER 与 GATHER 命令修改表 student.dbf。

```
USE student
LOCA FOR 姓名="李"
SCATTER TO arr1
? arr1（1）,arr1（2）,arr1（3）,arr1（4）
arr1（2） = "李庆"
GATHER FROM arr1
BROW
USE
```

说明：SCATTER 命令只能复制一条记录到数组，COPY TO ARRAY 命令可以复制多条记录到一个二维数组，且可以指定字段的个数。具体用法可以参见联机帮助。

7.6　多模块程序

7.6.1　模块化程序设计的概念

1．模块化

一个应用程序往往是复杂的，不可能编一个程序（或者说建立一个程序文件）就把所有工作都完成。因为那样势必导致程序过于庞大，层次不清，条理不明，给编程、调试和阅读都带来很大困难。通常采用化整为零的编程方法，把应用程序按功能划大为小，将整个系统分解为若干个相对独立的子功能模块。

另外，还可以把程序中多次重复的代码设计成能够完成一定功能的、可供其他程序使用（调用）的独立模块。这种模块称为子程序，它独立存在，但可以被多次调用，调用程序称为主程序。子程序的调用，即已经设置好的子程序被主程序或其他子程序‘借去使用”。

不但重复执行的程序段可以作为子程序独立出去，即使只执行一次的程序段也可以把它写成子程序，并把程序应该完成的主要功能都分配给各子程序去完成。

这就是模块化程序设计方法。VFP 支持模块化程序设计方法。

2．子程序

既然子程序只是一个相对独立的程序段，我们就可以仍然用顺序、选择、循环这三种基本结构去构造它。至于子程序的输入输出，应根据实际情况灵活掌握，一般表现为主程序与子程序间的数据传递。

VFP 中子程序的结构分为过程、函数与方法三类。一般来说，过程与函数的区别在于函数返回一个值而过程不返回值，而方法则是 VFP 中的一个新式的程序组装方式——限制在一个对象中的子程序。

本章主要介绍过程与函数。

7.6.2 过程与函数

过程与函数在使用上几乎没有什么区别，其结构有共同的本质特性，正是这种本质特性，使其具有极大的灵活性、方便性、有效性和清晰性。

1．过程与函数的定义

过程的定义格式为：

```
PROCEDURE 〈过程名〉
  〈命令序列〉
  [RETURN [〈表达式〉]]
ENDPROC
```

说明：

① PROCEDURE 指明过程的开始，并且定义过程名称。

②〈过程名〉必须以一个字母或下划线开头，可以包含字母、数字和下划线的任何组合。在 VFP 中，过程名最长可达 254 个字符。

③〈命令序列〉构成过程。

④ 在过程的任何地方都可以包含 RETURN 以返回控制到调用程序或另一个程序，还可以定义过程返回的值。如果没有包含 RETURN 命令，在函数退出时会自动执行一个隐含的 RETURN 命令。

⑤ 如果 RETURN 命令没有包含返回值（或执行一个隐含的 RETURN）时，VFP 将.T.（真）作为返回值。

⑥ 过程以 ENDPROC 命令结尾，此命令是可选的；过程遇到 PROCEDURE 命令、FUNCTION 命令或程序文件的结尾时会自动结束。

⑦ 不能在过程后的程序文件中包含普通的执行程序代码，在文件中的第一个 PROCEDURE 或 FUNCTION 命令后只能跟随过程、用户自定义函数和类定义。

⑧ 可在 PROCEDURE 和 ENDPROC 后的同一行放置注释语句。在编译和执行程序时，将忽略注释。

函数的定义格式为：

```
FUNCTION 〈函数名〉
  〈命令序列〉
  [RETURN [〈表达式〉]]
ENDFUNC
```

说明：

① FUNCTION 指明函数的开始，并且定义函数名称。

② 在 VFP 中，函数名最长不超过 254 个字符。如果要区分超过 10 个字符的程序文件名与以相同的 10 个字符开头的函数，可以使用引号将程序文件名包围起来或在程序文件名后包含一个扩展名。

③ 函数以 ENDFUNC 命令结尾，此命令是可选的；过程遇到 PROCEDURE 命令、FUNCTION 命令或程序文件的结尾时会自动结束。

其他参数的说明与过程定义相同。

2．过程与函数的调用

过程与函数的调用方式有两种：

（1）第一种调用方法

第一种方式与程序的执行命令相同，其语法格式是：

DO〈过程名或函数名〉 [IN 〈程序文件名〉]

说明：

① VFP 首先在当前执行的程序中查找此过程，如果在该程序中找不到此过程，则在用 SET PROCEDURE 命令打开的过程文件中查找。

② 如果包含 IN 〈程序文件名〉子句，则执行该程序文件中的过程。如果找不到该程序文件，就会显示“文件不存在”的信息；如果找到了程序文件，但指定的过程不存在，就会显示“找不到过程”信息。

（2）第二种调用方法

第二种方式类似于一般函数的执行，其语法格式是：

〈过程名或函数名〉( )

说明：本方式既可以作为命令使用（忽略返回值），也可以作为函数出现在表达式中。

【例 7-21】下面的代码演示过程的不同调用方式。

```
CLEAR
? "主程序开始"
? "调用 f1"
DO f1
? "调用 f2"
f2()
? "f3 的返回值：",f3()
PROCEDURE f1
   ? "子程序 f1 开始"
   ? "子程序 f1 结束"
ENDPROC
PROCEDURE f2
   ? "子程序 f2 开始"
   ? "子程序 f2 结束"
ENDPROC
```

```
PROCEDURE f3
  RETURN "这是子程序 f3"
ENDPROC
```

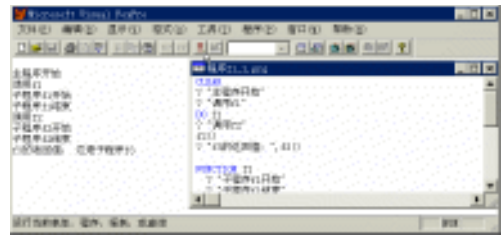


图7-39　子程序的调用

说明：

① 上面代码中主程序与子程序在同一个程序文件中，属于“依附型”的子程序。

② 上述子程序定义中使用的过程定义改为函数定义，结果是相同的，如图 7-39 所示。

7.6.3　内存变量的属性和作用域

对于一个大的应用系统，必然会用到许多内存变量。合理地利用不同属性的内存变量，会给设计程序带来很多方便。

VFP 中的内存变量按其属性可以分为私有型（Private）、全局型（Public）和局部型（Local）三种。不注意内存变量的属性，将会造成程序出错。

1．私有型内存变量

前面章节中，我们使用过许多变量，那些变量只简单地定义和使用，而未加任何说明。程序中使用的内存变量，凡未经特殊说明的均属于私有型内存变量，这些内存变量可以在本级程序及以下各级子程序中使用，其值可以在子程序中改变，返回主程序时保留改变后的值。因此私有型内存变量又称为主从型变量。

某一级程序定义的私有型内存变量，在本级程序结束时被清除，因此不能进入上一级程序。即若在子程序中定义它，它不能进入上一级程序；若在主程序中定义它，它可以到下级子程序中使用；当低层程序在执行过程中改变了该变量的值时，在返回本级主程序时被改变的值仍然保存，主程序可以继续使用子程序改变后的值。下面举例说明主从型内存变量。

【例 7-22】这是一个调用子程序计算圆面积的程序。在子程序中计算出圆面积 s 的值，返回主程序时输出 s。

主程序：

```
CLEAR
r = 10
s = 0
DO mj
? "s =",s
CANCEL
```

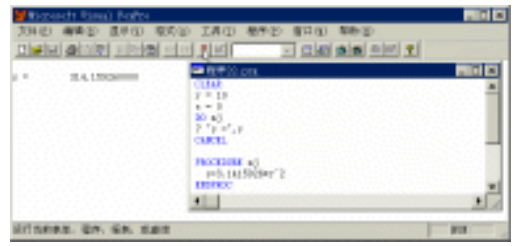


图7-40　私有型内存变量

计算圆面积子程序 mj：

```
PROCEDURE mj
  s = 3.1415926*r^2
ENDPROC
```

说明：如果把上面主程序改为：

主程序：

```
CLEAR
r = 10
DO mj
? "s =",s
CANCEL
```

执行主程序输出命令“？”时将出错，系统提示变量 s 未定义。产生这一情况的原因为 s 是子程序定义的变量，当子程序运行结束返回主程序时，s 即被子程序清除。

2．全局型内存变量

程序设计中，对有些子程序中定义的内存变量，我们希望在主程序中仍能使用，或者希望它在整个系统中均能使用，这时就可以定义全局型内存变量。

全局型内存变量是指在上、下各级程序中都可使用、修改的内存变量。全局型内存变量可以由主程序定义，也可以由子程序定义，全局变量就像在一个程序中定义的变量一样，可以任意改变和调用，当程序执行完后，其值仍然保存。若欲清除这种变量，必须用 RELEASE 命令。全局型内存变量的定义格式为：

```
PUBLIC 〈内存变量表〉
```

说明：

①〈内存变量表〉中可以是内存变量或数组变量。

② 任何全局内存变量或者数组必须先定义，后赋值。反之，如果程序中已给一个内存变量或数组赋值，或者已用 DIMENSION 建立了数组，再用 PUBLIC 将其定义为全局性数组，将产生错误。然而全局变量可以重复地进行 PUBLIC。

③ 定义后尚未赋值的全局变量其值为逻辑值.F.，若变量名为 FOX 和 FOXPRO 公用变量，则其初值为.T.。

④ 全局变量在程序结束时不能自动释放（即使主程序）。

⑤ 当进入下一级子程序时，已在上级由 PUBLIC 说明过的与之同名的内存变量可以用 PRIVATE，或者用 DO...WITH〈参数表〉传递参数的方法暂时隐藏起来，作为本级程序的局部变量使用。待本级子程序结束返回上一级程序后，便释放它们作为本级程序的局部变量特性，恢复它们全局变量的特性和内容。

⑥ 在命令窗口中建立的所有内存变量或者数组自动定义为全局型。

【例 7-23】在【例 7-22】中使用全局变量 s，则在主程序中不做定义即可使用。

主程序：

```
CLEAR
r = 10
s = 0
DO mj
? "s =",s
CANCEL
```

计算圆面积子程序 mj：

```
PROCEDURE mj
```

```
    PUBLIC s
    s = 3.1415926*r^2
ENDPROC
```

3．局部型内存变量

局部变量只能在建立它们的模块中使用，不能在上层或下层模块中使用。当建立它们的模块程序运行结束时，局部变量自动释放。局部型内存变量的定义格式为：

```
LOCAL 〈内存变量表〉
```

说明：

① LOCAL 创建的变量和数组都初始化为“假”(.F.)。

② 必须在赋值之前把变量或数组声明为局部。若在用 LOCAL 声明一个变量或数组为局部变量或数组之前，对该变量或数组进行赋值，则 VFP 会产生一个错误信息。

③ 局部变量可以由引用方式传递。

④ 不能缩写 LOCAL，因为 LOCAL 和 LOCATE 的前四个字母相同。

【例 7-24】局部型内存变量示例。

```
* 主程序
CLEAR
PUBLIC x1                    && 建立全局变量 x1
x1 = 10
LOCAL x2                     && 建立局部变量 x2
x2 = 20
x3 = 30                      && 建立私有变量 x3
? "主程序调用子程序前："
? "x1 = ", x1
? "x2 = ", x2
? "x3 = ", x3
DO p1
? "主程序调用子程序后："
? "x1 = ", x1
? "x2 = ", x2
? "x3 = ", x3
CANCEL
* 子程序
PROCEDURE p1
x1 = 40
x2 = 50
x3 = 60
? "在子程序中："
? "x1 = ", x1
? "x2 = ", x2
? "x3 = ", x3
ENDPROC
```

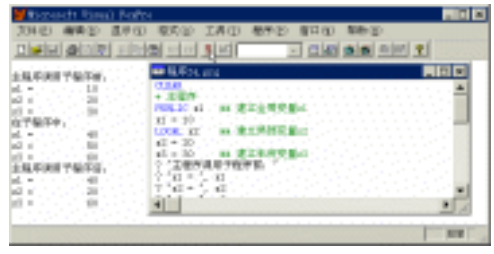


图7-41　局部型内存变量

运行程序，结果如图 7-41 所示。

4．内存变量的隐藏

如果某个程序中使用的局部变量与高层次程序中的局部变量或全局变量同名，为了避免同名变量在使用上的混淆，或者为了使主程序中的同名变量在子程序调用后返回时，仍保持原值，可在子程序中把它们定义为局部变量。局部型内存变量是在某一级程序中将上级程序定义的一些变量隐藏起来，此时就像这些内存变量不存在一样，可以再定义同名的内存变量，因而，该命令又称为公用内存变量隐蔽命令。但是，一旦返回上一级程序，则在下级程序中定义的同名变量即被清除，被隐藏的内存变量恢复原名，保持原值，丝毫不受子程序中同名变量的影响。

PRIVATE 〈内存变量表〉

说明：

①〈内存变量表〉指出需要隐藏的局部变量。

② 对 PRIVATE 中内存变量的修改并不影响上级程序中与之同名的内存变量的值。此命令只对本级程序及以下各级子程序有效，当返回到上级程序时，被 PRIVATE 隐蔽的当前程序中的内存变量自动被删除。

③ 在被隐蔽期间，程序就不能再调用这些被隐蔽的上级内存变量，但实际上它们仍然存在，一旦含有 PRIVATE 内存变量的子程序结束后，被 PRIVATE 隐蔽起来的那些以前建立的同名的上级内存变量自动恢复以前的内容和状态。

【例 7-25】隐藏内存变量的示例。

```
CLEAR
* 主程序
a = 10
? "111",a
DO jb1
? "222",a
a = 15
? "333",a
DO jb2
? "444",a
CANCEL

PROCEDURE jb1
   PRIVATE a
   a = 20
   ? "JB1",a
ENDPROC

FUNCTION jb2
   PRIVATE a
   a = 30
   ? "JB2",a
ENDFUNC
```

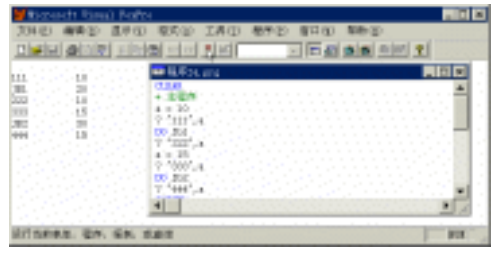


图7-42　隐藏内存变量

运行程序，结果如图 7-42 所示。

说明：在主程序中定义了 a，并被赋值。到了子程序，由于 a 被隐藏，又重新赋值后，内存中存在着两个名为 a 的变量且值不相同，返回主程序后，子程序定义的变量 a 被清除，而主程序的 a 被恢复了本来的属性并保持原值。

内存变量可隐蔽的特点使得我们在开发大型应用程序时，在不同层次的程序中可以使用同名内存变量而不致发生混乱。当多个程序员共同开发一个大型程序时，这种情况是经常发生的。这时，我们就可以利用 PRIVATE 命令，把本程序用到的某些变量定义成局部变量，如果其中有与上级程序同名的变量，则自动隐蔽起来。然后再对子程序中的内存变量进行数据操作，当程序返回到上级程序时，原来定义的变量内容保持不变，从而可以消除本级程序产生的某些副作用。下节介绍的用参数传递数据，是一种消除副作用的最好方法。

7.6.4 数据传递

在调用过程中，要考虑调用程序和被调用程序之间的数据是如何传递的。通常在编制一个子程序时，要考虑它需要输入哪些量，进行处理后输出哪些量。正确的提供一个子程序的输入数据和正确地引用其输出数据，是使用子程序的关键问题，也就是调用程序和被调用程序之间的数据传递。

调用程序向被调用的子程序传递数据通常是传递一组参数，而子程序向调用程序传递数据则通常是返回一个值。

子程序可以接收主程序传递的参数，也可以不接收参数（如上面的例子），可以有返回的值，也可以没有返回的值。

1．参数的传递

调用程序向被调用的子程序传递的参数通常称为“实际参数”，简称“实参”；而子程序中用来接受数据的变量通常称为“形式参数”，简称“形参”。

（1）子程序的命令形式

若想使子程序能够接收参数，只需在子程序代码的开始增加命令行：

PARAMETERS 〈形参表〉

或

LPARAMETERS 〈形参表〉

说明：

① LPARAMETERS 与 PARAMETERS 的区别在于：以 PARAMETERS 命令所接收的参数变量属于 PRIVATE（私有）性质，而以 LPARAMETERS 命令所接收的参数变量属于 LOCAL（局部）性质。

②〈形参表〉是用逗号分开的形参列表。

（2）主程序调用子程序的命令形式

调用子程序的方式仍有两种：

DO 〈过程名或函数名〉 WITH 〈实参表〉

或

〈过程名或函数名〉（〈实参表〉）

说明：

①〈实参表〉中实际参数的个数最多不能超过 27 个。

② 若〈形参表〉中形参的个数多于实际参数的个数，则多余的形参变量的值为.F.。若实际参数的个数多于〈形参表〉中形参的个数，则出现“程序错误”提示：必须指定额外参数，如图 7-43 所示。

图7-43　必须指定额外参数。

③ 在调用子程序时，无论指定或不指定实际参数，子程序名后都可以带一对括号。

④〈实参表〉中的实际参数可以是任何类型的变量、函数、数组、表达式，甚至是对象。

【例 7-26】计算矩形面积。用参数实现数据传送。

```
* 主程序
CLEAR
gao = 5
kuan = 6
mj = 0
DO jxmj WITH gao, kuan, mj
? "高为 5，宽为 6，矩形的面积为：" + ALLT（STR（mj））
CANCEL
* 子程序
PROCEDURE jxmj
   PARAMETERS g,k,m
   m = g * k
ENDPROC
```

图7-44　计算矩形面积

说明：

① 主程序中的参变量 gao、kuan、mj 与子程序 jxmj 中的参变量 g、k、m 相对应。当子程序中 m 的值变化后，子程序返回时，把 m 的值传给主程序中的对应变量 mj。

② 程序运行结果如图 7-44 所示。

2．参数传递的方式

参数传递的方式分为传址方式和传值方式。

传址方式是指主程序将实际参数在内存中的地址传给被调用的子程序，由形式参数接收，而形式参数也使用该地址。即实际参数与形式参数使用相同的内存地址，形式参数的内容一经改变，实际参数的内容也将跟着改变。

传值方式是指主程序将实际参数的一个备份传给被调用的子程序，这个备份可以被子程序改变，但在主程序中变量的原值不会改变。

传址或传值方式对于数组的影响较大，如果采用传值方式，只能传递数组的第一个元素的内容，其他元素无法传递。如果采用传址方式，则将整个数组的地址传给了被调用的子程序，形式参数会自动变成一个与实际参数同样大小的数组。

在调用子程序方式的第 1 种格式中，如果实参是常量或非变量形式的表达式时，系统会

自动计算实参的值，并将其传递给对应的形参，即为传值方式；如果实参是变量时，则将传递变量的地址，即为传址方式。

在调用子程序方式的第 2 种格式中，则总是默认为传值方式。

如果要改变参数的传递方式，可以采用以下两种方法：

- 使用 SET UDFPARMS TO VALUE | REFERENCE 命令来强制改变参数的传递方式。
- 使用@符号来强制 VFP 使用传址的参数传递方式。

下面的例子使用两种不同的参数传递方式，得到不同的结果。

【例 7-27】编写求最大公约数的自定义子程序，输入的两个整数按值传递，求出的最大公约数按地址传递。

程序代码如下：

```
* 主程序
CLEAR
INPUT "请输入一个整数:" TO x
INPUT "请输入一个整数:" TO y
a = 0
IF x * y <> 0
   hcf（x,y,@a）                          && 变量 a 按传址方式传递
   ? "两数的最大公约数是:"+ALLT（STR（a））
ENDIF
* 子程序
PROCEDURE hcf
   PARAMETERS m, n, Z
   IF m < n
    t = m
    m = n
    n = t
 ENDIF
 r = m % n
 DO WHILE r <> 0
    m = n
    n = r
    r = m % n
 ENDDO
 Z = n                                  && 将求出的最大公约数赋值给变量 Z
ENDPROC
```

图7-45 求最大公约数

程序运行的结果如图 7-45 所示。

3．子程序的返回值

若想使子程序能够返回一个值，只需在子程序代码的结束处（ENDPROC 或 ENDFUNC 命令之前）增加命令行：

```
RETURN [〈表达式〉]
```

如果缺省〈表达式〉，VFP 将自动返回.T.。

当代码执行到 RETURN 命令，就会立即返回到主程序中。

在主程序中可用以下形式调用子程序：

① 在表达式中调用子程序，如：

```
k = PI() * Demo（r）
```

② 在赋值语句中调用子程序，如：

```
k = Demo（r）
```

③ 以等号命令调用子程序，如：

```
= Demo（r）
```

注：用等号命令调用子程序将舍弃返回值。

【例 7-28】改写【例 7-27】中的子程序，使其能够返回值。然后通过在表达式中调用子程序，得到三个整数的最大公约数。

编写程序代码如下：

```
* 主程序
CLEAR
INPUT "请输入一个整数:" TO x
INPUT "请输入一个整数:" TO y
INPUT "请输入一个整数:" TO z
IF x * y * z <> 0
   a = hcf（x,y）
   b = hcf（a,z）
   ? "3 个数的最大公约数是:"+ALLT（STR（b））
ENDIF
* 子程序
PROCEDURE hcf
   PARAMETERS m, n
   IF m < n
      t = m
      m = n
      n = t
  ENDIF
  r = m % n
  DO WHILE r <> 0
      m = n
      n = r
      r = m % n
  ENDDO
  RETURN n                                  && 将求出的最大公约数返回
```

```
ENDPROC
```

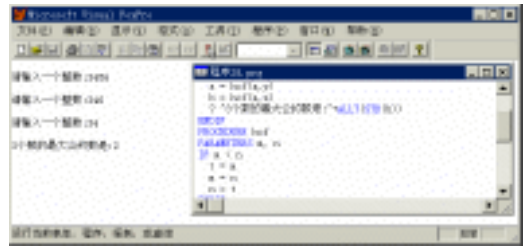


图7-46　求最大公约数

程序运行结果如图 7-46 所示。

【例 7-29】验证哥德巴赫猜想：一个不小于 6 的偶数可以分解为二个奇素数之和。设计一个程序，输入一个不小于 6 的偶数，然后由计算机将其分解为两个奇素数之和。

编写程序代码如下：

```
* 主程序
CLEAR
INPUT "请输入一个整数:" TO n
DO WHILE n % 2 != 0 OR n < 6
   MESSAGEBOX（'必须输入大于 6 的偶数，请重新输入！',64）
   CLEAR
   INPUT "请输入一个整数:" TO n
ENDDO
FOR x = 3 TO n / 2 STEP 2
   IF prime（x）
      y = n - x
      IF prime（y）
         ? ALLT（STR（n））  + ' = ' + ALLT（STR（x））  + '+' + ALLT（STR（y））
         EXIT
      ENDIF
   ENDIF
ENDFOR
* 子程序
PROCEDURE prime
   LPARAMETERS m
   f = .T.
   IF m > 3
      FOR I = 3 TO SQRT（m）
       IF m % I = 0
          f = .F.
          EXIT
      ENDIF
   ENDFOR
  ENDIF
  RETURN f
ENDPROC
```

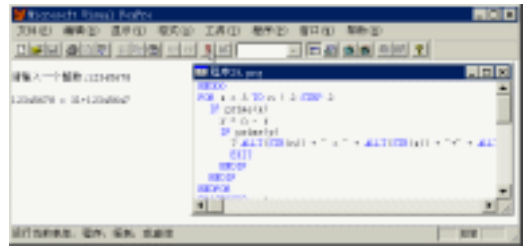


图7-47　验证哥德巴赫猜想

程序运行结果如图 7-47 所示。

7.6.5　子程序的递归调用

递归函数论是现代数学的一个重要分支，数学上常常采用递归的办法来定义一些概念，

例如，自然数 N 的阶乘定义为：

$$N!=\begin{cases}1 & N=0\\ N\times(N-1)! & N>0\end{cases}$$

上述阶乘的定义就是递归定义。

递归在算法描述中有着不可替代的作用。很多看似十分复杂的问题，使用递归算法来描述显得非常简洁与清晰。

一个问题要采用递归的算法来解决，必须能够做到：

① 该问题必须是根据给定的参数分支求解，其中一支能够直接求解。

② 其他分支的求解转化为原问题的求解，但求解时的参数改变。参数的变化最终能够满足可直接求解分支的条件。

其中，第一个条件保证了递归算法的可行性，即可终止性；第二个条件表明了算法的递归性。

VFP 支持递归算法。在子程序代码中直接或间接调用该子程序本身，称为子程序的递归调用。若在一个子程序的代码中调用其自身，则称为直接递归；若在甲子程序中调用了乙子程序，而乙子程序中又调用了甲子程序，则称为间接递归。

【例 7-30】利用递归调用计算 $n!$。

分析：使用递归算法求阶乘的流程图如图 7-48 所示。

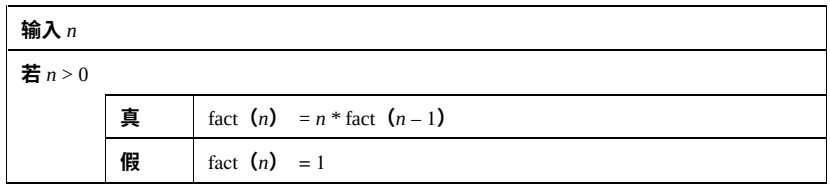


图7-48　递归算法求阶乘的流程图

程序代码如下：

```
* 主程序
CLEAR
INPUT "请输入一个整数:" TO m
DO WHILE m < 0 OR m > 20
   MESSAGEBOX（'必须输入小于 20 的正数，请重新输入！',64）
   CLEAR
   INPUT "请输入一个整数:" TO m
ENDDO
x = fact（m）
? ALLT（STR（m））  + '! = ' + ALLT（STR（x））
* 子程序
PROCEDURE fact
   LPARAMETERS n
   IF n > 0
      f = n * fact（n - 1）
```

```
    ELSE
      f = 1
    ENDIF
    RETURN f
ENDPROC
```

说明：当 *n* > 0 **时，在子程序** fact **中调用** fact **子程序，参数为** *n* - 1**，这种操作一直持续到** *n* = 1 **为止。**

例如，当 *n* = 5 **时，求** fact（5）**的值变为求** 5×fact（4）**；求** fact（4）**的值又变为求** 4×fact（3）**，…，当** *n* = 0 **时，**fact **的值为** 1**，递归结束，其结果为** 5×4×3×2×1**。如果把第一次调用子程序** fact **叫做** 0 **级调用，以后每调用一次级别增加** 1**，过程参数** *n* **减** 1**，则递归调用的过程如下：**

递归级别	执行操作
0	fact（5）
1	fact（4）
2	fact（3）
3	fact（2）
4	fact（1）
4	返回 1 fact（1）
3	返回 2　fact（2）
2	返回 6　fact（3）
1	返回 24　fact（4）
0	返回 120　fact（5）

程序执行结果，如图 7-49 **所示。**

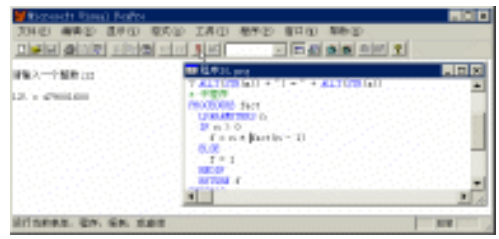


图7-49　**递归算法求阶乘**

7.6.6　过程文件

子程序的使用为程序的编制带来了很大的方便。按照存储方式的不同，子程序可以分为独立型和依附型两种。

1．独立型、依附型子程序与过程文件

子程序可以和其他程序文件一样单独以文件的形式存储在磁盘上，也可以如上面的例子那样放置在（主）程序代码的后面，还可以集中保存在称为过程文件的单独文件中。前者称为独立型的子程序，后两种则称为依附型的子程序。

独立型的子程序其实和一般的程序文件没有什么区别，程序首行无需有 PROCEDURE | FUNCTION **命令行，尾行也无需** ENDPROC | ENDFUNC **命令行。如果需要传递参数，则需要加上** PARAMETERS **或** LPARAMETERS **命令行，如果需要返回值，则需要加上** RETURN **命令行。**

独立型子程序的创建与一般程序文件完全相同。

在每次调用独立型子程序时，都要访问磁盘，进行一番查找。而访问磁盘的速度一般较慢，子程序分得越细，花费的代价就越高，而且每调用一个子程序，就打开一个文件。当调

用子程序的次数较多时，必将降低程序运行效率。为此，可以把多个子程序合并成一个大的文件，而在这个大的文件中，每个子程序仍然是独立的，这个大的公用程序文件被称作过程文件。由于过程文件是集中起来以一个文件的形式存储在磁盘上的，调用时作为一个文件一次打开，每个子过程可随时调用，从而大大减少访问磁盘的次数，提高程序运行速度。

过程文件也是程序文件，其扩展名为.PRG，创建的方法也同其他程序一样。

2．过程文件的组织与使用

过程文件由若干子程序组成，其中的每个子程序都以 PROCEDURE 〈过程名〉或 FUNCTION〈函数名〉命令开始，以 ENDPROC 或 ENDFUNC 命令结束。

过程文件只包含子程序，这些子程序能被任何其他程序调用，包括过程文件中的子程序自己。

但是，在过程文件被打开之前，过程中所包含的过程是不能被任何程序调用的，因此，在调用前，过程文件必须预先用过程打开命令打开。打开一个过程文件的命令是：

SET PROCEDURE TO [〈过程文件 1〉[,〈过程文件 2〉, ...]][ADDITIVE]

说明：

① 可以打开一个或多个过程文件，一旦过程文件被打开，其中的所有子程序就都可以被调用。

② 如果使用 ADDITIVE 短语，则不关闭原先已打开的过程文件。否则将关闭原先已打开的过程文件。

③ 使用不带任何文件名的 SET PROCEDURE TO 命令，将关闭已打开的所有过程文件。

④ 如要关闭个别过程文件，可以使用命令：

RELEASE PROCEDURE 〈过程文件 1〉[,〈过程文件 2〉, ...]

⑤ 存放在程序文件后面的子程序，也可以被其他程序所调用，但该程序文件必须处于执行状态或被 SET PROCEDURE TO 命令打开。

【例 7-31】将计算圆面积、圆周长和球体积的子程序组织到过程文件 yuan.prg 中，然后编写主程序调用过程文件中的子程序。

首先编写过程文件内容如下，并以 yuan.prg 为文件名存盘。

```
* yuan.prg 过程文件
***** 计算圆面积 *****
FUNCTION ymj
   LPARAMETERS r
   s = 3.1415926 * r^2
   RETURN s
ENDFUNC
* eof:ymj
***** 计算圆周长 *****
FUNCTION yzc
   LPARAMETERS r
   s = 2 * 3.1415926 * r
   RETURN s
```

```
ENDFUNC
* eof:yzc
***** 计算圆体积 ****
FUNCTION qtj
   LPARAMETERS r
   s = 4 / 3 * 3.1415926 * r^3
   RETURN s
ENDFUNC
* eof:qtj
```

然后编写主程序如下：

```
* 主控程序
SET PROCEDURE TO yuan
CLEAR
INPUT "请输入圆或球体的半径：" TO n
? "圆的面积是：" + STR（ymj（n）,8,2）
? "圆的周长是：" + STR（yzc（n）,8,2）
? "球的体积是：" + STR（qtj（n）,8,2）
RETURN
```

图7-50　过程文件的调用

程序运行结果如图 7-50 所示。

7.7　习题

一、选择题

1．将内存变量定义为全局变量的 VFP 命令是（　　）。

A）LOCAL　　B）PRIVATE　　C）PUBLIC　　D）GLOBAL

2．设当前盘目录下有数据库 db_stock，其中有数据库 stock.dbf，该数据库表的内容是：

股票代码	股票名称	单价	交易所
600600	青岛啤酒	7.48	上海
600601	方正科技	15.20	上海
600602	广电电子	10.40	上海
600603	兴业房产	12.76	上海
600604	二纺机	9.96	上海
600605	轻工机械	14.59	上海
000001	深发展	7.48	深圳
000002	深万科	12.50	深圳

执行下列程序段以后，内存变量 a 的内容是（　　）。

```
CLOSE DATABASE
a = 0
USE stock
GO TOP
```

```
DO WHILE .NOT. EOF()
  IF 单价 > 10
     a = a + 1
  ENDIF
  SKIP
ENDDO
```

A）1　　B）3　　C）5　　D）7

3．在 VFP 中，用于建立或修改过程文件的命令是（　　）。

A）MODIFY〈文件名〉　　B）MODIFY COMMAND〈文件名〉

C）MODIFY PROCEDURE〈文件名〉　　D）上面 B）和 C）都对

4．下面关于子程序调用的陈述中，哪个是正确的（　　）。

A）实参与形参的数量必须相等。

B）当实参的数量多于形参的数量时，多余的实参被忽略

C）当形参的数量多于实参的数量时，多余的形参取逻辑假

D）上面 B）和 C）都对

5．如果一个过程不包含 RETURN 语句，或者 RETUEN 语句中没有指定表达式，那么该过程（　　）。

A）没有返回值　　B）返回 0　　C）返回.T.　　D）返回.F.

6．有如下程序：

```
INPUT TO A
IF A=10
  S = 0
ENDIF
S=1
? S
```

假定从键盘输入的 A 的值一定是数值型，那么上面条件选择程序的执行结果是（　　）。

A）0　　B）1　　C）由 A 的值决定　　D） 程序出错

二、填空题

1．有一分支程序为：

```
IF S > 100
  DO P1.PRG
ELSE
  IF S>10
DO P2.PRG
  ELSE
IF S>1
  DO P3.PRG
ELSE
  DO P4.PRG
ENDIF
```

```
   ENDIF
ENDIF
```

分别写出执行 p2、p3、p4 **子程序的条件表达式：**

DO P1.PRG **条件为：** S > 100

DO P2.PRG **条件为：**_____

DO P3.PRG **条件为：**_____

DO P4.PRG **条件为：**_____

2．**有一个表文件** bhsl.dbf，**其内容如下：**

记录号	编号	数量
1	A1	10
2	A0	85
3	A2	67
4	A10	50
5	A12	65

写出下列程序的运行结果。

```
SET TALK OFF
USE BHSL
SET ORDER TO TAG 编号
STORE 0 TO S
LOCATE FOR 数量 > 10
DO WHILE .NOT.EOF()
   ?? 编号
   IF SUBSTR （编号, 2, 1） = "1"
      S = S + 数量
   ENDIF
   CONTINUE
ENDDO
? S
USE
```

运行结果：_____

3．**写出下列程序的运行结果。**

```
SET TALK OFF
DIMENSION A（6）
FOR k = 1 TO 6
   A（k） = 20 – 2 * k
ENDFOR
k = 5
DO WHILE k >= 1
   A（k） = A（k） – A（k+1）
   k = k - 1
```

```
ENDDO
? A（1）, A（3）, A（5）
SET TALK ON
```

运行结果：_____

三、上机题

1．有如下命令序列，其功能是根据输入的考试成绩显示相应的成绩等级：

```
SET TALK OFF
CLEAR
INPUT "输入考试成绩:" TO chj
Dj = IIF（chj < 60, "不合格", IIF（chj >= 90, "优秀", "通过"））
? "成绩等级:" + dj
SET TALK ON
```

请编写程序，用 DO CASE-ENDCASE **型分支结构实现该命令序列功能。**

2．修改下列程序的错误，使其能够计算出 30 **以内（含** 30**）能被** 5 **整除的正整数之和。注意，不要修改或删除** Y = Y + X **及其后面的命令和语句。**

```
CLEAR
STORE 0 TO X, Y
DO WHILE .T.
  X=X+1
  DO CASE
  CASE MOD（X,5）=0
    EXIT
  CASE X <=30
    LOOP
  ENDCASE
  Y=Y+X
ENDDO
? Y
```

3．在考生目录下的“订货管理”数据库中有两个表 order_detail、order_list，order_detail **表中包含“订单号”、“器件号”、“器件名”、“单价”、“数量”和“新单价”等字段，**order_list **表中包含“客户号”、“订单号”、“订购日期”和“总金额”等字段。编写满足如下要求的程序：根据** order_list **表中的“订购日期”字段的值确定** order_detail **表的“新单价”字段的值。原则是：“订购日期”为** 2001 **年的“新单价”字段的值为原单价的** 90%，**“订购日期”为** 2002 **年的“新单价”字段的值为原单价的** 110%**（注意:在修改操作过程中不要改变** order_detail **表记录的顺序），最后将程序保存为** prog1.prg，**并执行该程序。**

4．在考生目录下的“商品销售”数据库中有商品表和单价调整表，两个表中都包含“商品号”、“商品名”、“单价”、“出厂单价”和“产地”等字段。编写名称为 change_c **的命令程序并执行，该程序实现下面的功能：**

将“商品表”进行备份，备份文件名为“商品表备份.dbf **”；**

将“商品表”中“商品号”前两位编号为“10”的商品的“单价”修改为出厂单价的10%；使用“单价调整表”对商品表的部分商品出厂单价进行修改（按“商品号”相同）。

5．在考生文件夹下有数据表：雇员工资表 salarys 和“人事部”向“财务部”提供的雇员工资调整表 c_salary1。请编写名称为 change_c 的程序并执行；该程序实现下面的功能：

将雇员工资表 salarys 进行备份，备份文件名为 bak_salarys.dbf；利用雇员工资调整表 c_salary1 的“工资”，对 salarys 表的“工资”进行调整（请注意：按“雇员号”相同进行调整，并且只是部分雇员的工资进行了调整，其他雇员的工资不动）。

第 8 章　表单设计基础

建立一个应用程序最重要的内容之一是设计一个用户界面，表单（Form）就是 VFP 提供的应用程序界面。表单中可以包含命令按钮、文本框、列表框等各种 Windows 应用程序的基本元素。表单的使用也是从 DOS 应用程序向 Windows 应用程序转变的重要标志。

8.1　面向对象的概念

首先介绍在表单设计中涉及的面向对象的基本概念。

8.1.1　对象与类

对象与类是面向对象编程方法的两个最基本的概念。

1．对象

对象（Object）在现实生活中是很常见的，如一个人是一个对象，一台 PC 机是一个对象。如果将一台 PC 机拆开来看便有“显示器、机箱、软盘驱动器、硬盘、键盘、鼠标器……”，每一个又都是一个对象，即 PC 机对象是由多个“子”对象组成的。此时 PC 机又称为一个包容器（Container）对象。

对象的概念在 VFP 中无处不在，表单、控件是对象，数据库、数据表、字段也是对象。每个对象都有自己的特征和行为。对象的特征用数据来表示，称作对象的属性；对象的行为用对象中的代码来实现，称作对象的方法。此外，对象还可以识别和响应预定义的动作——事件。用户通过对象的属性、事件和方法程序来处理对象。

对象是被封装的，也就是说它同时包含其代码和数据，是代码与数据的组合体。这比传统的编写代码方法更容易维护。

在 VFP 中，对象是由类创建的，因此对象被说成是类的一个实例。

2．类

类和对象关系密切，但并不相同。类是描述对象特征以及对象外观和行为的模板。在 VFP 中，所有的对象都是由类定义的。例如，工具箱中的控件就是 VFP 的预定义类（控件类），在表单上画出的文本框、命令按钮等都是控件类的实例，这些类实例就是应用程序中引用的对象。

对象所具有的属性、方法和事件都是在类中定义的，类就像是一个制造对象的模板。

除了系统的预定义类之外，VFP 还允许用户编写自定义的类，通过对象变量来引用自定义的类。

8.1.2　对象的属性、事件与方法

从可视化编程的角度来看，对象是一个具有属性（数据）和方法（行为方式）的实体。一个对象建立以后，其操作就通过与该对象有关的属性、事件和方法来描述。

1．对象的属性

属性（Property）是指对象的一项描述内容，用来描述对象的一个特性，不同的对象有不

同的属性，而每个对象又都由若干属性来描述。在可视化编程中，常见的属性有标题（Caption）、名称（Name）、背景色（Backcolor）、字体大小（FontSize）、是否可见（Visible）等。通过修改或设置某些属性便能有效地控制对象的外观和操作。

属性值的设置或修改可以通过属性窗口（参见 2.1.2）来进行，也可以通过编程的方法在程序运行的时候来改变对象的属性。在程序中设置属性的一般格式是：

表单名.对象名.属性名 = 属性值

2．对象的事件

所谓事件（Event），是由 VFP 预先定义好的、能够被对象识别的动作，如单击（Click）事件、双击（DblClick）事件、装入（Load）事件、移动鼠标（MouseMove）事件等，不同的对象能识别的事件也不全相同。

对象的事件是固定的，用户不能建立新的事件。为此，VFP 提供了丰富的内部事件，这些事件足以应付 Windows 中的绝大部分操作需要。

事件过程（Event Procudure）是为处理特定事件而编写的一段程序。当事件由用户触发（如 Click）或由系统触发（如 Load）时，对象就会对该事件作出响应（Respond）。响应某个事件后所执行的程序代码就是事件过程。一个对象可以识别一个或多个事件，因此可以使用一个或多个事件过程对用户或系统的事件作出响应。

虽然一个对象可以拥有多个事件过程，但在程序中要使用多少事件过程，则由程序员根据程序的具体要求来确定。对于必须响应的事件需编写该事件的事件过程，而不必理会的事件则不需要编写事件过程，只要交给 VFP 的默认处理程序即可，例如命令按钮的 Click 事件是最重要的事件，而 MouseUp 事件则可有可无，全视设计人员的需要。

3．对象的方法

方法（Method）是与对象相关联的过程，但又不同于一般的 VFP 过程。方法程序紧密地和对象连接在一起，并且与一般 VFP 过程的调用方式也有所不同。

与事件过程类似，VFP 的方法也属于对象的内部函数，只是方法用于完成某种特定的功能而不一定响应某一事件，如添加对象（AddObject）方法、绘制矩形（Box）方法、释放表单（Release）方法等。方法也被“封装”在对象之中，不同的对象具有不同的内部方法。VFP 提供了百余个内部方法供不同的对象调用。

与事件过程不同的是，根据需要可以由用户自行建立新的方法。这里，表单对象的新方法可在编程时建立，而一般对象的新方法需要在定义类时建立。

事件过程由事件的激发而调用其代码，也可以在运行中由程序调用其代码，而方法的代码只能在运行中由程序调用。

在程序中调用对象方法的格式是：

[[变量名] =]对象名.方法名（）

8.1.3 VFP 的基类

VFP 基类（参见表 8-1）是系统本身含有的，并不放在某个类库中。可以基于这些基类生成所需要的对象，也可以在“类生成器”中创建大部分基类的子类。

表 8-1　Visual FoxPro 基类

类　名	说　明	类　名	说　明	类　名	说　明	类　名	说　明
ActiveDoc	活动文档	Custom	自定义	CommandButton	命令按钮	PageFrame	页框
CheckBox	复选框	EditBox	编辑框	CommandGroup	命令组	ProjectHook	项目挂钩
Column*	列[1]	Form	表单	Hyperlink Object	超级连接	Separator	分隔符
Label	标签	FormSet	表单集	ListBox	列表框	Shape	形状
Page*	页面[1]	Grid	表格	OLEBoundControl	OLE 绑定型控件	Spinner	微调
ComboBox	组合框	Header*	标头[1]	OLEContainerControl	OLE 容器控件	TextBox	文本框
Container	容器	Line	线条	OptionButton*	选项按钮[1]	Timer	计时器
Control	控件	Image	图像	OptionGroup	选项组	ToolBar	工具栏

[1] 这些类是父容器的集成部分，在“类设计器”中不能子类化。

所有VFP基类的最小事件集，见表8-2。

表 8-2　Visual FoxPro 基类的最小事件集

事　件	说　明
Init	当对象创建时激活
Destroy	当对象从内存中释放时激活
Error	当类中的事件或方法程序过程中发生错误时激活

所有VFP基类的最小属性集，见表8-3。

表 8-3　所有 Visual FoxPro 基类的最小属性集

属　性	说　明
Class	该类属于何种类型
BaseClass	该类由何种基类派生而来，例如 Form、Commandbutton 或 Custom 等
ClassLibrary	该类从属于哪种类库
ParentClass	对象所基于的类。若该类直接由 VFP 基类派生而来，则 ParentClass 属性值与 BaseClass 属性值相同

8.2　表单与控件

表单与控件都是 VFP 中最常用的对象。VFP 编程的最大特点，就是在可视的环境下以最快的速度和效率开发具有良好用户界面的应用程序，其实质就是利用 VFP 所提供的图形构件快速构造应用程序的输入输出屏幕界面。

控件（Control）是某种图形构件的统称，如“标签控件”、“文本框控件”、“列表框控件”等，利用控件创建对象则是构造应用程序界面的具体方法。

8.2.1　常用控件和内部对象

常用控件由 VFP 的基类提供，共 21 个，每个控件用“表单控件”工具栏中的一个图形按钮表示，其图标、名称及说明见表 8-4。

VFP 还提供了一些内部对象，如表单对象、表单集对象、页对象和工具栏对象等。内部

对象一般是可以直接被使用的，但某些对象是要在建立某对象之后才能被使用。例如：分隔符（Separafor）对象可以直接加入到一个工具栏（ToolBar）对象中当间隔；页（Page）对象只有在建立一个页框（PageFrame）对象之后才能使用，列（Column）对象和列表头（Header）对象都是在建立一个表格（Grid）对象之后才能被使用。

表 8-4　常用控件

图标	名　称	说　明
	标签（Label）	创建一个标签对象，用于保存不希望用户改动的文本，如复选框上面或图形下面的标题
	文本框（Text Box）	创建用于单行数据输入的文本框对象，用户可以在其中输入或更改单行文本
	编辑框（Edit Box）	创建用于多行数据输入的编辑框对象，用户可以在其中输入或更改多行文本
	命令按钮（Command Button）	创建命令按钮对象，用于执行命令
	命令按钮组（Command Group）	创建命令按钮组对象，用于把相关的命令编成组
	选项按钮组（Option Group）	创建选项按钮组对象，用于显示多个选项，用户只能从中选择一项
	复选框（Check Box）	创建复选框对象，允许用户选择开关状态，或显示多个选项，用户可从中选择多于一项
	组合框（Combo Box）	创建组合框或下拉列表框对象，用户可以从列表项中选择一项或人工输入一个值
	列表框（List Box）	创建列表框对象，用于显示供用户选择的列表项。当列表项很多，不能同时显示时，列表可以滚动
	微调（Spinner）	创建微调对象，用于接受给定范围之内的数值输入
	表格（Grid）	创建表格对象，用于在电子表格样式的表格中显示数据
	图像（Image）	创建图像对象，在表单上显示图像
	计时器（Timer）	创建计时器对象，以设定的时间间隔捕捉计时器事件。此控件在运行时不可见
	页框（Page Frame）	创建页框对象，显示多个页面
	ActiveX（ActiveX Control）	创建 OLE 容器对象，向应用程序中添加 OLE 对象
	ActiveX 绑定型（ActiveX Bound Control）	创建 OLE 绑定型对象，可用于向应用程序中添加 OLE 对象。与 OLE 容器控件不同的是，OLE 绑定型控件绑定在一个通用字段上
	线条（Line）	创建线条对象，设计时用于在表单上画各种类型的线条
	形状（Shape）	创建形状对象，设计时用于在表单上画各种类型的形状。可以画矩形、圆角矩形、正方形、圆角正方形，椭圆或圆
	容器（Container）	创建容器对象，在容器中可以包含其他的控件
	分隔符（Separafor）	创建分隔符对象，在工具栏的控制间加上空格
	超级链接（Hyper Link）	使用“超级链接”可以跳转到 Internet 或 Intranet 的一个目标地址上

8.2.2　表单对象

表单（Form）是应用程序的用户界面，也是我们进行程序设计的基础。各种图形、图像、数据等都是通过表单或表单中的对象显示出来，因此表单是一个容器对象。

在 FoxPro 的早期版本中表单被称为屏幕（Screen），在 Visual Basic 中则称为窗体。

1．表单的结构

VFP 的表单具有和 Windows 应用程序的窗口界面相同的结构特征，如图 8-1 的左图所示。

一个典型的表单有图标、标题、极小化按钮、极大化按钮、关闭按钮、移动栏、表单体及其周围的边框。其中除了表单体之外的所有特征都可以部分或全部从表单中被删除。例如，可以创建一个如图 8-1 的右图所示的没有标题的表单。

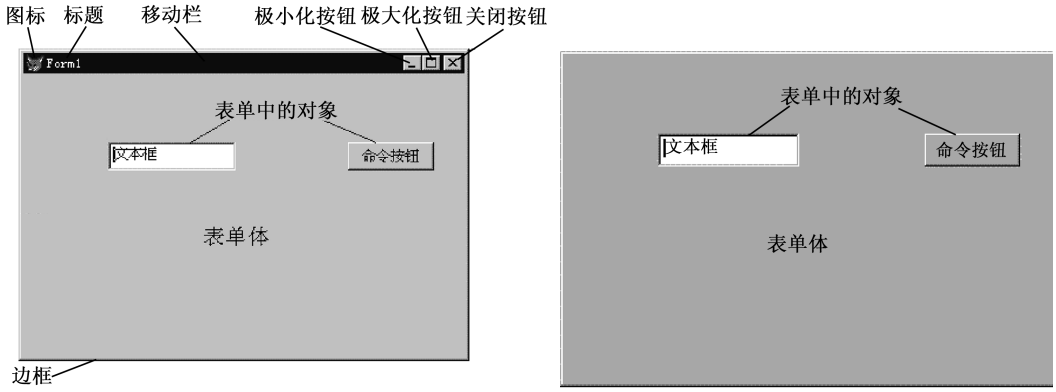


图 8-1　表单的结构

表单的移动栏用来将表单移放到屏幕的任何位置，如图 8-2 的左图所示。表单的可调边框用来在设计或程序运行时调整表单的大小，如图 8-2 的右图所示。

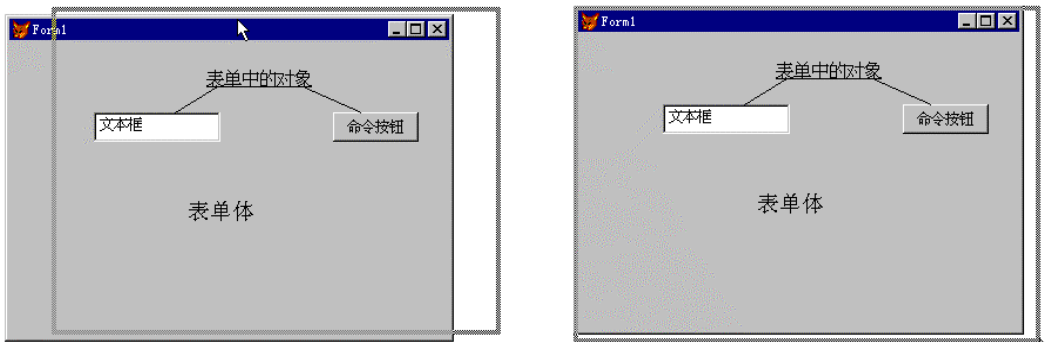


图 8-2　表单的移动与调整边界

单击图标可以打开表单的控制菜单。如图 8-3 所示，控制菜单中的选项与标题栏中的相应按钮功能相同。

表单体是表单的主体部分，用来容纳应用程序所必需的任何控件。本书下文所述在表单上画控件等操作，均指在表单体中的操作。

2．表单的属性

在 VFP 中，表单的属性就是表单的结构特征。通过修改表单的属性可以改变表单的内在或外在的特征。常用的表单属性见表 8-5。

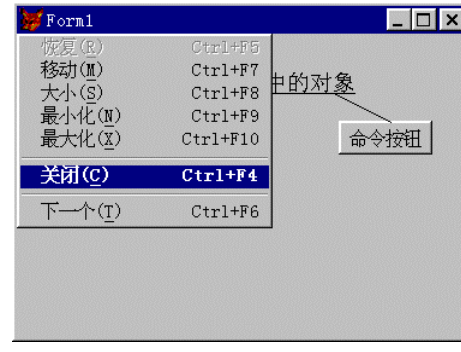


图 8-3　表单的控制菜单

表 8-5　常用的表单属性

属　性　名	作　　用
AutoCenter	用于控制表单初始化时是否总是位于 VFP 窗口或其父表单的中央
BackColor	用于确定表单的背景颜色
BorderStyle	用于控制表单是否有边框：系统（可调）、单线、双线
Caption	表单的标题
Closable	用于控制表单的标题栏中的关闭按钮是否能用
ControlBox	用于控制表单的标题栏中是否有控制按钮
MaxButton	用于控制表单的标题栏中是否有极大化按钮
MinButton	用于控制表单的标题栏中是否有极小化按钮
Movable	用于控制表单是否可移动
TitleBar	用于控制表单是否有标题栏
WindowState	用于控制表单是极小化、极大化还是正常状态
WindowType	用于控制表单是模式表单还是无模式表单（默认），若表单是模式表单，则用户在访问 Windows 屏幕中其他任何对象前必须关闭该表单

3．表单的事件与方法

就像属性那样，只有部分的表单事件与方法经常被使用，很多事件与方法绝少被使用，除非你在编写一个非常复杂的应用程序。我们可以在代码窗口的“过程”下拉列表框中看到所有表单事件与方法的列表，也可以在“属性”窗口的“方法程序”选项卡中看到所有表单事件与方法的列表，如图 8-4 所示。

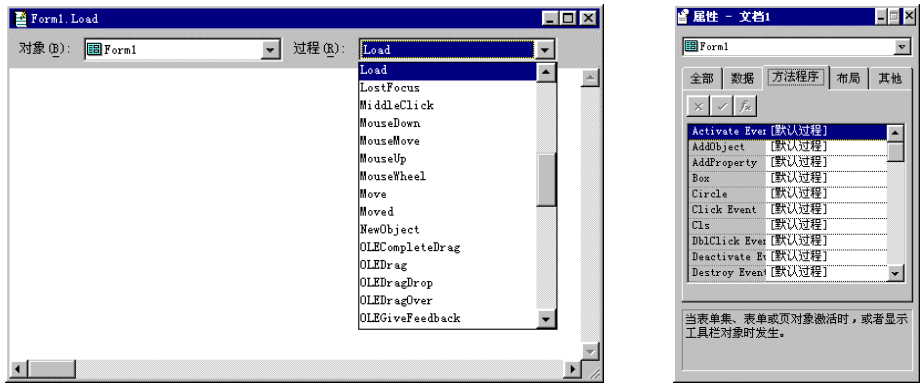


图 8-4　表单的事件与方法列表

下面我们只列举那些最常用的事件与方法。

（1）常用的表单事件

Load 事件：当表单被装入内存时发生。

Init 事件：当表单被初始化时发生。如果表单上包含其他控件，将先引发控件的 Init 事件。

Activate 事件：当表单被激活时发生。

上述事件被激发的顺序为 Load、（控件的 Init 事件）、Init、Activate。

Destroy 事件：当表单被释放时发生。如果表单上包含其他控件，将先引发表单的 Destroy

事件，然后再引发控件的 Destroy 事件。

Unload 事件：当表单被关闭时发生。

上述事件被激发的顺序为 Destroy、（控件的 Destroy 事件）、Unload。

Resize 事件：当用户或程序去改变表单的大小时发生。

（2）常用的表单方法

Hide 方法：通过将 Visible 属性设为.F.，隐藏表单。

Show 方法：通过将 Visible 属性设为.T.，显示表单，并使表单成为活动对象。

Release 方法：从内存中释放表单。

Refresh 方法：重画表单或控件，并刷新所有值。

8.2.3 对象的引用

1．对象的包容层次

VFP 中的对象根据它们所基于的类的性质可分为两类：容器类对象和控件类对象。

容器类对象可以包含其他对象，并且允许访问这些对象，例如表单集、表单、表格等。控件类对象只能包含在容器对象之中，而不能包含其他对象，例如命令按钮、复选框等。表 8-6 列出了每种容器类对象所能包含的对象。

表 8-6　容器类对象所能包含的对象

容　器	能包含的对象
命令按钮组	命令按钮
容器	任意控件
自定义	任意控件、页框、容器、自定义对象
表单集	表单、工具栏
表单	页框、任意控件、容器或自定义对象
表格列	标头对象以及除了表单集、表单、工具栏、计时器和其他列对象以外的任意对象
表格	表格列
选项按钮组	选项按钮
页框	页面
页面	任意控件、容器和自定义对象
工具栏	任意控件、页框和容器

当一个容器包含一个对象时，称该对象是容器的子对象，而容器称为该对象的父对象。所以

- 容器对象可以作为其他对象的父对象。如，一个表单作为容器，是放在其上的复选框的父对象。
- 控件对象可以包含在容器中，但不能作为其他对象的父对象。如复选框就不能包含其他任何对象。

2．对象的引用

作为应用程序的用户界面，表单上可以包含许多对象，而这些对象又有可能具有互相包含的层次关系。若要引用一个对象，需要知道它相对于容器层次的关系。例如，如果要在表

单集中处理一个表单的控件，则需要引用表单集、表单和控件。

在容器层次中引用对象恰似给 VFP 提供这个对象地址。例如，当您给一个外乡人讲述一个房子的位置时，您需要根据其距离远近，指明这幢房子所在的城市、街道，甚至这幢房子的门牌号码，否则将引起混淆。图 8-5 表示了一种可能的容器嵌套方式。

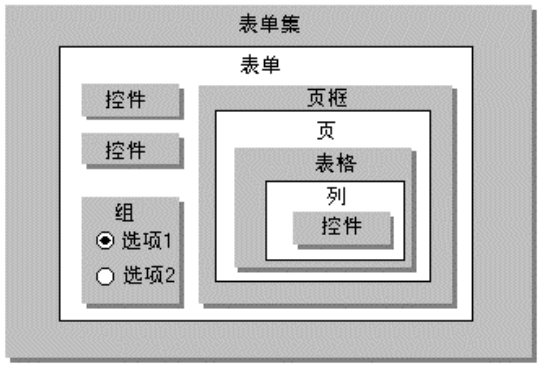


图 8-5　嵌套容器

（1）绝对引用

通过提供对象的完整容器层次来引用对象称为绝对引用。

若要使表列中的控件无效，需要提供以下地址：

```
Formset.Form.PageFrame.Page.Grid.Column.Control.Enabled = .F.
```

应用程序对象（_VFP）的 ActiveForm 属性允许在不知道表单名的情况下处理活动的表单。例如，下列代码改变活动表单的背景颜色，而不考虑其所属的表单集。

```
_VFP.ActiveForm.BackColor = RGB（255,255,255）
```

类似地，ActiveControl 属性允许处理活动表单的活动控件。

```
Name1 = _VFP.ActiveForm.ActiveControl.Name
```

（2）相对引用

在容器层次中引用对象时，可以通过快捷方式指明所要处理的对象，即所谓相对引用。例如：

```
THISFORMSET.Frm1.Cmd1.Caption = "关闭"
```

表示将本表单集的名为 Frm1 的表单中的 Cmd1 对象的标题（Caption）属性设为“ 关闭 ”。此引用可以出现在该表单集的任意表单中的任意对象的事件或方法程序代码中。

```
THISFORM.Cmd1.Caption = "关闭"
```

表示将本表单的名为 Cmd1 对象的标题（Caption）属性设为 “ 关闭 ”。 此引用可以出现在该 Cmd1 所在表单的任意对象的事件或方法程序代码中。

```
THIS.Caption = "关闭"
```

对于需要改变标题的控件，表示将本对象的标题（Caption）属性设为 “ 关闭 ”。此引用

可以出现在该对象事件或方法程序代码中。

THIS.Parent.BackColor = RGB（192,0,0）

表示将本对象的父对象的背景色设置为暗红色。此引用可以出现在该对象事件或方法程序代码中。

表 8-7 列出了一些属性和关键字，这些属性和关键字允许更方便地从对象层次中引用对象。

表 8-7　引用对象的属性和关键字

属性或关键字	引　　用
ActiveControl	当前活动表单中具有焦点的控件
ActiveForm	当前活动表单
ActivePage	当前活动表单中的活动页
Parent	该对象的直接容器
THIS	该对象
THISFORM	包含该对象的表单
THISFORMSET	包含该对象的表单集

说明：只能在方法程序或事件过程中使用 THIS、THISFORM 和 THISFORMSET。

8.3　创建与管理表单

创建表单有两种方法：

- 使用表单向导创建操作数据的表单。
- 使用表单设计器创建、设计新的表单或修改已有的表单。

8.3.1　表单向导

表单向导可以很方便地帮助我们设计一个操作数据的表单。无论对初学者还是程序员，使用向导创建表单，只需按照向导提示一步一步地操作，就可完成表单的设计，既直观又容易。

1．用表单向导创建表单

【例 8-1】设考生文件夹下的数据库学生管理.dbc 中包含表 student 和 cj，使用表单向导选择 student 表生成一个名为 form1 的表单。要求选择 student 表中所有字段，表单样式为“阴影式”，按钮类型为“图片按钮”，排序字段选择“学号”（升序），表单标题为“学生基本数据输入维护”。

设计步骤如下：

① 在“文件”菜单中选择“新建”命令或直接用鼠标单击“常用工具栏”中的“新建”按钮，打开新建对话框，选择“表单”选项，并且单击“向导”按钮，如图 8-6 所示。

② 打开“向导选取”对话框，这里我们选择“表单向导”，如图 8-7 所示。

图 8-6　新建对话框

图 8-7　“ 向导选取 ” 对话框

用 “ 表单向导 ” 为单个表创建操作数据的表单，只需要在一系列简捷的步骤中，回答一些简单的问题，就可以从中指定用于创建控件的表和字段，这些控件将出现在表单上。全部表单向导工作需要四个步骤，以填表选择的方式进行表单的定义。

● 步骤 1 —— 字段选取。

表单向导的第一步是要求用户选择包含在表单中的字段。单击 “ 数据库和表 ” 列表框右边的 “ … ” 按钮，在 “ 打开 ” 对话框中选择所需要的数据库（或表），即可在 “ 可用字段 ” 列表框中选取字段，如图 8-8 的左图所示。

字段的选取根据需要来决定，只能在单独的表或视图中选取字段。

在选取字段时，并不要求按照字段在 “ 可用字段 ” 列表框中的顺序来选择，要根据需要的顺序来选择，只有这样，才能设计出灵活美观的格式。图 8-8 的左图是利用 “ 全选 ” 按钮选择 student 表中字段的情况。单击 “ 下一步 ” 按钮进入样式对话框。

● 步骤 2 —— 选择表单样式。

完成字段选择后，第二步是选择显示风格样式和按钮类型，如图 8-8 的右图所示。这里的选择仅决定显示的样式，并不影响表单本身所起的功能。

“ 样式 ” 列表框中列出了 9 种样式选项，来决定不同的字段显示方式，分别是标准式、凹陷式、阴影式、边框式、浮雕式、新奇式、石墙式、亚麻式和彩色式。

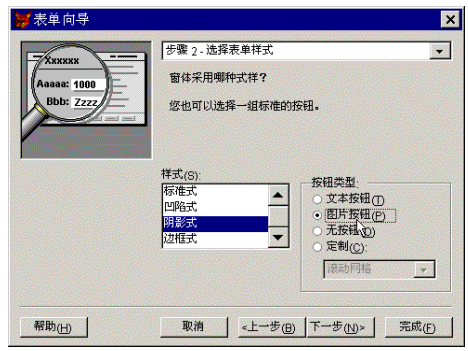


图 8-8　字段选取与选择表单样式

当单击 “ 样式 ” 框中的任何样式时，在向导对话框左上部框中的放大镜中显示一个图片，

可以预览所选择的式样。

“按钮类型”用于表单的定位按钮，分别是文本按钮、图形按钮、无按钮、定制。

向导在表单上创建的定位按钮，见表 8-8。

表 8-8　表单上的定位按钮

文本按钮	图形按钮	说　明
第一个		将记录指针移动到第一个记录
前一个		将记录指针移动到上一个记录
下一个		将记录指针移动到下一个记录
最后一个		将记录指针移动到最后一个记录
查找		显示“搜索”对话框
打印		打印报表
添加		在表末尾添加一个新记录
编辑		允许用户更改当前记录的值
删除		删除当前记录
退出		关闭表单

说明：

由“表单向导”和“表单生成器”创建的控件都保存在类库文件 WIZARDS\WIZSTYLE.VCX 中。如果想改变样式，可修改这个文件中的类。

本例选择“阴影式”及“图形按钮”。

- 步骤 3——排序次序。

如果表单是基于一个表而设计的，就会出现“排序次序”对话框，如图 8-9 的左图所示。“排序次序”对话框用来确定记录排列的顺序，按照记录的排序顺序选择字段。其操作方法与查询向导一样。如果表单是基于一个查询，会跳过这一步。

- 步骤 4——完成。

如图 8-9 的右图所示是向导最后一个对话框。在标题文本框中输入创建表单的标题“学生基本情况表”，它将出现在表单的顶部。对于表单的存储同其他向导一样，有三种选择：存储、存储并立即运行、存储后在表单设计器中修改。保存表单之后，可以像其他表单一样，在表单设计器中打开并修改它。

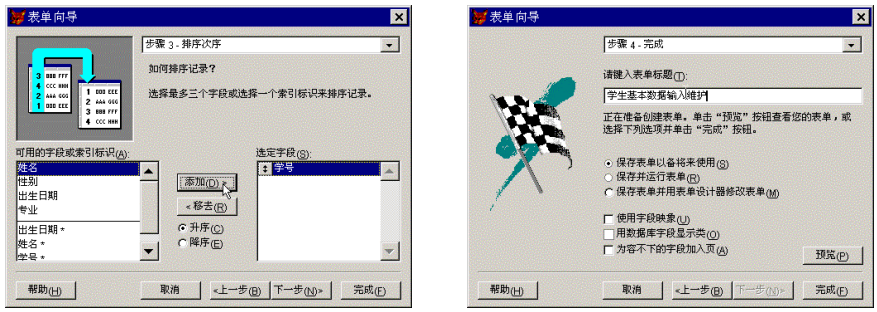


图 8-9　排序次序与完成

单击对话框左下角的“预览”按钮，可以预览设计的表单，如图 8-10 所示。

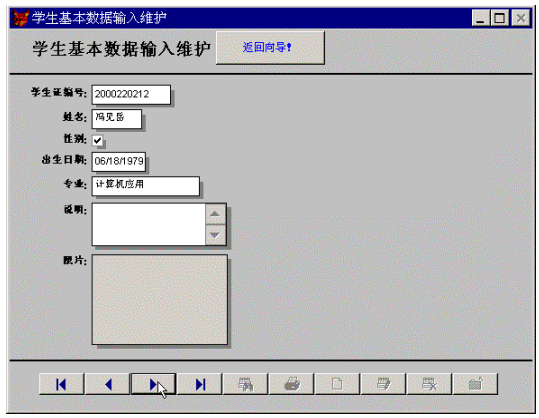


图 8-10 “预览”表单

这就是“表单向导”为我们设计的表单“学生基本情况表”。布局不太理想，所以我们选择“保存表单并修改于表单设计器中”。

最后，单击“完成”按钮，出现“另存为”对话框，存储设计的表单后，结束表单向导操作。

2．用一对多表单向导创建表单

用“一对多表单向导”创建一个表单的过程类似前面的“表单向导”，区别在于要从两个相关表中选取字段和设置关系。在向导中回答一些简单的问题，从中指定表和字段，用这些表和字段创建表单上的控件。

通过在“选项”对话框的表单选项卡中设置最大设计区，可以确定表单的大小。

【例 8-2】设数据库同上例，利用“一对多表单向导”设计操作多表的表单程序。

设计步骤为：

首先，用下列方法之一打开“向导选取”对话框：

① 在“工具”菜单中选择“向导”子菜单，再从子菜单中选择“表单”命令。

② 从“文件”菜单中选择“新建”命令，并依次选取“表单”、“向导”。

③ 选择主工具条上的“表单”按钮。

打开“向导选取”对话框后，选择“一对多表单向导”，如图 8-11 所示。单击“确定”按钮，开始为两个相关的表建立操作数据的表单程序。

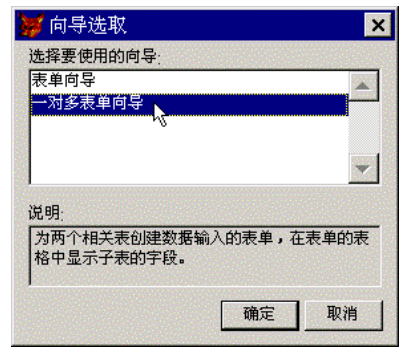


图 8-11 向导选取对话框

- 步骤 1——从父表中选定字段。

选择来自主表中的字段，即一对多关系中的“一”表，只能从单个的表或视图中选取字段，选择方法与前面用过的向导一样。如果需要的表不在“数据库/表”框中，可以按其右边的“...”钮，在“打开”对话框中选择需要的表，例如选择 xs.dbf。然后用字段选择器选择需要的字段。从这个表中选择的字段的记录会显示在表单的上部。其对话框如图 8-12 的左图所示。

图 8-12　从父表中与子表中选定字段

单击“下一步”，进入“从子表中选定字段”对话框。

- **步骤 2——从子表中选定字段。**

选择来自子表中的字段，即一对多关系中的“多”表，只能从单个的表或视图中选取字段。操作方法与步骤 1 相同，例如选择 cj.dbf，并从这个表中选择所需字段。对应于主表的那一个记录的所有记录都会显示在表单的下部。其对话框如图 8-12 的右图所示。

单击“下一步”，进入“建立表之间的关系”对话框。

- **步骤 3——建立表之间的关系。**

确定联系两个表的关键字段，这两个表的关键字段名并不要求相同，只要类型相同就可以联系，可以在下拉框中选择关键字。本例中关键字为“学号”，如图 8-13 的左图所示。

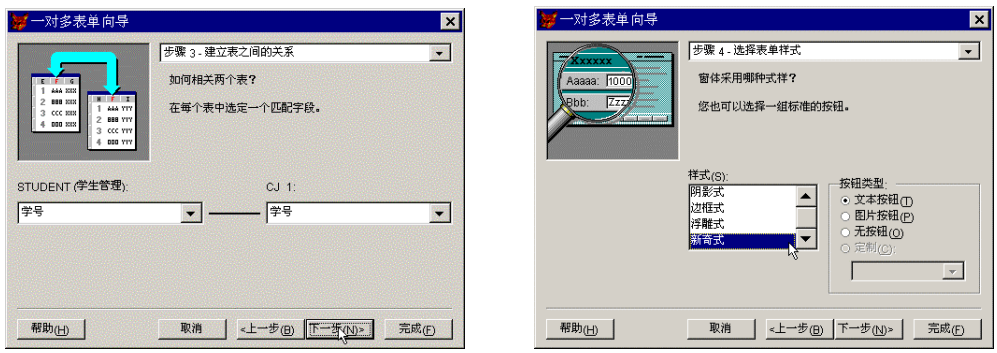


图 8-13　关联表与选择表单样式

单击“下一步”，进入“选择表单样式”对话框。

- **步骤 4——选择表单样式。**

在“样式”框中单击任何样式，向导会在放大镜中显示该样式的示例。这里我们选择“新奇式”，并且在“按钮类型”中选择“图形按钮”，如图 8-13 的右图所示，其操作与“表单向导”相同。

由“表单向导”和“表单生成器”创建的控制都保存在类库文件 WIZARDS\WIZSTYLE.VCX **中。如果想改变样式，应修改这个文件中的类。**

单击“下一步”，进入“排序次序”对话框。

- **步骤 5——排序次序。**

按照记录的排序顺序选择字段，仅对主表中字段的记录起作用，如图 8-14 的左图所示。

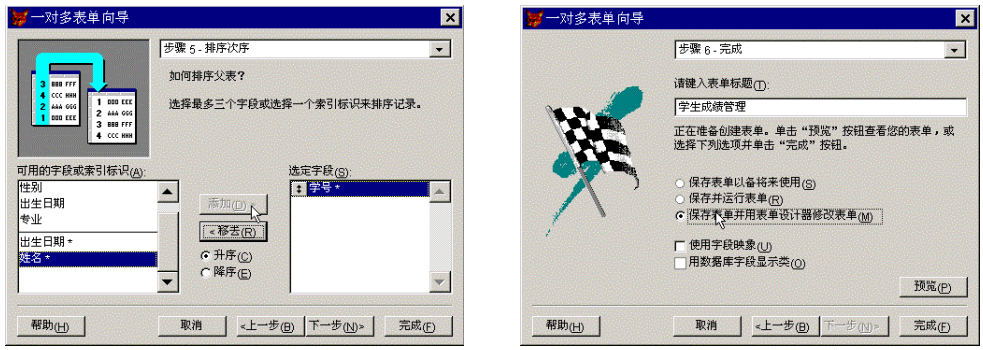


图 8-14　排序次序与完成

单击“ 下一步 ”，进入“ 完成 ”对话框。

●步骤 6 —— 完成。

最后一步是输入标题和生成表单。输入表单的标题为“ 学生成绩管理 ”，如图 8-14 的右图所示。在按下“ 完成 ”按钮之前，我们可以预览生成的表单。如果对其内容不满意，可以逐步返回，重新设置。如果对其格式不满意，可以选择“ 保存表单并修改于表单设计器中 ”。

假如我们选择‘ 保存并运行表单 ’选项，在‘ 另存为 ’对话框中输入表单文件名 student.scx，执行 student.scx 表单，显示如图 8-15 的左图所示的执行结果。

图 8-15　执行表单与添加记录对话框

保存表单之后，可以像其他表单一样，在表单设计器中打开并修改它。

3．一对多表单的使用

在一对多表单中不但查看记录容易，而且编辑、删除、追加操作方便。

如果单击“ 删除 ”按钮，则同时删除主表中的当前记录和子表中的所有相关记录。如果只删除子表中的记录，只需单击子表浏览窗口的删除栏。

如果要追加记录，单击“ 添加 ”按钮，显示如图 8-15 的右图所示的“ 添加记录 ”对话框。

在“ 添加记录 ”对话框中，如果选择：

- 仅对父表添加记录：则要在“ 关键值 ”文本框中输入关键字的值，以便进行惟一性检查。
- 仅对子表（表格）添加记录：“ 关键值 ”文本框中会自动出现父表中当前记录的关键

字值，也可输入新值。

- 两者都添加记录：则先在“关键值”文本框中输入关键字值后，然后输入两个表的记录。

在这种表单中输入记录，可以保证输入“一”表和“多”表中的记录都满足约束条件：

- 对于父表，原始关键字段不能有重复值的记录，由于在建立表结构时选择这个字段为“候选关键字”，系统将保证输入关键字值的惟一性，若输入的记录与前面的记录重复，将拒绝接受。
- 对于子表，每个记录的外部关键字段值与“一”表的原始关键字段值相同，都有相对应的值，不会出现孤记录。这时由于子表的所有记录都使用了一对多表单，找不到对应时系统会拒绝接受。

8.3.2 表单设计器简介

表单设计器是 VFP 提供的一个功能非常强大的表单设计工具，它是一种可视化（Visual）工具，表单的全部设计工作都在表单设计器中完成。

1．打开表单设计器

无论是建立新表单还是修改已有的表单，程序都要打开表单设计器。打开表单设计器有三种方法：

- 在“文件”菜单中选择“新建”命令或直接单击常用工具栏上的“新建”按钮，打开“新建”对话框，在对话框中选择“表单”单选钮并单击“新建文件”按钮。
- 在命令窗口中使用 CREATE FORM 命令。
- 在项目管理器中选择“文档”选项卡，用鼠标选中“表单”，再单击“新建”按钮。在弹出的“新建表单”对话框中单击“新建表单”按钮，如图 8-16 所示。

以上任何一种方法都可以打开“表单设计器”，开始设计新表单。

如图 8-17 所示，为进入表单设计器时的初始画面。

表单设计器中包含一个新创建的表单或是待修改的表单，可在其上添加和修改控件。表单可在表单设计器内移动或是改变其大小。

2．表单设计器工具栏

如果你的屏幕上没有出现“表单设计器”工具栏，可以将鼠标移到标准工具条上的任意位置，单击鼠标右键，从弹出的快捷菜单中选择“表单设计器”，如图 8-18 所示，即可得到“表单设计器”工具栏。

图 8-16 选择“新建表单”

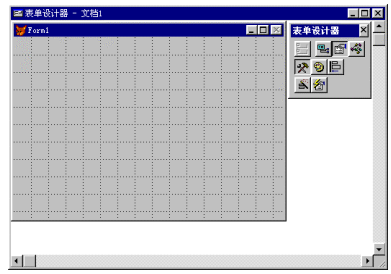


图 8-17 表单设计器窗口

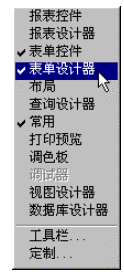


图 8-18 工具栏快捷菜单

或者从“显示”菜单中选择“工具栏”，如图 8-19 的左图所示，在“工具栏”对话框中选择“表单设计器”，然后单击“确定”按钮，如图 8-19 的右图所示，也可得到表单设计器工具栏。

表单设计器工具栏中包括了设计表单时的所有工具。把鼠标指针移到工具栏的某按钮上，就会出现该工具按钮的名称，如图 8-20 所示。各个工具按钮的功能说明见表 8-9。

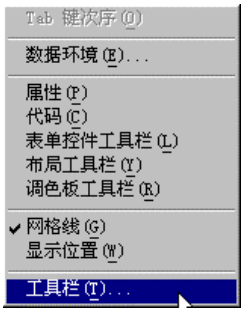


图 8-19　“显示”菜单与“工具栏”对话框图

图 8-20　表单设计器工具栏

表 8-9　工具按钮

图标	名　称	说　明
	设置 Tab 键次序	在表单设计过程中，单击此按钮，可以显示当按动 Tab 键时，光标在表单的各控件上移动的顺序。用键盘上的 Shift 键加上鼠标左键可以重新设置光标移动的顺序
	数据环境	在表单设计过程中，单击此按钮，可以结合用户界面同时设计一个依附的数据环境
	属性窗口	在表单设计过程中，单击此按钮，可以启动或关闭属性窗口，以便在属性窗口中查看和修改各个控件的属性
	代码窗口	在表单设计过程中，单击此按钮，可以启动或关闭代码窗口，以便在代码窗口中编辑各对象的方法及事件代码
	表单控件工具栏	在表单设计过程中，单击此按钮，可以启动或关闭表单控件工具栏，以便于利用各控件进行用户界面的设计
	调色板工具栏	在表单设计过程中，单击此按钮，可以启动或关闭调色板工具栏。利用调色板工具栏可以进行各对象前景与背景颜色的设置
	布局工具栏	在表单设计过程中，单击此按钮，可以启动或关闭布局工具栏。利用布局工具栏可以针对对象进行位置配置和对齐设置
	表单生成器	启动表单生成器，直接以填表的方式进行相关对象的各项设置，以方便快速建立表单
	自动格式	在表单设计过程中，单击此按钮，可以启动或关闭自动格式生成器，以对各控件进行格式设置

3．表单控件工具栏

单击“表单设计器”工具栏上的“表单控件工具栏”按钮，屏幕出现“表单控件”工具栏，可以把它拖放到适当的位置，如图 8-21 所示。

如前所述，“表单控件”工具栏中提供了 VFP 可视化编程的各种控件，利用这些控件，可以创建出所需要的对象。除了各种控件以外，工具栏中的其他按钮见表 8-10。

表 8-10　工具栏中的其他按钮

图标	名　称	说　明
	选定对象	选定一个或多个对象，移动和改变控件的大小。在创建了一个对象之后，“选择对象”按钮被自动选定，除非按下了“按钮锁定”按钮

（续）

图标	名　　称	说　　明
	查看类	单击可以激活此项，使用户可以选择显示一个已注册的类库。在选择一个类后，工具栏只显示选定类库中类的按钮
	生成器锁定	生成器锁定方式可以自动显示生成器，为任何添加到表单上的控件打开一个生成器
	按钮锁定	按钮锁定方式可以添加多个同种类型的控件，而不需多次按此控件的按钮

4．属性窗口

在设计时修改或设置属性，一般是在“属性窗口”中进行。

单击鼠标右键，在弹出的快捷菜单中选取“属性”，如图 8-22 所示，或单击“表单设计器”工具栏中“属性窗口”按钮，可打开“属性”窗口，如图 8-23 所示。

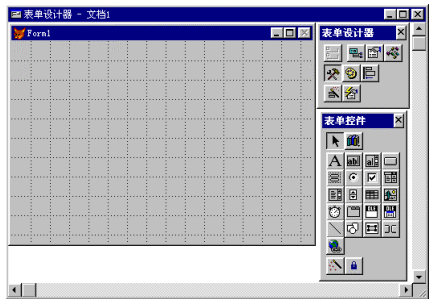


图 8-21　带有“表单控件”工具栏的表单设计器

图 8-22　选取“属性”

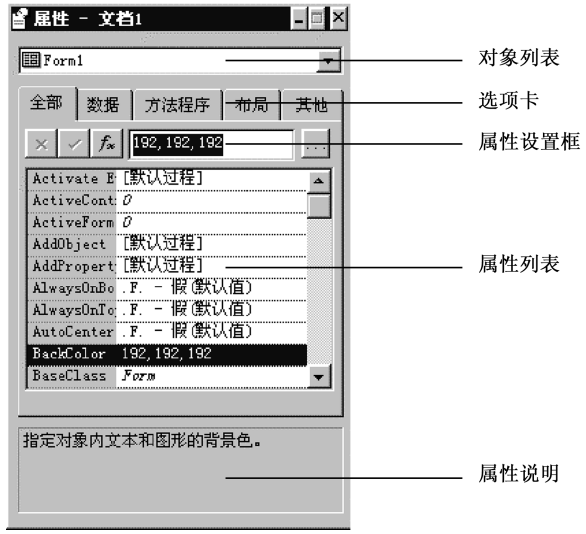


图 8-23　“属性”窗口

属性窗口包含选定对象（表单或控件）的属性、事件和方法列表。可在设计或编程时对这些属性值进行设置或更改。属性窗口从上到下依次包括：

（1）“对象”下拉列表框

标识当前选定的对象。单击右端的向下箭头，可看到包括当前表单（或表单集）及其所包含的全部对象的列表。可以从列表中选择要更改其属性的表单或对象。

（2）“ 选项卡 ”

按分类方式显示所选对象的属性、事件和方法。依次为：

全部：显示全部属性、事件和方法。

数据：显示所选对象如何显示或怎样操纵数据的属性。

方法程序：显示方法和事件。

布局：显示所有的布局属性。

其他：显示其他和用户自定义的属性。

（3）“ 属性设置框 ”

可以更改属性列表中选定的属性值。如果选定的属性具有预定义的设置值，则在右边出现一个向下箭头。如果属性设置需要指定一个文件名或一种颜色，则在右边出现三点按钮。单击“ 接受 ”按钮（√号）来确认对此属性的更改。单击“ 取消 ”按钮（×号）取消更改，恢复以前的值。有些属性（如背景色）显示一个三点按钮，允许从一个对话框中设置属性。单击“ 函数（fx）”按钮，可打开表达式生成器。属性可以设置为原义值，也可以设置为由函数或表达式返回的值。

“ 属性列表 ”:这个包含两列的列表显示所有可在设计时更改的属性和它们的当前值。对于具有预定值的属性，在“ 属性 ”列表中双击属性名可以遍历所有可选项。对于具有两个预定值的属性，在“ 属性 ’列表中双击属性名可在两者间切换。选择任何属性并按〈F1〉键可得到此属性的帮助信息。对于以表达式作为设置的属性，它的前面具有等号（=）。只读的属性、事件和方法以斜体显示。

在属性窗口中以上各项之外的地方单击鼠标右键，将弹出快捷菜单，如图 8-24 所示。

图 8-24　弹出式菜单

通过相应的选择可以改变“ 属性 ”窗口的外观。

5．代码窗口

“ 代码 ”（Code）窗口是编写事件过程和方法代码的地方。下述方法之一可以打开代码窗口：

- 在表单中用鼠标右键单击需要编写代码的对象，在弹出的快捷菜单中选择“ 代码... ”，如图 8-25 的左图所示。
- 单击表单设计器中“ 代码 ”按钮。
- 用鼠标左键双击需要编写代码的对象。

打开的代码窗口如图 8-25 的右图所示。

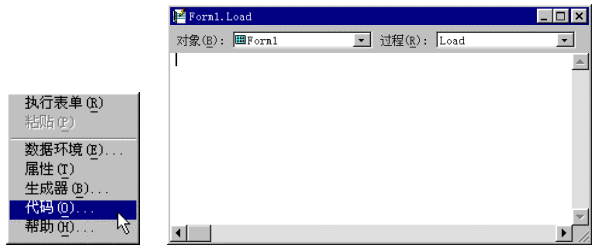


图 8-25　快捷菜单与代码窗口

8.3.3 使用表单设计器创建和修改表单

与传统的面向过程的编程方法不同，VFP 采用面向对象、事件驱动的编程方法。程序员不再以“过程”为中心思考应用程序开发的结构，而是面对可视的“对象”考虑如何响应用户的动作。也就是说，程序员不必编写一个大型的程序，而是建立若干“对象”以及相关的微小程序，这些微小程序都可以由用户启动的事件来激发。因此，VFP 可视化编程的一般步骤为：

① 建立应用程序的用户界面，主要是建立表单，并在表单上安排应用程序所需的各种对象（由控件创建）。

② 设置各对象（表单及控件）的属性。

③ 编写方法及事件过程代码。

当然，也可以边建立对象，边设置属性和编写方法及事件过程代码。下面将通过建立一个最简单的表单来介绍可视化编程的一般方法和表单设计器的使用。

【例 8-3】假设当前文件夹中存有数据表 order_detail.dbf，利用“表单设计器”设计可以浏览数据表的表单程序。

（1）添加控件

首先在表单上增加一个控件：

① 单击“表单控件”工具栏中的“命令按钮”。

② 将鼠标指向表单的左下部，按下鼠标左键并拖动鼠标的十字指针画出一个矩形框，松开左键即画出一个“命令按钮”，如图 8-26 所示。按钮内自动标有“Command1”，然后，用同样的方法再画一个命令按钮，按钮内自动标有“Command2”，其中序号自动增加。

（2）修改属性

设计时的设置和修改属性一般都在属性窗口进行，其操作步骤为：

首先，在表单上用鼠标单击命令按钮“Command1”或在“对象”下拉列表框中选择对象“Command1”，将其标题 Caption 属性改为“浏览”（原值为 Command1），如图 8-27 的左图所示。将其名属性 Name 改为“Browse”（原值为 Command1），如图 8-27 的右图所示。用同样的方法修改命令按钮 Command2 的属性：将其标题 Caption 属性改为“关闭”。

图 8-26　增加一个“命令按钮”

图 8-27　修改命令按钮 Command1 的属性

修改后的表单画面如图 8-28 所示。

（3）编写代码

编写代码就是为对象编写事件过程或方法，首先打开代码窗口。

窗口中的“对象”下拉列表框中列出当前表单及所包含的所有对象名 Form1、Browse、Command2。其中 Browse 与 Command2 对象前的缩进表示对象的包容关系，如图 8-29 所示。

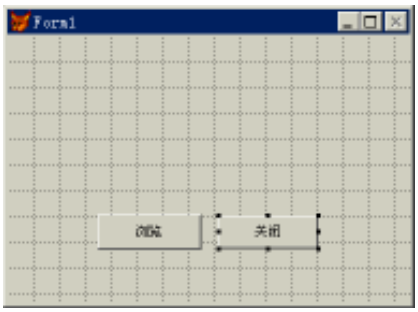


图 8-28　修改后的表单画面

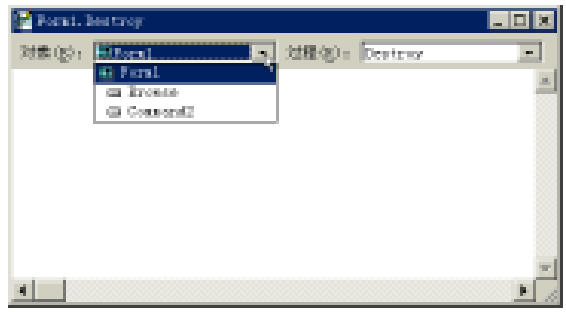


图 8-29　打开代码窗口

“过程”下拉列表框中可以列出所选对象的所有方法及事件名。

首先，在“对象”下拉列表框中选择“Form1”对象，在“过程”下拉列表框中选择“Load”，并在代码窗口输入代码：

```
USE order_detail
```

然后，在“过程”下拉列表框中选择“Destroy”，并在代码窗口输入代码：

```
USE
```

再在“对象”下拉列表框中选择“Browse”对象，在“过程”下拉列表框中选择“Click”，并在代码窗口输入代码：

```
BROWSE
```

如图 8-30 所示。

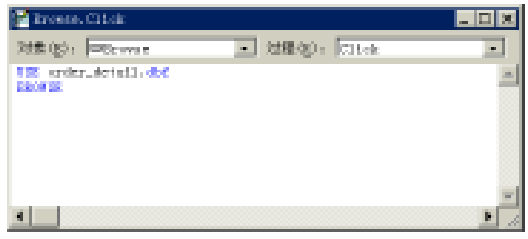


图 8-30　输入代码

最后，在“对象”下拉列表框中选择“Command2”对象，在“过程”下拉列表框中选择“Click”，并在代码窗口输入代码：

```
THISFORM.Release
```

该代码的作用是调用表单的 Release 方法来释放表单。

单击代码窗口右上角的关闭按钮，关闭代码窗口。然后，单击“表单设计器”窗口右上

角的“关闭”按钮，关闭表单设计器，此时，系统提示是否保存所作的改变，如图 8-31 所示。

选择“是”，打开“另存为”对话框，如图 8-32 所示。选择表单文件名为 MyForm。系统将以表单文件 MyForm.scx 存盘。

图 8-31　确认保存对话框

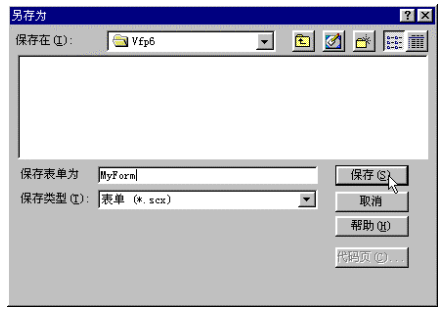


图 8-32　“另存为”对话框

（4）运行表单

有多种运行表单的方法：

- 在命令窗口键入 DO FORM〈表单名〉。
- 在程序代码中使用命令 DO FORM〈表单名〉。
- 在未退出“表单设计器”时，单击常用工具栏中的“运行”按钮，如图 8-33 所示。

表单的运行结果是否和预料的完全一样呢？我们只写了少许几行代码的“程序”，就具有标准的 Windows 风格：图标、标题、极小化按钮、极大化按钮、关闭按钮、移动栏、表单体及其周围的边框。试试极小化按钮、极大化按钮；试试移动表单到屏幕的其他位置。

最后用鼠标单击“浏览”按钮，打开浏览窗口，如图 8-34 所示。

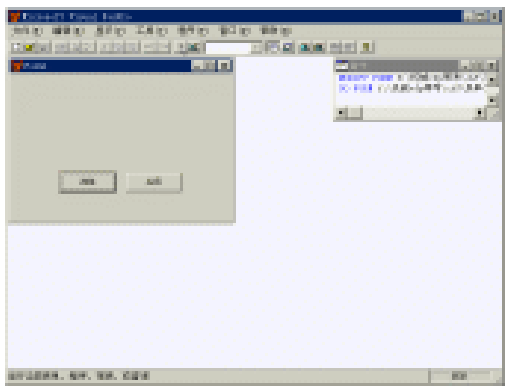


图 8-33　运行表单 MyForm

图 8-34　浏览数据表

单击表单右上角的“关闭”按钮，返回表单。单击命令按钮“关闭”，如果是用上述第三种方式运行表单的，此时将返回“表单设计器”，否则返回 VFP 主屏幕。

（5）修改表单

下面，来修改刚创建的表单，使之具有一个“编辑”按钮，如图 8-35 的左图所示。

图 8-35　增加“编辑”按钮

修改一个表单有三种方法：

- 在“文件”菜单中选择“打开”命令或直接单击常用工具栏上的“打开”按钮，在“打开”对话框的“文件类型”下拉列表框中选择“表单（*.scx）”，然后在列出的表单文件中选择所要的表单名，如图 8-36 所示。
- 在命令窗口中使用命令 MODIFY FORM〈表单名〉。
- 在项目管理器中选择所要修改的表单名称，再单击“修改”按钮，如图 8-37 所示。

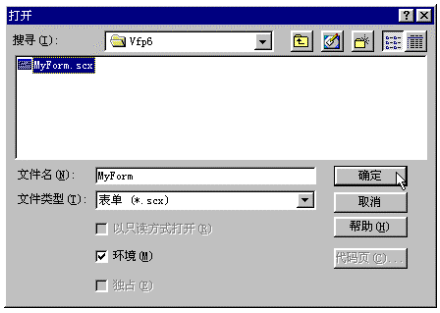


图 8-36　“打开”对话框

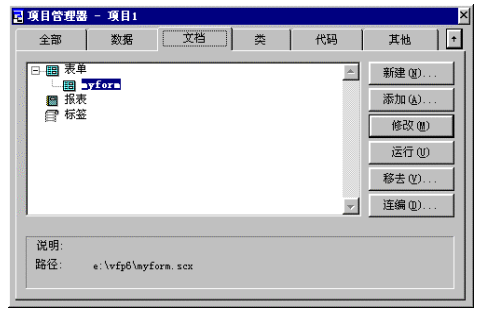


图 8-37　项目管理器

上述方法均可再次进入“表单设计器”。修改表单 MyForm 的步骤为：

① 在表单中再增加一个命令按钮 Command1，修改其 Caption 属性为编辑。

② 修改其 Name 属性为 Edit。

③ 打开代码编辑器，编写其 Click 事件代码：

```
EDIT
```

④ 保存表单。

⑤ 单击“常用工具栏”上的“运行”按钮，运行表单。屏幕显示如图 8-35 的左图所示的表单，单击“编辑”按钮，表单显示如图 8-35 的右图所示。

⑥ 关闭表单返回表单设计器。

8.3.4　数据环境

虽然在表单中可以使用 USE 命令来打开和关闭数据表，但是在一些较为复杂的情况下，比如使用多表或数据库时，不易协调各表之间的关系。比较可靠的办法是使用“数据环境”。

数据环境是一个对象，它包含与表单相互作用的表或视图，以及表单所要求的表之间的关系。可以在“数据环境设计器”中直观地设置数据环境，并与表单一起保存。

每一表单或表单集都可以包含一个数据环境。在表单运行时，数据环境可自动打开、关闭表和视图。

如果在表单或表单集中创建了数据环境，就可以通过“属性”窗口来设置控件的 ControlSource 属性。

【例 8-4】在上例的表单中使用数据环境。

设计步骤如下：

① 创建“数据环境”。选择新建表单，进入表单设计器。在系统菜单的“显示”子菜单中选择“数据环境”命令，或在表单设计器中单击鼠标右键，从弹出的快捷菜单中选择“数据环境”命令，或单击表单设计器中“数据环境”按钮，如图 8-38 所示，均可打开“数据环境设计器”窗口。

在“数据环境”窗口中单击鼠标右键，在快捷菜单中选择“添加”命令，可添加表单所要控制的数据表 order_detail，如图 8-39 所示。

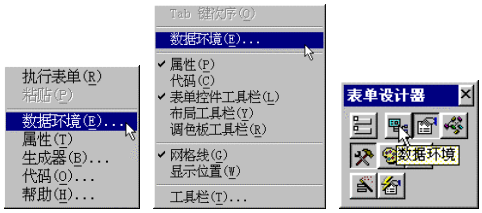


图 8-38　选择“数据环境”

图 8-39　“数据环境”设计器

② 修改代码。删除上例表单中的 Load 事件代码与 Destroy 事件代码，改由数据环境来处理数据表的打开与关闭。

③ 运行表单，结果完全相同。

8.3.5　控件的画法

在设计用户界面时，要在表单上利用 VFP 提供的可视化控件画出各种所需要的对象。为了与在表单运行时由程序添加的对象相区别，我们把由控件创建的对象仍称为控件，并且把由控件创建对象的过程称为“画控件”。

下面我们详细介绍控件的画法。

1．在表单上画一个控件

在表单上画一个控件有两种方法：

- 第一种方法前面已作介绍，即单击“表单控件工具栏”中的某个图标，然后在表单适当位置拖动鼠标画出控件。
- 第二种方法是单击“表单控件工具栏”中的某个图标，然后在表单适当位置单击鼠标左键即可在表单的相应位置画出该控件。

与第一种方法不同的是，用第二种方法所画控件的大小是固定的。当然，任何方法画出的控件都可以用下面介绍的方法改变其大小和位置。

2．控件的缩放和移动

在前面画控件的过程中，我们看到，刚画完的控件其边框上有 8 个黑色小方块，表明该控件是“活动”的，活动控件也称“当前控件”，如图 8-40 所示。当表单上有多个控件时，一般只有一个控件是活动的（除非进行了多重选定），对控件的所有操作都是针对活动控件进行的。所以，为了对一个不活动的控件进行指定的操作，必须将其变为活动控件。单击不活动控件（鼠标指向其内部）可使该控件成为活动控件，而单击活动控件的外部，则可使该控件变为不活动控件。

用鼠标拖拉活动控件边框上的小方块可以使控件在相应的方向放大与缩小。按下〈Shift〉键，用左右方向键可以调整控件的宽度，用上下方向键可以调整控件的高度。

当控件为活动控件时，用键盘的方向键可以使控件向相应的方向移动。也可以把鼠标指向控件内部，拖动控件到表单的任何位置。

除了以上方法外，还可以通过修改某些属性来改变控件的大小与位置。有 4 种属性与表单及控件的大小与方向有关，即 Width，Height，Top 和 Left。其中（Left，Top）是表单或控件左上角的坐标，Width 是其宽度，Height 是其高度。坐标的原点在 Windows 窗口或表单的左上角，单位由 ScaleMode 属性确定。在属性窗口的“Layout”栏中可以找到这些属性。

3．控件的复制与删除

可以对控件进行复制与删除的操作。先将所要操作的控件变为“活动控件”，按〈Ctrl〉+〈C〉键可将该控件复制到 Windows 的剪贴板中，按〈Ctrl〉+〈V〉键可以在表单中得到该控件的复制品。对于活动控件，只须按〈Delete〉键即可删除该控件。

还可以通过“编辑”菜单中的相应命令，或常用工具栏上的相应按钮，对控件进行复制与删除的操作。

最快的方法是直接在要操作的控件上单击鼠标右键，打开快捷菜单，在快捷菜单中选取需要的项。

4．在表单上画多个同类控件

若要在表单上画出多个同类的控件，可以利用“按钮锁定”功能。在表单控件工具栏中选择“按钮锁定”按钮，如图 8-41 所示。然后单击表单控件工具栏中的某个所需控件的图标，就可以在表单上连续画出控件（不必每画一个，单击一次图标），直到再用鼠标单击“按钮锁定”按钮而取消该功能。

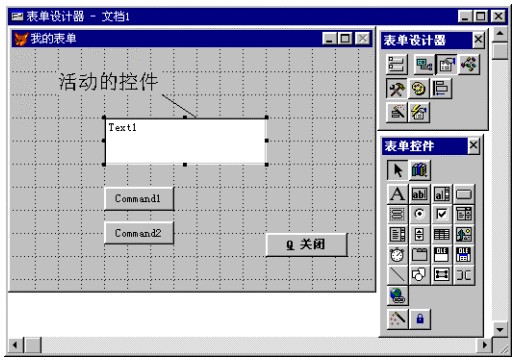


图 8-40　活动的控件

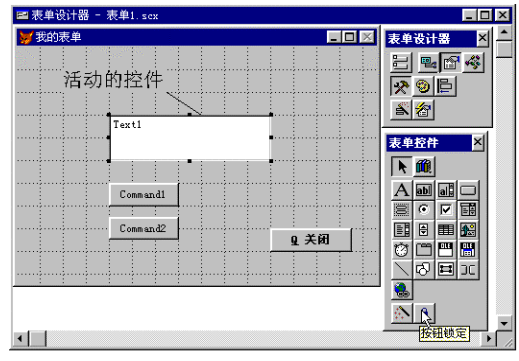


图 8-41　按钮锁定

5．布局工具栏

当表单上有多个控件时，可以使用“布局工具栏”对控件进行各种形式的对齐操作。在表单设计器工具栏中，单击“布局工具栏”按钮可打开“布局”工具栏，如图 8-42 所示。

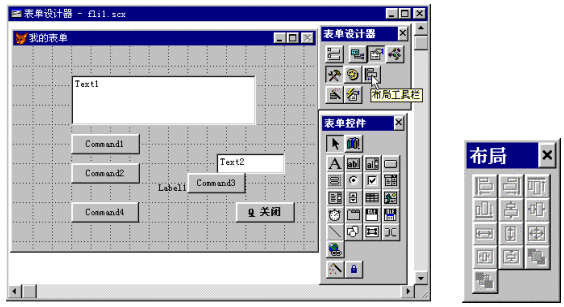


图 8-42　打开“布局”工具栏

此时布局工具栏中的按钮全都处于不可用的状态，原因是没有选定的控件。使一组控件同时被选定称为多重选定。经多重选定的控件才可调整其相互之间的位置。

进行多重选定时，先按下键盘上的〈Shift〉键，再用鼠标单击所要选择的控件。或者直接用鼠标在表单上拉出一个矩形，凡是与此矩形相交的控件均被选定，如图 8-43 所示。

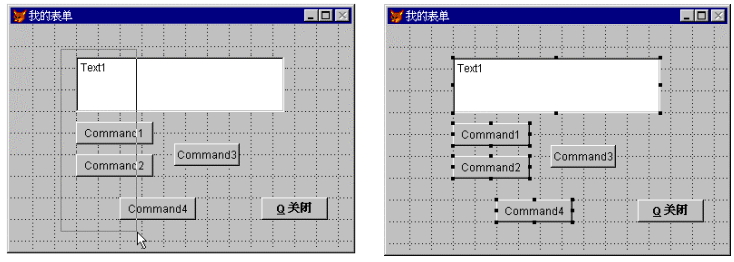


图 8-43　多重选定

布局工具栏有 13 个工具按钮，其名称与说明见表 8-11。

表 8-11　布局工具栏按钮

图　标	名　称	说　明
	左边对齐	被选择的控件靠左边对齐
	右边对齐	被选择的控件靠右边对齐
	顶边对齐	被选择的控件靠顶端对齐
	底边对齐	被选择的控件靠底端对齐
	垂直居中对齐	被选择的控件按其垂直的中心对齐
	水平居中对齐	被选择的控件按其水平的中心对齐
	相同宽度	被选择的控件按最宽的控件设置相同的宽度
	相同高度	被选择的控件按最高的控件设置相同的高度

（续）

图　标	名　称	说　明
	相同大小	被选择的控件设置相同的大小
	水平居中	被选择的控件按表单的水平中心线对齐
	垂直居中	被选择的控件按表单的垂直中心线对齐
	置前	被选择的控件设置为前景显示
	置后	被选择的控件设置为背景显示

布局工具栏的操作如图 8-44 所示。

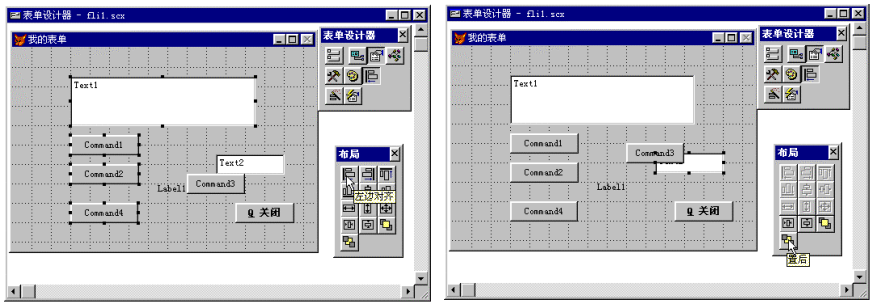


图 8-44　“靠左边对齐”与“置控件为背景显示”

对控件的置前、置后操作在设计表单的时候非常有用，比如，当表单上的控件设计完成后，我们希望用“方框”（形状控件）将其分组，但是后画的形状控件将覆盖原有的控件，按“置后”按钮即可将形状控件放置到原有控件的背后，如图 8-45 所示。

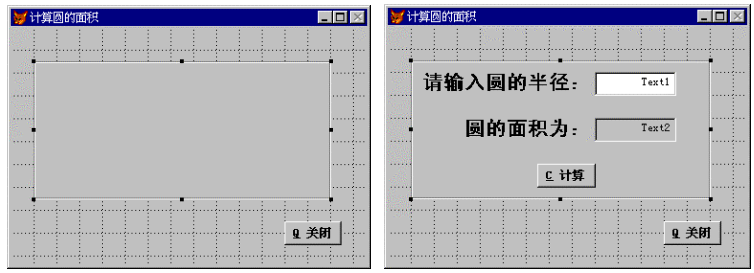


图 8-45　将形状控件放置到原有控件的背后

8.4　习题

一、选择题

1．能够将表单的 Visible 属性设为.T.，并使表单成为活动对象的方法是（　　）。

A）Hide　　B）Show　　C）Release　　D）SetFocus

2．下列关于属性、方法和事件的叙述中，哪个是错误的（　　）？

A）属性用于描述对象的状态，方法用于表示对象的行为

B）基于同一个类产生的两个对象可以分别设置自己的属性值

C）事件代码也可以像方法一样被显式调用

D）在新建一个表单时，可以添加新的属性、方法和事件

3．下面关于表单控件基本操作的陈述中，哪个是不正确的（　　）？

A）要在“表单控件”工具栏中显示某个类库文件中自定义类，可以单击工具栏中的“查看类”按钮。然后在弹出的菜单中选择“添加”命令

B）要在表单中复制某个控件，可以按住〈Ctrl〉键并拖放该控件

C）要使表单中所有被选控件具有相同的大小，可单击“布局”工具栏中的“相同大小”按钮

D）要将某个控件的 Tab 序号设置为 1，可在进入 Tab 键次序交互式设置状态后，双击控件的 Tab 键次序盒

4．下面关于数据环境和数据环境中两个表之间关系的陈述中，哪个是正确的（　　）？

A）数据环境是对象，关系不是对象

B）数据环境不是对象，关系是对象

C）数据环境是对象，关系是数据环境中的对象

D）数据环境和关系都不是对象

5．假定表单中包含一个命令按钮，那么在运行表单时，下面有关事件引发次序的陈述中，哪个是正确的（　　）？

A）先命令按钮的 Init 事件，然后表单的 Init 事件，最后表单的 Load 事件

B）先表单的 Init 事件，然后命令按钮的 Init 事件，最后表单的 Load 事件

C）先表单的 Load 事件，然后表单的 Init 事件，最后命令按钮的 Init 事件

D）先表单的 Load 事件，然后命令按钮的 Init 事件，最后表单的 Init 事件

二、上机题

1．考生目录下有一个 form1 表单文件，其中三个命令按钮 click 事件下的语句都是错误的，请按如下要求进行修改（最后保存所做的修改）：

（1）单击“刷新标题”命令按钮时，使表单的标题为“简单应用”，按钮的 Click 事件代码为：

```
thisform = "简单应用"
```

（2）单击“订单记录”命令按钮时，使表格控件中显示 order_list 表中的记录，按钮的 Click 事件代码为：

```
thisform.grid1="order_list.dbf"
```

（3）单击“关闭表单”命令按钮时，关闭表单，按钮的 Click 事件代码为：

```
thisform.close
```

注意：每处错误只能在原语句上进行修改，不可以增加语句。

2．在考生文件夹下有一个名称为 form1 的表单文件，该表单中的三个命令按钮的 click 事件下的语句有错误。请按如下要求进行修改，修改完成后保存所做的修改：

（1）单击“刷新标题”命令按钮时，将表单的标题改为“商品销售数据输入”，按钮的 Click 事件代码为：

```
ThisForm.标题 = "商品销售数据输入"
```

（2）单击“商品销售输入”命令按钮时，调用当前文件夹下的名称为 sellcomm 的表单文件打开数据输入表单，按钮的 Click 事件代码为：

```
DO sellcomm
```

（3）单击“输出销售报表”命令按钮时，调用当前文件夹下的名称为 print1 的报表文件对报表进行预览，按钮的 Click 事件代码为：

```
DO  print1 TO PREVIEW
```

注意：每处错误只能在原语句上进行修改，不可以增加语句行。

3．设计一个名称为 form2 的表单，上面有“调整”（名称为 Command1）和“退出”（名称为 Command2）两个命令按钮。

单击“调整”命令按钮时，调用 change_c 命令程序实现商品单价调整。

单击“退出”命令按钮时，关闭表单。

注意：以上两个命令按钮均只含一条语句，不可以有多余的语句。

第 9 章　常用标准控件

VFP 表单程序的设计方法属于“可视化编程”，其特点是无需编写大量的代码，只需从工具箱中选择所需对象（控件）在表单上画出，并为每个对象设置属性即可完成程序界面的设计。因此，使用和了解控件是可视化编程的最基本要求，本章将介绍控件工具栏中的一些基本控件，使大家对 VFP 的控件所具有的灵活性和通用性有初步的了解。

9.1　公共的属性与事件过程

在 VFP 所使用的对象（表单及控件）中，很多属性与事件是共有的，这些共有的属性与事件在不同对象中的功能和用法基本相同。

9.1.1　常用的公共属性

1．名称属性（Name）

该属性为字符型数据，用于确定对象的名字，即程序代码中使用的对象标识符。新建一个对象后，系统会为其规定一个默认的名字。用户可以更改也可保持默认名字不变。该属性只能在设计时更改，运行时只读。所有对象（表单及控件）都有名称属性。

2．标题属性（Caption）

该属性为字符型数据，用于返回或设置对象（如表单的标题栏、标签、命令按钮）所显示的文本信息。例如，下述语句将当前表单的标题栏内容设置为“商品销售数据输入”：

```
THISFORM.Caption = "商品销售数据输入"
```

该属性可以在设计阶段通过属性窗口或在运行阶段用程序代码设置。

在为命令按钮、标签、选项按钮、复选框定义标题时，可为其定义一个键盘操作的热键，其定义方法是：在输入标题内容时，将需要定义成热键的字母前加入一个“\<”符号。运行时该字母下面显示的下划线表明该字母为热键，用户只需按下〈Alt〉+该字母键，便可激活并执行该命令按钮。例如，将命令按钮的 Caption 属性设置为“取消（\<C）”，即为该按钮定义了一个热键“C”。

3．位置与大小属性（Left、Top、Width、Height）

该组属性为数值型数据，用于设置或返回对象在其容器中的显示位置和对象的大小。对于表单来说，Left 和 Top 属性用于设置表单左上角在屏幕中的坐标位置；Width 和 Height 属性用于设置表单的宽度和高度。对于控件来说，其容器可以是表单、容器控件或页框控件等。

此属性可以在设计阶段通过属性窗口或在运行阶段用程序代码设置。

4．与字体有关的属性

与字体相关的属性有 FontName、Fontsize、FontBold、FontItalic、FontStrikethru 和 FontUnderline，它们分别用于设置字体名、字体大小、粗体、斜体、删除线和下划线。其中，后 4 项属性为逻辑型数据.T.，表示该项生效；.F.表示该项不生效。例如：

```
THIS.FontName = "隶书"                    && 设置字体为隶书
```

```
THIS.FontSize = 24                  && 字体大小为 24 点
THIS.FontBold = .T.                 && 字体效果为粗体
THIS.FontStrikethru = .T.           && 字体效果为下划线
THIS.FontItalic = .T.               && 字体效果为倾斜
? "Now is the time!"                && 输出文本
```

在表单的 Click 事件过程中输入上述代码，即可验证以上设置。

上述属性均可以在设计阶段通过属性窗口或在运行阶段用程序代码设置。

5．前景色属性（ForeColor）与背景色属性（BackColor）

该组属性用于设置对象的前景色与背景色。在设计时，单击属性窗口中该项属性，“ 属性设置框 ”的右边将显示一个下拉式按钮“ … ”，单击该按钮，系统将弹出“ 颜色 ”对话框，从中选择所需的颜色即可。在运行时，可以用代码，通过 RGB 函数来设置所需的颜色。

对于文本框、编辑框、标签、命令按钮、复选框等控件，前景色即为显示文本的颜色。对于表单，前景色则是由“ ? ”输出文本的颜色。

例如，若要在当前表单中输出红色的“ Visual FoxPro ”字符串，代码为：

```
THISFORM.ForeColor = RGB（255, 0, 0）
? "Visual FoxPro"
```

其中 RGB 函数用来指定颜色。

该组属性可以在设计阶段通过属性窗口或在运行阶段用程序代码设置。

6．可用性属性（Enabled）

返回或设置一个逻辑型的值，该属性用于决定表单或控件是否响应用户所产生的事件。该属性默认值为.T.，VFP 的绝大多数控件均有该属性。若将该属性设置为.F.，则表单或控件将失效，即不能响应用户的任何事件或操作。

该属性可通过属性窗口设置或用代码进行修改。在程序设计中，常利用该属性使某个控件暂时失效。

7．可视性属性（Visible）

该属性为逻辑型数据，决定对象是否可见，默认值为.T.。若将其设置为.F.，则对象运行时不可见。

该属性可以在设计阶段通过属性窗口或在运行阶段用程序代码设置。

8．提示文本属性（ToolTipText）

该属性为字符型，用于设置当鼠标在控件上暂停时显示的提示性文本。在许多 Windows 应用程序中，当鼠标暂停于工具栏的某个按钮上时，系统会自动显示该按钮的功能说明，就是通过该属性来实现的。

该属性可以在设计阶段通过属性窗口或在运行阶段用程序代码设置，但要注意，该属性仅当表单的 ShowTips 属性设为.T.时才有效。

9．文本对齐属性（Alignment）

该属性用于设置文本的对齐方式，其取值与对应功能见表 9-1。

10．只读性属性（ReadOnly）

指定用户是否可以编辑控件、临时表对象相关联的表或视图。当该属性取值为“ 真 ”（.T.）时，用户不能编辑控件。不能修改与临时表对象相关联的表或视图。取值为“ 假 ”（.F.）（默

认值）时，用户能够编辑控件。可以修改与临时表对象相关联的表或视图。

该属性可以在设计阶段通过属性窗口或在运行阶段用程序代码设置。

表 9-1　Alignment 属性

值	说　明
0	对于文本框、组合框、编辑框、标题、标签或微调控件，文本左对齐；对于复选框和选项按钮，控件左对齐，文本在控件的右边。除文本框控件，该值为默认值
1	对于文本框、组合框、编辑框、标题、标签或微调控件，文本右对齐；对于复选框和选项按钮，控件右对齐，文本在控件的左边
2	对于文本框、组合框、编辑框、标题、标签或微调控件，文本居中显示
3	自动。对于文本框、组合框、编辑框、标题、标签或微调控件，根据控件的数据源类型对齐文本。该值为文本框控件的默认值

11．控制源属性（ControlSource）

表单中的控件可以分为两类：与表中数据绑定的控件和不与数据绑定的控件。当用户使用绑定型控件时，所输入或选择的值将保存在数据源中（数据源可以是表的字段、临时表的字段或内存变量）。要想绑定控件和数据，可以设置控件的 ControlSource 属性（见表 9-2）。如果要绑定“表格”和数据，则需要设置表格的 RecordSource 属性。

表 9-2　控件 ControlSource 属性设置的作用

控　件	作　用
复选框	如果 ControlSource 是表中的字段，当记录指针在表中移动时，ControlSource 字段中的 NULL 值、逻辑值“真”(.T.）或“假”(.F.）或数值 0，1 或 2 将分别代表复选框被选中、清除或灰色状态
列	如果 ControlSource 是表中的字段，当用户编辑列中的数值时，实际是在直接编辑字段。要将整个表格和数据绑定，可设置表格的 RecordSource 属性
列表框与组合框	如果 ControlSource 是一个变量，用户在列表中选择的值也保存在变量中；如果 ControlSource 是表中的字段，值将保存在记录指针所在的字段中。如果列表中一个项和表中字段的值匹配，当记录指针在表中移动时，将选定列表中的这个项
选项按钮	如果 ControlSource 是一个数值字段，根据按钮是否被选中，在字段中写入 0 或 1。如果 ControlSource 是逻辑型的，则根据按钮是否被选中，在字段中写入“真”(.T.）或“假”(.F.)。如果记录指针在表中移动，则更新选项按钮的值，以反映字段中的新值。如果选项按钮的 OptionGroup 控件（不是选项按钮本身）的 ControlSource 是一个字符型字段，当选择该选项按钮时，选项按钮的标题就保存在字段中。记住，一个选项按钮（与 OptionGroup 控件明显不同）的控件源不能是一个字符型字段，否则当运行表单时，Visual FoxPro 会报告数据类型不匹配
微调	微调控件可以反映相应字段或变量的数值变化，并可以将值写回到相应字段或变量中
文本框或编辑框	表字段中的值在文本框中显示，用户对这个值的改变将写回表中。移动记录指针将影响文本框的 Value 属性

如果没有设置控件的 ControlSource 属性，用户在控件中输入或选择的值只作为属性设置保存。在控件生存期之后，这个值并不保存在磁盘上，也不保存到内存变量中。

部分通过使用控件完成的任务需要将数据与控件绑定，其他任务则不需要。

9.1.2　常用的公共事件

VFP 采用了事件驱动的编程机制，一个程序员，要能灵活自如地编写 VFP 程序，充分利用好各控件的功能，就必须熟悉和掌握各控件所能响应的事件，以及各种事件所产生的背景和触发条件。

根据事件产生的来源，可分为鼠标事件、键盘事件和系统事件三类。鼠标事件是 Windows 应用程序中触发频率最高的事件，用户一般都习惯用鼠标来实现对应用程序的操作。有时也

需要用键盘来进行一些辅助性的操作，所以除鼠标事件外，还有键盘事件。在 VFP 中还有一类特殊的事件，该事件是由系统自动触发的，这类事件称为系统事件。比如，定时器控件，该控件可在一定的时间间隔内自动产生 Timer 事件，以实现一些周期性的操作。

1．Click 事件

当用户在对象上用鼠标左键单击时，当命令按钮、选项按钮或复选框有焦点时按空格键（SPACEBAR），当表单中某命令按钮的 Defualt 属性设为.T.时按〈Enter〉键，均会触发产生相应对象的 Click 事件。也可以在程序代码中触发 Click 事件：

```
THISFORM.Command1.Click
```

将触发当前表单上的命令按钮 Command1 的 Click 事件。

2．DblClick 事件

当用户在对象上用鼠标左键双击时产生 DblClick（双击）事件，也可以在程序代码中触发 DblClick 事件。

3．RightClick 事件

当用户在对象上按下并释放鼠标右键时此事件发生。

4．InteractiveChange 事件

当用户使用键盘或鼠标改变控件的值时，产生 InteractiveChange 事件。

说明：在每次交互地更改对象时，都要发生该事件。例如当用户在文本框中键入字符时，每一次击键都会触发 InteractiveChange 事件。

5．KeyPress 事件

当对拥有焦点的对象进行按下键后又释放的键盘操作时，将会在该对象上触发产生 KeyPress 事件。事件过程格式：

```
LPARAMETERS [nIndex,] nKeyCode, nShiftAltCtrl
```

说明：事件过程中必须包括 LPARAMETERS，并为每个参数指定名称。

该事件被触发时，被按键的 ASCII 码将自动传递给事件过程的 nKeyCode 参数；如果在按下 nKeyCode 所标识的键时，同时按下了修改键〈Shift〉(1)、〈Ctrl〉(2)、〈Alt〉(4)，将同时得到 nShiftAltCtrl 参数值。

在程序中，通过访问两个参数，即可获知用户按下了哪一个键以及与修改键的组合，并可识别字母的大小写。该事件在实际编程中，应用较广泛。

6．MouseDown、MouseUp、MouseMove 事件

当用户按下一个鼠标键时发生 MouseDown 事件，其语法格式为：

```
LPARAMETERS nIndex, nButton, nShift, nXCoord, nYCoord
```

当用户释放一个鼠标键时发生 MouseUp 事件，其语法格式为：

```
LPARAMETERS nIndex, nButton, nShift, nXCoord, nYCoord
```

当用户在一个对象上移动鼠标时发生 MouseMove 事件，其语法格式为：

```
LPARAMETERS nButton, nShift, nXCoord, nYCoord
```

说明：

① 事件过程中必须包括 LPARAMETERS 语句，并为每个参数指定名称。

② nIndex 存放一个数，它惟一标识控件数组中的一个控件。仅当控件是控件数组的一部分时，才传送 nIndex 参数。

③ nButton 存放一个数，它指定为引发事件而按下（或释放）哪个键：1（左），2（右）和 4（中）。

④ nShift 存放一个数，它指定当按下（或释放）nButton 参数指定的键时，〈Shift〉、〈Ctrl〉和〈Alt〉键的状态：1（Shift），2（Ctrl），4（Alt）。

如果按下鼠标时，有多于一个的修改键也被按下，则 nShift 参数是这些修改键的和。例如，按下鼠标按钮时，也按下〈Ctrl〉键，则 nShift 的值为 2。但是如果同时按下〈Ctrl〉和〈Alt〉键，那么 nShift 的值为 6。

⑤ nXCoord，nYCoord 存放鼠标指针在表单中当前的水平（nXCoord）和垂直（nYCoord）位置。这些坐标总是以 ScaleMode 属性的设置值为度量单位，按照指定的表单坐标系统表达的。

注意：MouseMove 使用的 nButton 参数与 MouseDown 和 MouseUp 使用的 nButton 参数不同。对于 MouseMove，nButton 参数表明了所有键的当前状态；一个单独的 MouseMove 事件可以表明部分、全部或没有按下任何键；对于 MouseDown 或 MouseUp，每个事件中 nButton 参数都确切地指明一个键。

7．init 事件

在创建对象时将发生 init 事件。

对于表单集和其他容器对象来说，容器中对象的 Init 事件在容器的 Init 事件之前触发，因此容器的 Init 事件可以访问容器中的对象。容器中对象的 Init 事件的发生顺序与它们添加到容器中的顺序相同。

8．GotFocus、LostFocus、Valid、When 事件

这是一组与焦点有关的事件。

（1）GotFocus 事件

当用户用鼠标或按〈Tab〉键将输入焦点移动到控件上时，将在该控件上触发 GotFocus（获得焦点）事件。

若文本框获得输入焦点时，会在文本框中出现一个“I”形状的光标；若焦点移动到命令按钮上，则在命令按钮的标题四周会出现一个虚线矩形框。获得焦点的先后顺序可由控件的 TabIndex 属性值决定。

（2）LostFocus 事件

当按下〈Tab〉键使焦点移动到其他控件，或用鼠标单击了其他控件或表单时，将会使此控件失去焦点。在控件失去焦点时，将会触发 LostFocus（失去焦点）事件。LostFocus 事件常用于自动判断文本框中所输入的内容是否合法，若输入内容不符合要求，则可在该事件中通过相关语句和方法（如 GotFocus 方法），禁止将焦点移交给下一控件，以增强程序的容错能力。

（3）Valid 事件

在控件失去焦点之前发生。

说明：

① 若 Valid 事件返回“真”(.T.)，表明控件失去了焦点；若返回“假”(.F.)，则说明控件没有失去焦点。

② Valid 事件也可以返回数值，若返回 0，则控件没有失去焦点；若返回正值，则该值指定焦点向前移动的控件数；若返回负值，则该值指定焦点向后移动的控件数。例如，若 Valid 事件返回-1，则焦点由上一个控件得到。

（4）When 事件

在控件接收焦点之前此事件发生。

说明：

① 如果 When 事件返回“真”(.T.)，默认控件接收到焦点；如果返回“假”(.F.)，控件未接收到焦点。

② 在控件获得焦点时，事件顺序为 When 事件和 GotFocus 事件。

③ 对于列表框控件，每当用户单击列表中的项或用箭头键移动使焦点在项之间移动时，When 事件发生。对所有其他控件，当试图把焦点移动到控件上时，When 事件发生。

9.1.3 常用的公共方法

1．Move 方法

Move 方法可以用代码实现对象的移动和缩放。其语法格式为：

```
[〈对象名〉.] Move left[, top[, width[, height]]]
```

其中〈对象名〉.可以是表单或其他大多数的控件，如果省略〈对象名〉.，则表示带有焦点的表单。

2．SetFocus 方法

对文本框而言，最常用的方法要算设置焦点的方法了，其语法为：

```
〈对象名〉.SetFocus
```

此语句可将控制焦点移交给方法作用的对象，如文本框、命令按钮、表单、图片框以及打印机对象（Printer）等。若在表单中有多个文本框，为使输入焦点定位到某指定的文本框，就可利用该方法。例如，将文本输入光标定位到文本框 Text3 中：

```
THISFORM.Text3. SetFocus
```

3．Refresh 方法

一般情况下，画表单或控件是在没有事件发生时自动处理的。需要立刻更新表单或控件时可使用 Refresh 方法。Refresh 方法可以重画表单或控件，并刷新所有值。其语法格式为：

```
[Form.] Object.Refresh
```

说明：

① 可使用 Refresh 方法强制完全重画表单或控件，并更新控件的值。若要在加载另一个表单的同时显示某个表单，或更新控件的内容时，Refresh 方法很有用。

② 要更新组合框或列表框的内容，请使用 Requery 方法。注意，刷新表单的同时，也刷新表单上所有的控件；刷新页框时，只刷新活动的页。

9.2 文本输入与输出控件

标签控件与文本框控件、编辑框控件是 VFP 中用于实现文本输入与输出的重要控件。标签常用于显示静态、不可修改的文本信息；文本框常用于建立文本输入区或编辑区，以实现数据的输入、编辑、修改等；而编辑框则用于多行文本的输入、编辑、修改。

9.2.1 标签

标签（Label）控件显示的文本用户不能直接修改。有些没有自己的标题（Caption）属性的控件（如 Text 和 Edit）可以用标签标识。在标签中显示的文本是由 Caption 属性控制的，该属性可以在设计时在“属性”窗口中设置或在运行时用代码赋值。

1．标签的属性

标签具备控件的一些共有属性，同时也具有一些自身的特殊属性，它们分别描述如下：

（1）Autosize 与 WordWrap 属性

设计时可在“属性”窗口中指定标签的单行标题。对于一个较长的或在运行时可能变化的标题，标签提供了两种属性 AutoSize 和 WordWrap 来改变控件尺寸以适应较长或较短的标题。

为使控件能够自动调整以适应内容多少，可将 AutoSize 属性设置为.T.–真，这样控件可水平扩充并垂直扩充以适应 Caption 属性内容。为使 Caption 属性的内容自动换行，应将 WordWrap 属性设置为.T.–真。

当 AutoSize 和 WordWrap 两种属性都设为.T.–真时，Caption 属性的内容将自动换行并主要在垂直方向上扩充以适应 Caption 属性内容。四种不同设置方法的结果如图 9-1 所示。

图 9-1 使标签适应文本

（2）Backstyle 属性

该属性用于设置标签的背景是透明还是不透明，其取值与对应功能见表 9-3。

表 9-3 Backstyle 属性

值	说 明
0	标签背景透明——在控件后的背景色和任何图片都是可见的。此时标签的 Backcolor 属性无效
1	（默认值）标签背景不透明——此时标签的 Backcolor 属性生效

（3）Borderstyle 属性

该属性用于设置标签的边框风格。其取值与对应功能见表 9-4。

表 9-4 Borderstyle 属性

值	说 明
0	标签无边框（默认值）
1	标签有单线边框

2．标签的使用

标签可用于显示静态的、不允许用户修改的文本信息。由于标签可很方便地进行输出定位，设置文本字体及颜色等，故比使用“?|??”与传统的 SAY 命令要灵活方便得多，是 VFP 中显示文本信息的主要控件。利用标签透明的特点，还可用来设计动态文字。

【例 9-1】利用标签制作阴影文字效果，如图 9-2 所示。程序启动后，在表单上显示出不带阴影的文字“全国计算机等级考试”。单击“效果 1”按钮后文字出现黑色的阴影，如图 9-2 的左图所示。单击“效果 2”按钮后文字阴影的间距加大，如图 9-2 的中图所示。

图 9-2　阴影文字效果

设计步骤如下：

① 设计程序界面及设置控件属性。选择“新建”表单，进入表单设计器，在表单中增加两个命令按钮 Command1、Command2 和一个标签 Label1。并设置属性见表 9-5。

表 9-5　设置对象属性

对　象	属　性	属 性 值	说　明
Command1	Caption	效果 1	
	FontSize	11	字体大小
Command2	Caption	效果 2	
	FontSize	11	字体大小
Label1	Caption	全国计算机等级考试	
	AutoSize	.T. – 真	自动大小
	BackStyle	0 – 透明	背景类型
	FontName	隶书	字体名
	FontBold	.T. – 真	粗体
	FontSize	22	字体大小

选定标签 Label1，按组合键〈Ctrl〉+〈C〉将标签复制到系统剪贴板中，然后再按组合键〈Ctrl〉+〈V〉，在表单上增加了一个标签 Label2。

修改标签 Label2 的 BackStyle 属性为 0（透明），ForeColor 属性改为黄色，如图 9-2 的右图所示。修改标签 Label1 的 Visible 属性为.F.，使之不可见。

② 编写程序代码。

编写 Command1 的 Click 事件代码：

```
THISFORM.Label1.Visible = .T.                    && 设置阴影可见
THISFORM.Label1.Top = THISFORM.Label2.Top + 2    && 阴影较文字向下偏移 2 个像素
THISFORM.Label1.Left = THISFORM.Label2.Left + 2  && 阴影较文字向右偏移 2 个像素
```

编写 Command2 的 Click 事件代码：

```
THISFORM.Label1.Visible = .T.
THISFORM.Label1.Top = THISFORM.Label2.Top + 4
THISFORM.Label1.Left = THISFORM.Label2.Left + 4
```

说明：代码中的：

```
THISFORM.Label1.Top = THISFORM.Label2.Top + 4
THISFORM.Label1.Left = THISFORM.Label2.Left + 4
```

也可以用 Move 方法来实现：

```
THISFORM.Label1.Move（THISFORM.Label2.Left + 4, THISFORM.Label2.Top + 4）
```

9.2.2 文本框

文本框（TextBox）是用来进行数据输入的，与 FoxPro 以前版本的@...GET 语句相比，文本框具有更大的灵活性，它可以用来向程序输入各种不同类型的数据，也可以被用来作数据的输出。

1．文本框的属性

文本框除具备控件的一些共有属性外，还具有一些自身的特有属性：

（1）Value 属性

该属性为字符型数据，用于返回或设置文本框中所显示的文本信息。文本框无 Caption 属性，它是利用 Value 属性来存放文本信息。

注意：文本框的 Text 属性也存放文本框中所显示的文本信息，但是 Text 属性是只读的。

（2）MaxLength 属性

该属性为数值型数据，用于设置文本框允许输入的最大字符数。其默认值为 0，表示对用户输入的字符数没有限制。若该属性设置为非零值，则表示用户可输入的字符数在该值范围之内，超出的字符系统不接受并发出嘟嘟声。

（3）InputMask 属性

该属性用于指定控件中数据的输入格式和显示方式。设计时和运行时都有效。表 9-6 给出了该属性的取值及说明。

表 9-6 InputMask 属性的取值

值	说　明
X	可输入任何字符
9	可输入数字和正负符号
#	可输入数字、空格和正负符号
$	在某一固定位置显示（由 SET CURRENCY 命令指定的）当前货币符号
$$	在文本框或微调控件中货币符号显示时不与数字分开
*	在值的左边显示*号
.	指定小数点的位置
,	用来分隔小数点左边的整数部分

（4）Format 属性

该属性用于指定控件中数据的输入和输出格式。设计时和运行时都有效。表 9-7 给出了该属性的取值及说明。

说明：Format 属性指定整个输入区域的特性，可以组合使用多个格式代码，它们对输入区域的所有输入都有影响；而 InputMask 属性中的每个掩码对应输入区域中的一个输入字符。

表 9-7　Format 属性的取值

值	说　明
!	将输入的字母字符转换为大写字母
$	显示货币符号
^	用科学记数法显示数值型数据
A	只允许字母字符
D	使用当前的 SET DATE 格式
K	当控件具有焦点时选择所有文本
L	显示前导零
R	显示文本框中由 InputMask 属性指定的格式掩码
T	删除输入字段的前导空格和结尾空格

（5）PassWordChar 属性

该属性用于为文本框设置一个占位符替代显示字符。如设置为星号（*）或井号（#）等。设置该属性值后，输入或显示到文本框中的字符将全部用“*”或“#”号替代显示，从而避免别人看到输入的真实内容，起到保密的作用。在实际应用中，常与 MaxLength 属性配合使用，用于设计密码输入框。

（6）ReadOnly 属性

该属性为逻辑型数据，用于决定文本框是否可以编辑修改。若设置为.T.，则文本框不可编辑修改，成为一个只读的文本框，此时仍可通过滚动条来滚动文本。系统默认值为.F.，表示文本框可以编辑修改。

（7）IMEMode 属性

该属性用于设置对输入法的操作方式，其取值有 11 种，分别为 0～10，对于中国地区，可用的是前 4 种，系统默认值为 0。其取值与对应含义见表 9-8。

表 9-8　IMEMode 属性的取值

值	说　明
0	不对输入法进行任何控制
1	打开输入法，当控件获得焦点时打开输入法窗口
2	关闭输入法，当控件获得焦点时关闭输入法窗口

设置该属性为非 0 值后，一旦文本框获得焦点，IMEMode 属性值所设定的操作将会自动被执行。利用这一特点，可实现当文本框需要输入时，自动打开输入法。

（8）SelStart、SelLength、SelText 属性

该组属性常用于对文本内容进行选定操作，如设置插入点位置、建立插入范围、选择子

串、清除文本等。

SelStart 属性设置或返回选定文本的起始位置，SelLength 属性设置或返回选定文本的长度、SelText 属性则返回选定的文本内容。

2．文本框的使用

文本框的编辑方法与标准文本编辑器相同，如自动支持剪切、复制、粘贴、撤销等操作。但是，以上操作要通过对应的快捷键来实现，其对应的快捷键分别是〈Ctrl〉+〈X〉、〈Ctrl〉+〈C〉、〈Ctrl〉+〈V〉和〈Ctrl〉+〈Z〉组合键。

【例 9-2】利用文本框输入圆的半径，然后单击“计算”按钮，得到圆的面积，如图 9-3 的左图所示。

图 9-3　计算圆的面积

设计步骤如下：

① 建立应用程序用户界面与设置对象属性。选择“新建”表单，进入表单设计器，增加两个文本框控件 Text1 和 Text2，两个标签控件 Label1 和 Label2，一个命令按钮 Command1，如图 9-3 的右图所示。并修改对象属性见表 9-9。

表 9-9　属性设置

对　象	属　性	属 性 值	说　明
Label1	Caption	请输入圆的半径：	
	AutoSize	.T. – 真	自动大小
	FontBold	.T. – 真	粗体字
	FontSize	16	字体大小
Label2	Caption	圆的面积是：	
	AutoSize	.T. – 真	自动大小
	FontBold	.T. – 真	粗体字
	FontSize	16	字体大小
Text1	InputMask	999.99	只能输入有两位小数且小于 1000 的数值
	FontSize	16	字体大小
	Value	0	可以使文本框接受数值型数据
Text2	InputMask	9999999.99	
	FontSize	16	字体大小
	ReadOnly	.T.	只读
	Value	0	可以使文本框接受数值型数据
Command1	Caption	计算（\<C）	具有块捷键
	Default	.T.	默认按钮

② 编写程序代码。

编写表单的 Activate 事件代码：

```
THIS.Text1.SetFocus
```

编写 Command1 的 Click 事件代码：

```
a = THISFORM.Text1.Value
THISFORM.Text2.Value = a * a * 3.14
THISFORM.Text1.SetFocus
```

在应用程序中，为了限制非法用户的使用，常常需要设计能够判断用户输入口令的功能。利用文本框 Text 的 PasswordChar 属性可以使输入的口令信息不在屏幕上显示出来。

【例 9-3】文本框的 PasswordChar 属性可以隐蔽用户通过键盘输入的字符。编写程序，利用文本框检查用户口令，如图 9-4 所示。

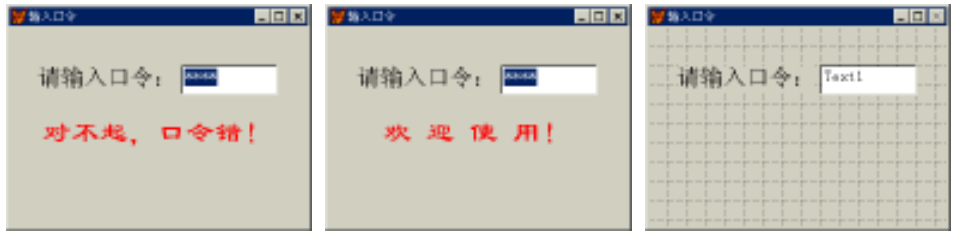


图 9-4　检查口令

设计步骤如下：

① 建立应用程序用户界面与设置对象属性。选择新建表单，进入表单设计器，增加一个文本框控件 Text1 和两个标签控件 Label1、Label2。并设置属性见表 9-10。

表 9-10　属性设置

对　象	属　性	属 性 值	说　明
Label1	Caption	请输入您的口令：	
	AutoSize	.T. – 真	自动大小
	FontBold	.T. – 真	粗体字
	FontSize	14	字体大小
Label2	Caption	圆的面积是：	
	AutoSize	.T. – 真	自动大小
	FontBold	.T. – 真	粗体字
	FontName	隶书	
	FontSize	22	字体大小
Text1	PasswordChar	*	只显示字符“*”
	Format	K	当控件具有焦点时选择所有文本
	FontSize	16	字体大小
Form1	Caption	输入口令	

② 编写程序代码。

编写表单的 Activate 事件代码：

```
THIS.Text1.SetFocus
```

编写 Text1 的 KeyPress 事件代码：

```
LPARAMETERS nKeyCode, nShiftAltCtrl
IF nKeyCode = 13
  a = LOWER（THIS.Value）
  IF a = "abcd "
    THISFORM.Label2.Caption="欢 迎 使 用！"
    W1 = THISFORM.Width
    W2 = THISFORM.Label2.Width
    THISFORM.Label2.Move（（W1 - W2）/ 2）
  ELSE
    THISFORM.Label2.Caption="对不起，口令错！"
    W1 = THISFORM.Width
    W2 = THISFORM.Label2.Width
    THISFORM.Label2.Move（（W1 - W2）/ 2）
  ENDIF
ENDIF
```

9.2.3 编辑框

在 VB 中，处理多行的文本可以利用文本框的“多行”属性，但在 VFP 中，文本框只能用来处理单行的文本，处理多行文本的工作要由“编辑框（Edit）”控件来完成。编辑框允许换行并能用方向键、翻页键以及滚动条来浏览文本。

编辑框可以用来编辑字符类型的变量、数组元素、字段或备注字段。所有标准的 Windows 编辑功能，如剪切、复制和粘贴，在编辑框中都可以使用。

1．编辑框的属性

（1）AllowTabs 属性

确定用户在编辑框中是否能插入〈Tab〉键，而不是移到下一个控件。如果允许插入〈Tab〉键，应提示用户可用〈Ctrl〉+〈Tab〉键移到下一个控件。

（2）HideSelection 属性

确定在编辑框没有获得焦点时编辑框中选定的文本是否仍然显示为选定状态。

（3）ScrollBars 属性

确定编辑框是否具有滚动条。编辑框中的文本可以自动换行，在默认的情况下，ScrollBars 属性为 2–垂直，编辑框具有一个垂直方向的滚动条，可以用方向键、翻页键以及滚动条来浏览文本。如果把 ScrollBars 属性设为 0–无，编辑框就像一个多行的文本框，没有滚动条，但是仍然可以用方向键和翻页键来浏览文本，如图 9-5 所示。

图 9-5　编辑框的垂直滚动条

在下面的例子中，给出一个简单的文本编辑器。将在以后的章节中逐渐增加其功能，逐渐完善之。

【例 9-4】设计一个文本文件的编辑器，可以新建或打开文件，并能在编辑后保存该文件，如图 9-6 所示。

图 9-6　打开文件与修改文件

设计步骤如下：

① 建立应用程序用户界面与设置对象属性。选择新建表单，进入表单设计器，增加一个编辑框控件 Edit1 和四个命令按钮 Command1～Command4，如图 9-7 所示。并设置对象属性见表 9-11。

表 9-11　属性设置

对　象	属　性	属 性 值	说　明
Command1	Caption	\<N 新建	按钮的标题
Command2	Caption	\<O 打开	按钮的标题
Command3	Caption	\<S 保存	按钮的标题
Command4	Caption	\<A 另存为	按钮的标题

② 编写程序代码。

编写表单的 Activate 事件代码：

```
WITH THIS.Edit1
    .Top = 0
    .Left = 0
    .Width = THIS.Width
ENDWITH
SET EXACT ON
THIS.Caption = "未命名"
THIS.Edit1.SetFocus
```

图 9-7　设计一个文本编辑器

编写 Command1 的 Click 事件代码：

```
THISFORM.Edit1.Value = ""
THISFORM.Refresh
THISFORM.Caption = "未命名"
```

```
THISFORM.Edit1.SetFocus
THISFORM.Command2.Enabled = .T.
THISFORM.Command3.Enabled = .F.
THISFORM.Command4.Enabled = .T.
```

编写 Command2 **的** Click **事件代码：**

```
cfile = GETFILE（""）
nhandle = FOPEN（cfile）
nend = FSEEK（nhandle,0,2）
= FSEEK（nhandle,0,0）
THISFORM.Edit1.Value = FREAD（nhandle,nend）
THISFORM.Caption = cfile
= FCLOSE（nhandle）
THISFORM.Edit1.SetFocus
THISFORM.Refresh
THISFORM.Command3.Enabled = .T.
```

编写 Command3 **的** Click **事件代码：**

```
cFile = THISFORM.Caption
nhandle = FOPEN（cfile,1）
= FWRITE（nhandle,THISFORM.Edit1.Value）
= FCLOSE（nhandle）
THISFORM.Refresh
THISFORM.Edit1.SetFocus
```

编写 Command4 **的** Click **事件代码：**

```
cfile = PUTFILE（""）
nhandle = FCREATE（cfile,0）
cc =FWRITE（nhandle,THISFORM.Edit1.Value）
= FCLOSE（nhandle）
THISFORM.Edit1.SetFocus
THISFORM.Refresh
THISFORM.Command3.Enabled = .T.
```

说明：

① 事件代码中的等号（=**）作为空操作符使用。**

② 代码中的GETFILE（）、PUTFILE（）、FOPEN（）、FSEEK（）、FCLOSE（）、FREAD（）、FWRITE（）、FCREATE（）**均为** VFP **特有的低级文件操作函数。利用这些函数我们可以方便地处理文本文件。**

2．与文件操作有关的函数

上例的事件代码中用到的一些低级文件操作函数说明如下：

（1）GETFILE（）**函数**

格式为：

```
GETFILE（[〈c1〉]）
```

功能：显示“打开”对话框，如图 9-8 所示，供用户选定一个文件并返回文件名。其中〈c1〉用于指定文件的扩展名。

（2）PUTFILE（）函数

格式为：

PUTFILE（[〈c1〉]）

功能：显示“另存为”对话框，如图 9-9 所示，供用户指定一个文件名并返回文件名。其中〈c1〉用于指定文件的扩展名。

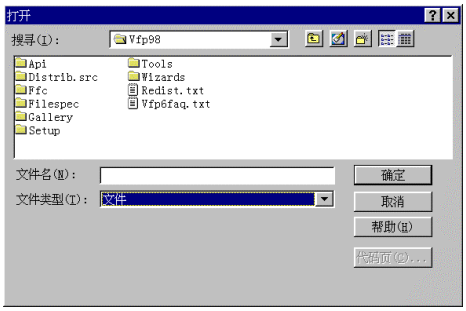


图 9-8 “打开”对话框

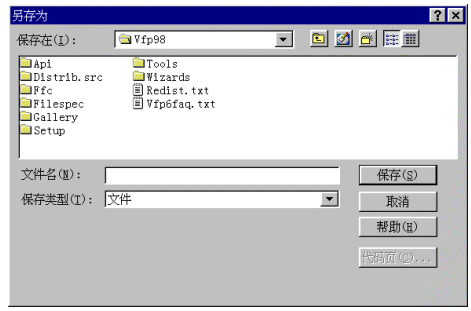


图 9-9 “另存为”对话框

（3）FOPEN（）函数

格式为：

FOPEN（〈文件名〉）

功能：打开指定文件，返回文件句柄（控制号）。

（4）FCREATE（）函数

格式为：

FCREATE（〈文件名〉）

功能：建立一个新文件，返回文件句柄（控制号）。

（5）FCLOSE（）函数

格式为：

FCLOSE（〈文件句柄〉）

功能：将文件缓冲区的内容写入文件句柄所指定的文件中，并关闭该文件。

（6）FREAD（ ）函数

格式为：

FREAD（〈文件句柄〉，〈字节数〉）

功能：从文件句柄所指定的文件中读取指定字节数的字符数据。

（7）FWRITE（ ）函数

格式为：

FWRITE（〈文件句柄〉，〈c 表达式〉）

功能：把〈c 表达式〉表示的数据写入文件句柄所指定的文件中。

（8）FSEEK（）函数

格式为：

FSEEK（〈文件句柄〉,〈移动字节数〉[,〈n〉]）

功能：在文件句柄所指定的打开的文件中移动文件指针，其中 n 表示移动的方式或方向：n = 0 为向文件首移动，n=1 为相对位置移动，n=2 为向文件尾移动。

9.3 控制类控件

在实际编程中，有时会遇到一些功能开关或功能选项要求用户作出选择，或要求用户在一个范围内对某些参数作出选择等，为此，VFP 提供了控制类控件命令按钮组、选项按钮组、复选框、列表框和组合框来实现上述功能。

命令按钮组与选项按钮组都属于容器类控件，它们分别包含一些命令按钮和选项按钮，为我们执行多种任务或提供选择。复选框也是经常成组使用，以实现多项选择。

对于可供选择的项目比较多的情况，使用选项按钮和复选框就显得力不从心了。列表框和组合框是向用户提供一系列数据的理想方式，用户既可以浏览列表框中的数据，也能从中选择一项或多项作为进一步处理的基础。组合框与列表框的区别在于：前者的数据可以被编辑，而后者无法直接编辑。

9.3.1 命令按钮组

1．命令按钮的属性、事件和方法

命令按钮（CommandButton）控件是使用最多的控件之一，常被用来执行某些代码，如开始计算、移动指针、关闭表单等。可以使用 Caption 属性指定在命令按钮上显示的文本，使用 Picture 属性指定命令按钮上显示的图片；使用 Click 事件来执行代码。除此之外，命令按钮还有下述一些有用的属性：

（1）Default 属性

默认值为.F.，取值为.T.的按钮称为“确认”按钮。一个表单内只能有一个“确认”按钮。当用户按下〈Enter〉键时，将激发活动表单中“确认”按钮的 Click 事件，而无需使用鼠标单击该按钮。

（2）Cancel 属性

默认值为.F.，取值为.T.的按钮称为“取消”按钮。一个表单内只能有一个“取消”按钮。当用户按下〈Esc〉键时，将激发活动表单中“取消”按钮的 Click 事件，而无需使用鼠标单击该按钮。

如果表单上有多个命令按钮，可以考虑使用命令按钮组（Commandgroup）。使用命令按钮组可以使代码更为简洁，界面更加整齐。命令按钮组是一个容器对象，其中包含命令按钮，即它具有如图 9-10 所示的层次性。

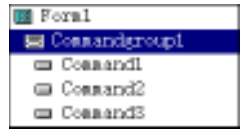


图 9-10　命令按钮组的层次性

命令按钮组中各命令按钮的用法和前面所述的单个命令

按钮的用法一样。此外，还可以将代码加入到命令按钮组的 Click 事件代码中，让组中所有命令按钮的 Click 事件使用同一个过程代码。

2．命令按钮组的属性

（1）Value 属性

指示单击了命令按钮组的哪一个按钮。

（2）ButtonCount 属性

用来设置命令按钮组中按钮的个数，默认值为 2。

3．使用命令按钮组

【例 9-5】使用命令按钮组的程序。设银行定期存款年利率为 1 年期 2.25%，2 年期 2.43%，3 年期 2.70%，5 年期 2.88%（不计复利）。今有本金 x 元，5 年以后使用，共有以下 6 种存法：

① 存一次 5 年期。

② 存一次 3 年期，一次 2 年期。

③ 存一次 3 年期，两次 1 年期。

④ 存两次 2 年期，一次 1 年期。

⑤ 存一次 2 年期，三次 1 年期。

⑥ 存五次 1 年期。

分别计算各种存法 5 年后到期时的本息合计，如图 9-11 所示。

图 9-11　使用命令按钮组

分析：设 x_1、x_2、x_3、x_5 分别表示 1 年、2 年、3 年、5 年定期储蓄的利息，a 表示本金，则定期的本息计算公式分别为：$(1 + x_1)$ a、$(1 + 2x_2)$ a、$(1 + 3x_3)$ a、$(1 + 5x_5)$ a。

设计步骤如下：

① 建立应用程序用户界面。选择新建表单，进入表单设计器，增加一个命令按钮组 Commandgroup1、一个文本框 Text1、两个标签控件 Label1 和 Label2，如图 9-12 的左图所示。

② 设置对象属性。

将命令按钮组 Commandgroup1 的 ButtonCount 属性改为 6。

命令按钮组是个容器类控件，用鼠标右键单击命令按钮组 Commandgroup1，在弹出菜单中选择“编辑”命令，“容器”Commandgroup1 的周围出现浅绿色的边界，表示开始编辑该容器。此时，可以依次选择其中的命令按钮，设置其各项属性。

各控件属性的设置可以参照图 9-12 的右图所示。

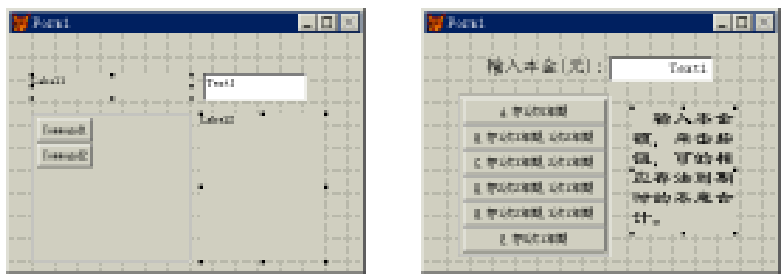


图 9-12　建立界面与设置属性

③ 编写程序代码。

编写表单的 Activate 事件代码：

```
THIS.Text1.SetFocus
```

编写命令按钮组 Commandgroup1 的 Click 事件代码：

```
a = THISFORM.Text1.Value
x1 = 0.0225
x2 = 0.0243
x3 = 0.027
x5 = 0.0288
n = THIS.Value
DO CASE
    CASE n = 1
       mes = "存 1 次 5 年期"
       y =  （1 + 5 * x5）* a
    CASE n = 2
       mes = "存 1 次 3 年期,1 次 2 年期"
       y =  （1 + 3 * x3）*  （1 + 2 * x2）* a
    CASE n = 3
       mes = "存 1 次 3 年期,2 次 1 年期"
       y =  （1 + 3 * x3）*  （1 + x1）^2 * a
    CASE n = 4
       mes = "存 2 次 2 年期,1 次 1 年期"
       y =  （1 + 2 * x2）^2 *  （1 + x1）* a
    CASE n = 5
       mes = "存 1 次 2 年期,3 次 1 年期"
       y =  （1 + 2 * x2）*  （1 + x1）^3 * a
    CASE n = 6
       mes = "存 5 次 1 年期"
       y =  （1 + x1）^5 * a
ENDCASE
mes = ALLT（STR（a））+ "元" + mes + CHR（13）+ "到期时,本息共计:" + ALLT（STR（y,12,2））
+ "元"
MESSAGEBOX（mes, 0 , "利息计算"）
```

说明：

① 由计算公式可知，各种存法的到期本息与存法中各定期的先后顺序无关。

② 可以分别为每一个命令按钮单独编写 Click 事件代码。

③ 如果为按钮组中的某个按钮的 Click 事件编写了代码，当选择这个按钮时，程序将优先执行该代码而不是命令组的 Click 事件代码。

4．按钮组生成器

使用按钮组生成器可以方便我们设计命令按钮组。在上面的例子中我们可以使用“按钮组生成器”来设置命令按钮组的各项属性。

用鼠标右键单击命令按钮组控件 CommandGroup1，在弹出菜单中选择“生成器”命令，

如图 9-13 所示，打开“按钮组生成器”。

在“命令组生成器”的“按钮”选项卡中，修改“按钮的数目”为 6，这相当于在属性窗口修改 ButtonCount 属性为 6。然后依次修改按钮的“标题”（Caption 属性），如图 9-14 的左图所示。如果要设计图文并茂的按钮，可以在“图形”栏中填入图形文件的路径与名称，或单击“图形”栏右边的“…”按钮，查找所需要的图形文件。

在“布局”选项卡中可以指定命令按钮组的排列方式，如水平或垂直、有无边框等。将“按钮间隔”微调器的值调整为 0，除去各命令按钮间的间隔，如图 9-14 的右图所示。最后按“确定”按钮退出命令组生成器。

图 9-13　打开“生成器”

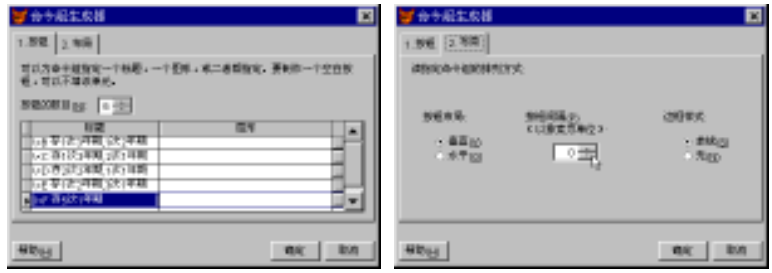


图 9-14　命令按钮组生成器

9.3.2　选项按钮组

选项按钮组也是一个容器类控件，它包含一组相互排斥的选项按钮（或称为单选按钮）。在选项按钮组中只能单击一个选项，即选项按钮组只允许用户从选择清单中选择一个选项。

当最初创建一个选项按钮组时，系统仅提供两个选项按钮。要增加更多的选项按钮可以改变按钮数（ButtonCount）属性。

1．选项按钮组的属性

（1）Value 属性

指示选项按钮组的哪一个按钮被选中。

（2）ButtonCount 属性

用来设置选项组中按钮的个数，默认值为 2。

（3）Style 属性

属于选项按钮的属性，用来设置选项按钮的样式。设计和运行时可用。取值有两个：0-标准方式（默认）和 1-图形方式。图形方式的选项按钮看起来像一个命令按钮，可以包含图形和文本。当按钮两者都包含时，文本总是在按钮底部居中对齐的。

2．使用选项按钮组

由于选项按钮组是一个容器类控件，在设计时，要用鼠标右键单击选项按钮组，并从弹出菜单中选择“编辑”命令。此时，选项按钮组的周围出现浅色边界，即可对选项按钮组内的选项按钮进行编辑了。当然，设计选项组最方便的办法是利用“生成器”。

【例 9-6】利用选项组控制【例 9-5】中存款利息的计算，如图 9-15 所示。

设计步骤如下：

① 建立应用程序用户界面。选择新建表单，进入表单设计器，增加一个选项按钮组控件

OptionGroup1、一个文本框 Text1、三个标签控件 Label1～Label3，如图 9-16 的左图所示。

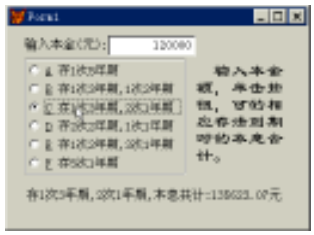


图 9-15　利用选项组计算存款利息

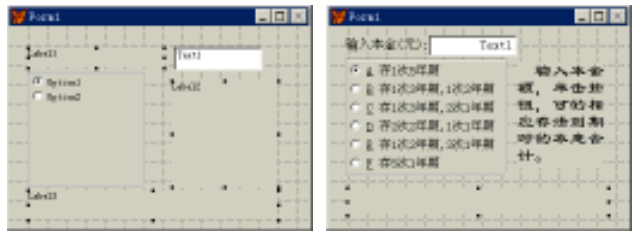


图 9-16　建立界面与设置属性

② 设置对象属性。

将选项按钮组控件 OptionGroup1 的 ButtonCount 属性改为 6。

用鼠标右键单击选项按钮组 OptionGroup1，在弹出菜单中选择“编辑”命令，选项按钮组 OptionGroup1 的周围出现浅绿色的边界，表示开始编辑该容器。此时，可以依次选择其中的选项按钮，设置其各项属性。各控件属性的设置可以参照图 9-16 的右图。

③ 编写程序代码。

编写表单的 Activate 事件代码：

```
THIS.Text1.SetFocus
```

编写选项按钮组 OptionGroup1 的 Click 事件代码：

```
a = THISFORM.Text1.Value
x1 = 0.0225
x2 = 0.0243
x3 = 0.027
x5 = 0.0288
n = THIS.Value
DO CASE
    CASE n = 1
       mes = "存 1 次 5 年期"
       y =  （1 + 5 * x5）* a
    CASE n = 2
       mes = "存 1 次 3 年期,1 次 2 年期"
       y =  （1 + 3 * x3）*  （1 + 2 * x2）* a
    CASE n = 3
       mes = "存 1 次 3 年期,2 次 1 年期"
       y =  （1 + 3 * x3）*  （1 + x1）^2 * a
    CASE n = 4
       mes = "存 2 次 2 年期,1 次 1 年期"
       y =  （1 + 2 * x2）^2 *  （1 + x1）* a
    CASE n = 5
       mes = "存 1 次 2 年期,3 次 1 年期"
       y =  （1 + 2 * x2）*  （1 + x1）^3 * a
    CASE n = 6
       mes = "存 5 次 1 年期"
```

```
        y =  （1 + x1）^5 * a
ENDCASE
mes = mes + ",本息共计:"+ALLT（STR（y,12,2））+"元"
THISFORM.Label3.Caption = mes
```

在表单中，还可以同时使用不同的选项按钮组来控制不同的选择。

【例 9-7】利用选项按钮组控制文本的对齐方式与字体，如图 9-17 所示。

设计步骤如下：

① 建立应用程序用户界面与设置对象属性。选择新建表单，进入表单设计器，增加一个文本框控件 Text1、三个标签控件 Label1～Label3 和两个选项按钮组 OptionGroup1、OptionGroup2。

用鼠标右键单击选项组控件 OptionGroup1，在弹出的快捷菜单中选择“生成器”命令，如图 9-18 所示。

图 9-17　使用多个选项按钮组

图 9-18　选择生成器

在“选项组生成器”的“按钮”选项卡中，修改“按钮的数目”为 3，这相当于在属性窗口中修改 ButtonCount 属性为 3。分别修改三个按钮的标题（Caption 属性）为左对齐、居中、右对齐，如图 9-19 的左图所示。

图 9-19　在生成器中设计选项按钮组

在“选项组生成器”的“布局”选项卡中，修改“按钮布局”为水平，并适当设置按钮间隔。然后单击“确定”按钮退出“选项组生成器”。

各控件属性的设置可以参照图 9-17。

② 编写程序代码。

编写表单的 Activate 事件代码：

```
THIS.Text1.SetFocus
```

编写 OptionGroup1 **的** Click **事件代码：**

```
n = THIS.Value
DO CASE
    CASE n = 1
       THISFORM.Text1.Alignment = 0
    CASE n = 2
       THISFORM.Text1.Alignment = 2
    CASE n = 3
       THISFORM.Text1.Alignment = 1
ENDCASE
```

编写 OptionGroup2 **的** Click **事件代码：**

```
n = THIS.Value
DO CASE
    CASE n = 1
       THISFORM.Text1.FontName = "宋体"
    CASE n = 2
       THISFORM.Text1.FontName = "隶书"
    CASE n = 3
       THISFORM.Text1.FontName = "黑体"
    CASE n = 4
       THISFORM.Text1.FontName = "楷体_GB2312"
ENDCASE
```

3．选项组的图形方式

可以将选项组设计成图形按钮的形式。

【例 9-8**】修改上例中的选项组成图形按钮的形式，如图** 9-20 **所示。**

图 9-20　**图形按钮的选项组**

设计步骤同上例，只介绍选项组的修改方法。

与修改命令按钮组类似，可以在“选项组生成器”中对各个选项按钮进行修改。下面我们通过属性窗口对选项按钮进行修改。

用鼠标右键单击选项组 OptionGroup1**，在弹出菜单中选择“编辑”命令，**OptionGroup1 **的四周出现浅色边界，开始对选项组（容器）中的按钮进行编辑。**

依次选中三个按钮 Option1~Option3**，将其标题（**Cpation**）属性改为空，自动大小（**AutoSize**）属性改为**.F. – **假，图片（**Picture**）属性通过浏览按钮“**…**”进行查找，并分别改为：**

\program files\microsoft visual studio\common\graphics\bitmaps\tlbr_w95\lft.bmp
\program files\microsoft visual studio\common\graphics\bitmaps\tlbr_w95\ctr.bmp
\program files\microsoft visual studio\common\graphics\bitmaps\tlbr_w95\rt.bmp

最后适当调整按钮的大小与相互位置。与之相仿可以将选项组 OptionGroup1 改为图形方式，如图 9-21 所示。

代码部分与上例完全相同。

图 9-21 修改按钮布局

9.3.3 复选框

复选框（CheckBox）控件相当于一个开关，用于表明某个特定状态是选定（ON）还是未选定（OFF）状态。可同时选中多个复选框。

1．复选框的属性

（1）Alignment 属性

该属性用于设置标题文字的对齐方式，默认值为 0，表示控件在左，标题文字在右；属性值取 1，表示控件在右，标题文字在左。

（2）Value 属性

该属性返回或设置复选框控件的状态。默认值为.F.或 0，表示复选框没有选中；属性值取.T.或 1 表示已经选中；属性值取 2 或.NULL.时复选框变灰（变暗）。该属性在设计和运行时均可设置，在程序中，通过访问该属性可获得复选框的状态。

（3）Style 属性

该属性用于决定复选框的风格，其取值有 0 和 1 两种。默认值为 0–标准，表示标准复选框；属性值取 1–图形，表示图形化的复选框，类似于图形化命令按钮。此时可在复选框中装入示意位图或图形，其外观与标准复选框有较大的区别，更像是一个状态命令钮。

2．使用复选框

选项按钮组属于多项中选择一项的选择，若需要选择多项的情况，可以采用多个复选框控件。

当复选框被选定时，复选框中出现一个“√”。复选框的 Caption 属性可以指定出现在复选框旁边的文本，而 Picture 属性用来指定当复选框被设计成图形按钮时的图像。

【例 9-9】利用复选框来控制输入或输出文本的字体风格，如图 9-22 的左图所示。

图 9-22 控制字体风格

设计步骤如下：

① 建立应用程序用户界面与设置对象属性。选择新建表单，进入表单设计器，增加一个

形状控件 Shape1、一个文本框控件 Text1、一个标签控件 Label1 以及三个复选框控件 Check1、Check2 和 Check3。

② 设置对象属性。设置对象属性见表 9-12 及图 9-22 的右图所示。

表 9-12　属性设置

对　象	属　性	属 性 值	说　明
Label1	Caption	请输入文本内容：	标签的内容
	AutoSize	.T. – 真	自动适应内容的大小
	FontName	隶书	字体名称
	FontSize	16	字体的大小
Text1	FontSize	18	字体的大小
Check1	Caption	粗体	标题的内容
	AutoSize	.T. – 真	自动适应标题内容的大小
Check2	Caption	斜体	标题的内容
	AutoSize	.T. – 真	自动适应标题内容的大小
Check3	Caption	下划线	标题的内容
	AutoSize	.T. – 真	自动适应标题内容的大小

③ 编写事件代码。

编写表单的 Activate 事件代码：

```
THIS.Text1.SetFocus
```

编写 Check1 的 Click 事件代码：

```
THISFORM.Text1.FontBold = THIS.Value
```

编写 Check2 的 Click 事件代码：

```
THISFORM.Text1.FontItalic = THIS.Value
```

编写 Check3 的 Click 事件代码：

```
THISFORM.Text1.FontUnderLine = THIS.Value
```

说明：可以分别选择粗体、斜体和下划线修饰，也可以选择其中的两项或三项。

9.3.4　列表框

列表框（ListBox）控件显示一个项目列表，用户可以从中选择一项或多项，但不能直接编辑列表框中的数据。当列表框不能同时显示所有项目时，它将自动添加滚动条，使用户可以上下或左右滚动列表框，以查阅所有选项。

1．列表框的属性

（1）List 属性

用来设置或返回列表中的选项，使用 List 属性可以得到列表中的任何选项。例如要显示列表框 List1 中第三项的值，用 List1.List（2）即可达到目的。

（2）Value 属性

用来得到列表中当前选项的值。

（3）ListCount 属性

用来得到列表框中的选项个数。

（4）ListIndex 属性

返回当前选项的索引号，如果没有选项被选中，该属性为 0。

（5）Selected 属性

可以在程序运行使用代码设置或返回选定列表中的选项，例如下面代码使得列表框 List1 中的第三条选项被选中：

```
THISFORM.List1.Selected（3）= .T.
```

（6）ColumnCount 属性

设置或返回列表框的列数。

（7）ControlSource 属性

设置用户从列表中选择的值保存在何处。

（8）MoverBars 属性

确定是否在列表项左侧显示移动按钮栏，这样有助于用户更方便地重新安排列表中各项的顺序。MoverBars 属性设置为真（.T.）时，允许用户拖动列表中数据项左边的按钮到新的位置来重新排序数据项。

（9）Multiselect 属性

确定用户能否从列表中一次选择一个以上的项。若取值为.T.–真，则可作多重选择。

（10）RowSource 属性

确定列表中显示的值的来源。

（11）RowSourceType 属性

确定 RowSource 是下列哪种类型：0–无、1–值、2–别名、3–SQL 语句、4–查询、5–数组、6–字段或 7–文件列表。

（12）Sorted 属性

取值为真（.T.）时，将按照字典顺序显示列表项。不过，仅在列表的 RowSourceType 属性设置成 0（无）或 1（值）时，Sorted 属性才有效。

2．列表框的方法

列表框支持的方法主要有 AddItem、RemoveItem、Clear 和 Requery。

前 3 种方法常用来在运行期间向列表框增加或删除列表项内容。

（1）AddItem 方法

该方法用于在运行时向列表框增加一列表项，语法为：

```
〈对象名〉.AddItem（〈要增加的列表项〉[,〈列表项序号〉] [,〈列序号〉]）
```

其中，对象可以是列表框或组合框控件。“ 列表项序号 ”用于指定新插入的项在列表框中的位置。若省略该参数，则新增加的列表项放在列表框的末尾。“ 列序号 ”用于指定新插入的项在列表框中的第几列，默认值为 1。

（2）RemoveItem 方法

该方法用于删除列表框中指定的列表项，其语法为：

〈对象名〉.RemoveItem（〈列表项序号〉）

其中，对象可以是列表框或组合框控件。“列表项序号”用于指定被移去项在控件中的显示顺序。对于列表框或组合框中的第一项，列表项序号等于 1。

（3）Clear 方法

该方法用于清除列表框中的所有列表项。执行该方法后，列表框的 Listcount 将被置为 0，其语法为：

〈对象名〉.Clear

其中，对象可以是列表框或组合框控件。

注意：只有当 RowSourceType 属性设置为 0 时，才可以使用 AddItem 方法、AddListItem 方法或 Clear 方法。

（4）Requery 方法

重新查询列表框或组合框控件所基于的行源（RowSource）。其语法格式为：

〈对象名〉.Requery

其中，对象可以是列表框或组合框控件。

使用 Requery 方法可以确保控件中包含最新的数据。Requery 方法重新查询 RowSource 属性，并且使用新的值更新列表。

3．使用列表框

【例 9-10】利用循环结构和列表框控件，设计一个“选项移动”表单。

所谓“选项移动”表单是指由两个列表框和四个命令按钮所构成的界面，在 Windows 程序中常见到此类窗口，如图 9-23 所示。

图 9-23 “选项移动”表单

设计步骤如下：

① 建立应用程序用户界面与设置对象属性。选择“新建”表单，进入表单设计器。增加一个标签 Label1、两个列表框控件 List1、List2 和一个命令按钮组 CommandGroup1。将按钮组的按钮个数属性 ButtonCount 改为 4，并设置其他对象属性如图 9-23 及表 9-13 所示。

表 9-13 属性设置

对　象	属　性	属 性 值	说　明
Label1	Caption	Shift 或 Ctrl +单击鼠标左键可选择多项	标签的内容
	WordWrap	.T. – 真	
List1	Multiselect	.T. – 真	选择多项

（续）

对　象	属　性	属 性 值	说　明
List2	Multiselect	.T. – 真	选择多项
	MoverBars	.T. – 真	可移动
CommandGroup1:			
Command1	Caption	>	标签的内容
	FontBold	.T. – 真	
Command2	Caption	>>	标签的内容
	FontBold	.T. – 真	
Command3	Caption	<	标签的内容
	FontBold	.T. – 真	
	Enabled	.F. – 假	
Command4	Caption	<<	标签的内容
	FontBold	.T. – 真	
	Enabled	.F. – 假	

② 编写事件代码。编写表单的 Init 事件代码：

```
THIS.List1.AddItem （"one"）
THIS.List1.AddItem （"two"）
THIS.List1.AddItem （"three"）
THIS.List1.AddItem （"four"）
THIS.List1.AddItem （"five"）
THIS.List1.AddItem （"six"）
THIS.List1.AddItem （"seven"）
THIS.List1.AddItem （"eight"）
THIS.List1.AddItem （"nine"）
THIS.List1.AddItem （"ten"）
```

编写命令按钮组 CommandGroup1 的 Click 事件代码：

```
DO CASE
  CASE THIS.Value = 1
    I = 0
    DO WHILE I <= THIS.Parent.List1.ListCount
      IF THIS.Parent.List1.Selected（i）
        THIS.Parent.List2.Additem （THIS.Parent.list1.List（i））
        THIS.Parent.List1.RemoveItem（i）
      ELSE
        I = I + 1
      ENDIF
    ENDDO
  CASE THIS.Value = 2
     DO WHILE THIS.Parent.List1.ListCount > 0
       THIS.Parent.List2.AddItem（THIS.Parent.List1.List（1））
       THIS.Parent.List1.RemoveItem（1）
```

```
        ENDDO
    CASE THIS.Value = 3
      I = 0
      DO WHILE I <= THIS.Parent.List2.ListCount
        IF THIS.Parent.List2.Selected（i）
            THIS.Parent.List1.Additem （THIS.Parent.List2.List（i））
            THIS.Parent.List2.RemoveItem（i）
        ELSE
          I = I + 1
        ENDIF
      ENDDO
    CASE THIS.Value = 4
        DO WHILE THIS.Parent.List2.ListCount > 0
          THIS.Parent.List1.AddItem（THIS.Parent.List2.List（1））
          THIS.Parent.List2.RemoveItem（1）
        ENDDO
ENDCASE
IF THIS.Parent.List2.ListCount > 0
    THIS.Command3.Enabled =.T.
    THIS.Command4.Enabled =.T.
ELSE
    THIS.Command3.Enabled =.F.
    THIS.Command4.Enabled =.F.
ENDIF
IF THIS.Parent.List1.ListCount = 0
    THIS.Command1.Enabled =.F.
    THIS.Command2.Enabled =.F.
ELSE
    THIS.Command1.Enabled =.T.
    THIS.Command2.Enabled =.T.
ENDIF
THISFORM.Refresh
```

说明：

① 本例演示了如何将数据项从一个列表框移到另一个列表框。用户可以选定一个或多个数据项并使用适当的命令按钮在列表之间移动数据项。

② 为了能从列表中添加和移去数据项，列表的 RowSourceType **必须设置成** 0–**无。**

③ 当列表框 List1 **中没有选项时，改变命令按钮的** Enabled **属性，使** Commad1 **和** Command2 **同时关闭，当列表框** List2 **中没有选项时，则关闭** Commad3 **和** Command4**。**

4．显示文件目录

当列表框的 RowSourceType **属性取值为** 7 **时，列表框可以显示文件目录列表。利用列表框可以设计显示文件目录的程序，并且可以在目录列表中方便地选定文件。**

【例 9-11**】显示文件目录的列表框程序，如图** 9-24 **所示。在列表框中选定文件后，用鼠标单击“ 打开选定文件 ”按钮可打开该文件进行察看或编辑。**

图 9-24　文件目录列表

设计步骤如下：

① 选择新建表单，进入表单设计器。增加一个列表框控件 List1、一个命令按钮 Command1、一个标签 Label1 和一个文本框 Text1。

② 设置 List1 和 Text1 的属性见表 9-14，其他控件的属性设置如图 9-24 所示。

表 9-14　属性设置

对　象	属　性	属 性 值
Text1	Value	*.txt
List1	RowSourceType	7 – 文件
	RowSource	*.txt

③ 编写事件代码。

编写表单的 Activate 事件代码：

```
THISFORM.List1.SetFocus
```

编写文本框 Text1 的 Valid 事件代码：

```
THISFORM.List1.RowSource = ALLTRIM（THIS.Value）
THISFORM.List1.Requery
```

编写“打开选定文件”按钮 Command1 的 Click 事件代码：

```
a = THISFORM.List1.ListIndex
MODIFY FILE （THISFORM.List1.List（2）+THISFORM.List1.List（a））
```

运行表单，在列表框中选定文件，按“打开选定文件”按钮，即可打开一个包含指定文本文件的编辑器，如图 9-25 所示。

图 9-25　打开选定文件

说明：

① a = THISFORM.List1.ListIndex 表示将 List1 中光标所在项的序号赋予变量 a。

其后的 THIS.List（a）表示在列表框 List1 中选定的项。

② 当 RowSourceType 属性设置为“7–文件”时：

List1.List（1）代表驱动器。

List1.List（2）代表路径。

List1.List（3）是一个分隔行。

List1.List（4）是 [..]。单击它，则返回到父目录。

③ 文本框 Text1 的 Valid 事件代码中调用了列表框的 Requery 方法，用于保证列表框中包含的数据都是最新的。这样，每当在文本框中改变“文件类型”后，列表框中都将列出相应的文件目录。

④ 命令按钮 Command1 的 Click 事件代码中的 MODIFY FILE 命令用于打开编辑窗口，使用户可以编辑或修改选定的文件项。

5．在列表框中显示多列

修改列表框的 ColumnCount 属性、ColumnWidths 属性可以在列表框中显示多列选项。

【例 9-12】简易数学用表。显示整数 1～100 的平方、平方根、自然对数和 e 指数，如图 9-26 所示。

图 9-26 简易数学用表

设计步骤如下：

① 设计程序界面与设置对象属性。选择新建表单，进入表单设计器。增加一个列表框控件 List1、一个命令按钮 Command1 和四个标签 Label1～Label4，如图 9-24 所示。

设置 List1 的属性见表 9-15，其他控件的属性设置如图 9-26 所示。

表 9-15 属性设置

对 象	属 性	属 性 值	说 明
List1	ColumnCount	5	列对象的数目
	ColumnLines	.F. – 假	列间的分割线
	ColumnWidths	35,45,55,65,110	各列的宽度

② 编写 Command1 的 Click 事件代码：

```
FOR n = 1 TO 100
  s = ALLT（STR（n））
  THISFORM.List1.AddlistItem（s,n,1）
  s = ALLT（STR（n^2））
  THISFORM.List1.AddlistItem（s,n,2）
  s = ALLT（STR（Sqrt（n）,10,3））
  THISFORM.List1.AddlistItem（s,n,3）
  s = ALLT（STR（LOG（n）,10,4））
  THISFORM.List1.AddlistItem（s,n,4）
  s = ALLT（STR（EXP（n）,14,4））
  THISFORM.List1.AddlistItem（s,n,5）
ENDFOR
```

说明：添加列表项方法 AddlistItem（s,n,1）的第三个参数表示列表项添加到列表中的列数。

9.3.5 组合框

有两种形式的组合框，即下拉组合框和下拉列表框，通过更改控件的 Style 属性可选择所需要的形式。

下拉列表框（即 Style 属性为 2 的组合框控件）和列表框一样，为用户提供了包含一些选项和信息的可滚动列表。在列表框中，任何时候都能看到多个项；而在下拉列表中，只能看到一个项，用户可单击向下按钮来显示可滚动的下拉列表框。

下拉组合框（即 Style 属性默认为 0 的组合框控件）则兼有列表框和文本框的功能。用户可以单击下拉组合框上的按钮查看选择项的列表，也可以直接在按钮旁边的框中直接输入一个新项。

1．组合框的属性

组合框的属性与列表框基本相同，另外增加了一些与文本框相关的属性，表 9-16 列出的组合框属性常在设计时使用。

表 9-16　组合框的常用属性

属　性	说　明
ControlSource	指定用于保存用户选择或输入值的表字段
InputMask	对于下拉组合框，指定允许键入的数值类型
IncrementalSearch	指定在用户键入每一个字母时，控件是否和列表中的项匹配
RowSource	指定组合框中项的来源
RowSourceType	指定组合框中数据源类型
Style	指定组合框是下拉组合框还是下拉列表

2．组合框的事件与方法

组合框能响应的事件和所支持的方法与列表框和文本框的事件、方法相同，用法也相似。

3．下拉列表框

如果想节省表单上的空间，并且希望强调当前选定的项，可以使用下拉列表框。

【例 9-13】在文本框输入数据，按回车添加到列表框中，在列表框中选定项目，按回车后可以移去选定项，如图 9-27 所示。

图 9-27　添加或移去文本

设计步骤如下：

① 选择新建表单，进入表单设计器，增加一个文本框 Text1、一个组合框 Combo1 以及两个标签 Label1、Label2。设置 Combo1 的 Style 属性为：2 – 下拉列表框，其他控件的属性

设置如图 9-27 所示。

② 编写代码。

编写表单的 Activate 事件代码：

```
PUBLIC a
a = 1
THIS.Text1.SetFocus
```

编写 Text1 的事件代码：

KeyPress 事件:

```
LPARAMETERS nKeyCode, nShIFtAltCtrl
IF nKeyCode = 13
  IF !EMPTY（THIS.Value）
    THISFORM.Combo1.AddItem （THIS.Value）
    THISFORM.Combo1.DisplayValue = THIS.Value
  ENDIF
  THIS.SelStart = 0
  THIS.SelLength = LEN（RTRIM（THIS.Text））
  a = 0                                    && 与 Valid 事件结合
ENDIF
```

Valid 事件:

```
IF a = 1
  RETURN .T.
ELSE                                       && 按回车键不离开文本框
  a = 1
  RETURN 0
ENDIF
```

编写 Combo1 的 RightClick 事件代码：

```
IF THIS.ListIndex > 0
  THISFORM.Text1.Value = THIS.List（THIS.ListIndex）
  THIS.RemoveItem （THIS.ListIndex）
  THIS.Value = 1
ENDIF
```

说明：

① 在文本框中输入数据后按〈Enter〉键，即可将数据添加到下拉列表框中。在下拉列表框中选中数据，然后用鼠标右键单击所选项，即可将数据移回文本框。

② Text1 的事件代码中 THISFORM.Combo1.AddItem（THIS.Value）表示将文本框 Text1 中的内容添加进组合框 Combo1 中。

③ Combo1 的事件代码中 THIS.RemoveItem（THIS.ListIndex）表示将组合框中的选项移走。

④ 在文本框中按〈Enter〉键时，全局变量 a = 0，Valid 事件返回 0，焦点不移出文本框，在 Valid 事件又令全局变量 a = 1，以便按〈Tab〉键时光标可以移出。

4．下拉组合框

下拉组合框看起来像是在标准的文本框右边加了个下拉箭头，用鼠标单击该箭头就在文本框下打开一个列表。用户从中选择一个选项，该选项就会进入文本框。

下拉组合框能实现上述表单中的文本框和下拉列表框的组合功能。即允许用户既可以输入数据又可以从列表中选择数据。

【例 9-14】在上例中使用下拉组合框来代替文本框和列表框，实现同样的功能：输入数据，按〈Enter〉键后可添加到列表中，在列表中选定项目，单击鼠标右键可移去选定项，如图 9-28 所示。

图 9-28　下拉组合框

设计步骤如下：

① 选择新建表单，进入表单设计器，增加一个组合框 Combo1、两个标签 Label1、Label2 和一个文本框 Text1。只需修改两个标签的属性，如图 9-28 所示，其他控件属性无需改动。

② 编写代码。

编写 Combo1 的事件代码：

KeyPress 事件：

```
LPARAMETERS nKeyCode, nShiftAltCtrl
IF nKeyCode = 13
  IF !EMPTY（THIS. DisplayValue）
    THIS.AddItem （THIS.DisplayValue）
    THISFORM.Text1.Value = THIS.ListCount
  ENDIF
  THIS.SelStart = 0
  THIS.SelLength = LEN（ALLT（THIS.Text））
  THIS.Tag = "N"
ENDIF
```

RightClick 事件:

```
IF THIS.ListCount > 0
  THIS.RemoveItem （THIS.ListIndex）
  THIS.Value = 1
  THISFORM.Text1.Value = THIS.ListCount
```

```
    ENDIF
```

Valid 事件:

```
IF THIS.Tag = "Y"
   RETURN .T.
ELSE
   THIS.Tag = "Y"
   RETURN 0
ENDIF
```

说明：

① 在组合框中输入数据后按回车，即可将数据添加到下拉列表框中，下边的文本框开始计数。在下拉列表框中选中数据，然后用鼠标右键单击所选项，即可将数据移出组合框，同时计数减一。

② 控件的 Tag **属性用来存放程序所需的字符型数据,这里用来代替上例中的全局变量** a**。**

9.3.6 微调器

微调器（Spinner**）控件可以在一定范围内控制数据的变化。除了能够用鼠标单击控件右边向上和向下的箭头来增加和减少数字以外，还能像编辑框那样直接输入数值数据。**

1．微调器的属性

（1） KeyboardHighValue **和** KeyboardLowValue **属性**

用来控制用户通过键盘输入的值。

（2） SpinnerHighValue **和** SpinnerLowValue **属性**

用来控制用户通过鼠标单击箭头获得的值。

（3） Interval **属性**

用来设定数值增加或减少的量，要颠倒箭头的功能（向上箭头减少，向下箭头增加）可以把 Interval **设为负数。**

2．微调器的事件

微调器的 InteractiveChange **事件当微调器的值发生改变时发生。程序运行时，无论是用鼠标单击箭头或是用键盘改变数值，都将发生** InteractiveChange **事件。**

说明：微调器控件的值一般为数值型，也可以使用微调器控件和文本框来微调多种类型的数值。例如，如果想让用户微调一定范围的日期，可以调整微调器控件的大小，使它只显示按钮，同时在微调器按钮旁边放置一个文本框，设置文本框的 Value **属性为日期，在微调器控件的** UpClick **和** DownClick **事件中增加或减少日期。**

【例 9-15**】使用微调器控制色彩，还可以返回色彩的** RGB **值，如图** 9-29 **的左图所示。**

设计步骤如下：

① 建立应用程序用户界面与设置对象属性。选择新建表单，进入表单设计器，依次增加两个文本框 Text1**、**Text2**、三个微调器** Spinner1**～**Spinner3**、一个标签控件** Label1 **和一个命令按钮** Command1**。并修改各对象属性见表** 9-17 **和图** 9-29 **的右图所示。**

图 9-29　调色盘

表 9-17　属性设置

对　象	属　性	属　性　值	说　明
Label1	Caption	用鼠标单击色彩微调器，可以改变颜色，还可在文本框中得到相应的 RGB 值	标签的内容
	WordWrap	.T. – 真	折行
	FontName	隶书	字体名称
	FontSize	14	字体的大小
Spanner1	BackColor	255,0,0	背景色改为红色
	KeyboardHighValue	255	
	KeyboardLowValue	0	
	Spinnerhigh	255.00	
	Spinnerlow	0.00	
	Value	255	
	Increment	16	改变量
Spanner2	BackColor	0,255,0	背景色改为绿色
	其余属性同 Spanner1		
Spanner3	BackColor	0,0,255	背景色改为蓝色
	其余属性同 Spanner1		
Command1	Caption	关闭	

② 编写程序代码。

编写 Spanner1 的 InteractiveChange 事件代码：

```
r = THISFORM.Spinner1.Value
g = THISFORM.Spinner2.Value
b = THISFORM.Spinner3.Value
THISFORM.Text2.BackColor = RGB（r,g,b）
THISFORM.Text1.Value =;
"Color = RGB（"+Allt（Str（r）） +","+Allt（Str（g）） +","+Allt（Str（b）） +"）"
```

编写 Spanner2 的 InteractiveChange 事件代码：

```
r = THISFORM.Spinner1.Value
g = THISFORM.Spinner2.Value
b = THISFORM.Spinner3.Value
THISFORM.Text2.BackColor = RGB（r,g,b）
```

```
THISFORM.Text1.Value =;
"Color = RGB（"+Allt（Str（r）） +","+Allt（Str（g）） +","+Allt（Str（b）） +"）"
```

编写 Spanner3 **的** InteractiveChange **事件代码：**

```
r = THISFORM.Spinner1.Value
g = THISFORM.Spinner2.Value
b = THISFORM.Spinner3.Value
THISFORM.Text2.BackColor = RGB（r,g,b）
THISFORM.Text1.Value =;
"Color = RGB（"+Allt（Str（r）） +","+Allt（Str（g）） +","+Allt（Str（b）） +"）"
```

编写 Command1 **的** Click **事件代码：**

```
THISFORM.Release
```

说明：

① RGB **函数的三个参数依次代表红、绿、蓝的色彩浓度，其变化范围是** 0～255。

② **三个** Spinner **控件的事件代码相同是为了编写代码方便，可以有所区别，例如** Spinner1 **的** InteractiveChange **事件代码可以改为下面代码，其余类推。**

```
r = THIS.Value
g = THISFORM.Spinner2.Value
b = THISFORM.Spinner3.Value
THISFORM.Text2.BackColor = RGB（r,g,b）
THISFORM.Text1.Value =;
"Color = RGB（"+Allt（Str（r）） +","+Allt（Str（g）） +","+Allt（Str（b）） +"）"
```

9.3.7 计时器控件

计时器是 VFP **提供的一个用于定时的特殊控件，当到达预定时间时，系统会自动触发其** Timer **事件，以便完成指定的操作。计时器会自动开始新一轮计时，因此，利用计时器可完成一些周期性的工作，如定时检测系统或控件的状态，控制控件的移动，设计时钟、倒计数器、秒表等。计时器控件可以在应用中以重复的时间间隔产生一个事件。这对不需要与用户交互的代码的执行非常有用。**

计时器控件在设计时显示为一个小时钟图标，而在运行时则不可见，常用来做一些后台处理。

1．计时器的属性与事件

计时器的属性和事件都相当少，其能响应的事件只有自身特有的一个 Timer **事件，常用的属性有** 3 **个，即** Name、Enabled **和** Interval。

（1**）**Enabled **属性**

该属性用于决定计时器是否生效，其取值为.T.**或**.F.**，系统默认值为**.T.。**可在属性窗口中设置，也可通过程序代码设置。**

（2**）**Interval **属性**

该属性用于设置定时器触发 Timer **事件的时间间隔，即决定每隔多少时间产生一次** Timer **事件。其单位为毫秒（**1/1000 **秒），取值范围为** 1～65535**，因此，最大可设置的时间间隔为** 1

分钟多一点。若将该属性设置为 1000，则运行时计时器每隔一秒触发一个 Timer 事件，通过编写代码，可以很轻松地完成有关时钟的设计。

2．计时器的使用

利用 VFP 的计时器控件，可以很方便的设计一个电子表。

【例 9-16】在表单上设计一个数字时钟，如图 9-30 所示。

图 9-30　数字时钟

设计步骤如下：

① 建立应用程序用户界面与设置属性。选择“新建”表单，进入表单设计器，增加一个选项按钮组 OptionGroup1、一个文本框 Text1、一个标签 Label1 和一个“计时器”控件 Timer1，并设置属性见表 9-18。由于计时器控件在运行时不可见，所以可以放在表单的任何位置。

表 9-18　属性设置

对　象	属　性	属 性 值	说　明
Label1	Caption	上午	标题的内容
	AutoSize	.T. – 真	自动适应标题内容的大小
	FontBold	.T. – 真	粗体
	FontName	黑体	字体名
Timer	Interval	1000	
Text1	Alignment	1 – 右	
	Value	00:00:00	
	FontSize	36	
	Enabled	.F. – 假	
	DisabledBackColOR	255, 255, 255	
	DisabledFOReColOR	0, 0, 0	
OptionGroup1	Value	2	选中的按钮
	AutoSize	.T. – 真	自动适应按钮的大小
Option1	Caption	24 小时制	
	Style	1 – 图形	风格
Option2	Caption	12 小时制	
	Style	1 – 图形	风格

② 编写程序代码。

编写表单的 Activate 事件代码：

```
SET HOURS TO 12
```

编写 OptionGroup1 **的** InteractiveChange **事件代码：**

```
IF THIS.Value=2
   SET HOURS TO 12
   THISFORM.Label1.Visible=.T.
ELSE
   SET HOURS TO 24
   THISFORM.Label1.Visible=.F.
ENDIF
```

编写 Timer1 **的** Timer **事件代码：**

```
IF HOUR（DATETIME（））>=12
   THIS.Parent.Label1.Caption='下午'
ELSE
   THIS.Parent.Label1.Caption='上午'
ENDIF
THIS.Parent.Text1.Value=SUBSTR（TTOC（DATETIME（））,10,8）
```

说明：

① **计时器的** Interval **属性指定两个计时器事件之间的毫秒数。这里设定为** 1000（= 1 **秒），计时器将每秒（近似等间隔）激发一次** Timer **事件。**

② **在** OptionGroup1 **的事件代码中，利用分支结构来判断、改变时间运行的格式以及标签** Label1 **的显示与否。**

③ **计时器** Timer1 **的** Timer **事件代码中：**

```
THIS.Parent.Label1
```

表示对该对象的父容器中的标签控件 Label1 **的调用，这是一种对象的相对引用格式。也可以改为：**

```
THISFORM.Label1
```

④ DATETIME（）**是日期时间函数，返回系统当前的日期与时间。**

⑤ **函数** HOUR（**日期时间表达式**）**返回日期时间表达式中的小时数。**

⑥ **函数** TTOC（**日期时间表达式**）**是类型转换函数，将日期时间表达式转换成：**

```
YY:MM:DD HH:MM:SS
```

格式的字符串。

⑦ SET HOURS TO **为设置时间格式命令。**SET HOURS TO 12 **将** DATETIME（）**函数的时间格式改为** 12 **小时制，**TIME（）**函数不受此设置的影响。**

9.4 容器类控件

容器类控件包括前面已经介绍过的命令按钮组、选项按钮组，以及表格控件、页框控件

和容器控件。

9.4.1 容器控件

容器（Container）控件具有封装性，像表单一样，可以在容器控件中加上一些其他控件。这些控件随容器移动而移动，其 Top 和 Left 属性均相对于容器而言，与表单无关。

1．容器控件的属性

除了一些公用的属性之外，容器控件的属性主要是控制其外形的 SpesialEffect 属性，其取值有：0–凸起，1–凹下和 2–平面（默认）。

2．使用容器控件

由于容器（Container）控件的封装性，而且外形更具有立体感，使得我们更加喜欢使用容器控件来对程序界面进行修饰。

【例 9-17】设计一个华氏温度和摄氏温度互相转换的程序，如图 9-31 所示。输入一个华氏温度可以得到相应的摄氏温度，而输入一个摄氏温度则可以得到其相应的华氏温度。

分析：摄氏温度与华氏温度的关系为华氏=摄氏×9/5+32，由此可得摄氏=（华氏–32）×5/9。

设计步骤如下：

① 建立应用程序用户界面与设置对象属性。选择“新建”表单，进入表单设计器，增加两个命令按钮 Command1、Command2 和两个“容器”控件 Container1、Container2，并把容器控件的 SpesialEffect 属性改为 0 – 凸起。

为了将标签和文本框包容在容器中，用鼠标右键单击容器控件 Container1，在弹出的快捷菜单中选择“编辑”命令，如图 9-32 的左图所示。

此时，Container1 的周围出现浅绿色的边界，表示可以编辑该容器了。在 Container1 中增加一个标签控件 Label1 和一个文本框控件 Text1，如图 9-32 的右图所示。

图 9-31 使用“容器”控件

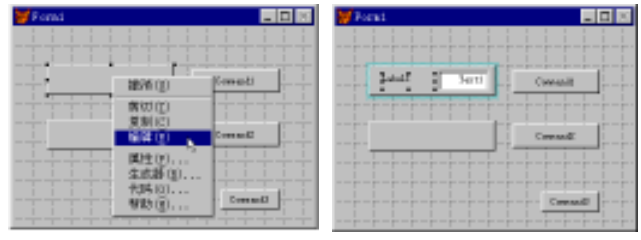


图 9-32 编辑容器控件

修改 Label1 和 Text1 的属性见表 9-19。

表 9-19 属性设置

对 象	属 性	属 性 值	说 明
Label1	Caption	摄氏温度	
	FontSize	12	字体大小
Text1	FontSize	12	字体大小

然后用同样的方法，设计 Container2 中的标签与文本。

设置其他对象属性见表 9-20。

表 9-20　属性设置

对　象	属　性	属 性 值	说　明
Command1	Caption	\<C 摄氏转华氏	按钮的标题
Command2	Caption	\<F 华氏转摄氏	按钮的标题
Command3	Caption	关闭（\<Q）	按钮的标题
	Cancel	.T. – 真	设为表单的取消按钮

② 编写程序代码。

编写表单的 Activate 事件代码：

```
THIS.Container1.Text1.SetFocus
```

编写 Command1 的 Click 事件代码：

```
c = THISFORM.Container1.Text1.Value
THISFORM.Container2.Text1.Value = c *  （ 9 / 5 ）+ 32
```

编写 Command2 的 Click 事件代码：

```
f = THISFORM.Container2.Text1.Value
THISFORM.Container1.Text1.Value =  （f - 32）*  （5 / 9）
```

编写文本框 Text1 的事件代码：

GotFocus 事件代码：

```
THIS.SelStart = 0
THIS.SelLength = LEN（THIS.Text）
```

InteractiveChange 事件代码：

```
THISFORM.Container2.Text1.Value = ""
```

编写文本框 Text2 的事件代码：

GotFocus 事件代码：

```
THIS.SelStart = 0
THIS.SelLength = LEN（THIS.Text）
```

InteractiveChange 事件代码：

```
THISFORM.Container1.Text1.Value = ""
```

说明：

① 不仅是容器控件 Container，容器类的控件一般都应按照上面的步骤进行设计，即用鼠标右键单击控件，在弹出菜单中选择“编辑”命令后，进入控件中进行设计。当然也可以在“属性”窗口中直接选择相应的控件，进行设计。

② 使用“表单向导”设计的“新奇式”表单，其中的各种部件都具有上述形式。

【例 9-18】设计一个电子游动标题板，标题“使用 VFP 设计动画”在表单的黄色区域（容器中）自右至左地反复移动。单击“暂停”按钮，标题停止移动，按钮变成“继续”。单击“继续”按钮，标题继续移动，按钮又变回“暂停”，如图 9-33 所示。

设计步骤如下：

① 建立应用程序用户界面。选择“新建”表单，进入表单设计器，增加一个命令按钮 Command1 和一个“容器”控件 Container1。用鼠标右键单击容器控件，在弹出的快捷菜单中选择“编辑”命令，开始对“容器”进行设计。在容器中增加一个标签 Label1 和一个“计时器”控件 Timer1，如图 9-34 所示。

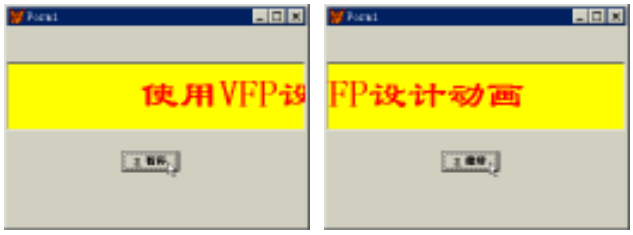


图 9-33　电子标题板

图 9-34　设计界面与设置属性

其中计时器控件 Timer1 可以放在容器中的任何位置。

② 设置对象属性见表 9-21。设置属性后的表单如图 9-34 所示。

表 9-21　属性设置

对　象	属　性	属 性 值	说　明
Command1	Caption	\<S 开始	
Container1	BackColOR	255, 255, 0	容器的背景颜色为黄色
	SpecialEffect	1 – 凹下	效果
Timer	Interval	100	
	Enabled	.F. – 假	可用性
Label1	Caption	使用 VFP 设计动画	标题的内容
	AutoSize	.T. – 真	自动适应标题内容的大小
	FontBold	.T. – 真	粗体
	FontName	隶书	字体名
	BackStyle	0 - 透明	标签的背景类型
	FOReColOR	255, 0, 0	标签的字体颜色为红色

③ 编写程序代码。

编写 Command1（开始/暂停）的 Click 事件代码：

```
IF THIS.Caption = "\<S 暂停"
   THIS.Caption = "\<S 继续"
   THISFORM.Container1.Timer1.Enabled = .F.
ELSE
   THIS.Caption = "\<S 暂停"
```

```
    THISFORM.Container1.Timer1.Enabled = .T.
ENDIF
```

编写 Timer1 **的** Timer **事件代码：**

```
IF THIS.Parent.Label1.Left + THIS.Parent.Label1.Width > 0
    THIS.Parent.Label1.Left = THIS.Parent.Label1.Left - 3
ELSE
    THIS.Parent.Label1.Left = THIS.Parent.Width
ENDIF
```

说明：

① THIS.Parent.Label1 **属于相对引用，指计时器对象的父对象容器** Container1 **中的标签对象** Label1**。**

② IF THIS.Parent.Label1.Left + THIS.Parent.Label1.Width > 0 **是判断对象** Label1 **的左上角** Left **位置加上其宽度是否大于零。若大于零，则重新定义其左上角的位置：**THIS.Parent.Label1.Left = THIS.Parent.Label1.Left - 3**（即向左移动** 3**）。否则（等于零），即整个** Label1 **已经移出容器的左端，定义** Label1 **左上角的位置为容器的最右端（重新出现）：**THIS.Parent.Label1.Left = THIS.Parent.Width**。**

9.4.2 页框

为了扩展应用程序的用户界面，常常使用带页框（PageFrame**）的表单。页框是一个可包含多个页面的容器控件，其中的页面又可包含各种控件。当有多个数据屏幕需要显示时页框很有用处，它使用户可以往前或往后“翻页”而开发者无须编写另外的程序。**

使用页框和页面，可以创建带选项卡的表单或对话框，就像“项目管理器”中见到的一样。

1．页框控件的属性

（1**）** Tabs **属性**

确定页面的选项卡是否可见。

（2**）** TabStyle **属性**

确定是否选项卡都是相同的大小，并且都与页框的宽度相同。

（3**）** PageCount **属性**

设置或返回页框的页面数。页框刚被创建时，只有两个“页面”（Page**），**PageCount **属性用来设置页面数。**

（4**）** ActivePage **属性**

设置或返回页框中活动页面的页码。可以从程序中使用 ActivePage **属性来激活一个页面。例如，下列代码，将表单中页框** PageFrame1 **的活动页面改为第三页面。**

```
THISFORM.PageFrame1.ActivePage = 3
```

（5**）** TabStretch**属性**

取值为“1 - **单行”（默认），只显示能放入选项卡中的标题字符；取值为“**0 - **多重行”，选项卡将层叠起来，以便所有选项卡中的整个标题都能显示出来。**

2．页框控件的方法

除了一些公共的方法之外，页框控件的常用方法是 Zorder 方法。其作用是将一个指定的页面放在前面或后面。其语法格式为：

[〈页面对象名〉.]ZOrder（[nOrder]）

其中参数 nOrder 为一个整数，表明一个对象相对于其他对象的位置。默认值为 0，表示对象位于前面；若取值为 1，则对象位于后面。

3．使用页框控件

和使用其他容器控件一样，向正在设计的页面中添加控件之前，必须先选择页框，并从用鼠标右键弹出的快捷菜单中选择“编辑”命令，或在“属性”窗口的“对象”下拉列表中选择该容器。这样，才能激活这个容器（具有宽边）。

在添加控件前，如果没有将页框作为容器激活，控件将添加到表单中而不是页面中，即使看上去好象是在页面中。

【例 9-19】带选项卡的表单。使用页框和页面，可以创建带选项卡的表单或对话框，和我们在“选项”中见到的一样。在表单中设计一个带选项卡的页框，其中有三个页面，分别放上一些不同的控件。

设计步骤如下：

① 选择新建表单，进入表单设计器，首先增加一个页框控件 PageFrame1，并修改其 PageCount 属性为 3，页框上出现三个页面。

用鼠标右键单击页框控件，在弹出的快捷菜单中选择“编辑”命令，或直接在属性对话框中选择 PageFrame1 的 Page1 对象。页框的四周出现淡绿色边界，可以开始编辑第一页。将 Page1 的 Caption 属性改为欢迎。然后，在 Page1 上增加一个标签 Label1。修改其属性，如图 9-35 所示。

图 9-35　编辑第一页

② 用鼠标单击 Page2，或在属性对话框中选择 PageFrame1 的 Page2 对象，开始编辑第二页。将 Page2 的 Caption 属性改为日期。然后，在 Page2 上增加一个文本框 Text1、一个标签 Label1。并修改属性见表 9-22，如图 9-36 的左图所示。

表 9-22　属性设置

对　象	属　性	属 性 值	说　明
Text1	Alignment	2 – 中间	文本居中
	Value	=DATE（）	使用日期函数
	DateFormat	14 – 汉语	设置日期格式
Label1	Caption	今天是：	

③ 用鼠标单击 Page3，或在属性对话框中选择 PageFrame1 的 Page3 对象，开始编辑第三页。将 Page3 的 Caption 属性改为时间。然后，在 Page3 上增加一个文本框 Text1、一个计时器 Timer1 和两个标签控件 Label1、Label2。并修改其属性，如图 9-36 的右图所示。

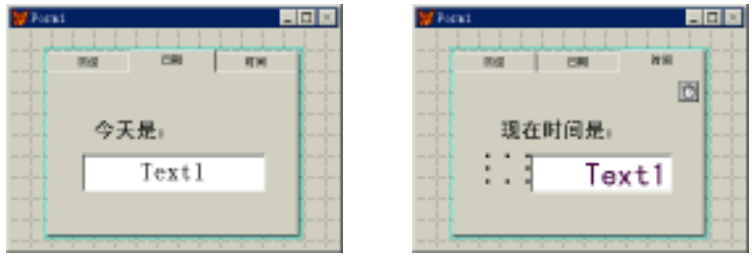


图 9-36　编辑第二页和第三页

④ 编写事件代码。

编写第二页中 Timer1 的 Timer 事件代码：

```
IF HOUR（DATETIME（））>= 12
   THIS.Parent.Label1.Caption = '下午'
ELSE
   THIS.Parent.Label1.Caption = '上午'
ENDIF
IF HOUR（DATETIME（））> 12
   hh = HOUR（DATETIME（））– 12
ELSE
   hh = HOUR（DATETIME（））
ENDIF
THIS.Parent.Text1.Value = STR（hh）+ SUBSTR（TIME（）,3）
```

说明：有时侯，由于个人的爱好，我们不喜欢选项卡的形式。可以设计用选项组或按钮组来控制页面的选择。

【例 9-20】不带选项卡的表单。将【例 9-19】中的页框架改为不带选项卡的形式，命令按钮组控制页面的选择，如图 9-37 所示。

设计步骤如下（在【例 9-19】的基础上进行修改，只给出修改的部分）：

① 选择“打开”表单文件，进入表单设计器，修改页框架控件 PageFrame1 的 Tabs 属性为.F. – 假，页框架改为不带选项卡的形式。然后，增加一个“命令按钮组”控件 CommandGroup1，并修改其属性。其中命令按钮组中的按钮 Command1 和 Command2 的 Enabled 属性改为.F. – 假，如图 9-37 所示。

图 9-37　不带选项卡的页框架

② 编写命令按钮组 CommandGroup1 的 Click 事件代码：

```
n = THIS.Value
```

```
k = THISFORM.PageFrame1.ActivePage
DO CASE
   CASE n=1
      THISFORM.PageFrame1.Pages（1）.Zorder
   CASE n=2
      THISFORM.PageFrame1.Pages（k-1）.Zorder
   CASE n=3
      THISFORM.PageFrame1.Pages（k+1）.Zorder
   CASE n=4
      THISFORM.PageFrame1.Pages（3）.Zorder
ENDCASE
k = THISFORM.PageFrame1.ActivePage
THIS.Command1.Enabled = IIF（k = 1, .F., .T.）
THIS.Command2.Enabled = IIF（k = 1, .F., .T.）
THIS.Command3.Enabled = IIF（k = 3, .F., .T.）
THIS.Command4.Enabled = IIF（k = 3, .F., .T.）
```

说明：

① ActivePage **表示被激活页（当前页）的页号。**

② Pages（ ）**表示页框架** PageFrame1 **中的页对象数组。**

9.4.3 表格

表格控件也是一个容器对象，和表单集包含表单一样，表格也能包含列，见表 9-23**。这些列除了包含标头和控件外，每一个列还拥有自己的一组属性、事件和方法程序，从而为表格单元提供了大量的控件。**

表格对象能在表单或页面中显示并操作行和列中的数据。使用表格控件的一个非常有用的应用程序是创建一对多表单，例如一个发票表单。

表 9-23　**表格作为一个容器**

容　器	可以包含
表格	列
列	标头，控件

1．常用的表格属性

（1）ChildOrder **属性**

与父表的主关键字相联接的子表中的外部关键字。

（2）ColumnCount**属性**

列的数目。如果 ColumnCount **设置为**-1**，表格将具有和表格数据源中字段数一样多的列。**

（3）LinkMaster**属性**

显示在表格中的子记录的父表。

（4）RecordSource**属性**

表格中要显示的数据。

（5）RecordSourceType**属性**

表格中显示数据来源于何处：表、别名、查询或用户根据提示选定的表。

（6）ActiveColumn 属性

返回一个整数，表明表格控件中包含活动单元的列的编号。设计时不可用，运行时只读。如果表格没有焦点，则 ActiveColumn 返回零。

（7）ActiveRow 属性

指定表格控件中包含活动单元的行。设计时不可用，运行时只读。

ActiveRow 属性的返回值与索引表中 RECNO () 函数的返回值不同。如果表格没有焦点，则 ActiveRow 返回零。

（8）ScrollBars 属性

指定表格所具有的滚动条类型。设计时可用，运行时可读写。取值说明为 0 - 无（默认值）；1 - 水平滚动条；2 - 垂直滚动条；3 - 水平滚动条和垂直滚动条。

2．常用的列属性

（1）ControlSource属性

在列中要显示的数据。常见的是表中的一个字段。

（2）Sparse属性

如果将 Sparse 属性设置为“真”(.T.)，表格中控件只有在列中的单元被选中时才显示为控件（列中的其他单元仍以文本形式显示）。将 Sparse 设置为“真”(.T.)，允许用户在滚动一个有很多显示行的表格时能快速重画。

（3）CurrentControl属性

确定表格中哪一个控件是活动的。默认值为“Text1”。如果在列中添加了一个控件，则可以将它指定为 CurrentControl。

注意：列中控件的ReadOnly属性被列的 ReadOnly 属性覆盖。若在和AfterRowColChange事件相关的代码中设置列中控件的 ReadOnly 属性，当指针位于该列时，新的设置有效。

3．常用的表格事件

（1）AfterRowColChange 事件

当用户移到表格的另一行或列时，新单元获得焦点以及新行或列中对象的 When 事件发生后，发生此事件。如果新行或列中对象的 When 事件不返回“真”(.T.)，则不触发 AfterRowColChange 事件。其语法格式为：

```
LPARAMETERS nColIndex
```

其中 nColIndex 返回最新选择的行或列的索引。

说明：触发 AfterRowColChange 事件的方法可以通过交互式使用鼠标或键盘，或用编程的方法，或调用 ActivateCell 方法。

（2）BeforeRowColChange 事件

当用户更改活动的行或列，而新单元还未获得焦点时发生该事件。也可以在表格列中当前对象和数据库中任何规则的 Valid 事件之前发生。使用 NODEFAULT 可以防止改变表格中活动的行和列。其语法格式为：

```
LPARAMETERS nColIndex
```

其中 nColIndex 返回新的活动行或活动列索引。

说明：可以采用鼠标、键盘交互方式或编程方式（如调用 ActivateCell **方法）触发** BeforeRowColChange **事件。**

（3） Scrolled **事件**

在表格控件（或表单）中，单击水平或垂直滚动条，或移动滚动条中的滚动块时，此事件发生。其语法格式为：

```
LPARAMETERS [nIndex], nDirection
```

其中 nIndex **惟一标识控制数组中的某个控件，** nDirection **指定用户如何滚动表格控制中的内容。可能的参数见表** 9-24**。**

表 9-24　**参数** nDirection **的取值**

取　值	说　明
0	使用上箭头键
1	使用下箭头键
2	单击垂直滚动条滚动块上面的区域
3	单击垂直滚动条滚动块下面的区域
4	使用左箭头键
5	使用右箭头键
6	单击水平滚动条滚动块左边的区域
7	单击水平滚动条滚动块右边的区域

说明：如果在程序代码中调用 DoScroll **方法，也发生** Scrolled **事件。**

4．常用的表格方法

（1） ActivateCell **方法**

激活表格控件中的一个单元。其语法格式为：

```
Grid.ActivateCell（nRow, nCol）
```

参数描述 nRow, nCol **指定活动单元所在的行和列。**

（2） DoScroll **方法**

滚动表格控件来模拟用户单击滚动条操作。其语法格式为：

```
Grid.DoScroll（nDirection）
```

其中参数 nDirection **的取值见表** 9-25**。**

表 9-25　**参数** nDirection **的取值**

值	滚 动 动 作
0	向上滚动
1	向下滚动
2	向上翻页滚动
3	向下翻页滚动
4	向左滚动
5	向右滚动

（续）

值	滚 动 动 作
6	向左翻页滚动
7	向右翻页滚动

说明：使用 DoScroll 方法后触发 Scrolled 事件。

5．使用表格

对表格可以有下列操作：

（1）将表格控件添加到表单

在“表单控件”工具栏中选择“表格”按钮，并在“表单”窗口中调整为期望的大小。

如果没有指定表格的RecordSource属性，同时在当前工作区中有一个打开的表，那么表格将显示这个表的所有字段。

（2）设置表格列数

在属性和方法程序列表中选择 ColumnCount 属性，在属性框中，键入需要的列数。

如果 ColumnCount 属性设置为-1（默认值），在运行时刻，表格将包含与其链接的表中字段同样数量的列。

（3）在设计时刻人工调整表格的显示效果

在表格中加入列后将改变列的宽度和行的高度。可以在“属性”窗口中人工设置列和行对象的高度和宽度属性，也可以在设计表格时以可视方式设置这些属性。

（4）切换到表格设计方式

从表格的快捷菜单中选择“编辑”命令。或者，在“属性”窗口的“对象”框中，选择表格的一列。

在表格设计方式下，表格周围将显示一个粗框。若要退出表格设计方式，只需选择表单或其他控件。

（5）调整表格中列的宽度

在表格设计方式下，将鼠标指针置于表格列的标头之间，这时指针变为带有左右两个方向箭头的竖条。然后，将列拖动到需要的宽度。或者，在“属性”窗口中设置列的 Width 属性。

（6）调整表格中行的高度

在表格设计方式下，将鼠标指针置于“表格”控件左侧的第一个按钮和第二个按钮之间，这时指针将变成带有向上和向下箭头的横条。然后，将行拖动到需要的宽度，或者在“属性”窗口中设置列的 Height 属性。

注意：将AllowRowSizing设置为“假”(.F.)，可以防止用户在运行时刻改变表格行的高度。

6．设置表格中显示的数据源

可以为整个表格设置数据源，也可以为每个列单独设置数据源。

（1）为整个表格设置数据源

选择表格，首先单击“属性”窗口的 RecordSourceType 属性。如果让 VFP 打开表，可将 RecordSourceType 属性设置为 0 - 表；如果在表格中放入打开表的字段，则将 RecordSourceType 属性设置为 1 - 别名。然后单击“属性”窗口中的 RecordSource 属性。键

入作为表格数据源的别名或表名。

（2）为列单独设置数据源

选择列，然后单击“属性”窗口的 ControlSource 属性。键入作为列的数据源的别名、表名或字段名。例如，可以键入 Orders.order_id。

7．向表格添加记录

将表格的AllowAddNew属性设置为“真”(.T.)，可以允许用户向表格中显示的表中添加新的记录。如果将 AllowAddNew 属性设置为真，当用户选中了最后一个记录，并且按下 DOWN ARROW 键时，就向表中添加了新记录。

如果需要进一步控制用户什么时候向表中添加新记录，可以将 AllowAddNew 属性设置为默认的“假”(.F.)，并使用APPEND BLANK或INSERT命令添加新记录。

8．使用表格控件创建一对多表单

表格最常见的用途之一是，当文本框显示父记录数据时，表格显示表的子记录；当用户在父表中浏览记录时，表格将显示相应的子记录。

如果表单的数据环境包含两表之间的一对多关系，那么要在表单中显示这个一对多关系非常容易。

（1）设置具有数据环境的一对多表单

将需要的字段从“数据环境”中的父表拖动到表单中。从“数据环境”中将相关的表拖动到表单中。

（2）创建没有数据环境的一对多表单

首先，将文本框添加到表单中，显示主表中需要的字段。设置文本框的ControlSource属性为主表。然后，在表单中添加一个表格。将表格的RecordSource属性设置为相关表的名称。设置表格的LinkMaster属性为主表名称。设置表格的ChildOrder属性为相关表中索引标识的名称，索引标识和主表中的关系表达式相对应。将表格的RelationalExpr属性设置为联接相关表和主表的表达式。

无论用哪种方法建立一对多表单，都可以添加定位控件，浏览父表并刷新表单对象。例如，在一个命令按钮的Click 事件中可包含下面的代码：

```
SELECT orders                    && 如果 orders 是父表
SKIP
IF EOF（）
   GO BOTTOM
ENDIF
THISFORM.Refresh
```

9．在表格列中显示控件

除了在表格中显示字段数据，还可以在表格的列中嵌入控件，这样就为用户提供嵌入的文本框、复选框、下拉列表框、微调按钮和其他控件。例如，如果表中有一个逻辑字段，当运行该表单时，通过辨认复选框可以判定哪个记录值是“真”(.T.)和哪个记录值是“假”(.F.)。修改这些值只需设置或清除复选框即可。

（1）在“表单设计器”中交互地向表格列中添加控件

可以在“表单设计器”中交互地向表格列中添加控件，也可以通过编写代码在运行时刻

添加控件。这里只介绍在“表单设计器”中交互地向表格列中添加控件的步骤：

① 在表单中添加一个表格。

② 在“属性”窗口中，将表格的 ColumnCount 属性设置为需要的列数。例如，如果需要一个两列的表格则键入 2。

③ 在“属性”窗口的“对象”框中为控件选择父列。例如，要选择 Column1 来添加控件，当选择这一列时，表格的边框发生变化，表明正在编辑一个包含其中的对象。

④ 在“表单控件”工具栏中选择所要的控件，然后单击父列。在“表单设计器”中，新控件不在表格列中显示，但在运行时刻会显示出来。

⑤ 在“属性”窗口中，要确保该控件缩进显示在“对象”框中父列下面。例如，添加到表格列中的新控件是一个复选框，应将复选框的 Caption 属性设置为“ ”，并将列的 Sparse 属性设置为假（.F.）。

⑥ 将父列的 ControlSource 属性设置为需要的表字段。

⑦ 将父列的 CurrentControl 属性设置为新加入的控件。

当运行表单时，这个控件将显示在表格列中。

说明：如果想让复选框在表格列中居中，可先创建一个容器类，将复选框添加到容器类中，并调整复选框在容器类中的位置。然后将容器类添加到表格列中，并将复选框的 ControlSource 属性设置为需要的字段。

（2）在“表单设计器”中移去表格列中的控件

若要在“表单设计器”中移去表格列中的控件，可以按照如下步骤：

① 在“属性”窗口的对象框中选择要移去的控件。

② 激活“表单设计器”。此时，如果“属性”窗口可见，控件的名称将显示在“对象”框中。

③ 按下〈Delete〉键。

10．在表格中进行有条件的格式设置

表格中的特定格式能让用户更容易浏览表格记录，并找出想要的信息。如果想进行有条件的格式设置，可使用列的动态字体和颜色属性。

例如，可以将表格添加到表单中，并将 ColumnCount 属性设置为 2。将第一列的 ControlSource 属性设置为 orders.to_name，第二列的 ControlSource 属性设置为 orders.order_net。如果想用黑色的前景色显示少于 500.00 的订货总计，用红色的前景色显示大于或等于 500.00 的订货总计，可在表格的 Init 事件代码中包含下列代码：

```
THIS.Column2.DynamicForeColor = "IIF（orders.order_net >= 500, RGB（255,0,0）, RGB（0,0,0）） "
```

9.5 图形与图像类控件

图形与图像类控件包括形状（Shape）、线条（Line）和图像（Image）控件。形状与线条可以将表单上的对象归并成组，这样将相关项联系起来有助于用户了解界面，更容易使用应用程序。而图像除了美化表单之外，还可以显示数据表中的图像数据。

9.5.1 形状

1．形状控件的属性

形状控件用于画矩形、正方形、圆角矩形、圆角正方形、圆和椭圆，如图 9-38 所示。其主要属性有下列几种：

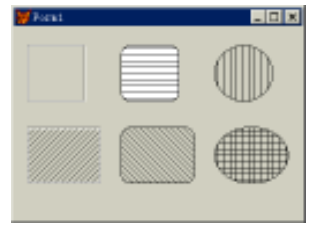


图 9-38 Shape 控件的各种形状

（1）Curvature 属性

确定角半径，有效的设置是从 0（方或长方形）到 99（圆或椭圆）。

（2）SpecialEffect 属性

确定边线是三维（0）还是平面（1）的。

（3）BorderColor 属性

确定边框的颜色。

（4）BorderStyle 属性

确定边框样式，取值有 0 – 透明、1 – 实线（默认）、2 – 虚线、3 – 点线、4 – 点划线、5 – 双点划线、6 – 内实线。

（5）BorderWidth 属性

确定边框宽度，取值为整数。

设置 BorderStyle 属性产生的效果取决于 BorderWidth 属性的设置。如果 BorderWidth 大于 1，则无论 BorderStyle 的设置如何，都将 BorderStyle 设置成 1。

（6）FillStyle 属性

用来确定填充形状的图案。

（7）FillColor 属性

确定填充形状的颜色。

可以在设计时通过设置其属性来确定显示某种图形，也可以在程序执行的时候动态地改变图形的形式。

2．使用形状

【例 9-21】利用“形状”控件修饰【例 9-1】中的表单，如图 9-39 所示。

在【例 9-1】的基础上进行设计，步骤如下：

在【例 9-1】的表单中画上一个“形状”控件 Shape1，如图 9-40 的左图所示。

修改 Shape1 的 SpesialEffect 属性为 0 – 三维，然后用鼠标单击“格式”菜单中的“置后”，将其置于原有控件的后边，如图 9-40 的右图所示。适当调整各控件的位置，即完成对原有表单的修饰。

图 9-39 使用“形状”控件

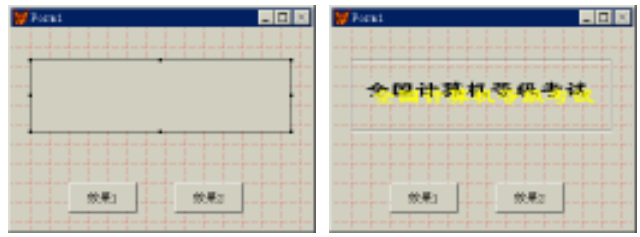


图 9-40 增加一个“形状”

【例 9-22】利用微调器控制图形的大小。如图 9-41 所示，用鼠标单击向下箭头时，图形中有色部分逐渐降低，反之，有色部分逐渐增高。

设计步骤如下：

① 建立应用程序用户界面。选择“新建”表单，进入表单设计器，增加一个“微调控件”Spinner1 和“容器”控件 Container1，先将容器（Container1）的属性 Backcolor 改为 0,255,0（粉绿色），SpecialEffect 改为 1 – 凹下（默认为 2 – 平面），把它的高度 Height 改为 123。然后，在容器中单击鼠标右键，打开快捷菜单，在快捷菜单中选择“编辑”命令，如图 9-42 所示。

图 9-41　利用微调器控制形状

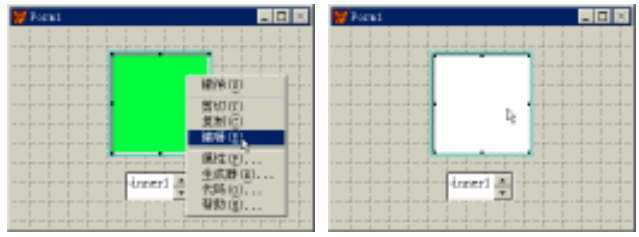


图 9-42　编辑容器

可以看到容器控件的四周出现浅绿色的边界，表示现在可以编辑该容器了。在容器中增加一个“形状”控件 Shape1，Shape1 覆盖了 Container1。将其属性 Backcolor 改为 255,255,255（白色），BorderStyle 改为 0 – 透明（默认为 1 – 实线）。然后用鼠标单击容器外表单内的任何地方，容器控件四周的浅绿色边界消失。

② 修改 Spinner1 的属性，见表 9-26。

表 9-26　设置属性

对　象	属　性	属 性 值	说　明
Spinner1	Spinnerhigh	120	数值变化的最大值
	Spinnerlow	0	数值变化的最小值
	KeyboardHighvalue	120.00	键盘可输入的最大值
	KeyboardLowvalue	0.00	键盘可输入的最小值

③ 编写程序代码。编写 Spinner1 的 InteractiveChange 事件代码：

```
THISFORM.Container1.Shape1.Top = 120 - THIS.Value
```

说明：运行表单，通过单击微调器可以控制容器中有色部分升高或降低。

9.5.2　线条

线条（Line）控件用于画各种直线段。它既可以在设计时通过设置线的端点坐标属性来画出线条，也可以在程序运行的时候动态地改变线条的各种属性。

在 FoxPro 的以前版本中，线条只能是水平或垂直的，而在 VFP 中，线条可以连接表单上的任意两点。

1．线条的属性

线条的属性主要有：

（1）BorderColor 属性

确定线条的颜色。

（2）BorderStyle 属性

确定线条样式，取值有 0 – 透明、1 – 实线（默认）、2 – 虚线、3 – 点线、4 – 点划线、5 – 双点划线、6 – 内实线。

（3）BorderWidth 属性

确定线条宽度，取值为整数。

设置 BorderStyle 属性产生的效果取决于 BorderWidth 属性的设置。如果 BorderWidth 大于 1，则无论 BorderStyle 的设置如何，都将 BorderStyle 设置成 1。

（4）LineSlant 属性

确定线条的倾斜方向，取值为“\”，表示线条从左上到右下倾斜（默认），取值为“/”，表示线条从左下到右上倾斜。

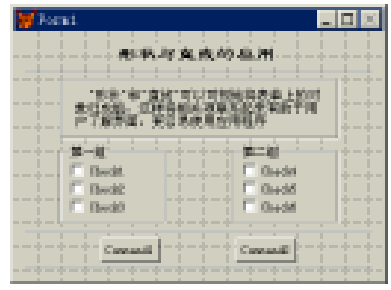


图 9-43　使用形状和线条

2．使用线条

线条控件主要被用来修饰表单，如图 9-43 所示。有时也是程序界面中必不可少的部分。下面的例子使用图形与计时器设计简单的动画。

【例 9-23】利用图形控件设计的“曲柄滑块机构的演示”，如图 9-44 所示。

设计步骤如下：

① 建立应用程序用户界面。选择“新建”表单，进入表单设计器，增加一个命令按钮 Command1、两个连杆（Line1、Line2）、三个“形状”控件 Shape1～Shape3、一个“计时器”、一个标签 Label1、一些直线（Line3～Line31），如图 9-45 所示。

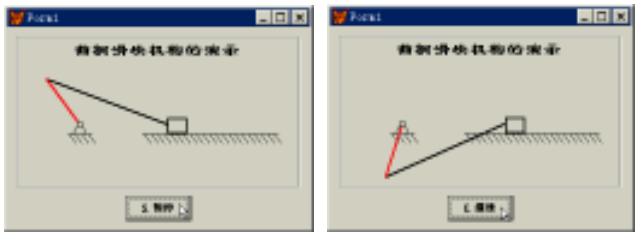


图 9-44　曲柄滑块机构的演示

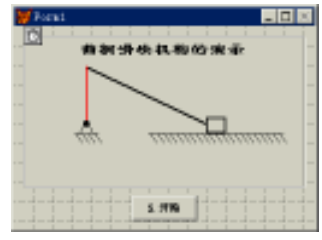


图 9-45　设计用户界面

② 设置对象属性，见表 9-27，其他控件的属性如图 9-45 所示。

表 9-27　设置属性

对　象	属　性	属 性 值	说　明
Shape1	SpecialEffect	0 – 3 维	
Line1	BorderColor	255,0,0（红）	连杆 Line1
	BorderWidth	2	

（续）

对　象	属　性	属 性 值	说　明
Line2	BorderWidth	3	
Shape2	Curvature	99	支点
	BorderWidth	2	
Shape3	BorderWidth	2	
Timer1	Interval	100	
	Enabled	.F. – 假	
Command1	Caption	\<S 开始	

③ 编写程序代码。

编写表单的 Activate 事件代码：

```
PUBLIC top1, left1
PUBLIC changdu[2]
PUBLIC t
t = 0
changdu（1）= THISFORM.Line1.Height
changdu（2）= Sqrt（THISFORM.Line2.Width^2 + THISFORM.Line2.Height ^ 2）
top1=120                    && 固定支点的纵坐标
left1=90                    && 固定支点的横坐标
```

编写命令按钮 Command1 的 Click 事件代码：

```
IF THIS.Caption = '\<S 暂停'
   THISFORM.Timer1.Enabled = .F.
   THIS.Caption = '\<C 继续'
ELSE
   THIS.Caption = '\<S 暂停'
   THISFORM.Timer1.Enabled = .T.
ENDIF
```

编写计时器的 Timer 事件代码：

```
t = t + 1
t = Mod（t, 60）
   kuan = changdu（1）*Cos（Pi（）*（15-t）/30）
   gao = changdu（1）*Sin（Pi（）*（15-t）/30）
   WITH THISFORM.Line1
      .Height = Abs（gao）
      .Width = Abs（kuan）
   ENDWITH
   DO CASE
      CASE t >= 0 .AND. t < 15
         THISFORM.Line1.Top = top1-THISFORM.Line1.Height
         THISFORM.Line1.Left = left1
         THISFORM.Line1.Lineslant = '/'
         THISFORM.Line2.Left = THISFORM.Line1.Left + THISFORM.Line1.Width
```

```
            THISFORM.Line2.Height = THISFORM.Line1.Height
            THISFORM.Line2.Top = THISFORM.Line1.Top
            THISFORM.Line2.Width = Sqrt（changdu（2）^2- changdu（1）^2）
            THISFORM.Line2.Lineslant = '\'
        CASE t>=15 .AND. t<30
            THISFORM.Line1.Top = top1
            THISFORM.Line1.Left = left1
            THISFORM.Line1.Lineslant = '\'
            THISFORM.Line2.Left = THISFORM.Line1.Left+THISFORM.Line1.Width
            THISFORM.Line2.Height = THISFORM.Line1.Height
            THISFORM.Line2.Top = THISFORM.Line1.Top
            THISFORM.Line2.Width = Sqrt（changdu（2）^2- changdu（1）^2）
            THISFORM.Line2.Lineslant = '/'
        CASE t >= 30 .AND. t < 45
            THISFORM.Line1.Top = top1
            THISFORM.Line1.Left = left1-THISFORM.Line1.Width
            THISFORM.Line1.Lineslant = '/'
            THISFORM.Line2.Left = THISFORM.Line1.Left
            THISFORM.Line2.Height = THISFORM.Line1.Height
            THISFORM.Line2.Top = THISFORM.Line1.Top
            THISFORM.Line2.Width = Sqrt（changdu（2）^2- changdu（1）^2）
            THISFORM.Line2.Lineslant = '/'
        CASE t >= 45 .AND. t < 60
            THISFORM.Line1.Top = top1-THISFORM.Line1.Height
            THISFORM.Line1.Left = left1-THISFORM.Line1.Width
            THISFORM.Line1.Lineslant = '\'
            THISFORM.Line2.Left = THISFORM.Line1.Left
            THISFORM.Line2.Height = THISFORM.Line1.Height
            THISFORM.Line2.Top = THISFORM.Line1.Top
            THISFORM.Line2.Width = Sqrt（changdu（2）^2- changdu（1）^2）
            THISFORM.Line2.Lineslant = '\'
    ENDCASE
THISFORM.Shape1.Left = THISFORM.Line2.Left + THISFORM.Line2.Width
```

9.5.3 图像

1．图像的属性

图像（Image）控件允许在表单中添加图片（.BMP、.ICO 文件）。图像控件和其他控件一样，具有一整套的属性、事件和方法程序。因此，在运行时可以动态地更改它。用户可以用单击、双击和其他方式来交互地使用图像。

表 9-28 列出了图像控件的一些主要属性。

表 9-28　图像控件的主要属性

属　性	说　明
Picture	要显示的图片（.BMP 或 .ICO 文件）

（续）

属　性	说　明
BorderStyle	决定图像是否具有可见的边框
BackStyle	决定图像的背景是否透明
Stretch	如果 Stretch 设置为 0 - 剪裁，那么超出图像控件范围的那一部分图像将不显示；如果 Stretch 设置为 1 - 恒定比例，图像控件将保留图片的原有比例，并在图像控件中显示最大可能的图片；如果 Stretch 设置为 2 - 伸展，将图片调整到正好与图像控件的高度和宽度匹配

2．使用图像

利用图像控件设计一个图片浏览器

【例 9-24】简单的图片浏览器程序。使用列表框控件与图像控件设计的“图片浏览器”可以方便地浏览各种图片，如图 9-46 所示。

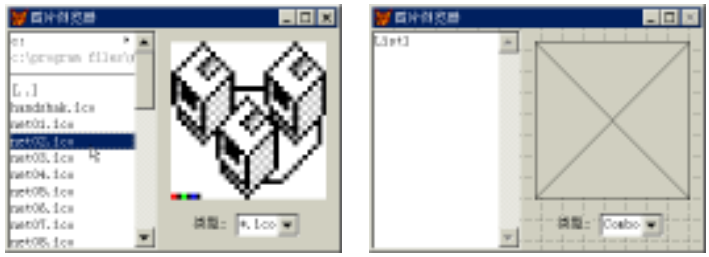


图 9-46　图片浏览器

① 选择新建表单，进入表单设计器，增加一个列表框控件 List1、一个标签 Label1、一个组合框 Combo1 和一个图像控件 Image1，如图 9-46 的右图所示。

② 修改属性见表 9-29。

表 9-29　设置属性

对　象	属　性	属 性 值	说　明
List1	RowSourceType	7 – 文件	
Combo1	RowSourceType	1 – 值	
	RowSource	*.bmp,*.ico	
	ControlSource	THISFORM.List1.RowSource	
Image1	Stretch	2 – Stretch	伸展
Label1	Caption	类型：	

③ 编写事件代码。

表单的 Activate 事件代码：

```
THIS.Combo1.DisplayValue = 2
```

列表框 List1 的 InteractiveChange 事件代码：

```
a = THIS.ListIndex
IF a > 3
   THISFORM.Image1.Picture = THIS.List（2）+ THIS.List（a）
   CD THIS.List（2）
```

```
ENDIF
```

说明：List1.List（2）**代表路径，参见【例** 9-11**】。**

9.6 习题

一、选择题

1．**决定标签内显示内容的属性是（　　）。**

A）Text　　B）Name　　C）Alignment　　D）Caption

2．**下面对编辑框（**EditBox**）控件属性的描述正确的是（　　）。**

A）SelLength **属性的设置可以小于** 0

B）**当** ScrollBars **属性值为** 0 **时，编辑框内包含水平滚动条**

C）SelText **属性在做界面设计时不可用，在运行时可读写**

D）ReadOnly **属性值为**.T.**时，用户不能使用编辑框上的滚动条**

3．**下面对控件的描述正确的是（　　）。**

A）**用户可以在组合框中进行多重选择**

B）**用户可以在列表框中进行多重选择**

C）**用户可以在一个选项框中选择多个选项按钮**

D）**用户对表单内的一组复选框只能选定其中一个**

4．**确定列表框内的某个条目是否被选定应使用的属性是（　　）。**

A）Value　　B）ColumnCount　　C）ListCount　　D）Selected

5．**在表单设计器环境下，要选定表单中某选项组里的某个选项按钮，可以（　　）。**

A）**单击选项按钮**　　B）**双击选项按钮**

C）**先单击选项组，并选择“编辑”命令，然后再单击选项按钮**

D）**以上** B）**和** C）**都可以**

6．**假定一个表单里有一个文本框** Text1 **和一个命令按钮组** Commandgroup1**，命令按钮组是一个容器对象，其中包含** Command1 **和** Command2 **两个命令按钮。如果要在** Command1 **命令按钮的某个方法中访问文本框的** Value **属性值，下面哪个式子是正确的（　　）？**

A）THIS.THISFORM.Text1.Value　　B）THIS.Parent.Parent.Text1.Value

C）Parent.Parent.Text1.Value　　D）THIS.Parent.Text1.Value

7．**下面关于列表框和组合框的陈述中，哪个是正确的（　　）？**

A）**列表框和组合框都可以设置成多重选择**

B）**列表框可以设置成多重选择，而组合框不能**

C）**组合框可以设置成多重选择，而列表框不能**

D）**列表框和组合框都不能设置成多重选择**

二、填空题

1．**用来确定复选框是否被选中的属性是**_____**，用来指定显示在复选框旁的文字的属性是**_____**。**

2．为了使标签能自动调整大小以显示全部文本内容，应把标签的_____属性设置为True。

3．假定表单中有一个文本框，其名称为 Text1，为了使该文本框具有焦点，应执行的语句是_____。

4．为了使一个标签透明且没有边框，必须把它的 BorderStyle 属性设置为_____，并把 Backstyle 属性设置为_____。

5．计时器事件之间的间隔通过_____属性设置。

6．有时候需要暂时关闭计时器，这可以通过_____属性来实现。

7．组合框有两种不同的类型，这两种类型是_____、_____，分别通过把_____属性设置为_____、_____来实现。

三、上机题

1．表单 form1 中包含两个文本框 Text1、Text2 和一个命令按钮“确定”Command1，打开并按如下要求修改 form1 表单文件（最后保存所做的修改）：

（1）在“确定”命令按钮的 click 事件（过程）下的程序有两处错误，请改正之;

```
&&**********Error**********
if thisform.text1 = thisform.text2
    wait "欢迎使用……" window timeout 1
&&**********Error**********
    thisform.close
else
    wait "用户名或口令不对，请重新输入……" window timeout 1
endif
```

（2）设置 Text2 控件的有关属性，使用户在输入口令时显示“*”（星号）。

2．设计一个“选取或替换”表单，如图 9-47 所示。单击“确定”按钮时，如果“选取或替换”复选框处于选中状态，那么就将文本框内容设置成编辑框中的选定内容，否则，就用文本框的内容去替换编辑框中的选定内容。

图 9-47　上机题 2

在题 3～5 中，假定在考生文件夹下，数据库 salary_db 中包含两个数据表 dept、salarys，一个视图 sview。

3．完成如下简单应用：设计一个名称为 form1 的表单，表单以表格方式（与 BROWSE

窗口方式相似，表格名称为 grdSalarys）显示 salary_db 数据库中 salarys 表的记录，供用户浏览。在该表单的右下方有一个命令按钮，名称为 Command1，标题为“退出浏览”，当单击该按钮时退出表单，如图 9-48 所示。

4．完成如下综合应用：设计一个名称为 form1 的表单，在表单上设计一个选项按钮组（名称为 Optiongroup1）及两个命令按钮“生成”（名称为 Command1）和“退出”（名称为 Command2）；其中选项按钮组有“雇员工资表”（名称为 Option1）、“部门表”（名称为 Option2）和“部门工资汇总表”（名称为 Option3）三个选项按钮（如图 9-49 所示）。然后为表单建立数据环境，并向数据环境添加 dept 表（名称为 Cursor1）和 salarys 表（名称为 Cursor2）。

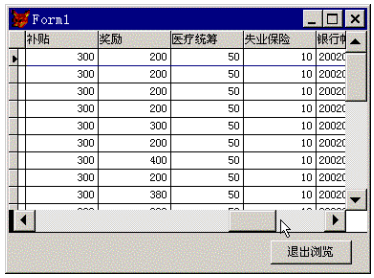


图 9-48　上机题 3

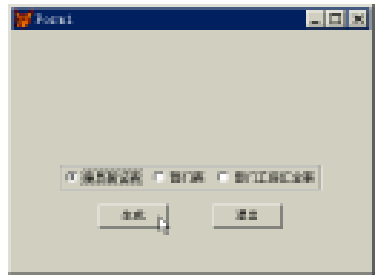


图 9-49　上机题 4

各选项按钮功能如下：

（1）当用户选择“雇员工资表”选项按钮后，再按“生成”命令按钮，查询显示 sview 视图中的所有信息并把结果存入表 gz1.dbf 中。

（2）当用户选择“部门表”选项按钮后，再按“生成”命令按钮，查询显示 dept 表中每个部门的部门号和部门名称并把结果存入表 bm1.dbf 中。

（3）当用户选择“部门工资汇总表”选项按钮后，再单击“生成”命令按钮，则按部门汇总，将该公司的部门号、部门名、工资、补贴、奖励、失业保险和医疗统筹的支出汇总合计结果存入表 hz1.dbf 中。请注意：字段名必须与原字段名一致。

（4）单击“退出”按钮，退出表单。

注意：以上各项功能必须调试、运行通过。

5．创建一个浏览数据表 salarys 的表单，如图 9-50 所示。

图 9-50　上机题 5

第 10 章　菜单设计与应用

使用菜单可以有效地组织应用程序的各项功能，使用户更加方便、迅速地使用应用程序。

10.1　VFP 菜单简介

在 VFP 中，可以创建多种菜单：可以添加新的菜单选项到菜单系统中——定制已有的 VFP 菜单系统；也可以创建一个全新的自定义菜单，以代替 VFP 的菜单系统；还可以创建依附于某个控件的快捷菜单。

10.1.1　VFP 菜单结构

1．条形菜单和弹出式菜单

VFP 的菜单分为两种类型：横向的条形菜单和纵向的弹出式菜单。条形菜单和弹出式菜单都有一个内部名字，并由一组菜单选项组成。每个菜单选项都有一个用作菜单标题的“菜单名称”。

条形菜单的每个菜单选项都有一个内部名字，弹出式菜单的每个菜单选项则都有一个选项序号。

内部名字或选项序号用于在代码中引用，而“菜单名称”则作为标题，显示于屏幕供用户识别。

每个菜单选项都可以设置一个访问键（热键）和一个快捷键。访问键通常是一个字符，当菜单激活时，通过按访问键可以快速选择该菜单项；快捷键通常是〈Ctrl〉键与另一个字符的组合，无论菜单是否激活，都可以通过快捷键快速选择该菜单项。

每个菜单选项都可以设置一个动作：执行一个命令、执行一个过程或激活另一个菜单。

2．VFP 的菜单系统

VFP 的菜单系统有两种形式：下拉式菜单和快捷菜单。

下拉式菜单系统由一个条形菜单（主菜单）和一组弹出式菜单（子菜单）组成。当选择一个条形菜单上的菜单选项时，激活相应的一个弹出式菜单。

快捷菜单系统则由一个或一组上下关联的弹出式菜单组成。

10.1.2　VFP 中的菜单

1．定制 VFP 菜单系统

VFP 内含一个包含众多菜单选项的菜单系统，这些菜单选项可以直接添加到应用程序的菜单系统中，而免去编写代码的麻烦。使用“快速菜单”的功能，可以定制已有的菜单系统。

2．自定义菜单

使用“菜单设计器”，VFP 可以方便地创建一个完全自定义的菜单系统。

3．快捷方式菜单

使用“快捷菜单设计器”，VFP 可以方便地创建一个快捷方式菜单。快捷方式菜单可以依附于某个控件，当用户在控件上单击鼠标右键时，将弹出快捷方式菜单。

10.2 菜单设计的步骤

无论是定制已有的 VFP 菜单系统，还是开发一个全新的自定义菜单，创建一个完整的菜单系统一般需要以下步骤：

① 规划系统：确定需要哪些菜单项、出现在界面的何处以及哪几个菜单项要有子菜单等。

② 创建菜单和子菜单。

③ 为菜单系统指定任务：指定菜单项所要执行的任务，例如显示表单或对话框等。另外，如果需要，还可以包含初始化代码和清理代码。

④ 选择“预览”按钮预览整个菜单系统。

⑤ 从“菜单”菜单上选择“生成”命令，生成菜单程序。

⑥ 运行生成的程序，测试菜单系统。

10.2.1 规划菜单系统

应用程序的实用性在一定程度上取决于菜单系统的质量。在设计菜单系统时，应考虑下列准则：

① 按照用户所要执行的任务组织系统，而不要按应用程序的层次组织系统。只要查看菜单和菜单项，用户就应该可以对应用程序的组织方法有一个感性认识。因此，要设计好这些菜单和菜单项，程序员必须清楚用户思考问题的方法和完成任务的方法。

② 给每个菜单一个有意义的菜单标题（菜单名称）。按照估计的菜单项使用频率、逻辑顺序或字母顺序组织菜单项。如果不能预计频率，也无法确定逻辑顺序，则可以按字母顺序组织菜单项。当菜单中包含有八个以上的菜单项时，按字母顺序特别有效。太多的菜单项需要用户花费一定的时间才能浏览一遍，而按字母顺序则便于查看菜单项。

③ 在菜单项的逻辑组之间放置分隔线。

④ 将菜单上菜单项的数目限制在一个屏幕之内。如果菜单项的数目超过了一屏，则应为其中的一些菜单项创建子菜单。

⑤ 为菜单和菜单项设置访问键或键盘快捷键。例如，快捷键〈Alt〉+〈F〉可以作为“文件”菜单的访问键。

⑥ 使用能够准确描述菜单项的文字。描述菜单项时，应使用日常用语，而不要使用计算机术语。同时，说明选择一个菜单项产生的效果时，应使用简单、生动的动词，而不要将名词当作动词使用。另外，用相似语句结构说明菜单项。例如，如果对所有的菜单项的描述都使用了同一个词，则这些描述应使用相同的语言结构。

⑦ 对于英文菜单，可以在菜单项中混合使用大小写字母。只有强调时才全部使用大写字母。

10.2.2 菜单设计器

“菜单设计器”是 VFP 提供的又一个可视化编程工具。使用“菜单设计器”可以添加新的菜单选项到 VFP 的系统菜单中——定制已有的 VFP 系统菜单，也可以创建一个全新的自定义菜单，以代替 VFP 的系统菜单。

1．打开“菜单设计器”

选择主菜单中“文件”菜单中的“新建”命令，打开“新建”对话框。选中“新建”对话框底部的“菜单”选项，单击“新文件”按钮，打开“新建菜单”对话框。在对话框中，单击“菜单”按钮，打开“菜单设计器”，如图 10-1 所示。

图 10-1　菜单设计器

2.“菜单设计器”简介

菜单设计器包含：

（1）菜单名称

菜单系统中菜单选项的标题。可为菜单中的各菜单选项定义一个访问键。当菜单项名是英文词汇时，若选首字母是访问键，在该选项的名字前加上“\<”，如“\<Edit”。若要另选访问键，则要在选项名后加“(\<字母)”，如“编辑(\<E)”。

一旦在“菜单名称”栏中输入了任何内容，“菜单名称”栏左边将出现一个上下双箭头按钮——移动控件，利用“移动控件”可以可视地调整菜单项之间的顺序。

（2）结果

指定用户在选择菜单标题或菜单项时，将执行的动作。例如，可执行一个命令、打开一个子菜单或运行一个过程。

单击“结果”下拉表，有四个选择，对添加的选择有四种处理方式，见表 10-1。

表 10-1　4 种菜单结果

选　项	功　能
子菜单	选择此项，右边出现“创建”钮，单击“创建”钮可以生成一个子菜单。一旦建立了子菜单，“创建”钮就变为“编辑”钮，用它修改已经定义的子菜单。这是最常用的方式，当用户选择主菜单上的某一选项时，就会出现下拉菜单，这个下拉菜单就是用“创建”定义的，因此，系统将“子菜单”作为默认选择
命令	选择此项，右边出现一个文字框，要在文字框中输入一条单命令，当在菜单中选择此项时，就会执行这个命令。如“结束”选项，在结果中选定命令，在文字框中输入 QUIT 命令
填充名称	选择此项，在右边显示一个文字框，要在文字框中输入一个用户自己定义的或者系统的菜单项名。在子菜单中，“填充名称”选项由“菜单项 #”代替，在这个选项中，既可以指定用户自己定义的项号，也可以是系统菜单的菜单项的名字
过程	选择此项，则在右边出现一个“创建”钮，单击此钮打开一个编辑窗口，可以编辑菜单过程代码

说明：在编辑窗口中输入一个过程文件，当选择该菜单选项时系统就会自动运行这个过程文件。由于在生成程序时系统会自动生成这个过程名，所以不需要再用 PROCEDURE 命令给这个过程命名。一旦生成了过程文件，“创建”钮就变为“编辑”钮。

（3）选项

单击“选项”按钮显示“提示选项”对话框，如图 10-2 所示。

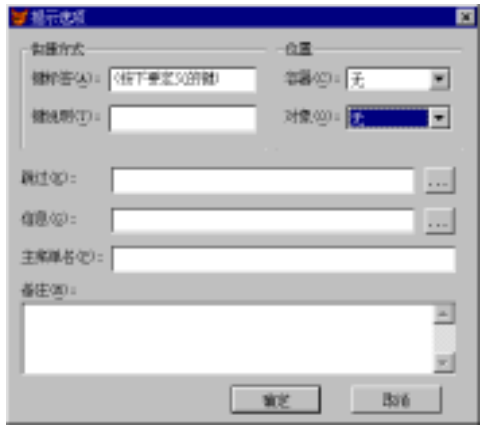


图 10-2 “提示选项”对话框

在“提示选项”对话框中可以定义键盘快捷键、确定废止菜单或菜单项的条件。当选定菜单或菜单项时，在状态栏中包含相应信息，指定菜单标题的名称以及在可视编辑期间控制菜单标题位置。提示选项对话框中的选择项见表 10-2。

表 10-2 “提示选项”对话框中的选项

选　项	功　能
快捷方式	显示键定义对话框，可在其中定义快捷键。在“键标签”框中按下一组合键，例如同时按下〈Ctrl〉键和〈C〉，“键标签”框中就会显示刚才按下的组合键，如〈CTRL〉+〈C〉。同时在“键说明”框中重复“键标签”框中的内容，用户可以修改，例如更改为^C。但要注意，〈Ctrl〉+〈J〉是无效的快捷键
位置	显示菜单位置对话框，可在其中指定当用户在应用程序中编辑一个 OLE 对象时，菜单标题的位置
跳过	显示表达式生成器。在“表达式生成器”的“跳过”框中，键入表达式来确定菜单或菜单项是否可用，当表达式值为.F.时，该菜单选项不可用
信息	显示“表达式生成器”。在“表达式生成器”的“信息”框中，可以键入用于说明菜单选择的信息，说明信息将出现在 Visual FoxPro 状态栏中
主菜单名	显示主菜单名对话框，可在其中指定可选的菜单标题。此选项仅在“菜单设计器”窗口的“结果”区域显示“命令”、“子菜单”或“过程”时可用
备注	输入用户自己使用的注释。在任何情况下注释都不影响所生成的代码，运行菜单程序时 Visual FoxPro 将忽略注释

（4）菜单级

显示当前正在设计的菜单级，在下拉列表框中还列出了当前子菜单的上级菜单名。选择上级菜单名可以返回上一级菜单栏对话框。

（5）插入

在当前行插入新的一行。

（6）插入栏

打开“插入系统菜单栏”对话框，选择在当前行插入系统菜单栏。

（7）删除

删除当前行的菜单定义。

（8）预览

显示正在创建的菜单。在菜单设计过程中，可随时单击“预览”钮显示当前创建的菜单，已经生成的菜单出现在原来系统菜单的地方。此外，“预览”对话框显示菜单的文件名或临时

文件的文件名。

10.2.3 主菜单中的有关选项

使用菜单设计器时，在“显示”菜单中将增加两个选项：常规选项与菜单选项，并且在主菜单中将增加一个“菜单”子菜单。

1．常规选项

选择主菜单中的“常规选项”，得到的是对应于整个菜单系统的选项。

选择“显示”菜单中的“常规选项”命令，显示“常规选项”对话框，如图 10-3 所示。

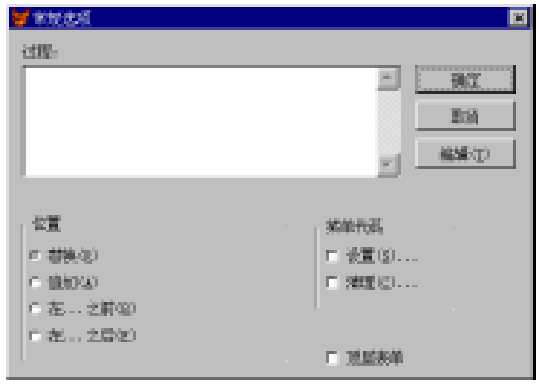


图 10-3 常规选项对话框

（1）“菜单代码”区

对话框的右下角有一个“菜单代码”区域，包含“设置”和“清理”两个复选框。

设置：打开一个编辑窗口，从中可以向菜单系统添加初始化代码。

清理：打开一个编辑窗口，从中可以向菜单系统添加清理代码。

在编辑窗口输入的命令，对于“设置”是在运行显示菜单的命令之前；而对于“清理”是在运行显示菜单的命令之后，并不是在使用完菜单之后。因此，它们分别称为初始代码和清理代码。

可通过向菜单系统添加初始代码而定制菜单系统。初始代码可以包含创建环境的代码、定义内存变量的代码、打开所需文件的代码以及使用 PUSH MENU 和 POP MENU 保存或还原菜单系统的代码。

可通过向菜单系统添加清理代码来裁减菜单系统。典型的清理代码包含初始时启用或废止菜单及菜单项的代码。在生成的菜单代码中，清理代码在初始代码及菜单定义代码之后，而在为菜单或菜单项指定的过程代码之前。

要激活编辑窗口，在“常规选项”对话框中选择“确定”按钮。

（2）“位置”区

对话框的左下角是位置区域，确定正在定义的菜单系统相对于激活菜单的位置，可以有以下几种选择。

替换：使用新的菜单系统替换当前系统菜单或活动菜单系统的原有内容。

追加：将新菜单系统添加在当前系统菜单的原有内容或活动菜单系统的右侧。

在...之前：将新菜单插入指定菜单的前面。这个选项显示一个包含活动菜单系统名称的下拉列表，要插入新菜单，选择希望新菜单在其前面的菜单名。

在...之后：将新菜单插入指定菜单的后面。这个选项显示一个包含活动菜单系统名称的下拉列表，要插入新菜单，选择希望新菜单紧跟其后的菜单名。

(3)“过程”文本框

“过程”文本框中，对于正在定义的菜单系统可以输入过程代码，它是作为某个指定菜单项的过程。使用中当选择这个菜单项时，此过程将被执行。如果这个菜单项原来已经生成了一个过程，则运行原来的那个过程，即这里输入的过程是在不存在其他过程时才运行。

(4)“顶层表单”复选框

由于“菜单设计器”创建的菜单系统默认位置是在 VFP 系统窗口之中，如果希望菜单出现在表单中，就需要选中“顶层表单”复选框，当然还必须将表单设置为“顶层表单”。

2．菜单选项

选择主菜单中的“菜单选项”，输入的过程仅属于主菜单条或一个指定的子菜单项。

选择“显示”菜单中的“菜单选项”命令，显示如图 10-4 所示的“菜单选项”对话框。

可以使用它对主菜单条或指定的子菜单添加过程，在“过程”文本框中或单击“编辑”按钮，都可在其中输入一个过程文件。当用户正在定义的是主菜单上的一个选项时，这个过程文件可以被主菜单上的所有选项调用。如果正在定义的是指定子菜单上的一个选项，则此过程可以被这个子菜单的所有选项调用。

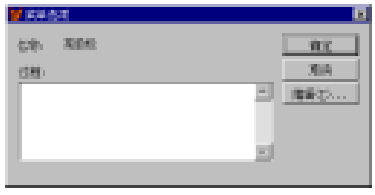


图 10-4　菜单选项对话框

图 10-5　生成菜单对话框

3．“生成”菜单码

完成菜单的定义后，还需生成菜单码。选择主菜单中的“菜单”菜单中的“生成”命令，选择“是”，在“另存为”对话框中输入菜单名，单击“确定”按钮后显示“生成菜单”对话框，如图 10-5 所示。在“输出文件”框中可改变输出文件名，系统默认扩展名为.MPR。单击“生成”按钮，则生成菜单程序，到此完成菜单的创建工作。

4．快速菜单

使用“快速菜单”功能可以将 VFP 的菜单系统放入菜单设计器中，供用户修改和操作。在其中可以增加用户自己的菜单和裁减、修改原来的系统菜单。这也是学习菜单设计的一种好方法。用“快速菜单”设计菜单的步骤为：

① 选择主菜单中的“文件”菜单中的“新建”命令，用鼠标单击“新建”对话框中的“菜单”按钮，单击“新文件”按钮，然后在打开的“新建菜单”对话框中单击“菜单”按钮，打开菜单设计器。

② 选择主菜单中的“菜单”菜单中的“快速菜单”命令，菜单设计器自动填充了系统主菜单中的所有选项，如图 10-6 所示。

③ 在菜单设计器中，用“插入”按钮插入新的菜单项，用“删除”按钮裁减菜单项，也可以通过选择主菜单上的选项调出相应子菜单。图 10-7 显示的是“文件”菜单的子菜单项。

图 10-6　快速菜单

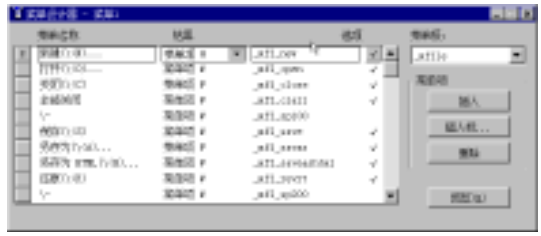


图 10-7　显示文件子菜单

④ 根据需要裁减、修改原来的系统菜单，具体操作步骤见下面的例子。

说明：如果在“常规选项”对话框中选择“替换”，将使用新设计的菜单系统替换当前系统菜单或活动菜单系统的原有内容。当退出应用程序时，需要恢复原有的系统菜单，可以使用如下命令：

```
SET SYSMENU NOSAVE
SET SYSMENU TO DEFAULT
```

10.2.4　在顶层表单中添加菜单

若要在顶层表单中添加菜单，可以按以下步骤操作：

① 创建顶层表单的菜单。即在“常规选项”对话框中，用鼠标选中“顶层表单”复选框。

② 将表单的 ShowWindow 属性设置为“2 – 作为顶层表单”。

③ 在表单的 Init 事件中，运行菜单程序并传递两个参数：

```
DO menuname.mpr WITH oForm, lAutoRename
```

说明：

① oForm 是表单的对象引用。在表单的 Init 事件中，THIS 作为第一个参数进行传递。

② lAutoRename 指定了是否为菜单取一个新的惟一的名字。如果计划运行表单的多个实例，则将.T.传递给 lAutoRename。

例如，可以使用下列代码调用名为 mySDImenu 的菜单：

```
DO mySDImenu.mpr WITH THIS, .T.
```

10.3　下拉式菜单的设计

使用“菜单设计器”可以创建菜单、菜单项、菜单项的子菜单和分隔相关菜单组的线条等。下面以一个具体实例来说明创建自定义菜单的方法。

10.3.1　创建一个自定义菜单

【例 10-1】在【例 9-18】中使用菜单来改变标题板中文本的字体与风格，如图 10-8 所示。

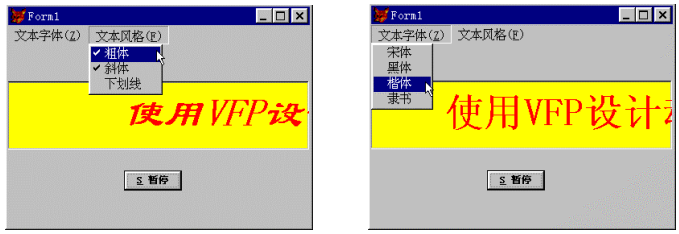


图 10-8　改变标题板中文本的字体与风格

设计步骤如下：

1．设计菜单

（1）规划菜单系统

见表 10-3。

表 10-3　菜单项的设置

菜 单 名 称	结　果	菜　单　级
文本字体（\<Z）	子菜单	菜单栏
宋体	过程	文本字体 Z
黑体	过程	文本字体 Z
楷体	过程	文本字体 Z
隶书	过程	文本字体 Z
文本风格（\<F）	子菜单	菜单栏
粗体	过程	文本风格 F
斜体	过程	文本风格 F
下划线	过程	文本风格 F

（2）创建菜单和子菜单

选择“文件”菜单中的“新建”命令，单击“新建”对话框底部的“菜单”按钮，单击“新文件”按钮，然后在打开的“新建菜单”对话框中单击“菜单”按钮，打开“菜单设计器”。

首先选择“显示”菜单中的“常规选项”命令，用鼠标选中“顶层表单”复选框，将菜单定位于顶层表单之中。单击“确定”按钮返回菜单设计器。

在菜单设计器中输入菜单名“文本字体（\<Z）”和“文本风格（\<F）”，如图 10-9 所示。

选中“文本字体”菜单项，单击“创建”按钮，输入子菜单项名，如图 10-10 所示。

返回上级菜单，选中“文本风格”菜单项，单击“创建”按钮，输入子菜单项名，如图 10-11 所示。

（3）编写菜单代码

在“菜单级”下拉列表框中选择“菜单栏”，回到顶层菜单。选择“文本字体”，用鼠标单击其右边“编辑”按钮，重新进入“文本字体 Z”的编辑对话框。

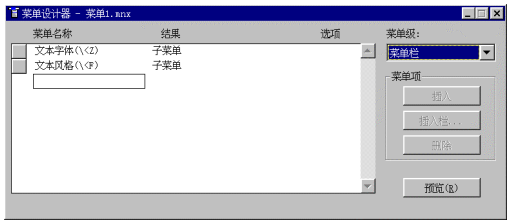


图 10-9　输入菜单名

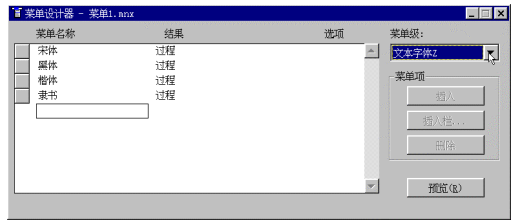


图 10-10　文本字体子菜单

依次单击“结果”右边的“创建/编辑”按钮，如图 10-12 所示，打开代码编辑器，为各菜单项输入过程代码。

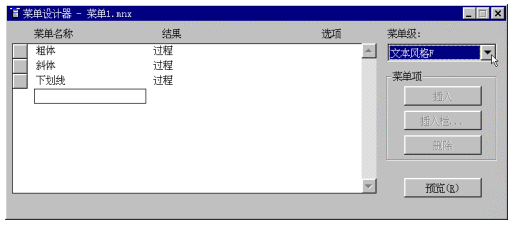


图 10-11　文本风格子菜单

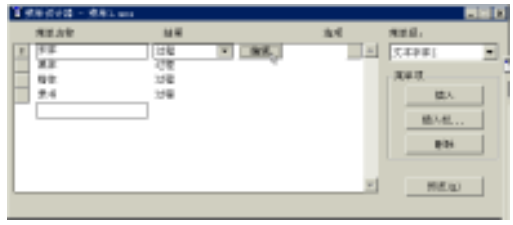


图 10-12　编辑过程代码

宋体”菜单项的过程代码：

```
_VFP.ActiveForm.Container1.Label1.FontName = "宋体"
```

“黑体”菜单项的过程代码：

```
_VFP.ActiveForm.Container1.Label1.FontName = "黑体"
```

“楷体”菜单项的过程代码：

```
_VFP.ActiveForm.Container1.Label1.FontName = "楷体_GB2312"
```

“隶书”菜单项的过程代码：

```
_VFP.ActiveForm.Container1.Label1.FontName = "隶书"
```

关闭编辑器，返回菜单设计器。

说明：代码中的_VFP**是系统变量，表示当前正在运行的** VFP **应用程序对象，**ActiveForm**则表示其中的活动表单对象。**

在“菜单级”下拉列表框中选择“菜单栏”，回到图 10-9 **中。再用鼠标单击“文本风格”子菜单的“编辑”按钮，进入“文本风格** F **”的编辑对话框。分别单击各菜单项的“创建”按钮，为其创建过程代码。**

“粗体”菜单项：

```
L = NOT _VFP.ActiveForm.Container1.Label1.FontBold
SET MARK OF BAR 1 OF "文本风格 F" L          && 为第 1 个菜单选项设置或清除标记符号
_VFP.ActiveForm.Container1.Label1.FontBold = L
```

“斜体”菜单项：

```
L = NOT _VFP.ActiveForm.Container1.Label1.FontItalic
SET MARK OF BAR 2 OF "文本风格 F" L          && 为第 2 个菜单选项设置或清除标记符号
_VFP.ActiveForm.Container1.Label1.FontItalic = L
```

“下划线”菜单项：

```
L = NOT _VFP.ActiveForm.Container1.Label1.FontUnderline
SET MARK OF BAR 3 OF "文本风格 F" L          && 为第 3 个菜单选项设置或清除标记符号
_VFP.ActiveForm.Container1.Label1.FontUnderline = L
```

（4）生成菜单码

完成菜单的定义后，选择主菜单“菜单”菜单中的“生成”命令，选择“是”，在“另存为”对话框中输入菜单名 Menu1，“确定”后显示“生成菜单”对话框，单击“生成”按钮，生成菜单程序 Menu1.mpr，至此完成菜单的创建工作。

2．修改表单

修改表单的 ShowWindow 属性为 2 – 作为顶层表单。

编写表单的 Init 事件代码：

```
DO menu1.mpr WITH THIS, .T.
```

3．运行表单

运行表单，即可修改标题板的文本字体与文本风格，如图 10-8 所示。

说明：也可以为子菜单编写通用的代码。在上例中，进入“文本字体子菜单”的编辑状态中，如图 10-10 所示。在主菜单中选择“显示”菜单中的“菜单选项”命令，打开菜单选项对话框，如图 10-4 所示，用鼠标依次单击“编辑”和“确定”按钮，打开编辑器，为“文本字体 Z”编写通用过程：

```
DO CASE
  CASE BAR（）= 1                    && 函数 BAR（）返回最近一次选择的菜单项的编号
    a = "宋体"
  CASE BAR（）= 2
    a = "黑体"
  CASE BAR（）= 3
    a = "楷体_GB2312"
  CASE BAR（）= 4
    a = "隶书"
ENDCASE
_VFP.ActiveForm.Container1.Label1.FontName = a
```

菜单运行结果是相同的。当菜单选项编有代码时，通用代码对该选项不起作用。

10.3.2 在自定义菜单中使用系统菜单项

在自定义菜单中使用系统菜单项，不仅设计出的菜单系统更为规范，而且使得菜单的设计更加简单、更加快速方便。

下面的例子，在“提示选项”对话框中为菜单项定义快捷键并设置条件。

【例 10-2】在【例 9-4】的文本编辑器中使用菜单代替命令按钮，并且使用系统菜单项设

计“编辑”子菜单，使之具有更加强大的功能，如图 10-13 所示。

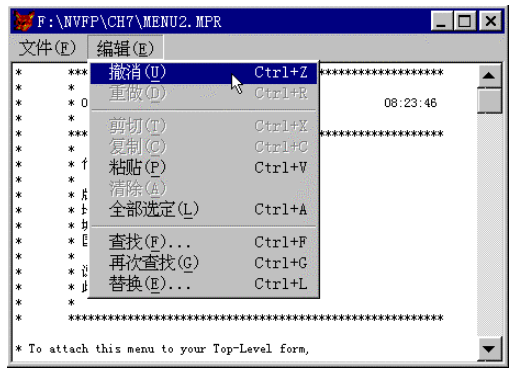


图 10-13　“编辑”菜单中的系统菜单项

设计步骤如下：

1．设计菜单

（1）规划菜单系统

本系统的菜单条由“文件”、“编辑”两项组成。其中“文件”菜单由“新建文件”、“打开文件”、“保存文件”、“文件另存’等菜单项组成；“编辑’菜单由‘剪切”、“复制”、“粘贴”、“删除”等系统菜单项组成。

（2）创建菜单和子菜单

选择“文件”菜单中的“新建”命令，单击“新建”对话框底部的“菜单”按钮，单击“新文件”按钮，打开“菜单设计器”。

首先选择“显示”菜单中的“常规选项”命令，在“常规选项”对话框中用鼠标选中“顶层表单”复选框，将菜单定位于顶层表单之中。单击“确定”按钮返回菜单设计器。

在菜单设计器中输入菜单名“文件（\<F）”和“编辑（\<E）”，如图 10-14 所示。

单击“文件”子菜单的“创建”按钮，依次输入子菜单项名，并在“选项”栏定义相应的快捷键。另外，为了增强可读性，在文件子菜单中有一项菜单名称为“\-”，结果为“菜单项 #”，将创建一条分隔线，如图 10-15 所示。

图 10-14　输入菜单名

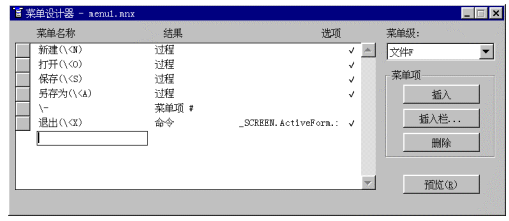


图 10-15　文件子菜单

选择“菜单级”下拉列表框中的“菜单栏”，回到图 10-14 的对话框。单击“编辑”子菜单的“创建”按钮，进入“编辑”菜单对话框。

用鼠标单击“插入栏…”按钮，打开“插入系统菜单栏”对话框，如图 10-16 所示。

依次插入所需的菜单项：撤销、重做、剪切、复制、粘贴、清除、全部选定、查找、再

次查找、替换等，适当插入一些分隔线，调整各菜单项的位置，如图 10-17 所示。

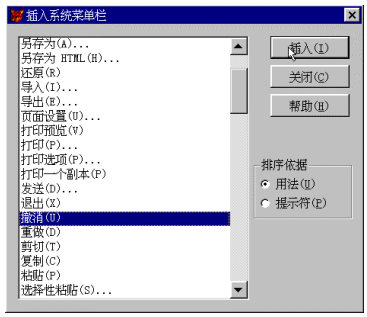


图 10-16　“ 插入系统菜单栏 ” 对话框

图 10-17　编辑子菜单

（3）编写菜单代码

在“ 菜单级 ”下拉列表框中选择“ 菜单栏 ”，回到如图 10-14 所示。在主菜单中选择“ 显示 ”菜单中的“ 菜单选项 ”命令，打开菜单选项对话框，如图 10-18 所示。

用鼠标依次单击“ 编辑 ”和“ 确定 ”按钮，打开编辑器，为菜单栏编写如下两个通用过程：

```
PROCEDURE Save0                          && 文件另存
cfile = PUTFILE（""）
IF cfile != ""
   nhandle = FCREATE（cfile,0）
   cc = FWRITE（nhandle,_VFP.ActiveForm.Edit1.Value）
   = FCLOSE（nhandle）
   IF cc < 0
      = MESSAGEBOX（'文件不能保存'）
   ELSE
      _VFP.ActiveForm.Caption = cfile
      _VFP.ActiveForm.Tag = ”
   ENDIF
ENDIF

PROCEDURE Save                           && 保存文件
cFile = _VFP.ActiveForm.Caption
IF cfile == "未命名"
   DO Save0
ELSE
   nhandle = FOPEN（cfile,1）
   = FWRITE（nhandle,_VFP.ActiveForm.Edit1.Value）
   _VFP.ActiveForm.Tag = ”
   = FCLOSE（nhandle）
ENDIF
```

关闭编辑器，返回菜单设计器后用鼠标单击“ 文件 ”子菜单的“ 编辑 ”按钮，进入“ 文件 ”子菜单对话框，如图 10-15 所示。在主菜单中选择“ 显示 ”菜单中的“ 菜单选项 ”命令，打开菜单选项对话框，如图 10-19 所示。

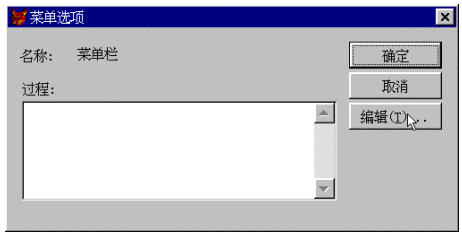


图 10-18　在菜单栏中编写通用过程

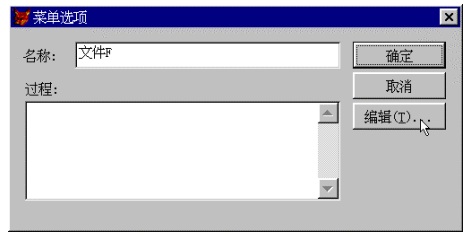


图 10-19　为“文件”菜单编写通用过程

用鼠标依次单击“编辑”和“确定”按钮，打开编辑器，为“文件”菜单编写如下通用过程：

```
DO CASE
  CASE BAR（）= 1
    IF _VFP.ActiveForm.Tag = "T"
      A = MESSAGEBOX（"是否保存编辑过的文件",4+48,"信息窗口"）
      IF A = 6
        DO Save
      ENDIF
    ENDIF
    _VFP.ActiveForm.Edit1.Value = ""
    _VFP.ActiveForm.Caption = "未命名"
  CASE BAR（）= 2
    IF _VFP.ActiveForm.Tag = 'T'
      A = MESSAGEBOX（"是否保存编辑过的文件",4+48,"信息窗口"）
      IF A = 6
        DO Save
      ENDIF
    ENDIF
    cfile = GETFILE（""）
    IF cfile != ""
      nhandle = FOPEN（cfile）
      nend = FSEEK（nhandle,0,2）
      IF nend <= 0
        = MESSAGEBOX（'文件是空的'）
      ELSE
        = FSEEK（nhandle,0,0）
        _VFP.ActiveForm.Edit1.Value = FREAD（nhandle,nend）
        _VFP.ActiveForm.Caption = cfile
        = FCLOSE（nhandle）
      ENDIF
    ENDIF
  CASE BAR（）= 3
    DO Save
  CASE BAR（）= 4
    DO Save0
ENDCASE
```

```
_VFP.ActiveForm.Refresh
_VFP.ActiveForm.Edit1.SetFocus
```

关闭编辑器，返回菜单设计器。在“退出”选项的右边填入命令：

```
_VFP.ActiveForm.Release
```

说明：“编辑”子菜单中的系统菜单项无须编写任何代码。

（4）生成菜单码

完成菜单的定义后，选择主菜单“菜单”菜单中的“生成”命令，选择“是”，在“另存为”对话框中输入菜单名 Menu1，“确定”后显示“生成菜单”对话框，单击“生成”按钮，生成菜单程序 Menu1.mpr，至此完成菜单的创建工作。

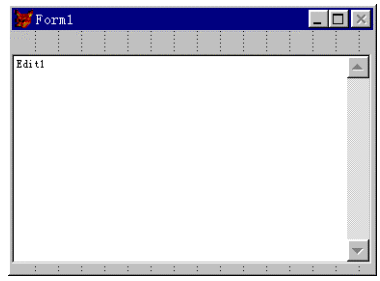


图 10-20　设计文本编辑器的界面

2．设计表单

（1）建立应用程序用户界面

选择“新建”表单，进入表单设计器，增加一个“编辑框”Edit1，如图 10-20 所示。

（2）设置对象属性

见表 10-4。

表 10-4　属性设置

对　象	属　性	属 性 值	说　明
Form1	Caption	未命名	表单的标题
	ShowWindow	2 – 作为顶层表单	

（3）编写程序代码

编写表单的事件代码：

Resize 事件：

```
WITH THIS.Edit1
   .Top = 0
   .Left = 0
   .Height = THIS.Height
   .Width = THIS.Width
ENDWITH
```

Init 事件：

```
SET EXACT ON
DO menu1.mpr WITH THIS, .T.
```

编写编辑框 Edit1 的 InteractiveChange 事件代码：

```
THISFORM.Tag = T’
```

3. 运行表单

运行表单即得到一个有相当功能的简易文本编辑器，如图 10-13 所示。

10.4 快捷方式菜单的设计

虽然下拉式菜单能够根据程序的运行情况动态地调整其可见性、有效性，也可以动态地增减菜单项，但其对用户的当前操作跟踪不够。

快捷方式菜单能以灵活的方式为用户提供更加便利的操作，它可以根据用户单击鼠标右键时的位置，动态地调整菜单项的显示位置，同时也改变菜单项显示内容，因此快捷菜单又称为“上下文菜单”。

10.4.1 快捷方式菜单的设计方法

利用系统提供的“快捷菜单设计器”可以方便地创建快捷方式菜单。与下拉式菜单相比，快捷方式菜单没有条形菜单，只有弹出式菜单。快捷方式菜单一般是一个弹出式菜单，或由几个具有上下级关系的弹出式菜单组成。

1．打开快捷菜单设计器

创建一个快捷方式菜单的步骤如下：

① 选择系统“文件”菜单中的“新建”菜单项。

② 在“新建”对话框中选择“菜单”，然后单击“新建文件”按钮，打开“新建菜单”对话框。

③ 在“新建菜单”对话框中单击“快捷菜单”按钮，打开“快捷菜单设计器”。

2．设计快捷式菜单

“快捷菜单设计器”设计快捷式菜单的方法与下拉式菜单的设计方法相似。其中需要注意的是：

① 选择“显示”菜单中的“常规选项”命令，打开“常规选项”对话框，选中“设置”复选框，在打开的“设置”代码编辑窗口输入接受当前表单对象引用的参数语句：

```
PARAMETERS mref
```

就可以在菜单的代码中以 mref 为名引用当前表单了。

② 选择“显示”菜单中的“常规选项”命令，打开“常规选项”对话框，选中“清理”复选框，在打开的“清理”代码编辑窗口输入清理代码，使得在选择、执行菜单命令后能及时清除菜单，释放其所占用的内存空间。其代码格式为：

```
RELEASE POPUPS 〈快捷菜单名〉
```

10.4.2 快捷方式菜单的使用

【例 10-3】如图 10-21 所示，在【例 10-1】中使用快捷方式菜单来改变标题板中文本的字体。

设计步骤如下：

1．设计菜单

（1）规划快捷菜单

规划的快捷菜单见表 10-4。

表 10-4　菜单项的设置

菜 单 名 称	结　　果	菜　单　级
宋体	过程	快捷菜单
黑体	过程	快捷菜单
楷体	过程	快捷菜单
隶书	过程	快捷菜单

（2）创建快捷菜单

选择“文件”菜单中的“新建”命令，单击“新建”对话框底部的“菜单”按钮，单击“新文件”按钮，在打开的“新建菜单”对话框中，单击“快捷菜单”按钮，打开“快捷菜单设计器”。

按照菜单规划，输入菜单选项的名称，并设置“结果”为“过程”，如图 10-22 所示。

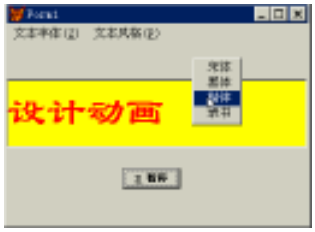


图 10-21　使用快捷菜单

图 10-22　输入菜单选项的名称

（3）编写代码

依次单击“结果”右边的“创建/编辑”按钮，打开代码编辑器，为各菜单项输入过程代码。

“宋体”菜单项的过程代码：

```
frm.ActiveForm.Container1.Label1.FontName = "宋体"
```

“黑体”菜单项的过程代码：

```
frm.ActiveForm.Container1.Label1.FontName = "黑体"
```

“楷体”菜单项的过程代码：

```
frm.ActiveForm.Container1.Label1.FontName = "楷体_GB2312"
```

“隶书”菜单项的过程代码：

```
frm.ActiveForm.Container1.Label1.FontName = "隶书"
```

关闭编辑器，返回菜单设计器。

选择系统菜单“显示”菜单中的“常规选项”命令，打开“常规选项”对话框。选中“设

置”复选框，如图 10-23 所示。单击“确定”按钮，打开编辑器窗口。在其中输入“设置”代码：

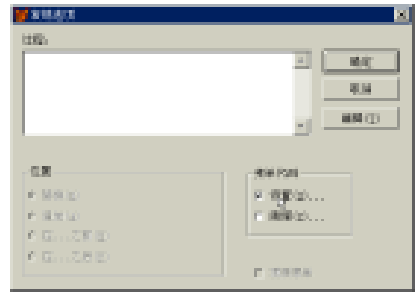


图 10-23　选中“设置”复选框

```
PARAMETERS frm
```

关闭编辑器，返回“快捷菜单设计器”。

然后选择“显示”菜单中的“常规选项”命令，打开“常规选项”对话框。选中“清理”复选框，单击“确定”按钮，打开编辑器窗口。在其中输入“清理”代码：

```
release popups 菜单 2
```

关闭编辑器，返回“快捷菜单设计器”。

（4）生成菜单码

完成菜单的定义后，选择主菜单“菜单”菜单中的“生成”命令，选择“是”，在“另存为”对话框中输入菜单名菜单 2，“确定”后显示“生成菜单”对话框，单击“生成”按钮，生成菜单程序菜单 2.mpr，至此完成菜单的创建工作。

2．修改表单

编写表单的 RightClick 事件代码：

```
DO 菜单 2.mpr WITH THIS, .T.
```

3．运行表单

运行表单，即可使用快捷菜单来修改标题板的文本字体，如图 10-21 所示。

说明：也可以为子菜单编写通用的代码。上例步骤（3）中，在如图 10-22 所示的快捷菜单设计器中。选择“显示”菜单中的“菜单选项”命令，打开“菜单选项”对话框。单击“编辑”按钮，如图 10-24 所示，然后单击“确定”按钮，打开编辑器窗口。在其中输入菜单过程代码：

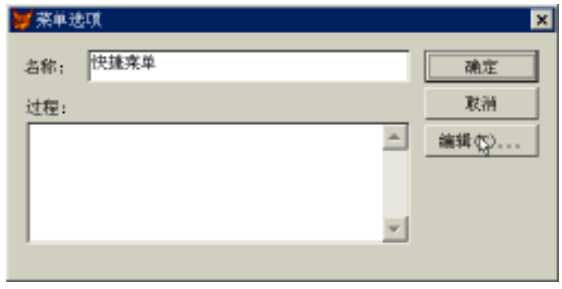


图 10-24　单击“编辑”按钮

```
DO CASE
  CASE BAR（）= 1
    a = "宋体"
  CASE BAR（）= 2
    a = "黑体"
  CASE BAR（）= 3
    a = "楷体_GB2312"
```

```
    CASE BAR（）= 4
       a = "隶书"
    ENDCASE
    frm.Container1.Label1.FontName = a
```

关闭编辑器，返回“快捷菜单设计器”即可，无需为每个菜单项编写代码。

10.5 习题

一、选择题

1．假设已经生成了名为 mymenu 的菜单文件，执行该菜单文件的命令是（　　）。

A）DO mymenu　　B）DO mymenu.mpr

C）DO mymenu.pjx　　D）DO mymenu.mnx

2．使用 VFP 的菜单设计器时，选中菜单项之后，如果设计其子菜单，应在“结果”中选择（　　）。

A）命令　　B）填充名称　　C）子菜单　　D）过程

二、填空题

1．典型的菜单系统一般是一个下拉式菜单，下拉式菜单通常由一个_____和一组_____组成。

2．要为表单设计下拉式菜单，首先需要在菜单设计时，在_____对话框中选择“顶层表单”复选框；其次要将表单的_____属性值设置为 2，使其成为顶层表单；最后需要在表单的_____事件代码中设置调用菜单程序的命令。

3．快捷菜单实质上是一个弹出式菜单。要将某个弹出式菜单作为一个对象的快捷菜单，通常是在对象的_____事件代码中添加调用该弹出式菜单程序的命令。

三、上机题

1．设计一个下拉菜单，如图 10-25 所示。各菜单选项的功能如下：

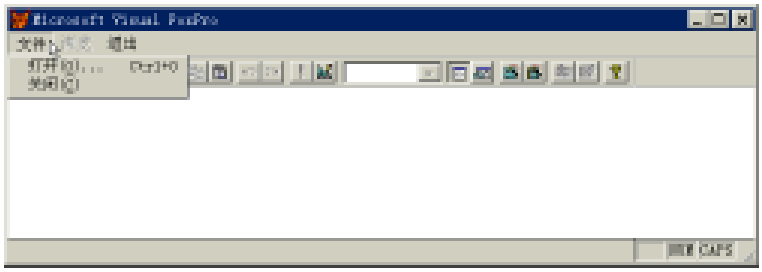


图 10-25　设计一个下拉菜单

1）“打开”选项用的是标准的系统菜单命令，可以调出“打开”对话框，打开一个文件。

2）“关闭”选项在当前工作区有表打开时有效。

3）“浏览”选项在当前工作区有表打开时有效，它用 BROWSE 命令浏览当前表的内容。

4）“退出”选项的功能是恢复标准的系统菜单。

2．为表单中的一个组合框设计一个快捷菜单，如图 10-26 所示。各菜单选项的具体功能如下：

1）选择“表文件名”选项时，组合框内显示表文件名列表。

图 10-26　设计一个快捷菜单

2）选择“学生表字段”和“选课表字段”选项时，组合框内分别显示学生表字段名列表和选课表字段名列表。

3）选择“组合/列表框”选项，组合框在下拉列表框和下拉组合框之间切换。

3．将第 1 题中设计的下拉菜单添加到一个表单里。

第 11 章　报　　表

报表设计是程序开发的一个重要组成部分。在数据库应用系统中，经常需要将数据以报表形式打印出来。

通过设计报表，可以用各种方式在打印页面上显示数据。设计报表有 4 个主要步骤：① 决定要创建的报表类型。② 创建报表布局文件。③ 修改和定制布局文件。④ 预览和打印报表。

11.1　数据源和报表布局

报表包括两个基本组成部分：数据源和布局。数据源通常是数据库中的表，也可以是视图、查询或临时表。视图和查询将筛选、排序、分组数据库中的数据，而报表布局定义了报表的打印格式。设计报表就是根据报表的数据源和应用需要来设计报表的布局。

11.1.1　决定报表的常规布局

创建报表之前，应该确定所需报表的格式。报表可能同基于单表的电话号码列表一样简单，或者复杂得像基于多表的发票那样。也可以创建特殊种类的报表。例如，邮件标签便是一种特殊的报表，其布局必须满足专用纸张的要求。如图 11-1 所示为常规报表布局。

图 11-1　常规报表布局

为帮助选择布局，在表 11-1 中给出常规布局的一些说明、它们的一般用途及示例。

表 11-1　常规布局说明、一般用途及示例

布 局 类 型	说　　明	示　　例
列	每行一条记录，每条记录的字段在页面上按水平方向放置	分组/总计报表、财政报表、存货清单
行	一列的记录，每条记录的字段在一侧竖直放置	列表
一对多	一条记录或一对多关系	发票、会计报表
多列	多列的记录，每条记录的字段沿左边缘竖直放置	电话号码薄、名片
标签	多列记录，每条记录的字段沿左边缘竖直放置，打印在特殊纸上	邮件标签、名字标签

当选定了满足需求的常规报表布局后，便可以创建报表布局文件。

11.1.2　报表布局文件

报表布局文件具有.FRX 文件扩展名，它存储报表的详细说明。每个报表文件还有具有.FRT 文件扩展名的相关文件。报表文件指定了想要的域控件、要打印的文本以及信息在页

面上的位置。若要在页面上打印数据库中的一些信息，可通过打印报表文件达到目的。报表文件不存储每个数据字段的值，只存储一个特定报表的位置和格式信息。每次运行报表，值都可能不同，这取决于报表文件所用数据源的字段内容的更改。

11.1.3 本章所涉及的数据源

本章使用第 4 章所创建的数据库学生管理和订单管理中的数据表作为数据源。

11.2 创建报表布局

在 VFP 中，有三种创建报表布局的方法：用“快速报表”从单表中创建一个简单报表，用“报表向导”创建简单的单表或多表报表，用“报表设计器”修改已有的报表或创建自己的报表。

以上每种方法创建的报表布局文件都可以用“报表设计器”进行修改。“报表向导”是创建报表的最简单途径，它自动提供很多“报表设计器”的定制功能。“快速报表”是创建简单布局的最迅速途径。如果直接在“报表设计器”内创建报表，“报表设计器”会提供一个空白布局。

11.2.1 快速报表

“快速报表”是一项省时的功能，可以创建一个格式极简单的报表。通常先使用快速报表功能来创建一个简单报表，然后在此基础上做修改，达到快速构造的目的。

下面通过实例说明创建快速报表的操作步骤。

【例 11-1】利用 VFP 的快速报表功能建立一个简单报表，报表的内容是 student 表的记录（全部记录，横向）。

操作步骤如下：

① 单击工具栏上的“新建”按钮，在弹出的“新建”对话框中选择“报表”，并单击“新建文件”按钮，自动打开报表设计器，并出现一个空白报表。

② 选择主菜单中“报表”菜单，从中选择“快速报表”选项，在“打开”对话框中选择数据源 student.dbf，并单击“确定”按钮。

③ 系统弹出“快速报表”对话框，如图 11-2 所示。

图 11-2 “快速报表”对话框

对话框中主要选项的功能如下：

字段布局：对话框中两个较大的按钮用于设计报表的字段布局，单击左侧按钮产生字段横向排列的报表，如果单击右侧的按钮，则产生字段竖向排列的报表。

“标题”复选框：选中复选框，表示在报表中为每一个字段添加一个字段名标题，否则，没有标题。

“添加别名”复选框：选中复选框，表示在字段前面添加表的别名。由于数据源是一个表，别名无实际意义，一般此项不选。

“将表添加到数据环境中”复选框，表示把打开的表文件添加到报表的数据环境中作为报表的数据源。

“字段”按钮：单击字段按钮，打开“字段选项器”为报表选择可用的字段，如图 11-3 所示。在缺省情况下，快速报表选择表文件中除通用字段以外的所有字段。单击“确定”按钮，关闭“字段选择器”返回“快速报表”对话框。

④ 在“快速报表”对话框中，单击“确定”按钮，快速报表便出现在“报表设计器”中，如图 11-4 所示。

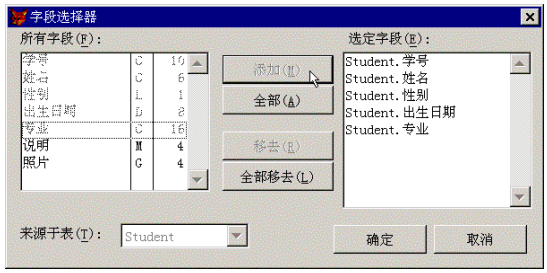


图 11-3 “字段选项器”对话框

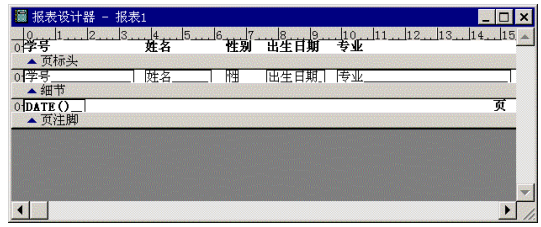


图 11-4 “报表设计器”窗口

⑤ 单击工具栏上的“打印预览”图标按钮，或者从“显示”菜单下选择“预览”命令，打开快速报表的预览窗口，如图 11-5 所示。

图 11-5 预览报表

⑥ 单击工具栏上的“保存”按钮，将该报表保存为默认的报表文件报表 1.frx。

11.2.2 使用向导创建报表

无论何时想创建报表，都可以使用“报表向导”。向导将提出一系列问题并根据回答创建报表布局。VFP 提供的报表向导有报表和一对多报表。应根据常规布局和报表的复杂程度选择向导。

使用报表向导可以建立两种类型的报表：每列一个字段，每行一条记录，字段名作为列标题放在每列的顶部，相当于“快速报表”中的“横向排列”；另一种报表是记录一个接一个列出，每个字段的左边是相应的字段名，相当于“快速报表”中的“竖向排列”。

1．启动“报表向导”

有多种启动“报表向导”的方法：

① 在“项目管理器”的“文档”选项卡中，选定“报表”，单击“新建”按钮，如图 11-6 的左图所示，在弹出的“新建报表”对话框中选择“报表向导”，如图 11-6 的右图所示。

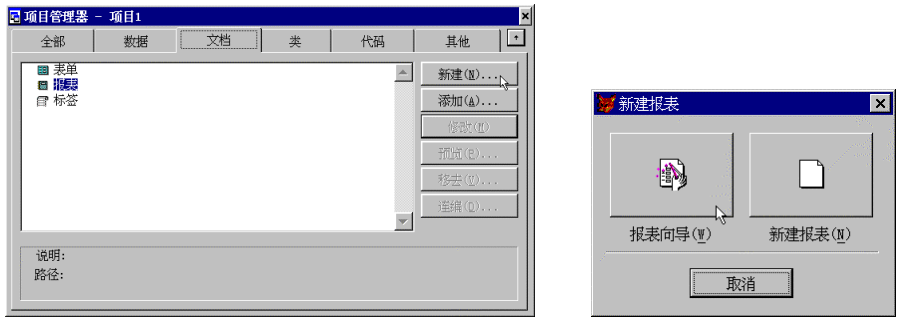


图 11-6 “项目管理器”的“文档”选项卡

② 在“文件”菜单中选择“新建”或直接单击“新建”按钮，在弹出的“新建”对话框中选择“报表”，然后单击“向导”按钮。

③ 在“工具”菜单中选择“向导”子菜单，然后选择“报表”菜单项。

④ 直接单击工具栏上的“报表向导”图标按钮。

上述操作都将打开“向导选取”对话框，如图 11-7 所示。

图 11-7 “向导选取”对话框

根据需要选取“报表向导”或“一对多报表向导”，即可启动相应的报表向导。

2．使用“报表向导”

【例 11-2】利用 VFP 的报表向导建立一个简单报表，要求选择 student 表中学号、姓名、性别、出生日期和专业等字段；记录按专业分组；报表样式为“随意式”；列数为 1，字段部局为“列”，方向为“纵向”；排序字段为“学号”（升序）；报表标题为“学生基本情况一览表”；报表文件名为 print1。

操作步骤如下：

首先启动“报表向导”，然后按向导的提示操作。

步骤 1–字段选取：本例选择 student 表中学号、姓名、性别、出生日期和专业等字段。

由于选择的是“报表向导”，只能从单个表或视图中选择输出到报表中的字段，如图 11-8 的左图所示。单击“数据库/表”框右边的“…”按钮，选择需要的表。通过字段选择器，选择报表中需要的字段和在报表中排列的顺序。

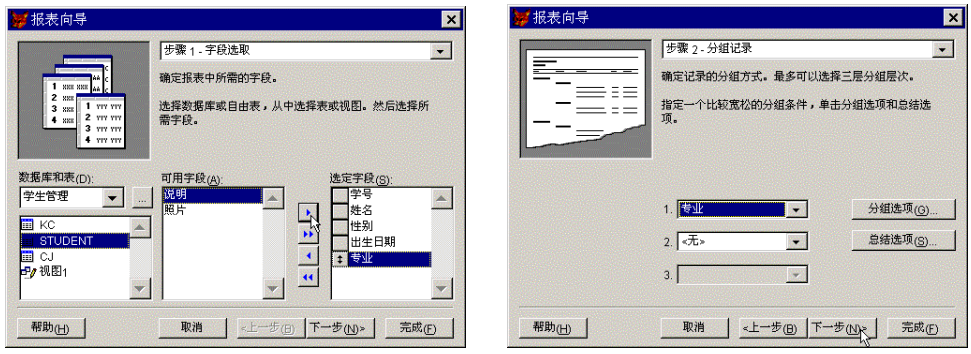


图 11-8　“ 字段选取 ” 与 “ 分组记录 ” 对话框

步骤 2 – 分组记录：本例选择按专业字段分组。

可以使用数据分组来分类并排序字段，这样能够方便读取。如图 11-8 所示，在某个 “ 分组类型 ” 框中选择了一个字段之后，可以选取 “ 分组选项 ” 和 “ 总结选项 ” 来进一步完善分组设置。

选择 “ 分组选项 ” 后将打开 “ 分组间隔 ” 对话框，从中可以选择与用来分组的字段中所含的数据类型相关的筛选级别。

选择 “ 总结选项 ” 将打开一个新的对话框，可以利用计算类型来处理数值型字段。

步骤 3 – 选择报表样式：本例选择 “ 随意式 ”。

当选择报表的任何一种样式时，向导都在放大镜中更新成该样式的示例图片，如图 11-9 的左图所示。

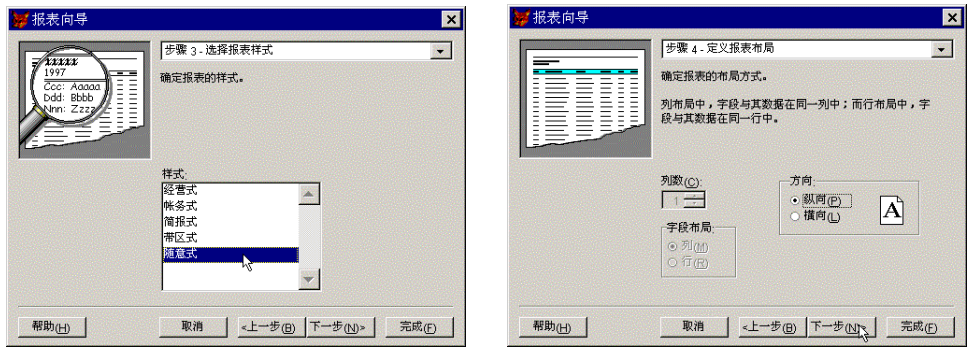


图 11-9　“ 选择报表样式 ” 与 “ 定义报表布局 ” 对话框

步骤 4 – 定义报表布局：本例不用选择。

如图 11-9 的右图所示，选择报表的版面布局。在指定列数或布局时，向导将在放大镜中更新成选定布局的实例图形。如果在步骤 2 中指定分组选项，则本步骤中的 “ 列数 ” 和 “ 字段布局 ” 选项不可用。

列数：设置每行放置几个记录，即分栏数。如果栏数不合适，将会造成重叠。

字段布局有两种排列方式：

- **列：按列排列，每页的字段名称在每列的顶部，每个记录占一行。**
- **行：按行排列，字段一个接一个，字段名在字段的左边。**

方向：用来设置纸张是竖放（纵向），还是横放（横向）。

由于版面样式、字段的多少、字段的宽度等原因，会造成字段内容重叠，要想看到所设计的最后效果，在最后一步中选择“预览”钮。若对设计的报表不满意，返回上一步修改。

步骤 5 - 排序记录：本例选择“学号”字段。

如图 11-10 的左图所示。按照视图查询结果排序的顺序选择字段，若要按原始顺序排列可不选择，直接按“下一步”。

步骤 6 - 完成：本例输入报表标题“学生基本情况一览表”。

如图 11-10 右所示是向导的最后一个对话框。在“报表标题”文本框中输入报表的标题。

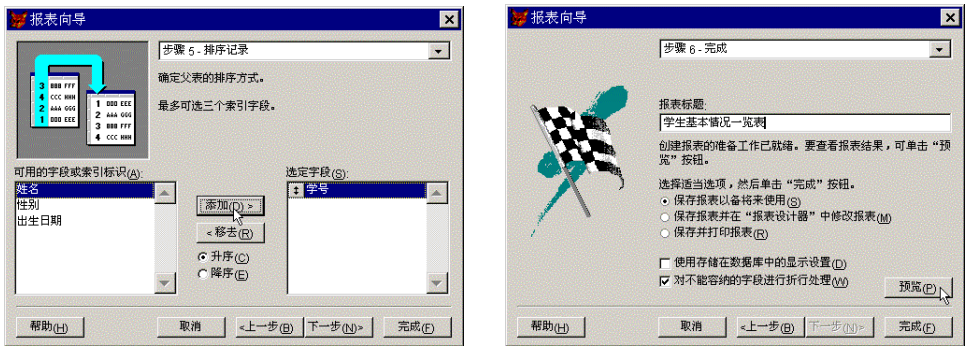


图 11-10 “排序记录”与“完成”对话框

由于报表的制作和纸张尺寸、报表式样、字段等有关，选择不当会出现报表超宽的情况。如果选定数目的字段不能放置在报表中单行指定宽度之内，字段将换到下一行上。如果不希望字段换行，清除“对不能容纳的字段进行折行处理”选项。为了不丢失数据，一般应选择本框。因此，应该合理地设计报表格式。

在完成对话框中单击“预览”按钮，刚才制作的报表将显示出来，如图 11-11 所示。若不满意可返回前面的步骤，或者调整纸张的尺寸及放置方向、或者删去那些可要可不要的字段。单击预览窗口可改变显示比例。

图 11-11 预览报表

使用了向导之后，就可以使用“报表设计器”来添加控件和定制报表。

执行前面几步骤后，单击“完成”按钮，将报表文件保存为 print1.frx，即可完成报表的设计。

11.2.3 启动“报表设计器”

如果不想使用“报表向导”或“快速报表”，也可以使用“报表设计器”从空白报表布局开始，然后添加控件来设计报表。启动“报表设计器”的步骤如下：

① 在“项目管理器”的“文档”选项卡中，选定“报表”，如图 11-6 的左图所示。

② 单击“新建”按钮。

③ 在如图 11-6 的右图所示的对话框中，单击“新建报表”按钮。此时显示“报表设计器”。可以使用“报表设计器”的任何功能来添加控件和定制报表。

11.3 设计报表

通过前面的学习，我们已经会制作出一些简单报表，但在实际应用中，还需要设计更为复杂的报表或对简单报表进行各种修饰和修改。VFP 所提供的报表设计器可以完成所有这些工作：设置报表数据源、更改报表的布局、添加报表的控件和设计数据分组等。

11.3.1 报表工具栏

与报表设计有关的工具栏主要包括“报表设计器”和“报表控件”两个工具栏。单击“显示”菜单中的“工具栏”项，在弹出的“工具栏”对话框中可以设置显示或隐藏相应的工具栏。

1．报表设计器工具栏

报表设计器工具栏中共有 5 个工具按钮，其功能说明见表 11-2。

表 11-2 报表设计器工具栏

图 标	名 称	说 明
	数据分组	在报表设计过程中，单击此按钮，显示“数据分组”对话框，用于创建数据分组并指定其属性
	数据环境	在报表设计过程中，单击此按钮，显示“数据环境设计器”窗口，可以结合用户界面同时设计一个依附的数据环境
	报表控件工具栏	在报表设计过程中，单击此按钮，可以启动或关闭报表控件工具栏，以便于利用各控件进行用户界面的设计
	调色板工具栏	在报表设计过程中，单击此按钮，可以启动或关闭调色板工具栏。利用调色板工具栏可以进行各对象前景与背景颜色的设置
	布局工具栏	在报表设计过程中，单击此按钮，可以启动或关闭布局工具栏。利用布局工具栏可以针对对象进行位置配置和对齐设置

2．报表控件工具栏

报表控件工具栏中共有 8 个工具按钮，其功能说明见表 11-3。

表 11-3　报表控件工具栏

图　标	名　　称	功 能 说 明
	选定对象	移动或更改控件的大小，在创建一个控件后，系统将自动选定该按钮，除非选中“ 按钮锁定 ” 按钮
	标签	在报表上创建一个标签控件，用于显示与记录无关的字符文本，例如标题
	域控件	在报表上创建一个域控件，用于显示字段、内存变量或其他表达式的内容
	线条	用于在报表上绘制垂直或水平直线
	矩形	用于在报表上绘制矩形
	圆角矩形	用于在报表上绘制圆角矩形
	图片/ActiveX 绑定控件	用于在报表上添加图片或包含 OLE 对象的通用型字段
	按钮锁定	用于添加多个同类型控件而不需要多次选中该按钮

11.3.2　报表的数据源

报表输出的数据来自于相关的表或视图，如果一个报表总是使用相同的数据源，就可以把数据添加到报表的数据环境中。当数据源中的数据更新之后，使用同一报表文件打印的报表将反映新的数据同内容，而报表的格式不变。

数据环境通过下列方式管理报表的数据源：打开或运行报表时打开表或视图，基于相关表或视图收集报表所需数据集合，关闭或释放报表时关闭表。

1．向数据环境中添加表或视图

向数据环境中添加表或视图的步骤如下：

① 从“ 显示 ” 菜单中，选择“ 数据环境 ”，打开“ 数据环境设计器 ”。

② 用鼠标右键单击“ 数据环境设计器 ”，从弹出的快捷菜单中选择“ 添加 ” 命令，如图 11-12 的左图所示。

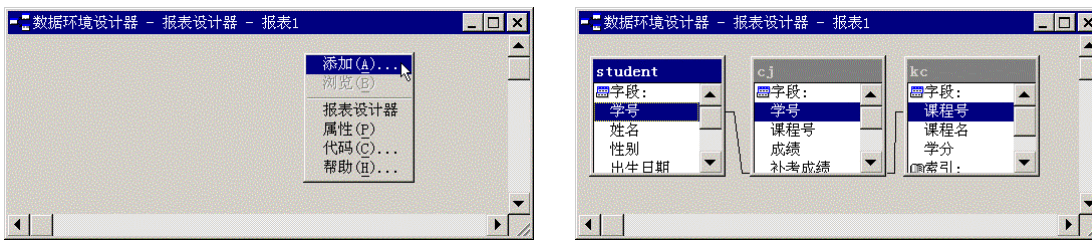


图 11-12　在“ 数据环境 ” 中添加表或视图

③ 在弹出“ 打开 ”和“ 添加表或视图 ”对话框中，依次选定表或视图，然后单击“ 添加 ” 按钮。

④ 最后单击“ 确定 ” 按钮，返回“ 数据环境 ” 设计器，如图 11-13 的右图所示。

2．为数据环境设置索引

为数据环境设置索引，可设置出现在报表中的记录顺序。为数据环境设置索引的步骤为：

① 从“ 显示 ” 菜单中，选择“ 数据环境 ” 命令。

② 用鼠标右键单击“数据环境设计器”，从弹出的快捷菜单中选择“属性”命令，如图 11-13 的左图所示。

③ 在打开的“属性”窗口中，选择“对象”框中的“Cursor1”。

④ 选择“数据”选项卡，然后选定“Order”属性。

⑤ 输入索引名。或者，从可用索引列表中选定一个索引，如图 11-13 的右图所示。

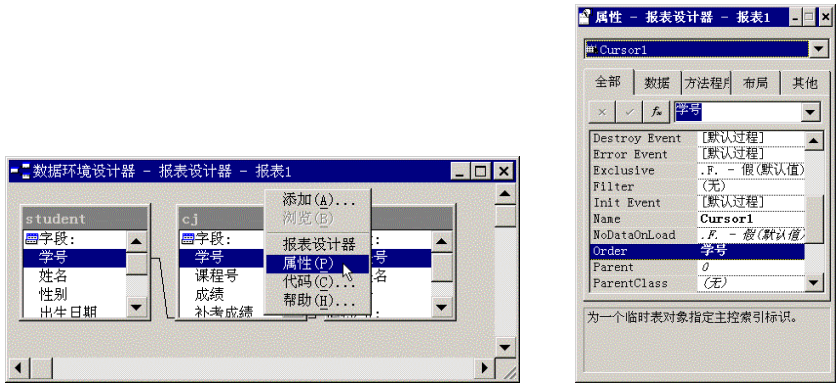


图 11-13　为数据环境设置索引

11.3.3　报表布局

一个良好的报表会把数据放在报表合适的位置上。在报表设计器中，报表包括若干个带区，带区的作用主要是控制数据在页面上的打印位置。在打印或预览报表时，系统会以不同的方式处理各个带区的数据。

1．报表的带区

报表中可能包含的一些带区以及每个带区的典型内容，如图 11-14 所示，注意每个带区下的栏标识了该带区。

使用“报表设计器”内的带区，可以控制数据在页面上的打印位置。表 11-4 列出了报表的一些常用带区以及使用情况。

表 11-4　常用的带区

带　区	打　印	使 用 命 令
标题	每报表一次	从“报表”菜单中选择“标题/总结”带区
页面标头	每页面一次	默认可用
列标头	每列一次	从“文件”菜单中选择“页面设置”设置“列数”>1
组标头	每组一次	从“报表”菜单中选择“数据分组”
细节带区	每记录一次	默认可用
组注脚	每组一次	从“报表”菜单中选择“数据分组”
列注脚	每列一次	从“文件”菜单中选择“页面设置”设置“列数”>1
页面注脚	每页面一次	默认可用
总结	每报表一次	从“报表”菜单中选择“标题/总结”带区

在“报表设计器”的带区中，可以插入各种控件，它们包含打印的报表中想要的标签、字段、变量和表达式。要增强报表的视觉效果和可读性，还可以添加直线、矩形以及圆角矩形等控件。也可以包含图片或 OLE 绑定型控件。

使用报表带区可以决定报表的每页、分组及开始与结尾的样式。可以调整报表带区的大小。在报表带区内，添加报表控件，然后移动、复制、调整大小、对齐以及调整它们，从而安排报表中的文本和域控件。

报表也可能有多个分组带区或者多个列标头和注脚带区。使用本章稍后的定义报表的页面和在布局上分组数据部分中提供的过程，可以添加这些带区。

可以在任何带区中设置任何“报表”控件。也可以添加运行报表时执行的用户自定义函数。

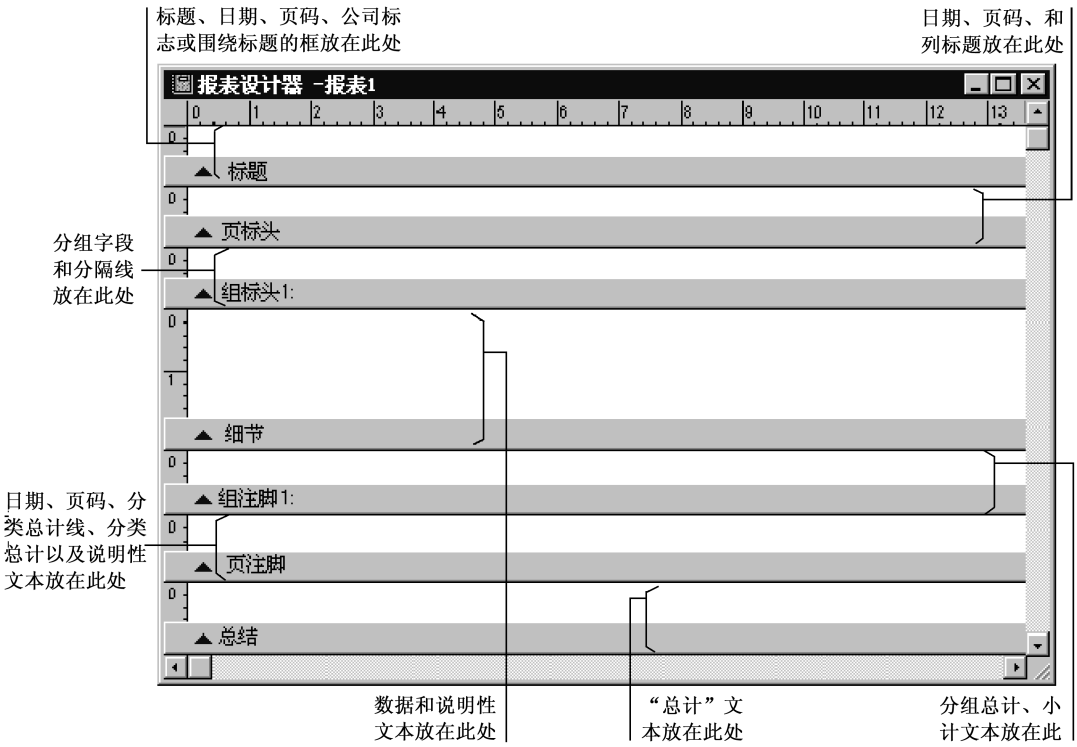


图 11-14 “报表设计器”中的报表带区

2．添加与调整报表带区

可以添加或调整报表带区。

① 在“报表设计器”中，可以添加所需要的带区。添加带区的方法见表 11-4 的右栏所示的“使用命令”。例如，单击“报表”菜单中的“标题/总结”菜单项，打开“标题/总结”对话框，在其中选定“标题带区”复选框，如图 11-15 所示。然后单击“确定”按钮，即可在报表中添加一个“标题带区”。

② 可以修改每个带区的尺寸和特征。例如，在带区中添加了控件后，带区高度不够，可以调整带区的高度以适应所放置的控件。

使用左侧标尺作为指导。标尺量度仅指带区高度，不包含页边距。不能使带区高度小于布局中控件的高度。可以把控件移进带区内，然后减少其高度。

调整带区高度的一种方法是用鼠标选中某一带区标识栏，然后上下拖动该带区，直至得到满意的高度为止。另一种方法是双击需要调整高度的带区的标识栏，系统将显示一个对话框。例如，双击“标题”带区的标识栏，系统弹出“标题”对话框，如图 11-16 所示。在对话框中直接输入所需的高度，或调整“高度”微调器中的数值即可。选中“带区高度保持不变”复选框，可以防止报表带区因容纳过长的数据或从中移去数据而改变其高度。

图 11-15 “标题/总结”对话框

图 11-16 调整带区的高度

11.3.4 报表中的控件使用

从面向对象的角度来看，报表可看成由诸多控件组合而成。因此，对报表的设计主要也是对控件及其布局的设计。这里还需要说明：

① 可以在任何带区加入任何报表控件。

② 相同的报表控件安置在不同的带区时，其输出效果也不一样，故使用带区可以控制数据在页面上的打印位置。

③ 可以调整带区大小，但不能使带区高度小于其内控件的高度。

④ 可以有多对组标头与组注脚带区。

1．标签控件

标签控件在报表的使用中相当广泛。例如，每一个报表都有一个标题、每一个字段前都要有说明性文字等，这些标题或说明性文字就是使用标签控件来实现的。

使用标签控件的操作很简单，只要在“报表控件”工具栏中单击“标签”控件，然后在报表指定位置上单击鼠标，便出现一个插入点光标，即可在当前位置上输入文本。

可以更改标签控件中文本的字体和大小，只要选定标签控件，从“格式”菜单中选择“字体”菜单项，在打开的“字体”对话框中选择适当的字体和大小。

若要更改标签控件的默认字体，可从“报表”菜单中选择“默认字体”菜单项，在打开的“字体”对话框中选择适当的字体和大小。

【例 11-3】为【例 11-1】中的报表添加一个“标题带区”，然后在该带区中放置一个标签控件，该标签控件显示报表的标题“学生情况表”。

操作步骤如下：

① 打开【例 11-1】中的报表文件报表 1.frx，系统自动打开报表设计器。

② 单击“报表”菜单中的“标题/总结”菜单项，打开“标题/总结”对话框。

③ 选中“标题带区”复选框，如图 11-15 所示，然后单击“确定”按钮。即可在报表中添加一个“标题带区”。

④ 在“标题带区”中添加一个标签控件，在光标处输入学生情况表。

⑤ 单击“格式”菜单中的“字体”菜单项，在打开的“字体”对话框中选择适当的字体及大小。

图11-17　添加一个标题

⑥ 重新选定该标签控件，单击“格式”菜单中的“文本对齐方式”子菜单，选择“居中”。修改后的报表如图 11-17 所示。

2．线条、矩形、圆角矩形控件

报表控件中线条、矩形、圆角矩形等控件的生成和设置比较简单，只需注意以下几点：

① 单击所需要的控件，在相应的带区拖动鼠标即可生成控件。

② 生成控件后，选中控件拖动鼠标可移动控件的位置。

③ 双击控件即可打开控件对话框，对控件进行相应的设置，包括布局、属性等。

注意：在报表中生成多个控件，这些控件有时按一点的布局排列，这时我们可以用布局工具栏对控件进行设置，布局的含义与表单上使用布局工具栏一样，可以参阅前面章节关于布局工具栏的使用。

3．域控件

域控件又称为表达式控件，用来表示表中字段、内存变量、系统变量、报表变量、表达式等计算结果。在报表中添加域控件有两种方法：一是从数据环境中添加，二是直接插入域控件。

若要从数据环境中添加表中字段，打开报表的数据环境，选择表或视图，拖放字段到布局上。

若要从工具栏添加表中字段，可按如下步骤操作：

① 从“报表控件”工具栏中，插入一个“域控件”。

② 在“报表表达式”对话框中，选择“表达式”框后的对话按钮。

③ 在“字段”框，双击所需的字段名。表名和字段名将出现在“报表字段的表达式”内。如果“字段”框为空，则应该向数据环境添加表或视图。不必保持表达式中表的别名。可以删除它或者清除“表达式生成器”对话框选项。

④ 单击“确定”按钮。

⑤ 在“报表表达式”对话框中，单击“确定”按钮。

4．报表表达式

当在报表中插入一个域控件之后，系统将打开“报表表达式”对话框，如图 11-18 所示。其各部分含义如下：

“表达式”文本框：用于键入表达式，也可通过其右侧的对话按钮打开表达式生成器来设置表达式。

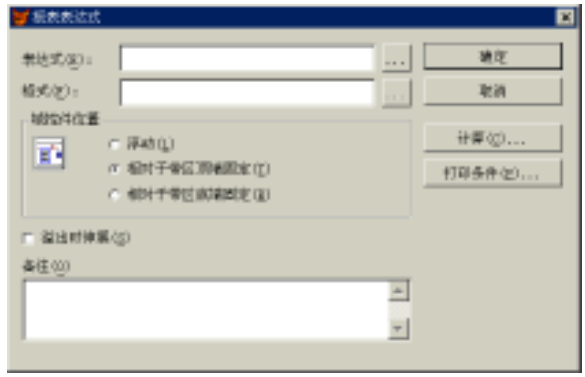


图 11-18　表达式对话框

“格式”文本框：用于为表达式键入输出格式符，也可通过其右侧的对话按钮打开格式对话框来指定格式。

“计算”按钮：用于打开计算字段对话框，以便为控件指定统计类型和范围，下文第 2 点详细说明。

“打印条件”按钮：用于打开打印条件对话框，以便为控件指定打印的时机，下文第 3 点详细说明。

“溢出时伸展”复选框：可用于数据的折行打印。当数据长于字段控件宽度时，多余部分能在垂直方向向下延伸打印。

5．格式对话框

插入“域控件”后，可以更改该控件的数据类型和打印格式。数据类型包括字符型、数值型和日期型，每一种数据类型都有自己的格式选项。在报表表达式中单击“格式”文本框右边的“...”按钮，打开“格式”对话框，如图 11-19 所示。在其中可以进行数据的格式设置。

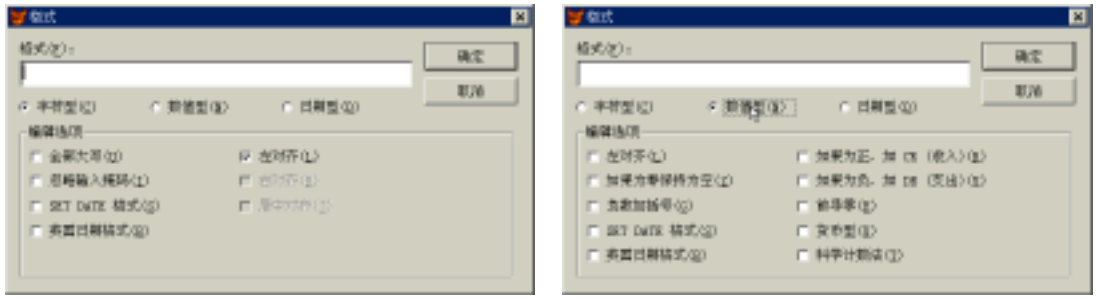


图 11-19　格式对话框

6．计算字段对话框

当在表达式对话框中单击“计算”按钮会弹出如图 11-20 所示的计算对话框，它允许为控件选择一项统计，主要由下面两部分组成。

（1）重置组合框

该组合框用于选定控件计算的复零时刻，包括的选项如下：

报表尾：此为默认值，表示在报表打印结束时将控件计算复零。

页尾：表示在报表每页打印结束时将控件计算复零。假定一个多页报表的页注脚带区已设置了价格字段控件，若要求每页价格分别小计，应选择页尾重置；如要求价格累计到底则应选报表尾重置。

列尾：在多列打印中，表示每一列打印结束时将控件计算复零。数据分组后分组表达式会自动添入重置组合框中，这一选项能使控件计算在组值变化时复零。

图 11-20　计算对话框

（2）计算区

该区包括 8 个选项按钮，用户可为控件从中选定一项要执行的计算，也可指定不进行不计算。

不计算：对控件不进行计算，直接打印表达式值。此为默认选项。

计数：用于计算并返回表达式出现的次数，此时不返回表达式的值。例如 DATE（）表达式放置在细节带区将根据表的记录数从 1 开始依次打印，放在页注脚带区则打印最大记录数。

总和：用于计算表达式值的总和。

平均值：用于计算表达式的算术平均值。

最小值：用于求表达式的最小值。

最大值：用于求表达式的最大值。

标准误差：用于计算表达式的方差的平方根。

方差：用于衡量各表达式值与平均值的偏离程序。

上述计算可用于整个报表、每组、每页或每列，计算范围与重置框中的选择有关。

7．打印条件对话框

在表达式对话框中单击“ 打印条件 ”按钮即可弹出如图 11-21 所示的对话框，说明如下：

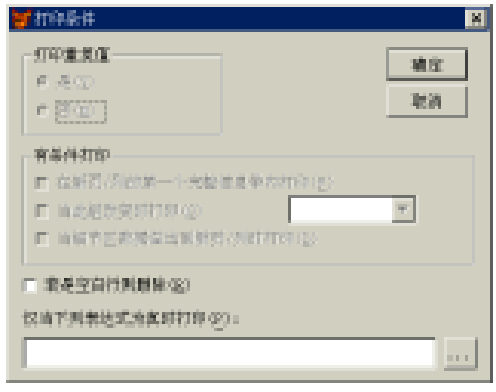


图 11-21　打印条件对话框

①“ 打印重复值 ”区：选定“ 是 ”选项按钮表示控件总是打印，此为默认状态。选定“ 否 ”选项按钮仅当控件值改变时才会打印，即不打印重复值，打印位置将留空。

②“ 有条件打印 ”区：“ 在新页/列的第一个完整信息带内打印 ”复选框用于指定在新页或新列的第一个完整信息带内打印，而不是在前一页或前一列的信息带内接着打印。

选定“当此组改变时打印”复选框后，再在其右边的组合框中选出一个组，则当组值改变时就会打印。

选定“当细节区数据溢出到新页/列时打印”复选框后，当细节带区中的打印内容已满一页或一列而换到另一页或另一列时就会打印。

③“若是空白行则删除”复选框：如果设置的条件使控件不被打印，并且又没有其他对象位于同一水平位置上，那么选定该复选框就会删除控件所在行；若该复选框未选定则打印一个空行。

④“仅当下列表达式为真时打印”文本框：用于键入一个表达式，或利用对话按钮显示表达式生成器来设置表达式。当表达式的值为真时控件才被打印，否则不打印。

11.3.5 报表变量

若要在报表中操作数据或显示计算结果，可以使用报表变量。使用报表变量，可以计算各种值，并且可以用这些值来计算其他相关值。

1．创建报表变量

报表菜单中的变量命令可用于创建与编辑报表变量。在“报表”菜单中选择“变量”将弹出“报表变量”对话框，如图 11-22 所示。

图 11-22　报表变量对话框

其中各组件的含义如下：

①“变量”列表区：用于显示已定义的报表变量，并可键入报表变量。拖动列表中变量名左边的上下双箭头按钮可改变报表变量的排列次序。

②“插入”按钮：用于在变量列表框中插入一个空文本框，以便定义新的变量。

③“删除”按钮：用于在变量列表框中删除选定的变量。

④“要存储的值”文本框：键入表达式，并将此表达式赋值给报表变量。例如图 11-22 中的报表变量为 xh，要存储的值是 xh+1，这相当于设置了命令 xh = xh + 1。

选定该文本框右侧的对话框会出现表达式生成器对话框，若变量已经赋值，它就会被列入该对话框的变量列表中。

⑤“初始值”文本框：键入变量的初始值。

⑥“重置”组合框：指定变量重置为初始值的位置。默认位置是报表尾，也可选择页尾或列尾。如果在报表中创建了组，重置框将为该组显示一个重置选项。

⑦“报表输出后释放”复选框：选定该复选框后，每当报表打印完毕报表变量即从内存中释放；如果未选定，除非退出 VFP 或使用 CLEAR ALL，CLEAR EMORY 等命令来释放，否则此变量一直保留在内存中。

⑧“计算”区：用来指定变量执行的计算操作。从其初始值开始计算，直到变量被再次重置为初始值为止。该区各选项按钮的功能前已介绍，这里不再重复。

2．创建报表变量控件

报表变量建立后，变量名即进入表达式生成器列表框，供创建域控件时选用。

创建报表变量控件的步骤：选定报表控件工具栏的“域控件”按钮，单击报表设计器窗口某处，在“报表表达式”对话框中选定“表达式”文本框右侧的“...”按钮，使出现表达式生成器对话框，在表达式生成器对话框的“变量列表”中双击某报表变量，单击“确定”按钮返回报表表达式对话框，单击“确定”按钮返回报表设计器窗口，报表变量控件便已产生。

【例 11-4】设计报表“成绩一栏表”，报表列出 cj 表中的所有成绩，如图 11-23 所示。

成绩一栏表

	专业	姓名	课程号	成绩
1	计算机应用	李富强	111101	72
2	计算机应用	李富强	141203	70
3	计算机应用	李富强	151101	70
4	计算机应用	冯见岳	151101	80
5	计算机应用	冯见岳	111101	82
6	计算机应用	冯见岳	141203	80
7	国际贸易	罗海燕	111101	67
8	国际贸易	罗海燕	151101	65
9	会计	张丽萍	151101	63
10	会计	张丽萍	111101	65
11	会计	张丽萍	621201	62
12	会计	刘刚	111101	50
13	会计	刘刚	151101	48
14	国际贸易	赵江山	631206	85
15	国际贸易	赵江山	111101	87
16	国际贸易	赵江山	151101	85
17	国际贸易	许海霞	111101	54
18	国际贸易	许海霞	631206	59
19	计算机应用	王春雷	111101	70
20	计算机应用	王春雷	151101	68
21	计算机应用	李家富	151101	81
22	计算机应用	李家富	111101	83
23	计算机应用	李家富	141203	81
			平均成绩	70.43

图 11-23　成绩一栏表

本例使用报表变量控件来输出序号和平均成绩，主要设计步骤如下：

① 新建一个空白报表，并增加标题带区和总结带区。

② 打开数据环境，添加 student 和 cj 表作为数据源，两个表按学号字段建立关系。用鼠标右键单击关系连线，选择快捷菜单中的“属性”项，打开属性窗口。设置关系 Relation1 的 OneToMany 属性为真（.T.），使得只有在“子表的记录指针遍历了所有相关记录之后，才移动父表的记录指针”。

③ 创建报表变量：单击“报表”菜单中的“变量”菜单项，打开“报表变量”对话框，在“变量列表区”编辑框中键入 xh，在“要存储的值”文本框中键入 xh+1；再创建 pj 变量，在“要存储的值”文本框中键入 cj.成绩，并在“计算”区选择“平均值”，如图 11-24 所示。

图 11-24　创建报表变量

④ 在各个带区添加如图 11-25 的左图所示的控件和字段。

⑤ 创建报表变量控件：在“报表控件”栏上单击“域控件”按钮，创建域控件，在弹出的“报表表达式”对话框中的“表达式”文本框中键入 xh，同理，创建 pj 报表变量控件，然后，将这个两控件分别拖入细节带区的最左面和总结带区，如图 11-25 的右图所示。

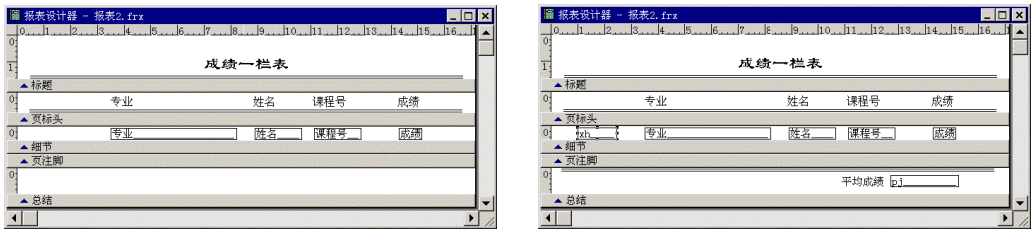


图 11-25　添加控件

⑥ 预览报表如上图 11-23 所示。将报表文件保存为报表 4.frx。

11.3.6　报表控件的布局

1．选择、移动及调整报表控件的大小

如果创建的报表布局上已经存在控件，则可以更改它们在报表上的位置和尺寸。可以单独更改每个控件，也可以选择一组控件作为一个单元来处理。

（1）移动一个控件

若要移动一个控件，可以选择控件并把它拖动到“报表”带区中新的位置上。

控件在布局内移动位置的增量并不是连续的。增量取决于网格的设置。若要忽略网格的作用，拖动控件时应按下〈Ctrl〉键。

（2）选择多个控件

若要选择多个控件，可以在控件周围拖动以画出选择框。

选择控点将显示在每个控件周围。当它们被选中后，可以作为一组内容来移动、复制或删除。

（3）将控件组合在一起

通过将控件标识在一个组中，可以为多个任务将一组控件关联在一起。例如，将标签控件和域控件彼此关联在一起，这样不用分别选择便可移动它们。当已经设置格式并且对齐控

件后，这个功能也有用，因为它保存了控件彼此间的位置。

若要将控件组合在一起：选择想作为一组处理的控件，从“格式”菜单中，选择“分组”。选择控点将移到整个组之外。可以把该组控件作为一个单元处理。

（4）对一组控件取消组定义

若要对一组控件取消组定义，可以选择该组控件，然后从“格式”菜单中，选择“取消组”命令。

（5）调整控件的大小

如果在布局上已有控件，则可以单独地更改它的尺寸，或者调整一组控件的大小使它们彼此相匹配。可以调整除标签之外任何报表控件的大小。标签的大小由文本、字体及磅值决定。若要调整控件的大小，可以选择要调整的控件，然后拖动选定的控点直到所需的大小。

（6）匹配多个控件的大小

若要匹配多个控件的大小，可以选择想使其具有同样大小的一些控件，从“格式”菜单中，选择“大小”。选择适当选项来匹配宽度、高度或大小，控件将按照需要进行调整。

2．复制和删除报表控件

可以单独或成组复制或删除布局上的任意控件。

（1）复制控件

选择要复制的控件，从“编辑”菜单中，选择“复制”，然后，选择“粘贴”。控件的副本将出现在原始控件下面，将副本拖动到布局上的正确位置。

（2）删除控件

选择要删除的控件，从“编辑”菜单中，选择“剪切”或按〈Delete〉键。

3．对齐控件

可以根据彼此间关系对齐控件，或者根据“报表设计器”提供的网格放置它们。可以沿某一侧或居中对齐控件。

若要对齐控件：选择想对齐的控件，从“格式”菜单中，选择“对齐”命令。从子菜单中，选择适当对齐选项。VFP 使用距离所选对齐方向最近的控件作为固定参照控件。

也可以使用“布局”工具栏。使用工具栏，可以同距离所选一侧最远的控件对齐，只要在单击对齐按钮时按下〈Ctrl〉键，如图 11-26 所示。

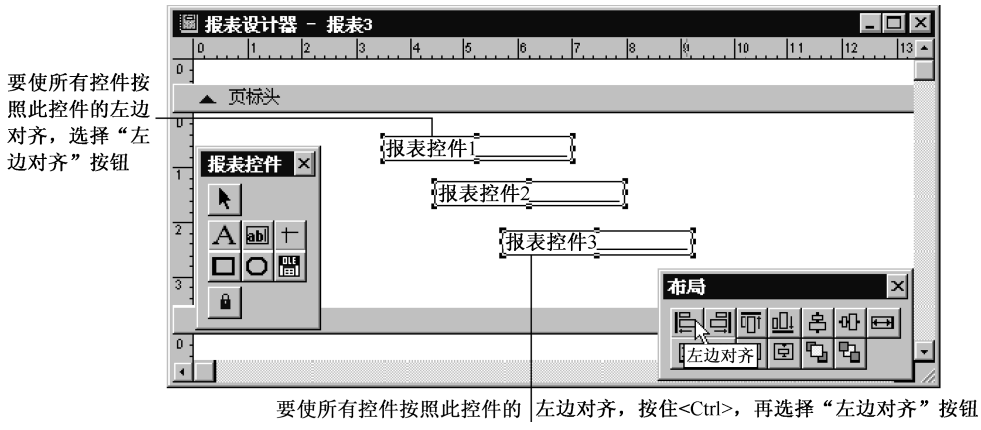


图 11-26　左对齐工具

选择对齐所有控件的边缘线时，应考虑到所有控件应彼此分开，而不应相互重叠。同一行上的控件如果沿它们右侧或左侧对齐，它们将彼此堆在一起。同样，同一竖线上的控件上、下对齐也会重叠。

若要居中对齐带区内的控件：选择想对齐的控件。从“格式”菜单中，选择“对齐”命令。从子菜单中，选择“垂直居中对齐”或“水平居中对齐”。控件将移动到各自带区的垂直或水平中心。

4．调整控件的位置

使用状态条或表格控件，可以将控件放置在报表页面上的特定位置。默认情况下，控件根据网格对齐其位置。可以选择关掉对齐功能和显示或隐藏网格线。网格线可以帮助您按所需布局放置控件。

若要将控件放置在特定的位置：从“显示”菜单中，选择“显示位置”命令。选择一个控件，然后使用状态栏上的位置信息将该控件移动到特定位置。

若要人工对齐控件：从“格式”菜单中，清除“对齐格线”。

若要显示网格线：从“显示”菜单中，选择“网格线”命令。网格将在报表带区中显示。

若要更改网格的度量单位：从“格式”菜单中，选择“设置网格刻度”命令。在“水平”、“垂直”框内，分别输入代表网格每一方块水平宽度和垂直高度的像素数目。

11.4 报表分组与多栏报表

在实际应用中，常需要把具有某种相同信息的数据组织在一起，使报表更易阅读。分组可以明显地分隔相关记录和为相关记录添加总结性数据。

11.4.1 报表分组

一个报表可以设置一个或多个数据分组，组的分隔基于分组表达式。

1．数据分组

如果数据源是表，记录的物理顺序可能不适合分组。为了使数据源适合于分组处理记录，必须对数据源进行适当的索引或排序。通过为表设置索引，或是在数据环境中使用视图、查询作为数据源，才能达到合理分组显示记录的目的。

如果数据表已经设有索引，可以在数据环境中设定索引：

① 为报表打开数据环境设计器。

② 在数据环境设计器中单击鼠标右键，从弹出的快捷菜单中选择“属性”命令，打开“属性”窗口。

③ 在“属性”窗口中选定对象“Cursor1”。

④ 修改其“Order”属性为相应的索引名。

2．单级分组报表

单级分组报表是指表达式进行单层（一级）数据分组。例如，可以按“客户号”分组，相同客户的记录在一起打印。

【例 11-5】在【例 11-4】的报表中，按“专业”分组打印，具体要求如下：

① 报表的内容（细节带区）是 student 表的姓名、cj 表的课程号和成绩。

② 增加数据分组，分组表达式是“ student.专业 ”，组标头带区的内容是“ 专业 ”，组注脚带区的内容是该组学生的“ 平均成绩 ”。

③ 标题带区中，标题是“ 分专业成绩汇总表 ”，要求是 3 号字、黑体，括号是全角符号。

④ 总结带区的内容是所有成绩的平均值。

为了正确处理分组数据，必须事先对报表文件的数据源进行索引，本例假定已经对“ 专业 ”建立索引，索引名为“ 专业 ”。

在【例 11-4】报表的基础上进行修改，主要步骤如下：

① 打开报表文件，并同时打开报表设计器。

② 设置当前索引：打开数据库环境设计器。单击鼠标右键，从快捷菜单中选择“ 属性 ”命令，打开“ 属性 ”窗口。确认对象框中为“ cursor1 ”，在“ 数据 ”选项卡中选定“ order ”属性，从索引列表中选择专业。

③ 添加数据分组：选择“ 报表 ”菜单中的“ 数据分组 ”菜单项，在弹出的“ 数据分组 ”对话框中单击第一个“ 分组表达式 ”框右侧的“ ... ”按钮，打开“ 表达式生成器 ”。

对话框中选择“ 专业 ”作为分组依据。单击“ 确定 ”按钮，报表设计器中添加了“ 组标头 1：专业 ”和“ 组注脚 1：专业 ”两个带区。

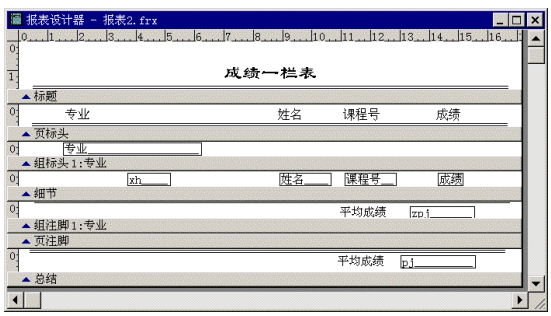
④ 添加报表变量：单击“ 报表 ”菜单中的“ 变量 ”菜单项，打开“ 报表变量 ”对话框，在“ 变量列表区 ”编辑框中键入 zpj，在“ 要存储的值 ”文本框中键入 cj.成绩，在“ 重置 ”组合框中选择“ student.专业 ”，并在“ 计算 ”区选择“ 平均值 ”。

⑤ 调整域控件：把“ 专业 ”字段的域控件从“ 细节 ”带区移动到“ 组标头 ”带区最左面；调整细节带区和页标头带区的其他控件，使它们上下对齐。

与【例 11-4】中总结带区的域控件 zj 相仿，在组注脚中添加域控件 zzj，如图 11-27 的左图所示。需要说明的是，将该控件添加到组注脚带区，计算出的平均值是本组的，而总结带区则是计算的全部。

最后，在组注脚带区添加一个线条控件。

⑥ 预览效果如图 11-27 的右图所示。



成绩一栏表

专业		姓名	课程号	成绩
国际贸易				
	1	罗海燕	111101	67
	2	罗海燕	151101	65
	3	赵江山	631206	85
	4	赵江山	111101	87
	5	赵江山	151101	85
	6	许海霞	111101	54
	7	许海霞	631206	52
			平均成绩	70.71
会计				
	8	张丽萍	151101	63
	9	张丽萍	111101	65
	10	张丽萍	621201	63
	11	刘刚	111101	50
	12	刘刚	151101	48
			平均成绩	57.80
计算机应用				
	13	李富强	111101	72
	14	李富强	141203	70
	15	李富强	151101	70
	16	冯见岳	151101	80
	17	冯见岳	111101	82
	18	冯见岳	141203	80

图 11-27　分组报表

3．多级分组报表

在 VFP 报表中，最多可以有 20 级数据分组。嵌套分组有助于组织不同层次的数据和总

计表达式，但在实际应用中往往只用到 3 级分组。在设计多级数据分组报表时，需要注意分组的级与多重索引的关系。

（1）多级数据分组基于多重索引

多级分组报表的数据源必须可以分出级别来，例如数据源中有“专业”、“姓名”和“课程号”等字段，可以按关键字表达式“专业 + 姓名 + 课程号”建立索引，也可以按关键字表达式“课程号 + 姓名 + 专业”建立索引。如何建立索引，应视具体情况而定。

（2）分组层次

一个数据分组对应于一组“组标头”和“组注脚”带区。数据分组的编号按其创建的顺序而定，编号越大的分组离细节带区越近。

（3）更改分组

在“数据分组”对话框中可以改变分组的顺序，更改或删除组带区。

当在“数据分组”对话框中移动组的位置而改变分组顺序时，组带区中定义的所有控件都将自动移到新的位置。对组重新排序并不更改以前定义的控件，例如，框或线条以前是相对于组带区的上部或底部定位的，它们仍将固定在组带区的原位置。

组被删除后，该组带区将从报表布局中删除。如果该组带区中含有控件，将提示同时删除控件。

需要特别注意的是，更改分组后，必须重新指定索引，才能够正确组织各组的数据。

【例 11-6】以“订货管理”数据库为数据源，设计一个二级分组报表“客户订单一览表”，效果如图 11-28 所示。

客户订单一览表

器件号	器件名		单价	数量	金额
客户号:A00112					
订单号: OR-21A		订购日期: 03/11/02			
S4911	声卡		390.00	2	780.00
P1005	CPU P4 1.5G		1,350.00	1	1350.00
			本订单金额合计:		2130.00
订单号: OR-22A		订购日期: 10/27/01			
M0256	内存		400.00	4	1600.00
			本订单金额合计:		1600.00
订单号: OR-33A		订购日期: 09/10/01			
E0032	E盘(闪存)		295.00	2	590.00
M0256	内存		405.00	6	2430.00
			本订单金额合计:		3020.00
订单号: OR-41A		订购日期: 04/01/02			
M0256	内存		380.00	10	3800.00
P1001	CPU P4 1.4G		1,100.00	4	4400.00
			本订单金额合计:		8200.00
			客户订单金额总计:		14950.00
客户号:B20001					

图 11-28　客户订单一览表

设计步骤如下：

① 假定“订货管理”数据库中数据表 order_list 以关键字表达式“客户号 + 订单号”建立普通索引“客户订单”，数据表 order_detail 以“订单号”建立普通索引“订单号”。

② 新建一个空白报表，并打开数据库环境设计器。在数据环境中依次添加数据表 order_list 和 order_detail，从表 order_list 中选定字段“订单号”，将其拖到表 order_detail 中，

如图 11-29 的左图所示，松开鼠标左键，建立一个关系 Relation1，如图 11-29 的右图所示。

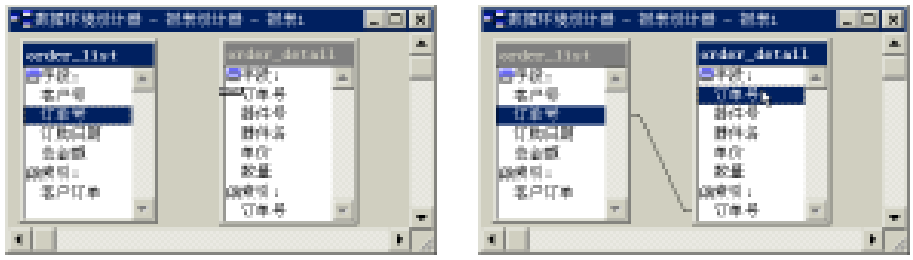


图 11-29　设置数据环境

用鼠标右键单击数据环境设计器，在弹出的快捷菜单中选择“ 属性 ”命令，打开属性窗口。在“ 对象 ”下拉框中选定关系 Relation1，修改其 OneToMany 属性为.T. – 真。

修改对象 Cursor1 （order_list）的 Order 属性为客户订单。

③ 添加数据分组：单击“ 报表设计器 ”工具栏上的“ 数据分组 ”按钮，打开“ 数据分组 ”对话框。

单击第一个“ 分组表达式 ”框右侧的对话框，在“ 表达式生成器 ”对话框中选择“ order_list.客户号 ”，单击“ 确定 ”按钮返回；相仿地，在第二个“ 分组表达式 ”框中选择“ order_list.订单号 ”。

单击“ 确定 ”按钮返回，报表设计器中添加了“ 组标头 1：客户号 ”、“ 组标头 2：订单号 ”、“ 组注脚 2：订单号 ”、“ 组注脚 1：客户号 ”4 个带区。

④ 添加“ 标题 ”带区：单击“ 报表 ”菜单中的“ 标题/总结 ”菜单项，在打开的“ 标题/总结 ”对话框中选中“ 报表标题 ”复选框，单击“ 确定 ”按钮返回。报表设计器中添加了“ 标题 ”带区。

⑤ 添加报表变量：单击“ 报表 ”菜单中的“ 变量 ”菜单项，打开“ 报表变量 ”对话框，依次添加表 11-5 中的变量。

表 11-5　报表变量

变 量 名 称	要存储的值	“ 重置 ”组合框	“ 计算 ”区
je	order_detail.单价* order_detail.数量		
zj1	je	订单号	总和
zj2	je	客户号	总和

⑥ 添加域控件：

在“ 组标头 1：客户号 ”带区添加字段域：order_list.客户号。

在“ 组标头 2：订单号 ”带区添加字段域：order_list.订单号、order_list.定购日期。

在“ 细节 ”带区添加字段域：order_detail.器件号、order_detail.器件名、order_detail.单价、order_detail.数量和报表变量域 je。

在“ 组注脚 2：订单号 ”带区添加报表变量域 zj1。

在“ 组注脚 1：客户号 ”带区添加报表变量域 zj2。

⑦ 添加标签控件：单击“ 报表控件工具栏 ”上的“ 标签 ”按钮，在“ 标题 ”带区中输

入“客户订单一览表”，单击“格式”菜单下的“字体”，选择黑体二号字，并设置为水平居中。

在页标头带区添加标签：器件号、器件名、单价、数量和金额。

在“组标头 1：客户号”带区添加标签：客户号。

在“组标头 2：订单号”带区添加标签：订单号、定购日期。

在“组注脚 2：订单号”带区添加标签：本订单金额总计。

在“组注脚 1：客户号”带区添加标签：客户订单金额总计。

将其设置为粗体字，其风格如图 11-30 所示。

图 11-30　设计报表

最后，在添加一些线条控件修饰报表。

至此，整个报表的设计已完成，其预览效果如图 11-28 所示。

11.4.2　报表分栏

多栏报表是指分为多列打印输出的报表。如果输出的字段较少，横向只占用部分页面，可以设计成多栏报表。

1．页面设置

从“文件”菜单中选择“页面设置”菜单项，打开“页面设置”对话框，如图 11-31 所示。

各部分的含义如下：

①“列数”：主要用于报表中的数据是 1 栏还多栏，默认为 1 栏。

②“宽度”：主要用于页面的可使用宽度。

③“间隔”：只有列数文本框中的值大于 1 时才起作用，用于设置栏与栏间的间隔。

④“打印顺序”：只有列数文本框的值大于 1 时才起作用，用于设置栏与栏数据是按从上到下还是从左到右的顺序打印。

⑤“左页边距”：设置页面左边距。

2．设置“打印顺序”

在打印报表时，“细节”带区中的内容依“自上向下”的顺序打印（默认）。对于多栏报

表，这种打印顺序只能靠左边距打印一个栏目，页面上的其他栏均为空白。为了能在页面上打印出多个栏，需要将“打印顺序”设置为“自左向右”。

3．设置“列标头”和“列注脚”带区

在“页面设置”对话框中设置“列数”，将改变报表分栏。例如设置列数为 2，则将整个页面平均分为两部分，设置列数为 3，则将整个页面平均分为三部分。单击“确定”按钮，在报表设计器中将添加一个“列标头”带区和一个“列注脚”带区，同时“细节”带区也将相应缩短，如图 11-32 所示。

图 11-31　“页面设置”对话框

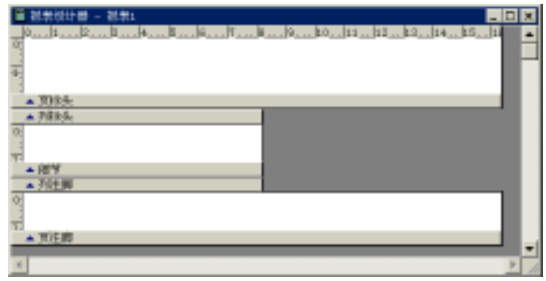


图 11-32　“列标头”和“列注脚”带区

下面以一个实例来说明多栏报表的设计。

【例 11-7】将【例 11-4】的报表设计成分栏报表。

① 打开报表文件：报表 4.frx。

② 设置多栏报表：在页面设置中将列数设为 2，在报表设计器中将添加二分之一的一对“列标头”带区和“列注脚”带区。

③ 设置打印顺序：在“页面设置”对话框中选择打印顺序为“自左至右”，以避免只靠左边距打印一个栏目，页面上其他栏目空白。单击“确定”按钮，关闭对话框。

④ 复制页标头带区的标签控件：专业、姓名、课程号、成绩，并调整其位置。

⑤ 调整“细节”带区各域控件的位置，如图 11-33 所示。

⑥ 调整“标题”标签的位置（水平居中），并调整各线条控件的长度。

⑦ 单击“文件”菜单中的“另存为”菜单项，在打开的“另存为”对话框中，将报表另存为报表 7.frx

预览效果如图 11-34 所示。

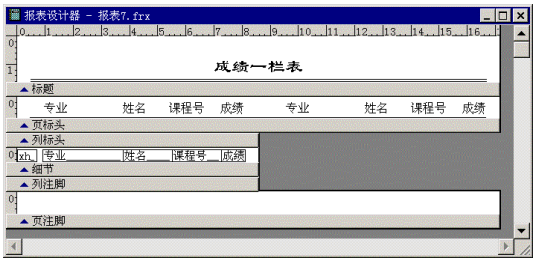


图 11-33　调整各域控件的位置

图 11-34　分栏报表

11.5 预览和打印报表

可以在定制报表布局的任何时候预览或打印报表。

11.5.1 预览结果

通过预览报表，不用打印就能看到它的页面外观。例如，可以检查数据列的对齐和间隔，或者看报表是否返回希望的数据。有两个选择，即显示整个页面或者缩小到一部分页面。

1．设计时的预览

如果报表已经在“报表设计器”中打开，那么任何时候都可以使用“预览”功能查看报表的打印效果。其操作十分方便：

① 可以从“显示”菜单中，选择“预览”菜单项。

② 可以从“文件”菜单中，选择“打印预览”菜单项。

③ 可以用鼠标右键单击“报表设计器”，从弹出的快捷菜单中选择“预览”命令。

④ 可以直接单击“常用”工具栏中的“打印预览”按钮。

2．使用命令预览

也可以使用命令来预览报表，命令的格式为：

```
REPORT FORM 〈报表文件名〉 PREVIEW
```

例如，在命令窗口输入如下命令，即可预览【例 11-7】中的分栏报表：

```
REPORT FORM report2.frx PREVIEW
```

说明：上述命令还可以使用 FoxPro 命令中的范围子句和 For 条件子句来控制报表所包含的记录。

预览窗口通常包含一个“打印预览”工具栏。在“打印预览”工具栏中，选择“前一页”或“下一页”可以切换页面；若要更改报表图像的大小，选择“缩放”列表；要打印报表，可以选择“打印报表”按钮；若想要返回，则选择“关闭预览”。

11.5.2 打印报表

在实际应用中，最为重要的是将报表通过打印机输出。

1．设计时打印

如果报表已经在“报表设计器”中打开，那么任何时候都可以打印报表。其操作十分方便：

① 可以从“文件”菜单中，选择“打印”菜单项或直接单击“常用”工具栏中的“打印”按钮。

② 可以从“报表”菜单中，选择“运行”菜单项或直接单击“常用”工具栏中的“运行”按钮。

③ 可以用鼠标右键单击“报表设计器”，从弹出的快捷菜单中选择“打印”命令。

④ 还可以通过预览，使用“打印预览”工具栏中的“打印”按钮。

2．使用命令打印

也可以使用命令来打印报表，命令的格式为：

REPORT FORM 〈报表文件名〉 TO Printer

例如，在命令窗口输入如下命令，即可打印【例 11-7】中的分栏报表：

REPORT FORM report2.frx TO Printer

说明：

① 上述命令还可以使用 FoxPro 命令中的范围子句和 For 条件子句来控制报表所包含的记录。

② 如果未设置数据环境，则显示“打开”对话框，并在其中列出一些表，从中可以选定要进行操作的一个表。VFP 把报表发送到打印机上。

11.6 习题

一、选择题

1．以下不是启动报表向导的途径的是（　　）。

A）打开“项目管理器”，在“文档”选项卡中，选择“报表”

B）从“文件”菜单的“新建”中选择“报表”

C）从“格式”菜单中选择“报表”

D）在“工具”菜单中选择“向导”子菜单，选择“报表”

2．下列哪个控件又称为“表达式控件”（　　）。

A）标签控件　　B）线条控件

C）图片/ActiveX 图片控件　　D）域控件

3．下列哪个带区不是快速报表默认的基本带区（　　）。

A）页标头　　B）标题　　C）细节　　D）页注脚

4．使用报表向导定义报表时，定义报表布局的选项是（　　）。

A）列数、方向、字段布局　　B）列数、行数、字段布局

C）行数、方向、字段布局　　D）列数、行数、方向

5．下列对于报表变量控件叙述正确的是（　　）。

A）报表变量控件和其他报表控件一样可以直接创建

B）报表变量控件必须先创建报表变量才能创建

C）可以先创建报表控件再创建报表变量

D）报表变量和报表变量控件没有关系

二、填空题

1．设计报表时包括两个部分：_____和_____。

2．在 Visual FoxPro 中提供的三种创建报表的方法是_____、_____和_____。

3．域控件可以打印表或视图中的_____、_____和_____。

4．数据分组之后会自动弹出两个带区是_____和_____。

5．数据分组的分组字段必须是_____。

6．数据分栏之后会自动添加的两个带区是_____和_____。

三、上机题

利用 Visual FoxPro 的快速报表功能建立一个满足如下要求的简单报表：

（1）报表的内容是 order_detail 表的记录（全部记录，横向）；

（2）增加“标题带区”，然后在该带区中放置一个标签控件，该标签控件显示报表的标题“器件清单”；

（3）将页注脚区默认显示的当前日期改为显示当前的时间；

（4）最后将建立的报表保存为 report1.frx。

第 12 章　上机考试技巧

上机考试是考生计算机综合应用的体现，因为上机考试时除了需要考生有扎实的基础知识外，上机操作的熟练程度也充分体现出来，所以考生要特别注意。

12.1　上机考试试题的题型与考试时间

全国计算机等级考试二级程序设计语言考试包括笔试和机试两部分。笔试全国按统一时间进行；机试是从笔试后的第二天开始，由各考点分批进行上机考试。

12.1.1　上机考试试题的题型

全国计算机等级考试二级 VFP 考试有以下三种类型的题目：

基本操作题（4 个小题，其中 1、2 题每题 7 分，第 3、4 题每题 8 分，共计 30 分）；

简单应用题（2 个小题，每题 20 分，共计 40 分）；

综合应用题（1 个小题，共计 30 分）。

1．基本操作题

测试考生对数据库、表、表结构、表记录和索引的使用能力；使用向导设计表单、菜单、查询、标签和报表的使用能力；建立表间关系和项目管理的使用能力。

2．简单应用题

测试考生对建立一个比较复杂的菜单的使用能力；自行设计表单、报表、查询和标签的使用能力；创建、修改和使用视图的使用能力。

3．综合应用题

测试考生对包括基本操作题和简单应用题操作的综合测试的使用能力、以及给定数据库或表，根据要求进行编程生成新的数据库或表的能力。

12.1.2　上机考试的时间

二级 VFP 上机考试时间定为 90min。考试时间由上机考试系统自动进行计时，提前 5min 自动报警来提醒考生及时存盘。考试时间用完，上机考试系统将自动锁定计算机，考生将不能再继续答题。

12.2　上机考试软件的使用

上机考试采用专门设计的软件：全国计算机等级考试上机考试系统。全国计算机等级考试（Windows 版）上机考试系统提供了开放式的考试环境，考生可以在 Windows 95/98 操作系统环境下自由地使用各种应用软件系统或工具，它的主要功能是考试项目的执行、控制上机考试的时间以及试题内容的显示。

根据目前考点的计算机情况，分别提供了网络和单机两种考试环境。具体使用哪种考试运行环境，由所报名的考点决定，因此，在报名时要咨询一下。一般考点在考前会举办上机

考试模拟学习班，提供给考生熟悉上机环境的机会，这样正式考试时就可以集中精力答题。下面将详细介绍最新上机考试系统 V4.2.3 版软件的使用及上机考试操作过程。如果所报名的考点不提供模拟考试的机会，上机考试时可参考下面步骤操作。

12.2.1 考试过程

考生提前 5min 进入机房，在抽签决定的工作站号（或微机号）上就坐，考试开始后请开启微机，进入 Windows 95/98 操作系统下。

从开始菜单的“程序”中选择“全国计算机等级考试”菜单项，启动“考试程序”。首先是一个登录过程，当考生登录成功后，上机考试系统将自动装载试题内容查阅工具，通过这个工具开始完成看题、做题。

考试过程分为登录、看题、做题几个阶段。

12.2.2 登录

启动考试程序，出现如图 12-1 的左图所示的登录界面（其中版本号可能会变动）。在“开始登录”按钮上单击鼠标左键或按〈Enter〉键出现考号输入窗口，如图 12-1 的右图所示。

图 12-1 登录界面

1．首次登录

① 输入准考证号，然后按回车或选择“考号验证”对输入的考号以及姓名、身份证号进行验证，如图 12-2 所示。如果考号不正确，可以选择“否（N）”重新输入；如果考号正确，则选择“是（Y）”继续。

图 12-2 考号验证

注意：如果在考试登录过程中，发现姓名或身份证号不相符时，应及时报告考点老师，请考点修改报名库中的姓名和身份证号，考试结束后把报名库交承办机构即可。考试可以继续进行，对考生的成绩也不会有影响。

② 在正确地输入了考号之后，选择“开始考试”按钮。系统出现如图 12-3 的左图所示的“考试须知”对话框。

注意：不管考点采用的是单机还是网络系统，具体的上机登录方法，以各考点指示的为准，监考人员会告诉考生操作方法。

2．二次登录

如果考生已经登录过，系统将提示输入密码，如图 12-3 的右图所示。这是在考试过程中

发生死机等意外情况，需要再次登录时出现的。在这里可以输入两种密码：

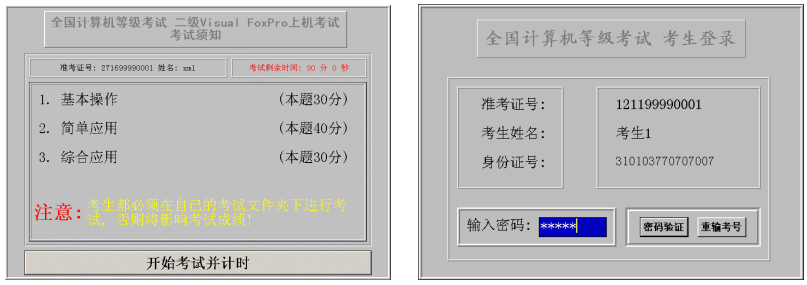


图 12-3　考试须知与二次登录密码输入

① 输入二次登录密码，将继续上一次的考试过程，从中断的地方继续考试，还是原先的题目，前面考过的时间也是继续累计。

如果考试中使用过“延时”密码，再进行二次登录，系统会给出一分钟的考试时间给考生进行考试。如果在这一分钟内退出考试，可以再进行二次登录，但系统只会给出一分钟内未使用掉的时间给考生进行考试。如果考生使用完了一分钟，屏幕会被锁住，这时只能使用“延时”密码，“结束”密码不可用，只要不进行“交卷”处理，可以几次“延时”。

在网络环境下中途更换机器进行二次登录时，考生在更换后的机器上进行登录时，其登录的用户名必须取更换前的机器上已登录的用户名。

② 输入重新抽题密码，系统会为考生重新抽取一套考题，并且与前一套考题不相同，考生的全部考试情况将从头开始，但系统会记录在案。

注意：在网络考试环境下，如果有多个考生同时用一个从未登录过的准考证号进行登录，那么只有一个考生可以正常登录，其余考生都不可以登录，并且在屏幕上会显示提示信息，提示已有一个考生正常登录，并指出它的网络服务器的登录用户名。在这种情况下，如果那个正常登录的考生确实不是这个准考证号的拥有者，只要找到这个准考证号的真正拥有者，在它的电脑上重新用这个准考证号进行登录，并输入重新抽题密码即可解决问题。

12.2.3　进入考生文件夹

在系统出现如图 12-3 所示的“考试须知”后，考生应该立即设置 VFP 的环境，以便进入考生文件夹。

1．考生文件夹

当考生登录成功后，上机考试系统将会自动产生一个考生文件夹，考生文件夹将存放该考生所有上机考试的考试内容。考生不能随意删除该文件夹以及该文件夹下与考试题目要求有关的文件及文件夹，避免在考试和评分时产生错误，从而导致影响考生的考试成绩。假设考生登录的准考证号为 121599990001，如果在单机上考试，则上机考试系统生成的考生文件夹将存放到 C 盘根目录下的 WEXAM 文件夹下，即考生文件夹为 C:\WEXAM\12150001；如果在网络上考试，则上机考试系统生成的考生文件夹将存放到 K 盘根目录下的用户目录文件夹下，即考生文件夹为 K:\ 用户目录文件夹\12150001。考生在考试过程中所操作的文件和文件夹都不能脱离考生文件夹，否则将会直接影响考生的考试成绩。

2．设置 VFP 的文件位置

设置 VFP 的文件位置可以使考试过程中所操作的文件和文件夹都不脱离考生文件夹。设置 VFP 的文件位置的步骤是：

（1）打开“选项”对话框

首先启动 VFP 系统，从“工具”菜单中选择“选项”命令，打开“选项”对话框。

（2）设置文件位置

选择“文件位置”选项卡，如图 12-4 所示。

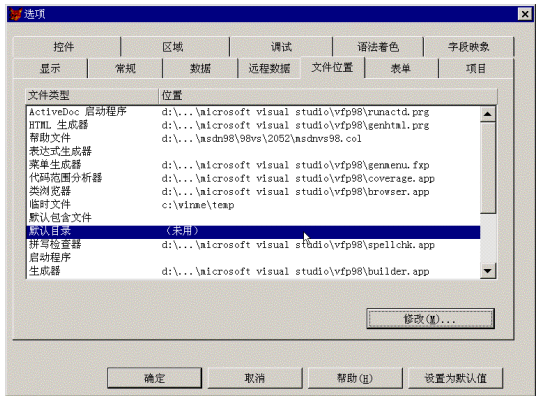


图 12-4　“文件位置”选项卡

在“文件类型”列表中选择“默认目录”，然后单击“修改”按钮。在弹出的“更改文件位置”对话框中选中“使用默认目录”复选框，如图 12-5 的左图所示。

输入考生文件夹的完整路径，或单击输入框右边的三点“…”按钮，在弹出的“选择目录”对话框中选定考生文件夹，如图 12-5 的右图所示，并单击“确定”按钮。再在“更改文件位置”对话框中单击“确定”按钮，返回“选项”对话框。

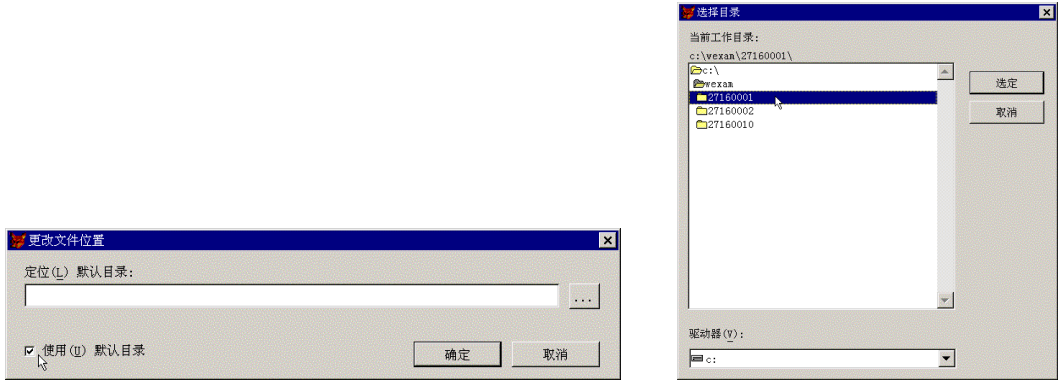


图 12-5　“更改文件位置”对话框与“选择目录”对话框

（3）保存设置

单击“设置为默认值”按钮，再按“确定”按钮。系统将把设置存储在 Windows 注册表中。

上述准备工作完成以后，就可以回到如图 12-3 的左图所示的“ 考试须知 ”对话框界面，准备进入考试界面，进行考试了。

12.2.4 考试界面

在如图 12-3 的左图所示的“ 考试须知 ”对话框中选择“ 开始考试并计时 ”，进入考试界面，就可以看题、做题，并开始计时。

考试界面是上机考试系统自动在屏幕中间生成的装载试题内容查阅工具的考试窗口，如图 12-6 所示。

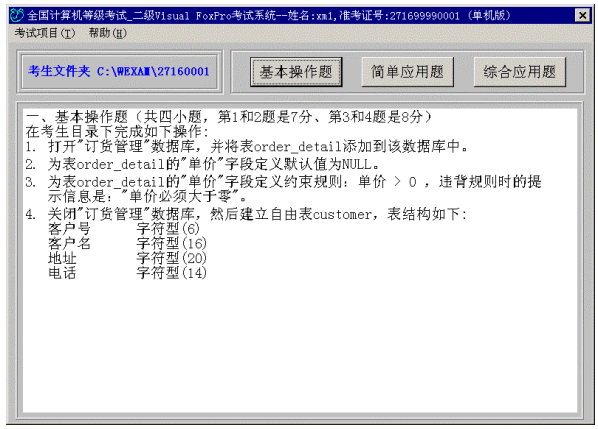


图 12-6 考试窗口

并在屏幕顶部始终显示着考生的准考证号、姓名、考试剩余时间以及可以随时显示或隐藏试题内容查阅工具和退出考试系统进行交卷的按钮的考生信息窗口，如图 12-7 所示。

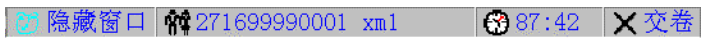


图 12-7 考生信息窗口

对于图 12-7 中最左面的“ 隐藏窗口 ”字符表示屏幕中间的考试窗口正在显示着，当用鼠标单击“ 隐藏窗口 ”字符时，屏幕中间的考试窗口就被隐藏，且“ 隐藏窗口 ”字符变成“ 显示窗口 ”。

考试窗口实质上只是一个试题显示窗口，答题操作均在 VFP 系统中进行。利用“ 隐藏/显示窗口 ”按钮，随时可以在 VFP 系统与考试窗口之间进行切换，以便查阅试题和答题。

12.2.5 查看试题要求

对于二级 VFP 考试，在考试窗口中选择工具栏中的试题选择按钮“ 基本操作题 ”、“ 简单应用题 ”和“ 综合应用题 ”可以查看相应题型的题目要求。

当试题内容查阅窗口中显示上下或左右滚动条时，表明该试题查阅窗口中试题内容不能完全显示，因此考生可用鼠标的箭头光标键并按鼠标的左键进行移动显示余下的试题内容，防止漏做试题从而影响考生考试成绩。

注意：在图 12-6 中标题栏的“（单机版）”字符表示现在的上机考试是单机考试，相应的

在图 12-6 中显示的考生文件夹的盘符就为 C；如果现在的上机考试是网络考试，则在图 12-6 中标题栏的“（单机版）”字符就会变成“（网络版）”字符，相应的在图中显示的考生文件夹的盘符就为 K。

12.2.6 寻求系统帮助

在“帮助”菜单栏中选择“等级考试系统帮助”可以启动考试帮助系统，并显示考试系统的使用说明，以及注意事项。

12.2.7 交卷

如果考生要提前结束考试进行交卷处理，则请在屏幕顶部始终显示着的“考生信息窗口”中选择“交卷”按钮，如图 12-7 所示，上机考试系统将显示是否要交卷处理的提示信息框，如图 12-8 所示。

此时考生如果单击“确定”按钮，则退出上机考试系统进行交卷处理，由系统管理员进行评分和回收。如果考生还没有做完试题，则单击“取消”按钮继续进行考试。

如果进行交卷处理，系统首先锁住屏幕，并显示“系统正在进行交卷处理，请稍候!”，当系统完成了交卷处理，在屏幕上显示“交卷正常，请输入结束密码：”或“交卷异常，请输入结束密码：”，这时只要输入正确的“结束”密码便可结束考试。

注意：这个过程不删除考生文件夹中的任何数据。

如果出现“交卷异常，请输入结束密码：”，说明这个考生有可能得零分或者考生文件夹有问题，所以要检查这个考生的实际考试情况是否正常。

如果在交卷过程中死机，可以重新启动计算机，再进行二次登录后进行“交卷”处理。

考试过程中，系统会为考生计算剩余考试时间。在剩余 5 分钟时，系统会显示一个提示信息，如图 12-9 所示，提示考生将应用程序的数据存盘，作最后的准备工作。

图 12-8 退出系统进行交卷处理提示

图 12-9 剩余考试时间提示

考试时间用完后，系统会锁住计算机并提示输入“延时”密码。这时考试系统并没有自行结束运行，它需要键入延时密码才能解锁计算机并恢复考试界面，而且考试系统会自动再运行五分钟，这时便可以单击“交卷”按钮进行退出系统的交卷处理。如果没有进行交卷处理，考试系统运行到五分钟后，系统又会锁住计算机并提示输入“延时”密码，这时还可以使用延时密码。只要不进行“交卷”处理，可以“延时”几次。

注意：在考试过程中，若遇到问题，应立即报告监考人员，请监考人员帮助解决。

在考试过程中，考生可以判断难易程度任意选择答题顺序，中间可以中断并允许考生重新答题。

12.2.8 文件的恢复

考生在考试过程中，若所操作的文件不能复原或误操作删除时，考生自己可把相应的文

件从考生文件夹下的 WARN 子文件夹中复制回来，然后可以继续考试。例如，给定的 DATAIN1.TXT 文件被更改后无法继续做题，这时可把 WARN 文件夹中的同名文件复制到当前文件夹中。

12.2.9 查分

另外还要注意，由于存在机器故障等原因，可能会发生考试内容没有被考试系统回收的情况，如果有考生得零分，考点可根据其实际情况决定该考生是否进行重考。所以在考试完后，应向监考人员咨询一下。

12.3 上机考试试题的操作

当选择“开始考试并计时”，进入考试界面后，系统立即开始计时，而上机一般都在 VFP 环境中操作，因此最好在此之前，先启动 VFP，并对 VFP 作相应的设置（见 12.2.2）。进入考试界面查阅试题后，切换到 VFP 中，根据题目要求，单击“新建”或“打开”按钮进行操作。

如果题目指出：在考生文件夹中有一个项目、数据库、表、或表单，则单击“打开”按钮，在“打开”对话框中找到试题给出的文件，如图 12-10 所示。

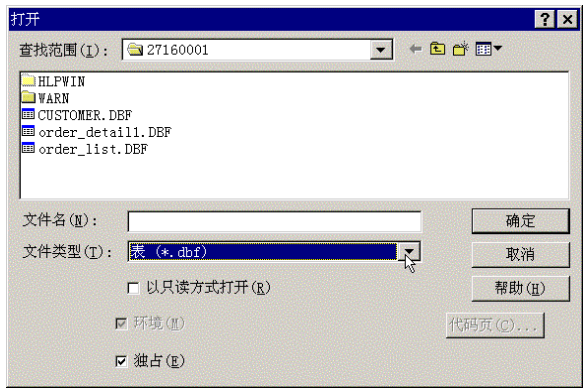


图 12-10 打开试题给出的表文件

如果题目指出：创建一个项目、数据库、表、表单或报表，则需要新建一个相应的文件，单击“新建”按钮，在“新建”对话框中选择相应的文件类型，进行操作。

12.3.1 基本操作题

测试考生对数据库、表、表结构、表记录和索引的使用能力；使用向导设计表单、菜单、查询、标签和报表的使用能力；建立表间关系和项目管理的使用能力。

【例 12-1】在考生目录下完成如下操作：

1．创建一个新的项目 sdb_p，并在该项目中创建数据库 sdb；

2．将考生文件夹下的自由表 student 和 sc 添加到 sdb 数据库中；

3．在 sdb 数据库中建立表 course，表结构见表 12-1。

表 12-1　course.dbf 的结构

字 段 名	类　型	宽　度
课程号	字符型	2
课程名	字符型	20
学时	数值型	2

随后向表中输入 6 条记录，记录内容见表 12-2（注意大小写）。

表 12-2　course.dbf 中记录

课 程 号	课 程 名	学　时
c1	C++	60
c2	Visual FoxPro	80
c3	数据结构	50
c4	JAVA	40
c5	Visual BASIC	40
c6	OS	60

4．为 course 表创建一个主索引，索引名为 cno、索引表达式为“课程号”。

操作步骤如下：

① 单击“新建”按钮或选择“文件”菜单中的“新建”项，在弹出的“新建”对话框中，选择文件类型项目，然后单击“新建文件”按钮。

在弹出的“创建”对话框中输入项目文件名 sdb_p（扩展名为.PJX），按“保存”按钮即完成项目 sdb_p 的创建，并打开“项目管理器”。

② 选择“项目管理器”中的“数据”选项卡，选中列表框中的“数据库”项，单击“新建”按钮，打开“新建数据库”对话框，在对话框中单击“新建数据库”按钮。

在弹出的“创建”对话框中输入数据库文件名 sdb（扩展名为.DBC），单击“保存”按钮，建立数据库 sdb.dbc，并打开“数据库设计器”。

③ 选择“数据库”菜单中的“添加表”项，或者在数据库设计器中单击鼠标右键，选择快捷菜单中的“添加表”项。

在弹出的“打开”对话框中选择考生文件夹中的表 student，然后单击“确定”按钮，将 student 添加到数据库中。用同样的方法，将表 sc 也添加到数据库中。

④ 选择“数据库”菜单中的“新建表”项，或者在数据库设计器中单击鼠标右键，选择快捷菜单中的“新建表”项，打开“新建表”对话框，在对话框中单击“新建表”按钮。

在弹出的“创建”对话框中输入表文件名 course（扩展名为.DBF），单击“保存”按钮，打开“表设计器”。

按照表 12-1 建立 course.dbf 的结构，然后按照表 12-2 输入 6 条记录。

⑤ 在表设计器中选择“索引”选项卡，在“索引名”栏输入 cno，在“类型”列表框中选择“主索引”，在“表达式”栏中输入索引关键字“课程号”。

单击“确定”按钮，并在弹出的对话框中单击“是”，返回“数据库设计器”。

12.3.2 简单应用题

简单应用题主要测试考生对建立一个比较复杂的菜单的使用能力；自行设计表单、报表、查询和标签的使用能力；创建、修改和使用视图的使用能力。包括使用向导、设计器建立简单的查询、视图、表单、报表、菜单设计与改错题。

无论设计还是改错题，必须要运行该程序，判断其运行结果是否正确。如果其运行结果不正确，则说明该程序还不正确，考生应该对该程序继续进行修改，直到运行结果正确为止。

【例 12-2】在考生文件夹下完成如下简单应用：

1．根据 sdb 数据库中的表用 SQL-SELECT 命令查询学生的学号、姓名、课程名和成绩，结果按“课程名”升序排序，“课程名”相同时按“成绩”降序排序，并将查询结果存储到 sclist 表中。

2．使用表单向导选择 student 表生成一个名为 form1 的表单。要求选择 student 表中所有字段，表单样式为“阴影式”；按钮类型为“图片按钮”；排序字段选择“学号”（升序）；表单标题为“学生基本数据输入维护”。

说明：题 1 判分依据是查询结果 sclist 表，因此，可以使用查询设计器生成查询，也可以用 SQL 命令设计查询，还可以在命令窗口使用 SQL 语句。当然，最直观的方法是使用查询设计器。题 2 只需按要求使用表单向导即可。

题 1 的操作步骤如下：

① 打开项目 sdb_p。选择“项目管理器”中的“数据”选项卡，选中列表框中的“查询”项，单击“新建”按钮，打开“新建查询”对话框，在对话框中单击“新建查询”按钮。

在弹出的“添加表或视图”对话框中依次添加表 student、cs、course，并按“学号”字段建立表 student 与 cs 的联接，按“课程号”字段建立表 cs 与 course 的联接关系。单击“关闭”按钮，打开“查询设计器”，如图 12-11 所示。

图 12-11 查询设计器

注意：如果“查询设计器”上部的数据表之间没有连线，则最好此时建立表间的联系（连线）：拖动“一”表相应字段到“多（或一）”表的字段上即可。

② 在“字段”选项卡中依次添加字段 student.学号、student.姓名、course.课程名、cs.成绩，如图 12-11 所示；在“排序依据”选项卡中依次添加字段 course.课程名、cs.成绩，并将

后一字段的排序选项设为降序，如图 12-12 所示。

图 12-12　排序依据

③ 在“查询”菜单中选择“查询去向”，打开“查询去向”对话框。单击“表”按钮，在“表名”栏中输入 sclist，如图 12-13 所示。

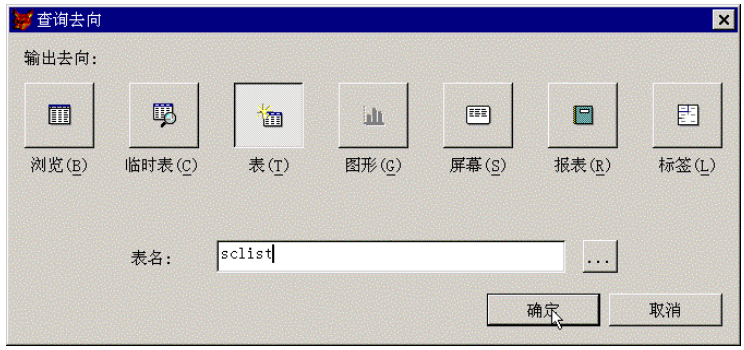


图 12-13　查询去向

单击“确定”按钮，返回“查询设计器”。

④ 最后，单击运行按钮“ !”，运行查询。关闭查询设计器，并保存查询。

在“数据工作期”对话框中，浏览表 sclist，如图 12-14 所示。

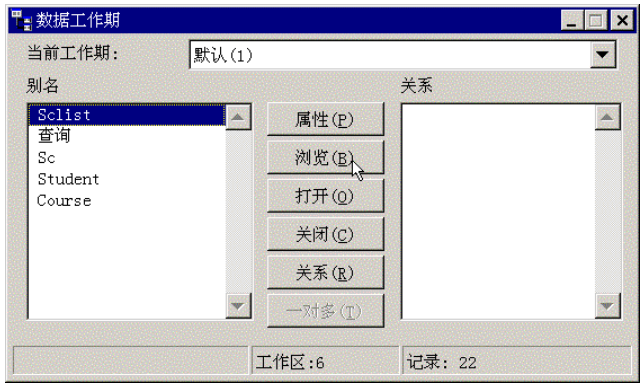


图 12-14　浏览表 sclist

说明：本例也可以编写程序或直接在命令窗口执行下述代码：

```
SELECT 学号,成绩,课程名 FROM sc;
  inner JOIN course AS cs ON  cs.课程号 = sc.课程号 INTO CURSOR tt
SELECT student.学号,姓名,课程名,成绩 FROM tt, student;
  WHERE tt.学号 = student.学号 INTO DBF sclist1;
  ORDER BY 课程名, 成绩 DESC
```

或

```
SELECT student.学号,姓名,成绩,课程名 FROM sc,course AS cs,student;
  WHERE sc.学号 = student.学号 AND  cs.课程号 = sc.课程号 INTO DBF sclist2;
  ORDER BY 课程名, 成绩 DESC
```

题 2 的操作步骤如下：

打开项目 sdb_p，**选择“项目管理器”中的“文档”选项卡，选中列表框中的“表单”项，单击“新建”按钮，打开“新建表单”对话框，在对话框中单击“表单向导”按钮。**

在弹出的“向导选取”对话框中直接单击“确定”按钮，打开表单向导。然后，按向导提示操作。

① 步骤 1 **–字段选取：单击“数据库和表”列表框右边的“…”按钮，在“打开”对话框中选择所需要的数据表** student，**即可在“可用字段”列表框中选取字段，如图** 12-15 **的左图所示。**

图 12-15　**字段选取与选择表单样式**

单击“全选”按钮选择 student **表中所有字段，然后单击“下一步”按钮进入样式对话框。**

② 步骤 2 **–选择表单样式：本例选择“阴影式”及“图片按钮”。单击“下一步”按钮进入排序对话框。**

③ 步骤 3 **–排序次序：本例选择“学号”字段作为排序依据。单击“下一步”按钮进入完成对话框。**

④ 步骤 4 **–完成：在标题文本框中输入创建表单的标题“学生基本数据输入维护”，如图** 12-16 **的右图所示，它将出现在表单的顶部。单击对话框左下角的“预览”按钮，可以预览设计的表单，如图** 12-17 **的左图所示。**

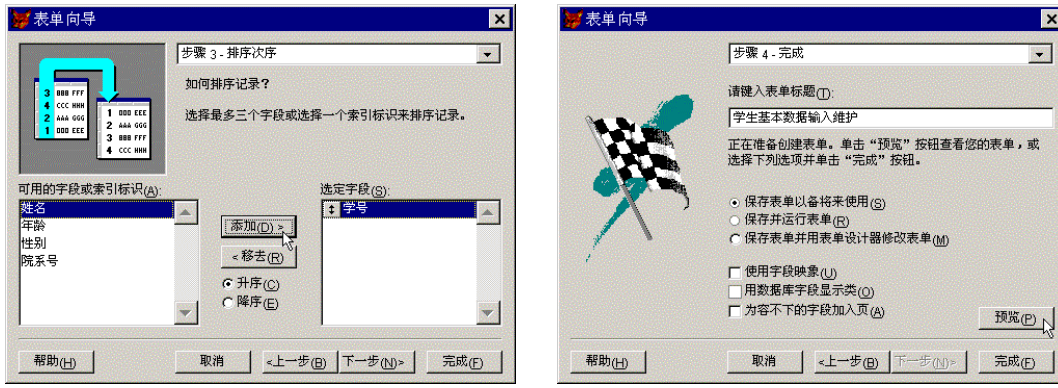


图 12-16　排序次序与完成

⑤ 最后，单击“完成”按钮，出现“另存为”对话框，将表单以 Form1（扩展名为.SCX）为名存储在考生文件夹中，如图 12-17 的右图所示。单击保存按钮结束表单向导操作。

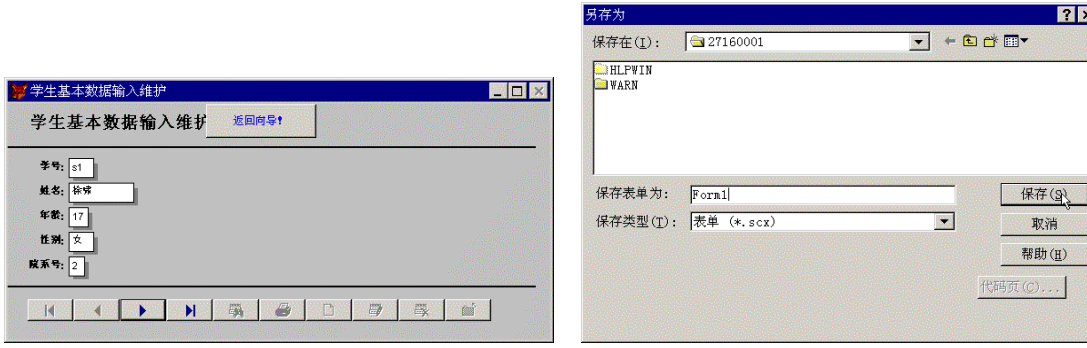


图 12-17　“预览”表单

12.3.3　综合应用题

综合应用题主要测试考生对包括基本操作题和简单应用题操作的综合测试的使用能力、以及给定数据库或表，根据要求进行编程生成新的数据库或表的能力。

【例 12-3】打开基本操作中建立的数据库 sdb，使用 SQL 的 CREATE VIEW 命令定义一个名称为 SVIEW 的视图，该视图的 SELECT 语句完成查询：选课门数是 3 门以上（不包括 3 门）的每个学生的学号、姓名、平均成绩、最低分和选课门数，并按“平均成绩”降序排序。最后将定义视图的命令代码存放到命令文件 T1.PRG 中并执行该文件。

接着利用报表向导制作一个报表。要求选择 SVIEW 视图中所有字段；记录不分组；报表样式为“随意式”；排序字段为“学号”（升序）；报表标题为“学生成绩统计一览表”；报表文件名为 p_student。

设计一个名称为 form2 的表单，表单上有“浏览”（名称为 Command1）和“打印”（Command2）两个命令按钮。鼠标单击“浏览”命令按钮时，先打开数据库 sdb，然后执行 SELECT 语句查询前面定义的 SVIEW 视图中的记录（两条命令，不可以有多余命令）；鼠标单击“打印”命令按钮时，调用报表文件 p_student 浏览报表的内容（一条命令，不可以有多

余命令)。

说明：本例需要完成的结果是一个程序文件 T1.PRG，其内容是定义视图 SVIEW；一个视图 SVIEW，包含在数据库中；一个报表 p_student，输出视图 SVIEW；一个表单 form2，调用视图和报表。

首先使用程序设计一个视图，然后利用视图制作报表和表单。视图的设计可以直接使用代码，也可以通过视图设计器得到 SQL 代码，然后建立程序文件。这里介绍利用视图设计器或的代码的方法。

操作步骤如下：

① 打开项目 sdb_p，选择“项目管理器”中的“数据”选项卡，展开列表框中的“数据库”项，再展开其中的数据库 sdb，选中其中的“本地视图”项，单击“新建”按钮，如图 12-18 的左图所示。

在弹出的“新建本地视图”对话框中，单击“新建视图”按钮，打开“视图设计器”。在图 12-18 的右图所示的“添加表和视图”对话框中添加表 student 和 sc，并在弹出的“联接条件”对话框中选择“学号”作为两个表的联接条件。

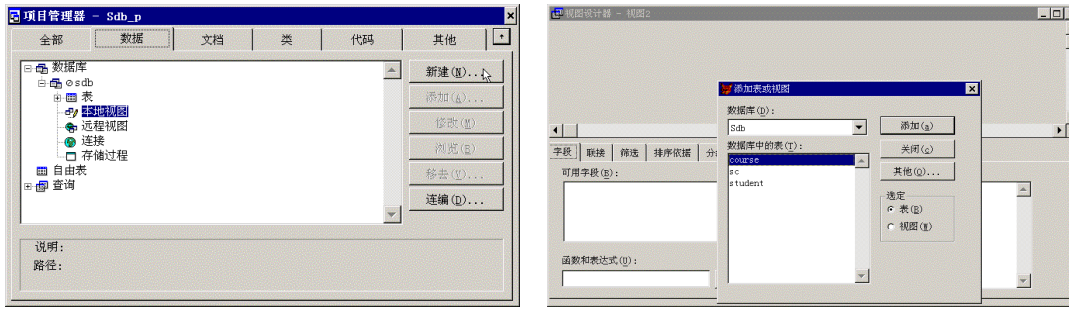


图 12-18　打开“视图设计器”

关闭“添加表和视图”对话框，进入视图设计器。

② 在视图设计器中选择“字段”选项卡，依次添加字段 student.学号、student.姓名，然后，依次构造并添加表达式 AVG（Sc.成绩）as 平均成绩、MIN（Sc.成绩）as 最低分、COUNT（Sc.课程号）as 选课门数，如图 12-19 所示。

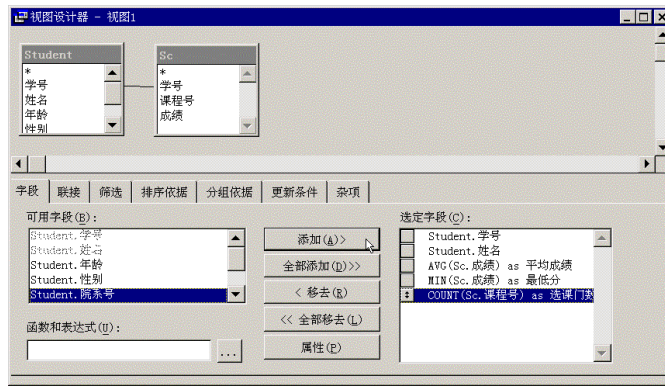


图 12-19　选定字段

在“排序依据”选项卡中，添加排序条件 AVG（Sc.成绩）as 平均成绩，并选中“降序”选项。

在“分组依据”选项卡中，添加分组字段 student.学号。

③ 选择“查询”菜单中的“查看 SQL”选项，打开编辑窗口，将其中的代码复制到系统剪贴板中（选中代码，然后按〈Ctrl〉+〈C〉）。关闭视图设计器。

返回“项目管理器”，选择“代码”选项卡，选中“程序”项，单击“新建”按钮，打开程序编辑器，将系统剪贴板中代码粘贴到编辑器中（按〈Ctrl〉+〈V〉），并添加打开数据库及创建视图的代码：

```
OPEN DATABASE sdb                                    && 打开数据库
CREATE SQL VIEW sview AS;                            && 创建视图
SELECT Student.学号, Student.姓名, AVG（Sc.成绩）as 平均成绩,;
   MIN（Sc.成绩）as 最低分, COUNT（Sc.课程号）as 选课门数;
 FROM    sdb!student INNER JOIN sdb!sc ;
    ON    Student.学号 = Sc.学号;
 GROUP BY Student.学号;
 HAVING 选课门数 > 3;
 ORDER BY 3 DESC
```

④ 保存程序为 T1.PRG，在“项目管理器”中运行程序 T1.PRG。然后在“项目管理器”的“数据”选项卡中可以看到“本地视图”中多了一个视图 sview，如图 12-20 的左图所示。

⑤ 在“项目管理器”中选择“文档”选项卡。选定其中的“报表”项，单击“新建”按钮，在弹出的“新建报表”对话框中，单击“报表向导”按钮，然后在弹出的“向导选取”对话框中直接单击“确定”按钮，打开报表向导。

在步骤 1 中，添加视图 sview 中的所有字段，单击“下一步”按钮，如图 12-20 的右图所示。

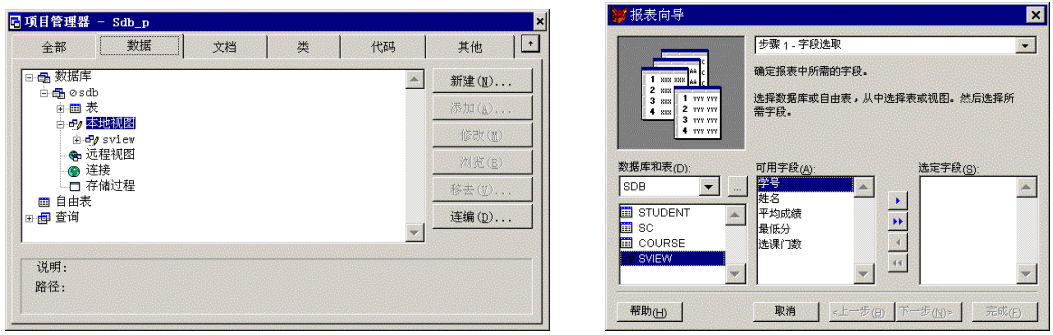


图 12-20　本地视图 sview

在步骤 2 中，单击“下一步”按钮；

在步骤 3 中，选择“随意式”，单击“下一步”按钮；

在步骤 5 中，添加“学号”为排序字段，单击“下一步”按钮；

在步骤 6 中，输入报表标题学生成绩统计一览表。单击“完成”按钮。将报表文件保存为 p_student（扩展名.frx）；

在“项目管理器”的“文档”选项卡中可以看到多了一个报表 p_student，如图 12-21 的左图所示。单击“预览”按钮，可以看到设计的报表如图 12-21 的右图所示。

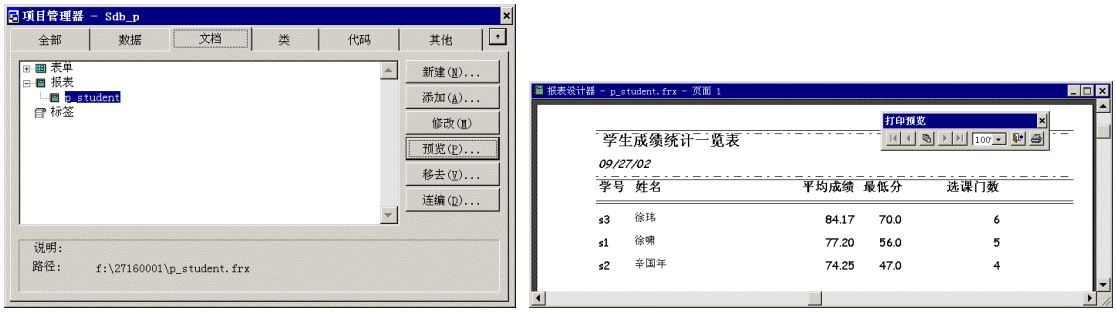


图 12-21　报表 p_student

⑥ 在“项目管理器”中选择“文档”选项卡。选定其中的“表单”项，单击“新建”按钮，在弹出的“新建表单”对话框中，单击“新建表单”按钮，打开表单设计器。

在表单上添加两个命令按钮 Command1 和 Command2，将其 Caption 属性分别改为“浏览”和“打印”，如图 12-22 所示。

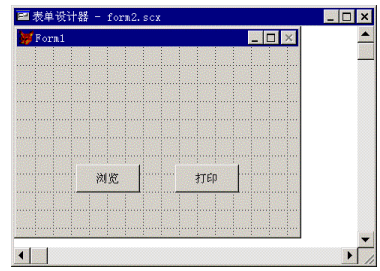


图 12-22　设计表单

编写 Command1 的 Click 事件代码：

```
OPEN DATABASE sdb
SELECT * FROM sview
```

编写 Command2 的 Click 事件代码：

```
REPORT FORM p_student PREVIEW
```

将表单以 Form2 为名，保存在考生文件夹中。

在“项目管理器”中选择“文档”选项卡。展开“表单”项，选择表单 Form2，单击“运行”按钮，表单运行如图 12-23 的左图所示，单击表单上的“浏览”按钮，表单显示如图 12-23 的右图所示的查询结果；单击“打印”按钮，则显示报表预览窗口，如图 12-21 的右图所示。

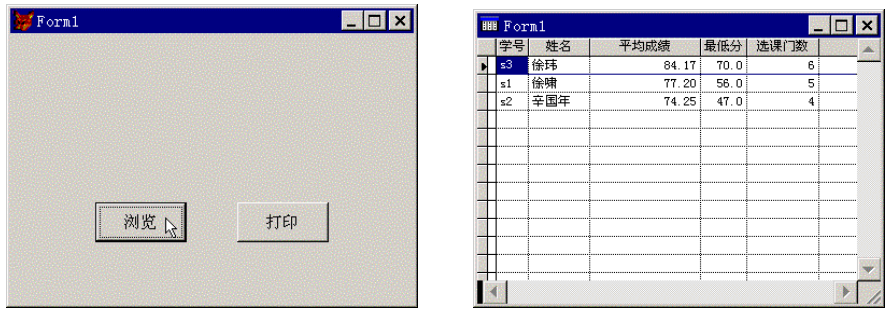


图 12-23　运行表单

注意：程序编写后，一定要运行程序。如果程序编写正确，而没有运行程序，同样会影

响成绩。

12.4 习题

一、基本操作题

在考生文件夹下的“雇员管理”数据库中完成如下操作：

1．为“雇员”表增加一个字段名为 EMAIL**、类型为“字符”、宽度为** 20 **的字段。**

2．设置“雇员”表中“性别”字段的有效性规则，性别取“男”或“女”，默认值为“女”。

3．在“雇员”表中，将所有记录的 EMAIL **字段值使用“部门号”的字段值加上“雇员号”的字段值再加上“** @xxxx.com.cn **”进行替换。**

4．通过“部门号”字段建立“雇员”表和“部门”表间的永久联系。

二、简单应用题

在考生文件夹下完成如下简单应用：

1．请修改并执行名称为 form1 **的表单，要求如下：**

（1）为表单建立数据环境，并将“雇员”表添加到数据环境中；

（2）将表单标题修改为“ XXX **公司雇员信息维护”；**

（3）修改命令按钮“刷新日期”的 click **事件下的语句，使用** SQL **的更新命令，将“雇员”表中“日期”字段值更换成当前计算机的日期值。注意：只能在原语句上进行修改，不可以增加语句行。**

2．建立一个名称为 menu1 **的菜单，菜单栏有“文件”和“编辑浏览”两个菜单。“文件”菜单下有“打开”、“关闭退出”两个子菜单；“浏览”菜单下有“雇员编辑”、“部门编辑”和“雇员浏览”三个子菜单。**

三、综合应用题

在考生文件夹下，对“雇员管理”数据库完成如下综合应用:

1．建立一个名称为 VIEW1 **的视图，查询每个雇员的部门号、部门名、雇员号、姓名、性别、年龄和** EMAIL**。**

2．设计一个名称为 form2 **的表单，表单上设计一个页框，页框有“部门”和“雇员”两个选项卡，在表单的右下角有一个“退出”命令按钮。要求如下**:

（1）表单的标题名称为“商品销售数据输入”；

（2）单击选项卡“雇员”时，在选项卡“雇员”中使用“表格”方式显示 VIEW1 **视图中的记录（表格名称为** grdView1**）；**

（3）单击选项卡“部门”时，在选项卡“部门”中使用“表格”方式显示“部门”表中的记录（表格名称为“ grd **部门”）；**

（4）单击“退出”命令按钮时，关闭表单。

附　录

附录 A　各章习题参考答案

第 1 章参考答案

一、选择题

1．D）　2．A）　3．B）　4．D）　5．A）　6．B）　7．C）　8．A）

二、填空题

1．32
2．**自然连接**
3．**事物之间的联系，即数据的结构**
4．**关系模型，属性，元组**
5．**选择运算，连接运算，投影运算**

第 2 章参考答案

一、选择题

1．D）　2．A）　3．D）　4．B）　5．A）　6．D）

二、填空题

1．**工具、选项**
2．**区域**
3．**代码**
4．**从项目中移去，从磁盘上删除**

第 3 章参考答案

一、选择题

1．C）　2．C）　3．C）　4．D）　5．B）　6．D）　7．D）　8．A）　9．D）　10．B）　11．C）　12．A）　13．C）

二、填空题

1．123456
2．65.00
3．123.457
4．（s * （s – a）* （s – b）* （s – c））^0.5

或 （s * （s – a）* （s – b）* （s – c））^（1 / 2）

或 SQRT（s * （s – a）* （s – b）* （s – c））

5．$\cfrac{a}{b+\cfrac{c}{d+\cfrac{e}{\sqrt{f}}}}$

6．7.7500，5.2500

7．**定义该变量的过程以及该过程所调用的子过程范围**

第 4 章参考答案

一、选择题

1．A） 2．C） 3．C） 4．A） 5．D） 6．B） 7．D） 8．C） 9．B）

二、填空题

1．dbf

2．.cdx，**结构复合压缩索引**

3．**实体**

4．SET RELATION

5．**逻辑表达式**

三、上机题

（1）依次执行如下的命令：

```
SELECT 2
USE customer
LOCATE FOR 客户名 = "三益贸易公司"
no = 客户号
USE order_list
INDEX ON 订单号 TAG ddh
SET ORDER TO ddh
SELECT 1
USE order_detail
SET RELATION TO 订单号 INTO b
INDEX ON 订单号+STR（100000/单价）TAG ddh
COPY TO results1 FOR b->客户号 = no
```

（2）依次执行如下的命令：

```
SELECT 2
USE order_list
INDEX ON 客户号 TAG 客户号
SET ORDER TO 客户号
```

```
SELECT 1
USE customer
SET RELATION TO 客户号 INTO b
INDEX ON 客户号 TAG 客户号
COPY TO result2 FOR 客户号=b->客户号
```

（3）依次执行如下的命令：

```
SELECT 2
USE order_list
INDEX ON 订单号 TAG ddh
SET ORDER TO ddh
SELECT 1
USE order_detail
SET RELATION TO 订单号 INTO b
INDEX ON 订单号 TAG dz
COPY TO results3 FIELD b->订单号,b->订购日期,器件号,器件名,b->总金额
```

（4）依次执行如下的命令：

```
SELECT 2
USE customer
INDEX ON 客户号 TAG kh
SET ORDER TO kh
SELECT 1
USE order_list
SET RELATION TO 客户号 INTO b
INDEX ON 总金额 DESC TAG zj
COPY TO results4 FIELD b->客户号,b->客户名,订单号,总金额
```

（5）依次执行如下的命令：

```
ALTER TABLE order_detail ADD COLUMN 新单价 F（10.2）
SELECT 2
USE order_list
SET ORDER TO 订单号
SELECT 1
USE order_detail
SET RELATION TO 订单号 INTO b
REPL ALL 新单价 WITH 单价 * 0.9 FOR YEAR（b->订购日期）= 2001
REPL ALL 新单价 WITH 单价 * 1.1 FOR YEAR（b->订购日期）= 2002
```

第 5 章参考答案

一、选择题

1．D） 2．C） 3．D） 4．C） 5．A）

二、填空题

1．过滤条件

2．更新

三、上机题

1．

1）创建新查询，依次添加表 customer、order_list 和 order_detail，前两个表的连接条件为 customer.客户号 = order_list.客户号，后两个表的连接条件为 order_list.订单号 = order_detail 订单号；

2）选择显示字段，添加 order_detail 中的所有字段；

3）筛选条件，customer.客户名 = "三益贸易公司"；

4）排序依据，订单号（升序）、单价（降序）；

5）查询去向，表 results5_1。

6）查看 SQL：

```
SELECT Order_detail.订单号, Order_detail.器件号, Order_detail.器件名,;
  Order_detail.单价, Order_detail.数量;
 FROM  订货管理!customer INNER JOIN 订货管理!order_list;
    INNER JOIN 订货管理!order_detail ;
   ON  Order_list.订单号 = Order_detail.订单号 ;
   ON  Customer.客户号 = Order_list.客户号;
 WHERE Customer.客户名 = "三益贸易公司";
 ORDER BY Order_detail.订单号, Order_detail.单价 DESC;
 INTO TABLE results5_1.dbf
```

2．

1）创建新查询，依次添加表 customer、order_list，连接条件为 customer.客户号 = order_list.客户号；

2）选择显示字段，添加表 customer 中的所有字段*；

3）排序依据，customer.客户号；

4）杂项，复选“无重复记录”；

5）查询去向，表 results5_2。

6）查看 SQL：

```
SELECT DISTINCT Customer.*;
 FROM  订货管理!customer INNER JOIN 订货管理!order_list ;
   ON  Customer.客户号 = Order_list.客户号;
 ORDER BY Customer.客户号;
 INTO TABLE results5_2.dbf
```

3．

1）创建新查询，依次添加表 order_list 和 order_detail，两个表的连接条件为 order_list.

订单号 = order_detail **订单号；**

2）**选择显示字段，**order_list.**订单号**、order_list.**订购日期**、order_detail.**器件号**、order_detail.**器件名**、order_list.**总金额**

3）**排序依据，**order_list.**订单号；**

4）**查询去向，表** results5_3。

5）**查看** SQL：

```
SELECT Order_list.订单号, Order_list.订购日期, Order_detail.器件号,;
  Order_detail.器件名, Order_list.总金额;
 FROM  订货管理!order_list INNER JOIN 订货管理!order_detail ;
   ON  Order_list.订单号 = Order_detail.订单号;
 ORDER BY Order_list.订单号;
 INTO TABLE results5_3.dbf
```

4．

1）**创建新查询，依次添加表** customer、order_list，**连接条件为** customer.**客户号** = order_list.**客户号；**

2）**选择显示字段，**customer.**客户号**、customer.**客户名**、order_list.**订单号**、order_list.**总金额；**

3）**排序依据，**order_list.**总金额（降序）；**

4）**查询去向，表** results5_4。

5）**查看** SQL：

```
SELECT Customer.客户号, Customer.客户名, Order_list.订单号,;
  Order_list.总金额;
 FROM  订货管理!customer INNER JOIN 订货管理!order_list ;
   ON  Customer.客户号 = Order_list.客户号;
 ORDER BY Order_list.总金额 DESC;
 INTO TABLE results5_4.dbf
```

5．

1）**创建新查询，添加表** order_detail **；**

2）**选择显示字段，**RIGHT（Order_detail.**订单号**,1） as **订单号**、order_detail.**器件号**、order_detail.**器件名**、MIN（Order_detail.**单价**）as **单价**、SUM（Order_detail.**数量**）as **数量；**

3）**排序依据，**RIGHT（Order_detail.**订单号**,1）as **订单号**、order_detail.**器件号；**

4）**分组依据，**RIGHT（Order_detail.**订单号**,1）as **订单号**、order_detail.**器件号；**

5）**查询去向，表** results5_5。

6）**查看** SQL：

```
SELECT RIGHT（Order_detail.订单号,1） as 订单号, Order_detail.器件号,;
  Order_detail.器件名, MIN（Order_detail.单价） as 单价,;
  SUM（Order_detail.数量） as 数量;
 FROM 订货管理!order_detail;
 GROUP BY 1, Order_detail.器件号;
```

```
ORDER BY 1, Order_detail.器件号;
INTO TABLE results5_5.dbf
```

6．

1）打开数据库：学生管理.dbc；

2）创建新视图，添加表 student.dbf 和 cj.dbf；

3）选择显示字段或函数表达式：student.学号、student.姓名、AVG（cj.成绩）AS 平均成绩、MIN（cj.成绩）AS 最低分、COUNT（cj.课程号）AS 选课门数；

4）联接：inner jion，student.学号 = cj.学号

5）排序依据：AVG（cj.成绩）AS 平均成绩；

6）分组依据：cj.学号，满足条件：COUNT（cj.课程号）> 2

7）查看 SQL：

```
SELECT Student.学号, Student.姓名, AVG（Cj.成绩）, MIN（Cj.成绩）,;
  COUNT（Cj.课程号）;
FROM  学生管理!student INNER JOIN 学生管理!cj ;
  ON  Student.学号 = Cj.学号;
GROUP BY Cj.学号;
HAVING COUNT（Cj.课程号）> 2;
ORDER BY 3 DESC
```

8）保存视图名为 sview5_6。

7．

1）打开数据库：salary_db.dbc；

2）创建新视图，添加表 salarys；

3）选择显示字段或函数表达式：部门号、雇员号、姓名、工资、补贴、奖励、失业保险、医疗统筹、（工资+补贴+奖励）– （失业保险+医疗统筹）AS 实发工资；

4）排序依据：部门号（降序）；

5）查看 SQL：

```
SELECT Salarys.部门号, Salarys.雇员号, Salarys.姓名, Salarys.工资,;
  Salarys.补贴, Salarys.奖励, Salarys.失业保险, Salarys.医疗统筹,;
  Salarys.工资+ Salarys.补贴+ Salarys.奖励- Salarys.医疗统筹- Salarys.失业保险 as 实发工资;
FROM salary_db!salarys;
ORDER BY Salarys.部门号 DESC
```

6）保存视图名为 sview5_8。

第 6 章参考答案

一、选择题

1．D） 2．B） 3．C） 4．A） 5．A） 6．A） 7．B） 8．A） 9．D） 10．C） 11．B） 12．B） 13．B） 14．D）

二、填空题

1．INTO，VALUES
2．SUM，student
3．UPDATE，WHERE，FROM
4．INTO DBF 或 INTO TABLE
5．SUM，VAG

三、上机题

1．

```
SELECT * FROM stock WHERE 单价 BETWEEN 7.48 AND 6.50
```

说明：上式如用 VFP 的条件为：

```
SELECT * FROM stock WHERE 单价>=7.48 AND 单价<=6.50
```

2．

```
SELECT * FROM stock WHERE 单价 IN （14.59,15.20 ）
```

说明：上式改为 VFP 条件为：

```
SELECT * FROM stock WHERE 单价=14.59 OR 单价=15.20
```

3．

```
SELECT * FROM stock WHERE 交易所 != "上海"
```

或

```
SELECT * FROM stock WHERE NOT（交易所 = "上海"）
```

4．

```
OPEN DATABASE stock
CREATE VIEW v_sal AS SELECT 股票名称, 交易所 FROM stock
```

5．

```
OPEN DATABASE stock
DROP VIEW v_sal
```

6．

```
SELECT * FROM order_list INTO DBF results;
  WHERE 总金额 > （SELECT AVG（总金额）FROM order_list）;
  ORDER BY 客户号
```

7．

```
SELECT * FROM order_list WHERE 客户号 IN;
  （SELECT 客户号 FROM customer WHERE 客户名 = "三益贸易公司"）;
```

```
   INTO CURSOR tt
SELECT * FROM order_detail WHERE 订单号 IN;
   （SELECT 订单号 FROM tt）;
   ORDER BY 订单号,单价 DESC;
   INTO DBF results
```

或

```
SELECT a.* FROM order_detail a, order_list b, customer customer;
   WHERE 客户名 = "三益贸易公司" ;
      AND a.订单号 = b.订单号 AND b.客户号 = customer.客户号;
   ORDER BY a.订单号,单价 DESC;
   INTO DBF results2_2
```

8．

```
SELECT * FROM customer WHERE 客户号 IN;
   （SELECT 客户号 FROM order_list）;
   ORDER BY 客户号;
   INTO DBF results3
```

9．

```
SELECT Order_list.订单号, Order_list.订购日期, Order_detail.器件号,;
   Order_detail.器件名, Order_list.总金额;
   FROM   Order_list INNER JOIN Order_detail ;
   ON   Order_list.订单号 = Order_detail.订单号;
   ORDER BY Order_list.订单号 , Order_list.总金额 DESC;
   INTO TABLE results.DBF
```

10．

```
SELECT CUST.客户号,CUST.客户名,ol.订单号,ol.总金额 ;
   FROM customer CUST inner JOIN order_list ol ON CUST.客户号=ol.客户号 ;
   ORDER BY 总金额 DESC
```

11．

```
OPEN DATA salary_db
* 将两个表 salarys 与 c_salary1 左联接为临时表
SELECT a.*,b.雇员号 AS b 雇员号,b.工资 AS b 工资 FROM;
   salarys a LEFT JOIN c_salary1 b ON a.雇员号=b.雇员号 INTO DBF ls
* 对临时表中的工资进行更新
UPDATE ls SET 工资 = b 工资 WHERE 雇员号=b 雇员号
* 删除临时表中多余的字段
ALTER TABLE ls DROP b 雇员号
ALTER TABLE ls DROP b 工资
* 删除表 salarys
DROP TABLE salarys
```

```
* 重新生成表 salarys
SELECT * FROM ls INTO DBF salarys
* 关闭表 salarys
SELECT salarys
USE
* 将表 salarys 添加到数据库中
ADD TABLE salarys
```

第 7 章参考答案

一、选择题

1．C）　2．C）　3．B）　4．C）　5．C）　6．B）

二、填空题

1．10 < S <= 100、1 < S <= 10、S <= 1、

2．　A0　A10　A12　A2
　　115

3．　6　4　2

三、上机题

1．编写程序如下：

```
SET TALK OFF
CLEAR
INPUT "输入考试成绩:" TO chj
DO CASE
   CASE chj < 60
      Dj = "不合格"
   CASE chj < 90
      Dj = "通过"
   OTHERWISE
      Dj = "优秀"
ENDCASE
? "成绩等级:" + dj
SET TALK ON
```

2．修改程序如下：

```
CLEAR
STORE 0 TO X, Y
DO WHILE .T.
   X=X+1
   DO CASE
      CASE MOD（X,5）=0
```

```
        CASE X>30
          EXIT
        CASE X <=30
          LOOP
      ENDCASE
      Y=Y+X
    ENDDO
    ? Y
```

3．编写程序如下：

```
SELECT 2
USE order_list
INDEX ON 订单号 TO li
SELECT 1
USE order_detail
SET RELATION TO 订单号 INTO order_list
REPL ALL 新单价 WITH 单价 * 0.9 FOR Year（order_list.订购日期）== 2001
REPL ALL 新单价 WITH 单价 * 1.1 FOR Year（order_list.订购日期）== 2002
```

4．编写程序如下：

```
SELECT 2
USE 单价调整表
INDEX ON 商品号 TO sy
SELECT 1
USE 商品表
REPL ALL 单价 WITH 出厂单价 * .1 FOR Left（商品号,2）= "10"
SET RELATION TO 商品号 INTO b
REPL ALL 出厂单价 WITH b->出厂单价 FOR 商品号 = b->商品号
```

5．编写程序如下：

```
SELECT 2
USE C_SALARY1
INDEX ON 雇员号 TO li
SELECT 1
USE SALARYS
COPY ALL TO BAK_SALARYS
SET RELATION TO 雇员号 INTO B
REPL ALL 工资 WITH B->工资 FOR 雇员号 = B->雇员号
```

第 8 章参考答案

一、选择题

1．B）　2．D）　3．B）　4．C）　5．D）

二、上机题

1．（1）按钮 Command1 的 Click 事件代码改为：

thisform.Cption = "简单应用"

（2）按钮 Command2 的 Click 事件代码改为：

thisform.grid1.RecordSource = "order_list.dbf"

（3）按钮 Command3 的 Click 事件代码改为：

thisform.Release

2．（1）按钮 Command1 的 Click 事件代码改为：

thisform.Cption = "商品销售数据输入"

（2）按钮 Command2 的 Click 事件代码改为：

DO Form sellcomm

（3）按钮 Command3 的 Click 事件代码改为：

REPORT FORM print1 PREVIEW

3．（1）按钮 Command1 的 Click 事件代码为：

DO change_c

（2）按钮 Command2 的 Click 事件代码为：

THISFORM.Release

第 9 章参考答案

一、选择题

1．D）　2．C）　3．B）　4．D）　5．C）　6．B）　7．B）

二、填空题

1．Value，Caption
2．AutoSize
3．THISFORM.Text1.Setfocus
4．0 – （无）、0 – 透明
5．Interval
6．Enabled
7．下拉组合框、下拉列表框、Style、0、2

三、上机题

1．（1）修改代码如下：

```
&&**********改正**********
if thisform.text1.Text = thisform.text2.Text
    wait "欢迎使用……" window timeout 1
&&**********改正**********
    thisform.Release
else
    wait "用户名或口令不对，请重新输入……" window timeout 1
endif
```

（2）在属性窗口设置 Text2 控件的 PasswordChar 属性值为：*。

2．表单设计如图所示，其中命令按钮“确定”Command1 的 Click 事件代码为：

```
IF THISFORM.Check1.value = 1
  THISFORM.Text1.value = THISFORM.Edit1.SelText
ELSE
  THISFORM.Edit1.SelText = THISFORM.Text1.value
ENDIF
```

3．创建一个新表单，打开表单设计器。（1）添加数据环境：在表单上单击鼠标右键，选择快捷菜单中的“数据环境”命令。然后在打开的数据环境中单击鼠标右键，选择快捷菜单中的“添加”命令。在“添加表或视图”对话框中选择表 salarys。（2）添加表格：将添加到数据环境中的表 salarys 拖到表单中，松开鼠标，表单上出现一个名为 grdSalarys 的表格控件。（3）添加命令按钮：单击“表单控件”工具栏中的“命令按钮”，在表单上画出一个命令按钮，名为 Command1，将其 Caption 属性改为“退出浏览”；为其编写 Click 事件代码：

```
THISFORM.Release
```

4．设计表单如图 9-49 所示。其中命令按钮的事件代码如下：

“生成”（Command1）命令按钮的 Click 事件代码：

```
DO CASE
CASE THISFORM.optiongroup1.VALUE = 1
  SELECT * FROM salary_db!sview
  SELECT * FROM salary_db!sview INTO TABLE gz1.DBF
CASE THISFORM.optiongroup1.VALUE = 2
  SELECT * FROM salary_db!dept
  SELECT * FROM salary_db!dept INTO TABLE bm1.DBF
CASE THISFORM.optiongroup1.VALUE = 3
  SELECT dept.*, SUM（Salarys.工资）AS 工资, SUM（Salarys.补贴）AS 补贴,;
    SUM（Salarys.奖励）AS 奖励, SUM（Salarys.医疗统筹）AS 医疗统筹,;
    SUM（Salarys.失业保险）AS 失业保险;
    FROM  salary_db!dept INNER JOIN salary_db!Salarys ;
    ON  dept.部门号 = Salarys.部门号;
    GROUP BY dept.部门号;
    INTO TABLE bm1.DBF
ENDCASE
```

“退出”（Command2）命令按钮的 Click 事件代码：

```
THISFORM.Release
```

5．创建表单如图 9-50 所示，其中各个文本框的 ControlSource 属性均设为相应的字段（与之绑定）。编写命令按钮的事件代码如下：

“向前”按钮 Command1 的 Click 事件代码：

```
IF NOT EOF（）
  SKIP 1
ENDIF
THISFORM.Refresh
```

“向后”按钮 Command2 的 Click 事件代码：

```
IF NOT BOF（）
  SKIP -1
ENDIF
THISFORM.Refresh
```

第 10 章参考答案

一、选择题

1．B）　2．C）

二、填空题

1．条形菜单，弹出式菜单

2．常规选项、ShowWindow、Init

3．RightClick

三、上机题

1．设计方法如下：

（1）创建菜单和子菜单

选择“文件”菜单中的“新建”命令，单击“新建”对话框底部的“菜单”按钮，单击“新文件”按钮，然后在打开的“新建菜单”对话框中单击“菜单”按钮，打开“菜单设计器”。

在菜单设计器中输入菜单名“文件”、“浏览”和“退出”，如图 1 所示。并为“浏览”菜单项设置“跳过”条件：DBF（）== ""。

选中“文件”菜单项，将“结果”设为“子菜单”，单击“创建”按钮，编辑子菜单项，用鼠标单击“插入栏…”按钮，打开“插入系统菜单栏”对话框，如图 2 所示。

插入所需的菜单项打开；再增加一个菜单选项关闭，其结果设为“命令”，并编写命令 USE，如图 3 所示。

（2）编写菜单代码

在“菜单级”下拉列表框中选择“菜单栏”，回到顶层菜单。分别为“浏览”和“退出”菜单项编写过程代码：

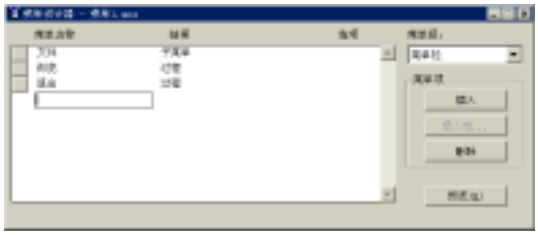


图 1　输入菜单名

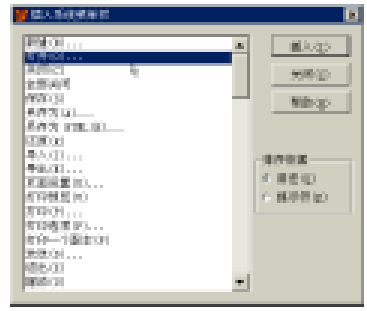


图 2　“ 插入系统菜单栏 ” 对话框

“ 浏览 ”:

```
BROWSE
```

“ 退出 ”:

```
SET SYSMENU NOSAVE
SET SYSMENU TO DEFAULT
```

（3）最后生成菜单文件 “ 菜单 1.mpr ” 即可。

2．设计方法如下：

（1）设计快捷菜单

选择 “ 文件 ” 菜单中的 “ 新建 ” 命令，单击 “ 新建 ” 对话框底部的 “ 菜单 ” 按钮，单击 “ 新文件 ” 按钮，在打开的 “ 新建菜单 ” 对话框中，单击 “ 快捷菜单 ”，打开 “ 快捷菜单设计器 ”。

按照题目要求，输入菜单选项的名称，并设置 “ 结果 ” 为 “ 过程 ”，如图 4 所示。其中名称为 “ \- ” 的菜单项表示分隔线。

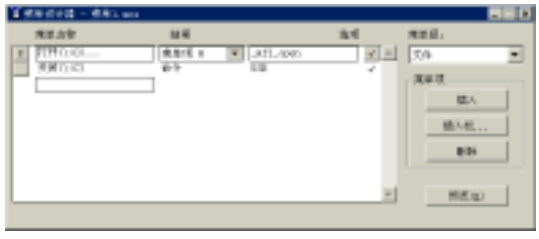


图 3　文件子菜单

图 4　输入菜单选项的名称

依次单击 “ 结果 ” 右边的 “ 创建/编辑 ” 按钮，打开代码编辑器，为各菜单项输入过程代码。

“ 表文件名 ” 菜单项的过程代码：

```
My1.RowSourceType = 7
My1.RowSource = "*.dbf"
My1.Requery
```

“ 表字段名 ” 菜单项的过程代码：

```
a = My1.value
use My1.List（2）+ a
My1.RowSourceType = 8
My1.RowSource = My1.List（2）+ a
My1.Requery
```

“ 组合/列表框 ” 菜单项的过程代码：

```
IF My1.Style = 0
   My1.Style = 2
ELSE
   My1.Style = 0
ENDIF
```

关闭编辑器，返回菜单设计器。

选择系统菜单“ 显示 ”菜单中的“ 常规选项 ”命令，打开“ 常规选项 ”对话框。选中“ 设置 ” 复选框，如图 5 **所示，单击“ 确定 ” 按钮，打开编辑器窗口。在其中输入“ 设置 ” 代码：**

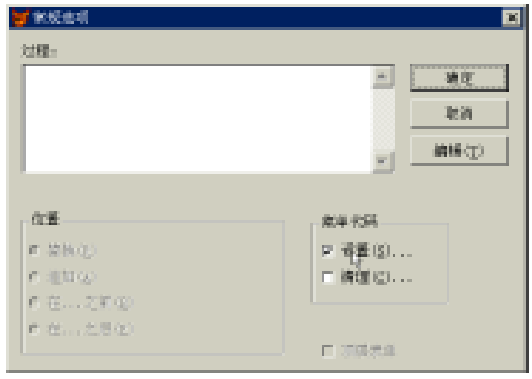


图 5　**选中“ 设置 ” 复选框**

```
PARAMETERS My1
```

关闭编辑器，返回“ 快捷菜单设计器 ”。

然后选择“ 显示 ”菜单中的“ 常规选项 ”命令，打开“ 常规选项 ”对话框。选中“ 清理 ” 复选框，单击“ 确定 ” 按钮，打开编辑器窗口。在其中输入“ 清理 ” 代码：

```
release popups 菜单 2
```

关闭编辑器，返回“ 快捷菜单设计器 ”。

完成菜单的定义后，选择主菜单“ 菜单 ”菜单中的“ 生成 ”命令，选择“ 是 ”，在“ 另存为 ” 对话框中输入菜单名菜单 2，**“ 确定 ” 后显示“ 生成菜单 ” 对话框，单击“ 生成 ” 钮，生成菜单程序：菜单** 2.mpr，**至此完成菜单的创建工作。**

（2**）设计表单**

在表单中增加一个组合框 Combo1，**然后编写组合框的** RightClick **事件代码**:

```
DO 菜单 2.mpr WITH THIS, .T.
```

3．在表单的 Init 事件代码中添加如下代码：

DO 菜单 1.mpr

第 11 章参考答案

一、选择题

1．C）　2．D）　3．B）　4．A）　5．B）

二、填空题

1．报表的数据源、报表布局
2．快速报表、报表向导、报表设计器
3．字段、变量、表达式的值
4．组标头带区、组注脚带区
5．索引字段
6．列标头、列注脚

三、上机题

操作步骤如下：

（1）单击工具栏上的“新建”按钮，在弹出的“新建”对话框中选择“报表”，并单击“新建文件”，自动打开报表设计器，并出现一个空白报表。

（2）选择“报表”菜单中的“快速报表”选项，在“打开”对话框中选择数据源 order_detail.dbf，系统弹出“快速报表”对话框。

（3）在“快速报表”对话框中，单击字段按钮，打开“字段选择器”为报表选择可用的字段，如图 6 所示。单击“确定”按钮，关闭“字段选择器”返回“快速报表”对话框。

图 6　“字段选项器”对话框

（4）在“快速报表”对话框中，单击“确定”按钮，快速报表便出现在“报表设计器”中。

（5）选择“报表”菜单中的“标题/总结”选项，打开“标题/总结”对话框。在其中选中“标题带区”复选框，并单击“确定”按钮，关闭对话框。

（6）在添加的“标题带区”放置一个标签控件，输入标签的内容器件清单，作为报表的标题，并且通过“格式”菜单中的“字体”选项修改标题的字体。

（7）用鼠标右键单击页注脚区的 DATE（）域控件，选择快捷菜单中的“属性”选项，弹出“报表表达式”对话框。将原来的 DATE（）改为 TIME（）以显示当前的时间。

（8）两次单击“确定”按钮，返回“报表设计器”。单击工具栏上的“保存”按钮，将该报表保存为 report1.frx 文件。

第 12 章参考答案

一、基本操作题

1．在“性别”字段的有效性规则中输入表达式：性别="男".OR.性别="女"；

2．在命令窗口执行命令：repl all EMAIL with 部门号 + 雇员号 + "@xxxx.com.cn"

二、简单应用题

1．单击“打开”按钮，在“打开”对话框中选择考生文件夹的表单 form1，单击“确定”按钮，打开表单设计器。

① 在表单中单击鼠标右键，选择快捷菜单中的“数据环境”项，打开数据环境设计器。添加表：雇员，然后关闭数据环境设计器；

② 在属性窗口中选中表单 Form1 的 Caption 属性，改为：XXX 公司雇员信息维护；

③ 修改命令按钮 Command1（刷新日期）的 Click 事件代码，将原代码：

```
UPDATE ALL 日期 WITH DATE()
```

改为：

```
UPDATE 雇员 SET 日期 = DATE()
```

2．单击“新建”按钮，在“新建”对话框中选择“菜单”项，单击“新建文件”按钮，在弹出的“新建菜单”对话框中单击“菜单”按钮，打开菜单设计器。

① 在“菜单名称”栏中输入文件，“结果”栏选择子菜单，单击“创建”按钮，进入子菜单的设计：在“菜单名称”栏依次输入“打开”和“关闭退出”。

在“菜单级”列表框中选择“菜单栏”，返回顶级菜单。

② 在“菜单名称”栏中输入浏览，“结果”栏选择子菜单，单击“创建”按钮，进入子菜单的设计：在“菜单名称”栏依次输入“雇员编辑”、“部门编辑”和“雇员浏览”。

在“菜单级”列表框中选择“菜单栏”，返回顶级菜单。

③ 在“菜单”菜单中选择“生成”项，生成菜单文件 menu1.mnt、menu1.mnx、menu1.mpr。

三、综合应用题

1．打开“雇员管理”数据库，使用视图设计器设计视图 VIEW1，或使用如下代码生成视图：

```
OPEN DATABASE 雇员管理
CREAT VIEW view1 AS;
  SELECT 雇员.部门号, 部门.部门名, 雇员.雇员号, 雇员.姓名, 雇员.性别,;
  雇员.年龄, 雇员.email;
```

```
FROM    雇员管理!部门 INNER JOIN 雇员管理!雇员 ;
ON    部门.部门号 = 雇员.部门号
```

2．选择“文件”菜单中的“新建”项，在“新建”对话框中选择“表单”项，单击“新建文件”按钮，打开表单设计器。

① 在表单上添加一个页框控件 PageFrame1 和一个命令按钮 Command1。

修改表单的 Caption 属性为商品销售数据输入；

修改页框 PageFrame1 中两个页对象 Page1、Page2 的 Caption 属性分别为部门、雇员。

修改命令按钮 Command1 的 Caption 属性为退出。

② 在表单中单击鼠标右键，选择快捷菜单中的“数据环境”，打开数据环境设计器。

在数据环境设计器中添加视图 view1 和“部门”表。

用鼠标右键单击页框控件，选择快捷菜单中的“编辑”项，使页框控件处于编辑状态。然后，在数据环境中用鼠标左键按住“部门”表的蓝色标题栏，拖向页框中，松开鼠标后，在“部门”页中出现表格控件 grd 部门；

选定“雇员”页，在数据环境中用鼠标左键按住“view1”视图的蓝色标题栏，拖向页框中，松开鼠标后，在“雇员”页中出现表格控件 grdview1。

调整两个表格的大小和位置，如图 7 所示。

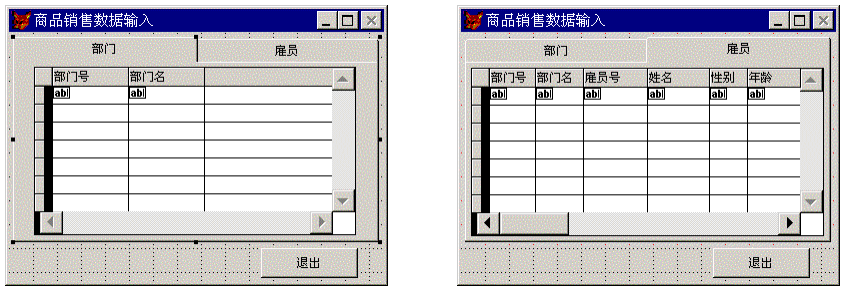


图 7　设计表单

③ 编写命令按钮 Command1 的 Click 事件代码：

```
THISFORM.RELEASE
```

将表单以 Form2 为名，保存在考生文件夹中。

④ 运行表单，表单显示如图 8 所示。

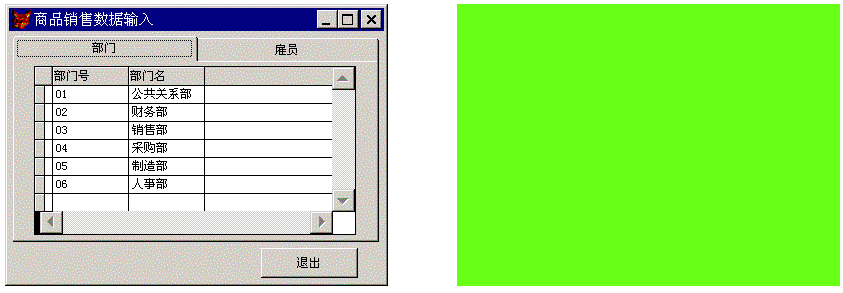


图 8　运行表单

附录 B　全国计算机等级考试二级笔试模拟试卷

Visual FoxPro 程序设计

（考试时间 90 分钟，满分 100 分）

一、选择题（(1) ～ (35) 每小题 2 分，共 70 分）

下列各题 A)、B)、C)、D) 四个选项中，只有一个选项是正确的，请将正确选项填涂在答题卡相应位置上，答在试卷上不得分。

(1) 在 Visual FoxPro 系统主菜单下，以下（　　）操作不能对已经进行逻辑删除的记录进行恢复。

A）在【显示】菜单下选择【浏览】子菜单，并在浏览窗口移动鼠标，在要进行恢复的每个记录上，将删除标记栏用鼠标左键去掉

B）在命令窗口使用 RECALL 命令

C）在【显示】菜单下选择【浏览】子菜单，在浏览窗口选择【程序】菜单，再选择【恢复记录】子菜单

D）在【显示】菜单下选择【浏览】子菜单，在浏览窗口选择【表】菜单，再选择【恢复记录】子菜单

(2) 在两个数据表中选一个相同的字段作为关键字段，把其中一个数据表关键字段称为原始关键字段，该字段值是重复的，而把另一个数据表中的关键字段称为外来关键字段，该字段也是重复的，则这两个表之间建立了（　　）关系。

A）多对多　　B）一对一　　C）一对多　　D）多对一

(3) 关系模型的一个关系可用一张二维表来表示，它相当于 Visual FoxPro 中的一个（　）。

A）数据库　　B）记录　　C）字段　　D）表

(4) 项目管理器可以有效地管理数据表、表单、数据库、菜单、类、程序和其他文件，并且把它们编译成（　　）文件。

A）全部扩展名为.APP 的文件　　B）全部扩展名为.EXE 的文件

C）扩展名为.APP 或.EXE 的文件　　D）全部扩展名为.PRG 的文件

(5) 数据库中有“完成定额否”(逻辑型）字段，完成定额其值为真，否则其值为假。另外有“奖金”(数值型）字段，现给完成定额者发奖金 200 元，没完成者不发，错误的命令是（　　）。

A）REPLACE 奖金 WITH 200 FOR 完成定额否 = .T.

B）REPLACE ALL 奖金 WITH 200 FOR 完成定额否 = .T.

C）REPLACE 奖金 WITH 200 FOR 完成定额否

D）REPLACE 奖金 WITH 200 FOR !完成定额否

（6）选择当前未使用的工作区中最小编号的工作区的命令是（　　）。

A）SELECT 0　　B）SELECT 1　　C）SELECT MIN　　D）SELECT -1

（7）以下关于视图的说法中，错误的是（　　）。

A）视图可以对数据库中数据按指定内容和顺序进行检索

B）试图可以更新数据

C）视图可以脱离数据库单独存在

D）视图必须依赖数据库而存在

（8）在参照完整性生成器中选择“删除规则”选项卡，可以设置关联数据表间的删除规则，其中（　　）按钮可以删除子表中的所有相关记录。

A）关联　　B）忽略　　C）限制　　D）级联

（9）由如下程序：

```
SET TALK OFF
CLEAR
AA = "全国计算机等级考试"
BB = "2003"
CC = "一"
?AA
??BB + "年第" + CC + "次考试"
```

执行程序后，屏幕显示（　　）。

A）全国计算机等级考试
　　2003 年第一次考试

B）全国计算机等级考试 2003 年第一次考试

C）全国计算机等级考试 BB 年第 CC 次考试

D）全国计算机等级考试 BB+年第+CC+次考试

（10）假定字符串 A = "123"，B = "234"。则下列表达式的运算结果为逻辑假的是（　　）。

A）.NOT.（A = B）.OR. B $（"13579"）

B）.NOT. A $ （"ABC"）.AND.（A< >B）

C）.NOT.（A < > B）

D）.NOT.（A >= B）

（11）将“复选框”控件的 Value 属性设置为（　　）时，复选框显示灰色。

A）0　　B）1　　C）2　　D）3

（12）设学生数据表当前记录的“计算机”字段值是 89，执行以下程序段后，屏幕输出（　　）。

```
DO CASE
  CASE 计算机 < 60
    ? "计算机成绩是：" + "不及格"
  CASE 计算机 >= 60
    ? "计算机成绩是：" + "及格"
  CASE 计算机 >= 70
```

```
    ? "计算机成绩是：" + "中"
  CASE 计算机 >= 80
    ? "计算机成绩是：" + "良"
  CASE 计算机 >= 90
    ? "计算机成绩是：" + "优"
ENDCASE
```

A）计算机成绩是：不及格　　B）计算机成绩是：及格

C）计算机成绩是：良　　D）计算机成绩是：优

（13）Visual FoxPro 中的 DO CASE – ENDCASE 语句属于（　　）。

A）顺序结构　　B）选择结构　　C）循环结构　　D）模块结构

（14）打开 Visual FoxPro “项目管理器”中的“文档”选项卡，其中包含（　　）。

A）表单（Form）文件　　B）报表（Report）文件

C）标签（Label）文件　　D）以上三种文件

（15）在 Visual FoxPro 中进行逻辑删除时，“删除”窗口中“作用范围”Next 的作用是（　　）。

A）对数据表中的全部记录进行逻辑删除

B）对数据表中的指定的某条记录进行逻辑删除

C）对数据表中从当前记录开始以后（含当前记录）的 N 条记录进行逻辑删除

D）对数据表中从当前记录开始以后（不含当前记录）的 N 条记录进行逻辑删除

（16）假定 Student 数据表中前 6 条记录均为男生的记录，执行以下命令序列后，记录指针定位在（　　）。

```
USE Student
GOTO 3
LOCATE NEXT 3 FOR 性别 = "男"
```

A）第 3 条记录上　　B）第 4 条记录上

C）第 5 条记录上　　D）第 6 条记录上

（17）用 DIMENSION Q（3, 5）命令定义了一个数组，该数组的下标变量数目是（　　）。

A）15　　B）24　　C）8　　D）10

（18）当前数据库中有 5 个字段：学号（C, 4）、姓名（C, 6）、年龄（N, 2）、性别（L）、联系电话（C, 12），记录指针指向一个非空的记录。要使用 SCATTER TO X 命令把当前记录的字段值存到数组 X 中，数组 X（　　）。

A）不必事先定义

B）必须用 DIMENSION X 命令事先定义

C）必须用 DIMENSION X（5）命令事先定义

D）必须用 DIMENSION X（1）, X（2）, X（3）, X（4）, X（5）命令事先定义

（19）下列叙述中，错误的是（　　）。

A）属性用于描述对象的状态，方法用于表示对象的行为

B）基于同一个类产生的两个对象可以分别设置自己的属性值

C）事件代码也可以像方法一样被显式调用

D）在新建一个表单时，可以添加新的属性、方法和事件

（20）在 SQL 查询中，使用 WHERE 子句指出的是（　　）。

A）查询目标　　B）查询结果　　C）查询条件　　D）查询视图

（21）～（35）使用的数据如下：

当前盘当前目录下有数据库 db_人事，其中有数据库表：教师.dbf，该数据库表的内容是：

职工号	姓名	职称	年龄	工资	系编号
11020001	肖天海	副教授	35	2000.00	01
11020002	王岩盐	教授	40	3000.00	02
11020003	刘星魂	讲师	25	1500.00	01
11020004	张月新	讲师	30	1500.00	03
11020005	李明玉	教授	34	2000.00	01
11020006	孙民山	教授	47	2100.00	02
11020007	钱无名	教授	49	2200.00	03

（21）执行如下 SQL 语句后

```
SELECT * FROM 教师 INTO DBF 教师 ORDER BY 年龄
```

A）系统会提示出错信息

B）会生成一个按“年龄”升序排序的表文件，将原来的教师.dbf 文件覆盖

C）会生成一个按“年龄”降序排序的表文件，将原来的教师.dbf 文件覆盖

D）不会生成排序文件，只在屏幕上显示一个按“年龄”升序排序的结果

（22）有如下 SQL SELECT 语句：

```
SELECT * FROM 教师 WHERE 工资 BETWEEN 2000 AND 2500
```

与该语句等价的是

A）SELECT * FROM teacher WHERE 工资 <= 2500 .AND. 工资 >= 2000

B）SELECT * FROM teacher WHERE 工资 < 2500 .AND. 工资 > 2000

C）SELECT * FROM teacher WHERE 工资 >= 2500 .AND. 工资 <= 2000

D）SELECT * FROM teacher WHERE 工资 > 2500 .AND. 工资 < 2000

（23）有如下 SQL SELECT 语句：

```
SELECT COUNT（DISTINCT 工资）INTO ARRAY a FROM 教师
```

执行该语句后（　　）。

A）a[1]的内容是 7　　B）a[1]的内容是 5

C）a[0]的内容是 7　　D）a[0]的内容是 5

（24）有如下 SQL SELECT 语句：

```
SELECT DISTINCT 系编号 FROM 教师 WHERE 工资 > 2000
```

执行该语句后，屏幕显示的查询记录个数是（　　）。

A）0　　B）1　　C）2　　D）3

（25）有如下 SQL SELECT 语句：

SELECT 系编号, AVG（工资）AS 平均工资 FROM 教师 GROUP BY 系编号 INTO DBF temp

执行该语句后 temp 表中第 2 条记录的“平均工资”字段的内容是（　　）。

A）1833.33　　B）2550.00　　C）1850.00　　D）3000.00

（26）有如下 SQL SELECT 语句：

```
SELECT * FROM 教师 WHERE 系编号 = "02" ;
    AND 工资> ANY （SELECT 工资 FROM 教师 WHERE 系编号 = "01"）
```

执行该语句后，屏幕显示的查询记录个数是（　　）。

A）1　　B）2　　C）3　　D）4

（27）有如下 SQL SELECT 语句：

```
CREAT VIEW 教师_v AS SELECT * FROM 教师 WHERE 系编号 = "01"
```

执行该语句后产生的视图中包含的记录个数是（　　）。

A）1　　B）2　　C）3　　D）4

（28）有如下 SQL SELECT 语句：

```
CREAT VIEW v_教师 AS;
    SELECT AVG（年龄）AS 平均年龄, AVG（工资）AS 平均工资 ;
    FROM 教师 GROUP BY 系编号
```

执行该语句后产生的视图中包含的字段名称是（　　）。

A）年龄、工资　　B）平均年龄、平均工资

C）年龄、工资、系编号　　D）平均年龄、平均工资、系编号

（29）有如下 SQL SELECT 语句：

```
CREAT VIEW v_教师 AS;
    SELECT DISTINCT 年龄 FROM 教师 ;
    WHERE 年龄 =（SELECT MIN（年龄）FROM 教师）
```

执行该语句后产生的视图中包含的记录个数是（　　）。

A）1　　B）2　　C）3　　D）4

（30）求每个系的平均年龄的 SQL 语句是（　　）。

A）SELECT 系编号, AVG（年龄）FROM 教师 GROUP BY 年龄

B）SELECT 系编号, AVG（年龄）FROM 教师 ORDER BY 年龄

C）SELECT 系编号, AVG（年龄）FROM 教师 GROUP BY 系编号

D）SELECT 系编号, AVG（年龄）FROM 教师 ORDER BY 系编号

（31）将教师表的“工资”字段名称改为“基本工资”，应使用的 SQL 语句是（　　）。

A）ALTER TABLE 教师 工资 WITH 基本工资

B）ALTER TABLE 教师 ALTER 工资 TO 基本工资

C）ALTER TABLE 教师 RENAME 工资 TO 基本工资

D）ALTER 教师 RENAME 工资 TO 基本工资

（32）在当前盘当前目录下删除教师表的命令是（　　）。

A）DROP 教师　　B）DELETE TABLE 教师

C）DROP TABLE 教师　　D）DELETE 教师

（33）在 Visual FoxPro 中，以下说法错误的是（　　）。

A）字符型字段中可以存放空格

B）日期型字段中专门存放日期数据

C）备注型字段中可以一个字都不存放

D）逻辑字段中可以存放数值

（34）在下面的 DO 循环中，一共要循环（　　）次。

```
x = 7
y = 12
DO WHILE y >= x
    y = y - 1
ENDDO
```

A）4　　B）5　　C）6　　D）7

（35）在数据库设计器中，建立两个表之间的一对多联系是通过（　　）实现的。

A）“一方”表的主索引或候选索引，“多方”表的普通索引

B）“一方”表的主索引，“多方”表的普通索引或候选索引

C）“一方”表的普通索引，“多方”表的主索引或候选索引

D）“一方”表的普通索引，“多方”表的候选索引或普通索引

二、填空题（每空 2 分，共 30 分）

请将正确答案分别写在答题卡中序号为【1】至【15】的横线上，答在试卷上不得分。

（1）关系型数据库管理系统的三种基本关系运算是选择、投影和__【1】__。

（2）表达式 LEN（SPACE（5）– SPACE（3））的结果是__【2】__。

（3）备注文件的扩展名是__【3】__。

（4）可以伴随着表的打开而自动打开的索引文件是__【4】__，其扩展名是__【5】__。

（5）下列程序代码的功能是求 1 到 100 之间所有整数的平方和并输出结果，请在__【6】__【7】__处填空。

```
SET TALK OFF
CLEAR
s = 0
x = 1
DO WHILE x <= 100
    __【6】__
    __【7】__
ENDDO
? s
```

（6）假设学生选课管理数据库中有 3 个表：学生.DBF、课程.DBF 和选课.DBF，其结构分别如下：

学生（学号 C（6），**姓名** C（8），**系名** C（30），**性别** L）
课程（课程号 C（6），**课程名** C（20））
选课（学号 C（6），**课程号** C（6），**成绩** N（3））
下面是查询每个学生及其选课情况的 SQL **语句，请在【8】【9】处填空。**

```
SELECT 学生.学号, 学生.姓名, 学生.系名, 选课.课程号, 选课.成绩;
    FROM 【8】, 选课;
    WHERE 【9】= 选课.学号
```

（7）下面是查询选修课程号为“ 102024 **”的学生姓名的** SQL **语句，请在【10】【11】处填空。**

```
SELECT 学生.姓名 FROM 学生;
    WHERE 学号 in （select 【10】 from 选课 where 【11】）
```

（8）使用 SQL **语句将一条新的记录插入学生表**

```
INSERT 【12】 学生（学号, 姓名, 性别） 【13】 （"020117", "王刚", .t.）
```

（9）使用 SQL **语句完成如下操作（将所有课程号为“** 102024 **”的成绩提高** 2%）

```
【14】 选课 SET 成绩 = 成绩 * 1.02 【15】 课程号 = "102024"
```

参考答案

一、选择题

（1）C （2）A （3）D （4）C （5）D （6）A （7）C （8）D （9）B
（10）C （11）C （12）B （13）B （14）D （15）C （16）A （17）A （18）A
（19）D （20）C （21）A （22）A （23）B （24）C （25）B （26）B （27）C
（28）B （29）A （30）A （31）C （32）C （33）D （34）C （35）A

二、填空题

（1）【1】**连接**
（2）【2】8
（3）【3】.DBT
（4）【4】**结构化复合索引文件，**【5】.CDX
（5）【6】s = s + x^2，【7】x = x + 1
（6）【8】**学生，**【9】**学生.学号**
（7）【10】**学号，**【11】**课程号** = "102024"
（8）【12】INTO，【13】VALUES
（9）【14】UPDATE，【15】WHERE

附录 C　全国计算机等级考试大纲（二级）

Visual FoxPro 程序设计

基本要求

1．具有数据库系统的基础知识。
2．基本了解面向对象的概念。
3．掌握关系数据库的基本原理。
4．掌握数据库程序设计方法。
5．能够使用 Visual FoxPro 建立一个小型数据库应用系统。

考试内容

一、Visual FoxPro 基础知识

1．基本概念

数据库、数据模型、数据库管理系统、类和对象、事件、方法。

2．关系数据库

（1）关系数据库：关系模型、关系模式、关系、元组、属性、域关键字和外部关键字。

（2）关系运算：选择、投影、联接。

（3）数据的一致性和完整性：实体完整性、域完整性、参照完整性。

3．Visual FoxPro 系统特点与工作方式

（l）Windows 版本数据库的特点。

（2）数据类型和主要文件类型。

（3）各种设计器和向导。

（4）工作方式:交互方式（命令方式、可视化操作）和程序运行方式。

4．Visual FoxPro 的基本数据元素

（1）常量、变量、表达式。

（2）常用函数：字符处理函数、数值计算函数、日期时间函数、数据类型转换函数、测试函数。

二、Visual FoxPro 数据库的基本操作

1．数据库和表的建立、修改与有效性检验

（1）表结构的建立与修改。

（2）表记录的预览、增加、删除与修改。

（3）创建数据库，向数据库添加或从数据库删除表。

（4）设定字段级规则和记录级规则。

（5）表的索引：主索引、候选索引、普通索引、惟一索引。

2．多表操作

（1）选择工作区。

（2）建立表之间的关联:一对一的关联；一对多的关联。

（3）设置参照完整性。

（4）表的联接 JOIN

内部联接。

外部联接：左联接、右联接、完全联接。

（5）建立表间临时关联。

3．建立视图与数据查询

（1）查询文件的建立、执行与修改。

（2）视图文件的建立、查看与修改。

（3）建立多表查询。

三、关系数据库标准语言 SQL

1．SQL 的数据定义功能

（l）CREATE TABLE-SQL。

（2）ALTER TABLE-SQL。

2．SQL 的数据修改功能

（1）DELETE-SQL。

（2）INSERT-SQL。

（3）UPDATE-SQL。

3．SQL 的数据查询功能

（1）简单查询。

（2）嵌套查询。

（3）联接查询。

（4）分组与计算查询。

（5）集合的并运算。

四、项目管理器、设计器和向导的使用

1．使用项目管理器

（1）使用“ 数据 ”选项卡。

（2）使用“ 文档 ”选项卡。

2．使用表单设计器

（1）在表单中加入和修改控件对象。

（2）设定数据环境。

3．使用菜单设计器

（1）建立主选项。

（2）设计子菜单。

（3）设定菜单选项程序代码。

4．使用报表设计器

（1）生成快速报表。

（2）修改报表布局。
（3）设计分组报表。
（4）设计多栏报表。
5．使用应用程序向导。
五、Visual FoxPro 程序设计
1．命令文件的建立与运行
（1）程序文件的建立。
（2）简单的交互式输入输出命令。
（3）应用程序的调试与执行。
2．结构化程序设计
（1）顺序结构程序设计。
（2）选择结构程序设计。
（3）循环结构程序设计。
3．过程与过程调用
（1）子程序设计与调用。
（2）过程与过程文件。
（3）局部变风和全局变量、过程调用中的参数传递。

考试方式

1．笔试：90min。
2．上机操作：90min。
上机操作包括：
（1）基本操作。
（2）简单应用。
（3）综合应用。