

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

**А.И. Муравьев**

# **БАЗЫ ДАННЫХ**

**Руководство  
к организации самостоятельной работы  
для студентов специальности 210106**

**2006**

Федеральное агентство по образованию

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

**Кафедра промышленной электроники**

**Утверждаю**

**Зав. кафедрой ПрЭ**

\_\_\_\_\_ А.В. Кобзев

# **БАЗЫ ДАННЫХ**

**Руководство**

**к организации самостоятельной работы  
для студентов специальности 210106**

Разработчик:

Ст. пр. каф. ПрЭ

\_\_\_\_\_ А.И. Муравьев

«\_\_\_» \_\_\_\_\_ 2006 г.

**2006**

## СОДЕРЖАНИЕ

1 Введение .....	4
2 Содержание лекционного курса .....	4
3 Список рекомендуемой литературы .....	5
4 Рейтинговая раскладка .....	6
5 Система Visual Foxpro .....	7
5.1 Системное меню и системные кнопки .....	11
5.2 Краткий обзор меню Visual FoxPro .....	13
5.2.1 Выполнение команд меню .....	14
5.2.2 Отказ от выполнения команды меню .....	14
5.2.3 Команды меню File .....	15
5.2.4 Команды меню Edit .....	15
5.2.5 Команды меню View .....	16
5.2.6 Команды меню Format.....	16
5.2.7 Команды меню Tools .....	17
5.2.8 Команды меню Program .....	18
5.2.9 Команды меню Windows.....	18
5.2.10 Выход из Visual FoxPro .....	18
5.3 Проект как средство объединения элементов приложения ...	19
5.3.1 Создание проекта приложения.....	20
5.3.2 Управление проектом с помощью меню .....	24
5.4 Создание базы данных проекта .....	26
5.4.1 Создание базы данных в окне проекта .....	26
5.5 Таблицы — основа базы данных .....	30
5.5.1 Имена таблиц.....	31
5.5.2 Окно конструктора таблицы.....	32
5.5.3 Определение полей таблицы .....	33
5.5.4 Ввод наименований полей .....	33
5.5.5 Типы полей .....	34
5.5.6 Задание свойств полей .....	35
5.5.7 Свойство NULL.....	36
5.5.8 Назначение кнопок окна конструктора таблицы .....	36
5.5.9 Определение свойств таблицы .....	37
5.5.10 Создание таблицы в конструкторе таблицы.....	38
5.5.11 Режимы просмотра таблицы.....	40

5.5.12 Модификация таблицы .....	42
5.5.13 Изменение наименований полей и их типов .....	43
5.5.14 Добавление и удаление полей .....	43
5.5.15 Изменение порядка расположения полей .....	43
5.5.16 Индексы .....	44
5.5.17 Создание первичного ключа для таблицы .....	47
5.5.18 Создание составного индекса.....	49
5.6 Форма как средство ввода и редактирования данных.....	49
5.6.1 Запуск формы .....	51
5.6.2 Объекты формы .....	51
5.6.3 Конструктор форм .....	55
5.6.4 Настройка параметров формы.....	60
5.6.5 Расположение формы .....	61
5.6.6 Заголовок формы .....	61
5.6.7 Среда окружения формы.....	62
5.6.8 Размещение полей таблиц и надписей к ним .....	65
5.6.9 Текстовая информация.....	66
5.6.10 Поле ввода .....	67
5.6.11 Поле редактирования .....	70
5.6.12 Кнопки управления.....	71
5.6.13 Размещение одиночных кнопок .....	71
5.6.14 Редактирование составных объектов .....	73
5.6.15 Создание группы кнопок .....	74
5.6.16 Объект Grid.....	77

## **1 ВВЕДЕНИЕ**

Дисциплина «Базы данных» включает изучение основных понятий реляционной модели данных, реляционной алгебры, как средства манипулирования данными, проектирования баз данных, стандарта языка запросов SQL, моделей серверов баз данных и языка программирования сервера ORACLE баз данных PL/SQL. Основной целью является приобретение навыков создания запросов и манипулирования данными, создание структуры базы данных, создание хранимых процедур и триггеров баз данных. Данное пособие включает материалы, необходимые для организации самостоятельной работы, рейтинговую раскладку изучения курса в восьмом семестре.

Изучение дисциплины требует знания информатики и теории множеств, как основы реляционной алгебры.

## **2 СОДЕРЖАНИЕ ЛЕКЦИОННОГО КУРСА**

2.1 Введение: Краткое содержание курса. Актуальность изучения курса «Базы данных». Численные и информационные прикладные системы. Администратор данных и администратор базы данных, его функции. Словарь Базы данных. Основная терминология.

2.2 Древовидно-иерархическая модель. Сетевая модель. Реляционная модель. Определение отношения, кортежа, домена, ключа, атрибута. Ограничение целостности баз данных (ограничения таблиц и ссылочная целостность).

2.3 Базисные средства манипулирования реляционными данными. Реляционная алгебра. Общая интерпретация реляционных операций.

2.4 Методика проектирования Баз данных. Этапы проектирования баз данных. Концептуальное и логическое проектирование.

2.5 Теория нормальных форм. Функциональные зависимости атрибутов. Неполные и транзитивные зависимости. Нормализация и свойства отношений. 1-я, 2-я, 3-я нормальные формы и форма Бойса-Кодда. Многозначные зависимости. 4-я нормальная форма.

2.6 Модель «Сущность-Связь» — ER-модель.

2.7 Язык SQL. Создание таблиц, декларативные ограничения данных целостности.

2.8 Запросы к базе данных, вложенные подзапросы. Использование агрегированных функций.

2.9 Операторы модификации базы данных. Понятие транзакции. Свойства транзакции. Фиксация и откат.

2.10 Модели серверов базы данных. Защита, восстановление и сохранность баз данных.

2.11 Сервер баз данных Oracle. Структура сервера, типы используемых файлов (файлы баз данных, управляющие файлы, журналы обновлений, словарь данных).

2.12 Структура памяти сервера, системные процессы (обязательные и необязательные), пользовательские процессы. PL/SQL — процедурный язык. Типы и управляющие структуры.

2.13 Курсоры, объявление и управление курсорами, параметры курсоров. Циклы по курсору.

2.14 Процедуры и функции PL/SQL. Создание хранимых процедур и функции, отличия от обычных процедур и функций.

2.15 Триггеры баз данных, виды триггеров, использование триггеров для поддержки целостности баз и ведения аудита.

2.16 Защита данных. Целостность данных. Привилегии системного и объектного уровня. Пользователи и роли. Резервное копирование и восстановление баз данных. Виды резервного копирования.

### **3 СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ**

3.1 Петров В.Н. Информационные системы — СПб: Питер, 2002. — 688 с: ил.

3.2 Муравьев А.И. Системы управления базами данных: Учебное пособие. — Томск: ТУСУР, 1999.— 93 с.

3.3 Дейт Л. Дж. Введение в системы баз данных: Пер. с англ. — 6-е изд. — К.: Диалектика, 1998. — 784 с.: ил.

3.4 Базы данных: модели, разработка, реализация / Т.С. Карпова — СПб.: Питер, 2001. — 304 с.: ил.

3.5 Шкарина Л. Язык SQL: учебный курс. — СПб.: Питер, 2001. — 592 с.: ил.

3.6 Джеймс Р. Грофф, Пол Н. Вайнберг SQL: полное руководство: Пер. с англ. — К.: Издательская группа BHV, 1998. — 608 с.

3.7 Попов А.А. Программирование в среде СУБД FoxPro 2.0. — М.: Радио и связь, 1993. — 352 с.: ил.

3.8 Oracle8. Энциклопедия пользователя.: Пер. с англ. / Компания Advanced Information System и др. — К.: Изд-во «ДиаСофт», 1998. — 864 с.

3.9 Введение в Oracle PL/SQL / В.А. Кравчук. — К.: Изд-во «ДиаСофт», 1998. — 400 с.

3.10 Озкарахан Э. Машины баз данных и управление базами данных: Пер. с англ. — М.: Мир, 1989. — 695 с.

#### **4 РЕЙТИНГОВАЯ РАСКЛАДКА**

Контрольные работы (3 работы по 10 баллов)	30 баллов.
Лабораторные работы (8 работ по 10 баллов)	80 баллов
Творческое задание	10 баллов
Максимальный рейтинг	120 баллов

Перечень, описание и варианты лабораторных работ приведены в руководстве к выполнению лабораторных работ. Для выполнения первых четырех работ необходимо самостоятельное изучение правил создания таблиц, индексов, и форм, которые рассмотрены в главе 4 данного руководства. Изучение правил необходимо выполнить до начала выполнения лабораторных работ, с тем условием, чтобы студент к началу лабораторных работ был подготовлен. Для выполнения лабораторных работ №5—№8 достаточно изучения лекционного материала.

#### **Перечень контрольных работ:**

1. По заданной предметной области выделение атрибутов, определение возможных ключей, выделение первичного ключа, определение функциональных зависимостей и нормальных форм
2. Составление заданных связанных таблиц с декларативными ограничениями.

3. Создание анонимного блока, реализующего выборку и вывод на экран данных по заданным параметрам

Творческое задание могут получить те, кто набирает не менее 80 баллов до начала зачетной недели. Можно не сдавать экзамен, если студент набирает не менее 80 баллов (оценка «хорошо») и пропущено не более трех лекций. Если студент набирает 100 баллов и пропущено не более двух лекций, получает оценку «отлично»

Для допуска к экзамену необходимо выполнение не менее 6 лабораторных работ и выполнение всех контрольных работ.

Экзамен по дисциплине принимается в письменном виде, билет включает два вопроса по теоретической части и две задачи.

## **5 СИСТЕМА VISUAL FOXPRO**

В Visual FoxPro реализованы все атрибуты реляционных систем управления базами данных. В базе данных поддерживается целостность данных с помощью первичных ключей и связей между таблицами. Для обработки событий добавления, удаления или изменения записей таблиц средствами самой базы можно использовать триггеры и хранимые процедуры.

Visual FoxPro — это объектно-ориентированный, визуально-программируемый язык, управляемый по событиям, который в полной мере соответствует новым требованиям, предъявляемым к современным средствам проектирования. Состоит из отдельных компонентов, которые используются для хранения информации, ее отображения и редактирования.

Все данные хранятся в базе данных, которая состоит из таблиц, отношений между таблицами, индексов, триггеров и хранимых процедур. Каждая таблица имеет уникальное имя и хранится в отдельном файле, наименование которого совпадает с именем таблицы. Созданный файл имеет расширение DBF.

Каждая создаваемая таблица может иметь несколько связанных с ней индексов, используемых для упорядочения данных, быстрого поиска необходимых записей и связи между таблицами.

Для хранения значений полей типа Memo и General применяются отдельные файлы. Memo-поля таблиц содержат текстовую информацию, а поля типа General используются, как прави-



ло, для хранения двоичной информации, данных других приложений, работающих в среде Windows.

В Visual FoxPro реализованы триггеры, которые позволяют централизованно обрабатывать события, возникающие при любых изменениях в базе данных. Также можно создавать хранимые процедуры которые являются частью базы данных и могут использоваться при описании таблиц для проверки введенных данных, определения значения по умолчанию и т.п.

Чрезвычайно удобным и полезным средством доступа к базе данных являются представления. Представления данных позволяют объединять данные таблиц и отображать их в более удобном виде.

Для отображения и редактирования данных используются формы, отчеты, запросы и программы. При создании форм, отчетов и запросов применяются конструкторы или построители. Поэтому эти компоненты часто называют конструкторскими объектами. Формы и отчеты являются составными объектами, т.к. они состоят из более мелких объектов (таких как поля, кнопки, диаграммы, рамки, OLE — компоненты и т.п.), которые называются объектами интерфейса.

Формы используются для просмотра или ввода данных в таблицы. Данные можно вводить непосредственно в таблицы, но использование формы является более быстрым, наглядным и эффективным способом ввода. Форма содержит некоторые или все поля таблиц, в которые вы вводите информацию. Для создания форм вы можете использовать мастер создания форм или конструктор форм. Мастер форм содержит целый ряд шаблонов, которые определяют соотношение между помещаемыми в форму таблицами, вид отображения данных и порядок размещения полей. Для создания сложных форм применяется конструктор форм.

Отчеты используются для просмотра и вывода на печать содержащейся в базе данных информации. Примерами отчетов являются прайс-лист товаров, список покупателей, оборотная складская ведомость. Как правило, отчеты создаются в том случае, если информацию необходимо передавать кому-либо в печатном виде. Для создания отчетов в Visual FoxPro, как и для форм, используются мастер и конструктор отчетов. С помощью мастера отчетов вы можете быстро создать собственный отчет на

основе имеющихся шаблонов. Применение конструктора отчетов позволяет создавать отчеты произвольной сложности, включая многоуровневую группировку данных и размещение вычисляемых полей.

Каждый компонент хранится в отдельном файле, причем имена файлов, содержащих основные компоненты, вы задаете самостоятельно, а наименования файлов, содержащих объекты, связанные с таблицей, совпадают с именем таблицы. В зависимости от типа содержащегося в нем объекта Visual FoxPro автоматически присваивает каждому файлу расширение, которое помогает в идентификации объекта. Список расширений файлов приведен в табл. 1.

Таблица 1 — Расширения имен файлов объектов Visual FoxPro

Наименование объекта	Расширения
Проект	PRJ, FPC, CAT, PJX, PJT
База данных	DBC
Таблица Visual FoxPro	DBF
Составной индексный файл	CDX
Мето-поле и поле типа General	FRT
Форма	SCX
Запрос	QPR
Отчет	FRX, FRM
Почтовая наклейка	LBX, LBL
Меню	MNX
Библиотека класса	VCX
Программа, библиотека программ, текст меню, текст запроса	PRG, FXP, MPR, QPR
Рисунок	BMP, JPG, GIF, ICO, DIB, CUR, ANI
Звукозапись	WAV

Для инсталляции Visual FoxPro в папке FoxPro установочного диска запустите программу VFP98.EXE. Появится диалоговое окно, в котором будет по умолчанию предложен путь установки

c:\documents and settings\name\_user\local settings\temp

где `name_user` — имя пользователя при входе в Windows. Можно изменить путь установки (рекомендуется), например на `c:\foxpro`. При нажатии на кнопку Unzip система будет установлена в эту папку (если папка не существовала, она будет создана). Запускается Visual FoxPro выполнением программы `vfp6.exe`. Желательно на Рабочий Стол установить ярлык Visual FoxPro (при этом создать рабочую папку, например `C:\VFP` и в свойствах ярлыка прописать путь этой рабочей папки, тогда все файлы при работе с Visual FoxPro будут находиться в рабочей папке).

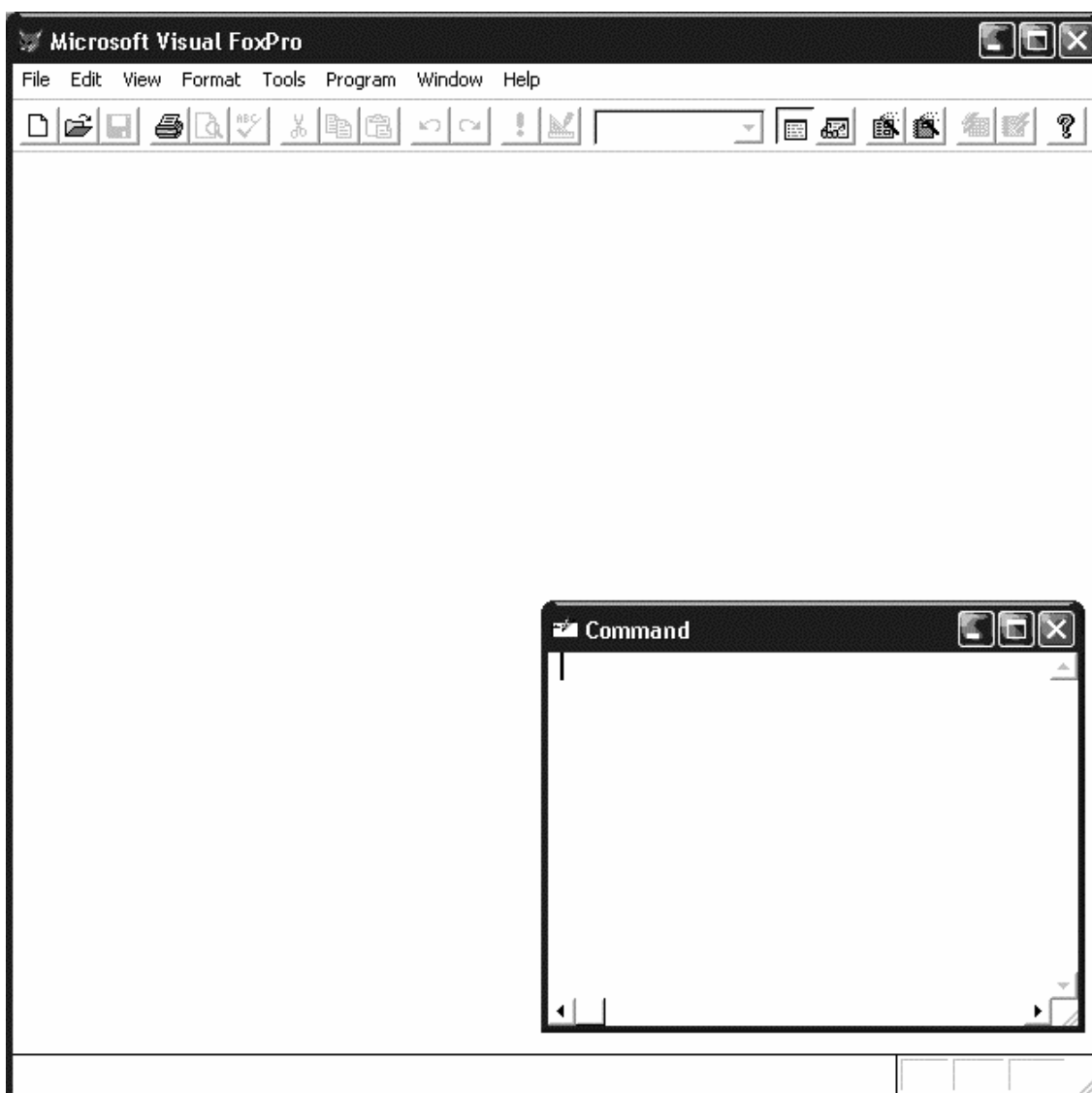


Рис. 1

На рис. 1 показано главное окно Visual FoxPro, каким оно выглядит после запуска программы. В нем можно выделить

шесть основных областей: заголовок окна, строка меню, панель инструментов, строка состояния, рабочая область и командное окно Command (Команда), позволяющее вводить одиночные команды.

Заголовок находится в верхней части главного окна и содержит три объекта: копию значка программы Visual FoxPro, заголовок главного окна и значки управления этим окном.

Строка меню содержит иерархическое меню Visual FoxPro. Список пунктов меню зависит от типа активного окна.

Visual FoxPro предоставляет в распоряжение пользователя много самых разнообразных панелей инструментов для работы с базами данных, формами, отчетами и запросами. Эти панели инструментов содержат набор кнопок, причем этот набор зависит от назначения конкретной панели инструментов. При первом запуске Visual FoxPro после его установки на экране отображается стандартная панель инструментов.

В нижней части главного окна расположена строка состояния, в которую выводится информация о текущем состоянии активного окна, краткое описание выбранной кнопки панели инструментов или пункта меню. Вид информации, отображаемой в строке состояния, зависит от типа активного окна. Например, в окне конструктора форм в строке состояния выводится наименование активного объекта формы, а при просмотре таблицы в режиме Browse в левой части строки состояния отображается наименование открытой таблицы, в середине строки состояния — номер текущей записи и индикатор режима работы.

В процессе работы в главном окне Visual FoxPro открываются окна конструкторов форм, отчетов, запросов, проекта и другие окна. Все эти окна являются дочерними по отношению к главному окну и закрываются при его закрытии.

## **5.1 Системное меню и системные кнопки**

Заголовок главного окна содержит уменьшенную копию значка программы, текст «Microsoft Visual FoxPro» и кнопки управления размерами главного окна. При щелчке на значке пиктограммы открывается системное меню, содержащее команды перемещения и изменения размеров главного окна.

Назначение кнопок стандартной панели инструментов описано в табл. 2.

Таблица 2 — Назначение кнопок стандартной панели инструментов

<b>Название</b>	<b>Назначение</b>
New (Новый)	Создает новый файл
Open (Открыть)	Открывает существующий файл
Save (Сохранить)	Сохраняет открытый файл
Print One Copy (Печатать одну копию)	Печатает содержимое активного окна
Print Preview (Просмотр печати)	Открывает окно предварительного просмотра печатаемого документа
Cut (Вырезать)	Удаляет выделенный текст или выделенные объекты и помещает их в буфер
Copy (Копировать)	Копирует в буфер выделенный текст или выделенные объекты, не удаляя их
Paste (Вставить)	Вставляет содержимое буфера. Действие этой кнопки зависит от того, в каком из режимов работаете
Undo (Отменить)	Отменяет последнее выполненное действие
Redo (Восстановить)	Восстанавливает отмененные действия
Run (Выполнить)	Запускает на выполнение
Database (База данных)	Содержит список открытых баз данных
Command Window (Командное окно)	Открывает командное окно
Data Session Window (Окно сеанса данных)	Открывает окно Data Session (Сеанс данных)
Properties Window (Окно свойств)	Открывает окно Properties (Свойства), используемое для настройки свойств
Document View Windows (Окно просмотра документа)	Открывает окно Document View (Просмотр документа), в котором отображается список процедур, функций открытой в окне редактора программы
Class Browser (Обзор классов)	Открывает окно Class Browser (Обзор классов), предназначенное для работы с классами
Object Browser (Обзор объектов)	Открывает окно Object Browser (Обзор объектов), позволяющее просматривать классы, свойства, методы, события и константы выбранных библиотек

### **Замечание**

Если по внешнему виду кнопки вы не можете определить ее назначение, установите курсор мыши на кнопку. При этом под курсором появится всплывающая подсказка с ее наименованием, а в левой части строки состояния будет выведено краткое описание указанной кнопки.

По умолчанию в главном окне программы Visual FoxPro всегда присутствует стандартная панель инструментов, если только вы не удалили ее с экрана. Если для работы вам необходима панель инструментов, а она не отображается на экране, выполните следующие действия:

- 1) выберите в меню View (Вид) команду Toolbars (Панели инструментов). Откроется диалоговое окно Toolbars (Панели инструментов), в котором расположен список всех панелей инструментов программы Visual FoxPro;
- 2) установите флажки слева от наименований тех панелей, которые вы хотите разместить в главном окне;
- 3) нажмите кнопку ОК.

После установки Visual FoxPro стандартная панель инструментов размещается в верхней части главного окна, но она, как и все остальные панели инструментов, может перемещаться в любое место экрана. Для этого установите курсор мыши в любое свободное от кнопок место на панели инструментов, нажмите кнопку мыши и, не отпуская ее, переместите панель на новое место. Если вы поместите панель инструментов не по краям главного окна Visual FoxPro, то сможете изменять ее размер.

## **5.2 Краткий обзор меню Visual FoxPro**

При работе с Visual FoxPro вы можете использовать как кнопки панели инструментов, так и строку меню, все команды которого являются иерархическими. При выборе определенной команды открывается ее подменю.

При отсутствии в главном окне Visual FoxPro других открытых окон, строка меню содержит пункты File (Файл), Edit (Правка), View (Вид), Tools(Сервис), Program (Программа), Windows

(Окна) и Help (Справка). При открытии окон в меню добавляются команды, применимые только к объектам данного окна. Например, при работе с таблицами в главное меню добавляется пункт меню Table (Таблица), при открытии на экране формы в меню появляется пункт Form (Форма).

### 5.2.1 Выполнение команд меню

Выполнение команд меню в Visual FoxPro осуществляется стандартными для Windows действиями. Вы можете использовать как мышь, так и клавиатуру. Перечислим возможные способы.

При использовании мыши установите курсор на пункт меню, содержащий выбираемую команду, и нажмите кнопку мыши. Если команда содержит подменю, выберите в нем требуемую команду.

При использовании клавиатуры сначала активизируйте строку меню нажатием клавиши <F10> или <Alt>. Затем установите указатель на требуемую команду с помощью стрелок и нажмите <Enter>.

Клавиша используется для перемещения на команду, находящуюся левее текущей. При нажатии на эту клавишу, когда текущая команда является самой левой, вы переместитесь в системное меню. Дальнейшее нажатие на клавишу установит указатель на самую правую команду строки меню. Остальные клавиши используются аналогично.

Для наиболее часто выполняемых команд вы можете запомнить связанные с ними «горячие» клавиши, которые состоят из комбинации символа <Alt> и подчеркнутого символа в имени команды. При использовании «горячих» клавиш не требуется нажимать клавишу <Enter>.

### 5.2.2 Отказ от выполнения команды меню

Если строка меню активизирована, но вы решили не выполнять команды меню, воспользуйтесь одним из следующих способов:

- Нажмите клавишу <F10>.

- Щелкните кнопкой мыши в любом свободном месте основного окна Visual FoxPro.
- Если вы находитесь в подменю, для возврата в меню предыдущего уровня нажмите клавишу <Esc>.

### 5.2.3 Команды меню File

Меню File (Файл) содержит команды, связанные с доступом к файлам. Эти команды позволяют создавать новые файлы разных типов, открывать существующие файлы, закрывать, сохранять и печатать их.

Команда Save As HTML (Сохранить файл в формате HTML) открывает соответствующее диалоговое окно, позволяющее сохранить текущую форму, меню, отчет или таблицу в формате HTML (Hyper Text Markup Language).

Очень удобна команда Revert (Возвратить), позволяющая отменить все изменения файла, сделанные в текущем сеансе редактирования.

В нижней части меню располагаются имена последних четырех открывавшихся проектов, которые предоставляют возможность быстрого открытия любого из них. Последней командой этого меню является команда Exit (Выход). Она аналогична действию команды QUIT (Выход) в командном окне и предназначена для выхода из Visual FoxPro.

### 5.2.4 Команды меню Edit

Меню Edit (Правка) имеется во многих приложениях Windows. В Visual FoxPro команды этого меню используются при создании форм и отчетов, редактировании программ.

Команды Paste Special (Специальная вставка), Insert Object (Вставить объект), Object (Объект) и Links (Связи) применимы только к полям типа General и OLE-объектам. Остальные команды применяются при редактировании как программ, так и полей таблиц.

Незнакомой для вас по другим приложениям Windows может оказаться команда Go to Line (Переход на строку), предна-



значенная в Visual FoxPro для работы с программами. С помощью данной команды можно переходить к строке с указанным номером в тексте программы.

Последняя в меню — команда Properties (Свойства), при выборе которой открывается диалоговое окно Edit Properties (Редактирование свойств), позволяющее задавать параметры сохранения программных файлов, просматривать их свойства, дату последнего редактирования и занимаемый объем на диске.

### **5.2.5 Команды меню View**

Пункт меню View (Вид) содержит команды, управляющие просмотром текущего открытого файла. Если вы не просматриваете файл, то меню содержит всего одну команду Toolbars (Панели инструментов). Эта команда открывает диалоговое окно, в котором перечислены панели инструментов, используемые при работе в Visual FoxPro.

При работе в конструкторах в меню View (Вид) содержатся команды, управляющие отображением используемых в данном режиме работы панелей инструментов. Так, например, при работе в конструкторе форм в меню появляются команды для выбора следующих панелей: Color Palette (Цветовая палитра), Layout (Расположение), Form Designer (Конструктор форм) и Form Controls (Элементы управления формы).

При просмотре таблиц или редактировании форм, меню или отчетов в меню появляются дополнительные команды. Например, если вы работаете с таблицами, в меню появляется несколько новых команд. Команды Edit (Редактирование) и Browse (Обзор) переводят таблицу в режим просмотра и редактирования содержащихся в ней записей. Отличие между этими двумя командами состоит в стиле отображения содержимого таблицы.

### **5.2.6 Команды меню Format**

Меню Format (Формат) содержит команды, управляющие характеристиками шрифта, выравниванием текста и объектов, определением интервалов между объектами. Однако при работе с

различными конструкторами становятся доступными и дополнительные команды.

### 5.2.7 Команды меню Tools

Это меню содержит мастера и средства отладки программ.

Первой в меню Tools (Сервис) располагается команда Wizards (Мастера), открывающая список мастеров Visual FoxPro. Мастер — это программа, которая помогает пользователю в решении таких задач, как создание форм, отчетов, таблиц. Процесс создания при этом заключается в ответах на вопросы мастера относительно создаваемого объекта.

Далее в меню расположена команда Macros (Макрос), предназначенная для создания макросов.

Команда Debugger (Отладчик) предназначена для отладки. Она открывает диалоговое окно Visual FoxPro Debugger, в котором вы можете просмотреть значения переменных, элементов массива, полей, возвращаемые значения функциями Visual FoxPro и т.д.

Команда Component Gallery (Галерея компонентов) открывает окно Catalogs — Component Gallery (Каталоги — галерея компонентов) для работы в Галерее компонентов Visual FoxPro. Команда Coverage Profiler (Анализ выполнения) позволяет просмотреть информацию о выполненной программе. Команда Run Active Document (Запуск активного документа) открывает одноименное диалоговое окно для выбора активного документа и его запуска.

Команда Object Browser (Браузер объектов) открывает одноименное окно, позволяющее просматривать классы, свойства, методы, события и константы выбранных библиотек.

Команда IntelliSense Manager (Менеджер IntelliSense) открывает окно настройки параметров, управляющих анализом синтаксиса вводимых в окне редактора программных конструкций.

Команда Task List (Список заданий) позволяет запомнить место, в которое вы можете вернуться впоследствии нажатием оперативной клавиши. Кроме того, данная команда позволяет вводить список необходимых задач, аналогично Microsoft Outlook.

Команда Document View (Просмотр документа) открывает диалоговое окно, в котором отображается список процедур, функций и операторов для открытой в окне редактора программы. При выборе элемента списка осуществляется переход на этот элемент в окне редактора.

Последняя команда в меню Tools (Сервис) — Options (Параметры). С ее помощью можно настраивать конфигурацию Visual FoxPro.

### **5.2.8 Команды меню Program**

Меню Program (Программа) содержит команды, связанные с выполнением и компиляцией программ.

### **5.2.9 Команды меню Windows**

В меню Windows (Окна) имеются команды, которые управляют открытыми на экране окнами. С помощью этих команд можно упорядочивать, скрывать, очищать окна и переходить из одного окна в другое. Кроме того, команды данного меню позволяют активизировать любое открытое окно.

Команда Command Window (Командное окно) из меню Windows (Окно) активизирует командное окно Command (Команда), позволяющее использовать команды при работе с Visual FoxPro (можно использовать любые команды, за исключением структурных, т.е. состоящим из нескольких строк).

Команда Properties Window (Окно свойств) открывает диалоговое окно Properties (Свойства), позволяющее просмотреть и изменить свойства объектов, расположенных в форме курсоров и отношений.

### **5.2.10 Выход из Visual FoxPro**

Чтобы завершить работу с программой Visual FoxPro, вам необходимо ее закрыть. Для этого воспользуйтесь одним из следующих способов:

- выберите в меню File (Файл) команду Exit (Выход);

- выберите команду **Закр $\ddot{y}$ ть** в системном меню главного окна Visual FoxPro;
- щелкните на значке Visual FoxPro в заголовке главного окна;
- введите в командном окне команду **QUIT**.

Если вы не сохранили форму, отчет, запрос или какой-либо другой объект приложения, Visual FoxPro предложит вам сохранить его. Для этого на экране появляется диалоговое окно, аналогичное представленному на рис. 2.



Рис. 2 — Диалоговое окно, появляющееся при попытке выхода из Visual FoxPro без сохранения информации

Для продолжения операции нажмите одну из кнопок (табл. 3).

Таблица 3 — Назначение кнопок диалогового окна, появляющегося при попытке выхода из Visual FoxPro без сохранения информации

Кнопка	Назначение
Yes (Да)	Сохранение файла и выход из Visual FoxPro. Если файлу еще не присвоено имя, Visual FoxPro открывает диалоговое окно Save As(Сохранить как)
No (Нет)	Выход из Visual FoxPro без сохранения файла
Cancel (Отмена)	Закрытие текущего диалогового окна и возвращение в VisualFoxPro

### 5.3 Проект как средство объединения элементов приложения

При создании приложения используется проект, который объединяет элементы приложения Visual FoxPro и группирует их по типам. Информация о проекте хранится в специальной табли-

це, которая, в отличие от обычных таблиц Visual FoxPro, имеет расширение PJX. Метод-поля таблицы содержат наименование элемента проекта, его описание и другие текстовые атрибуты. Файл с Метод-полями таблицы имеет расширение PJT. Использование проекта упрощает разработку приложения, т.к. в проекте базы данных, программы, формы, отчеты, запросы и другие элементы приложения располагаются в соответствующих разделах, а также запоминается расположение каждого включенного в проект элемента.

Создав проект и определив входящие в него элементы, вы можете использовать его для сборки приложения, построив файл с расширением APP, или для создания исполняемого файла с расширением EXE. При построении приложения из проекта осуществляется поиск и сборка файлов, на которые ссылаются элементы приложения, отслеживаются версии файлов, входящих в проект.

### **5.3.1 Создание проекта приложения**

Приступая к разработке нового приложения, прежде всего создайте проект приложения. В дальнейшем вы будете добавлять в него созданные вами элементы приложения. Для создания нового проекта вы можете использовать мастер Application Wizard (Мастер приложения) или команду New (Новый) из меню File (Файл).

При выполнении команды New (Новый) на экране открывается соответствующее диалоговое окно с перечислением всех типов элементов приложения, которые возможны в Visual FoxPro (рис. 3). По умолчанию установлена опция Project (Проект).

Для создания нового проекта выполните следующие действия:

1. Нажмите кнопку New file (Новый файл).
2. В поле ввода Enter project (Введите имя проекта) диалогового окна Create (Создать) задайте имя создаваемого проекта, убедившись, что в поле Тип файла установлен тип сохраняемого файла Project (Проект), а в поле Папка правильно выбрана папка, в которой вы хотите расположить свой проект (рис. 4).

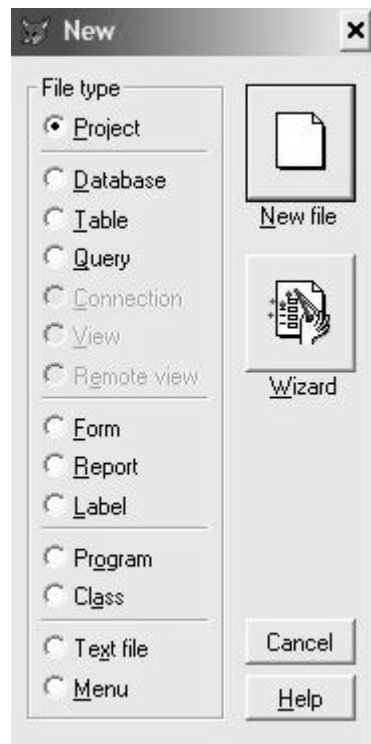


Рис. 3 — Диалоговое окно New, в котором можно указать тип создаваемого файла

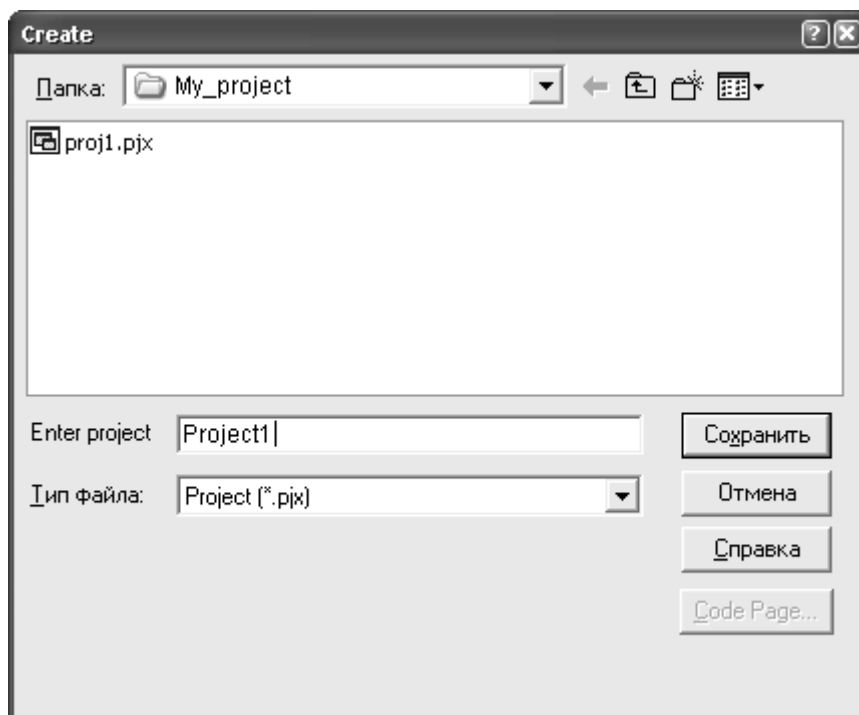


Рис. 4 — Диалоговое окно Create для задания имени создаваемого проекта

4. Для сохранения созданного проекта нажмите кнопку Сохранить. Visual FoxPro создаст файлы проекта и запишет их в указанное место. После этого откроется окно проекта Project Manager (Менеджер проекта) (рис. 5).

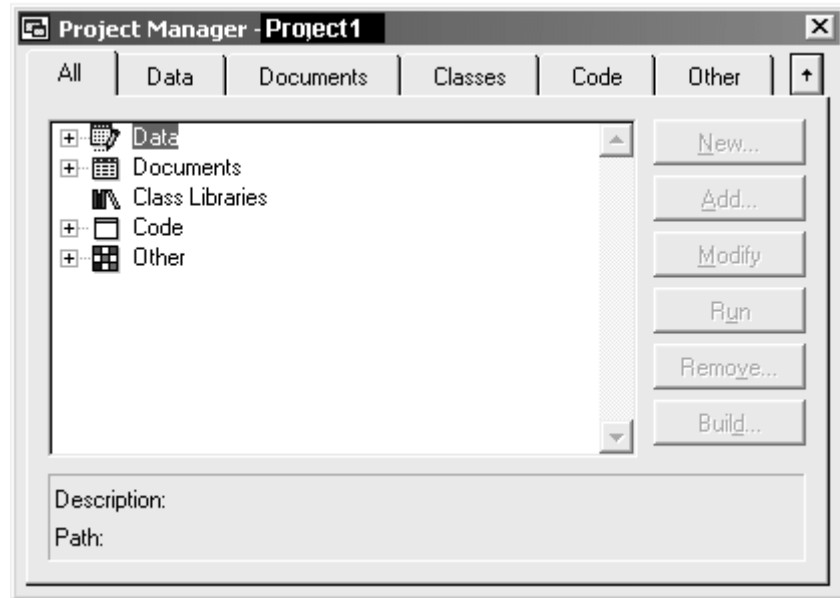


Рис. 5 — Окно проекта Project Manager

В верхней части окна проекта находятся вкладки, которые управляют отображением определенных типов файлов проекта, значительно облегчая работу по разработке приложения (табл. 4).

Таблица 4 — Вкладки окна Project Manager

Вкладка	Отображаемые файлы
All (Все)	Все файлы
Data (Данные)	Базы данных, таблицы, запросы, представления данных, хранимые процедуры
Documents (Документы)	Формы, отчеты и этикетки
Classes (Классы)	Классы
Code (Коды)	Программы и библиотеки
Other (Остальные)	Меню, файлы переменных памяти, клавишные макрокоманды

В Visual FoxPro используется иерархическое отображение элементов проекта. На самом верхнем уровне находятся катего-

рии элементов проекта: Data(Данные), Documents (Документы), Class Libraries (Библиотека классов),Code (Исходный код) и Other (Другие).

### **Примечание**

Для каждой категории элементов в окне проекта имеется отдельная вкладка.

На следующем уровне находятся типы файлов данной категории. Например, для категории Documents (Документы) имеются следующие типы файлов:

- Forms — формы;
- Reports — отчеты;
- Labels — этикетки.

Некоторые типы файлов могут иметь следующие уровни иерархии. Например, база данных может содержать таблицы, представления данных, а каждая таблица — поля.

Каждый иерархический список может находиться в свернутом или развернутом состоянии. Если элемент иерархического списка содержит элементы следующего уровня, то в свернутом состоянии с левой стороны наименования элемента отображается знак плюс. На рис. 3.3 видно, что все элементы списка, кроме Class Libraries (Библиотеки классов), содержат вложенные уровни элементов. Для раскрытия списка элементов следующего уровня необходимо щелкнуть мышью на знаке плюс. При раскрытии списка знак плюс изменяется на знак минус. На рис. 6 приведен пример иерархического списка Data (Данные), имеющего глубину в четыре уровня. Для свертывания вложенного списка элементов необходимо щелкнуть мышью на знаке минус.

В окне проекта имеется одна очень интересная и полезная кнопка. Расположена она в правом верхнем углу окна проекта и содержит направленную вверх стрелку. При нажатии на эту кнопку окно проекта сворачивается, остаются видимыми только наименования вкладок, а значок кнопки меняется на стрелку, направленную вниз. Тем самым на экране освобождается место для других окон (рис. 7).



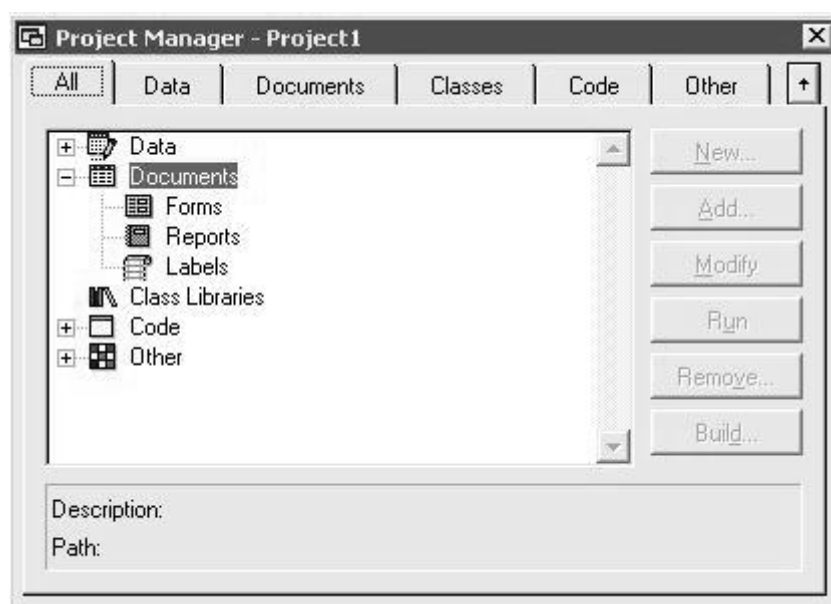


Рис. 6 — Иерархический список элементов проекта



Рис. 7 — Свернутое окно проекта

При этом вкладки проекта остаются активными, и при нажатии на них открывается часть окна проекта, содержащая информацию выбранной вкладки. Для возврата окна проекта в развернутый вид нажмите кнопку, содержащую стрелку, направленную вниз.

Visual FoxPro предоставляет возможность перемещать открытую вкладку по экрану. Для этого щелкните на ярлыке и, не отпуская кнопки мыши, переместите вкладку в любое место главного окна Visual FoxPro.

### 5.3.2 Управление проектом с помощью меню

При открытии окна проекта в основное меню Visual FoxPro добавляется новый пункт меню Project (Проект), который содержит команды, позволяющие работать с файлами, входящими в проект. Назначение этих команд описано в табл. 5.

Таблица 5 — Назначение команд меню Project

<b>Команда</b>	<b>Назначение</b>
New File (Новый файл)	Создает новый файл, который автоматически добавляется в проект
Add File (Добавить файл)	Добавляет созданные ранее файлы в проект
Modify File (Модифицировать файл)	Модифицирует выбранный файл проекта
Browse File (Просмотр файла в табличном виде)	Открывает таблицу в режиме Browse
Preview File (Просмотр файла)	Открывает файл в окне предварительного просмотра
Remove File (Удалить файл)	Удаляет файл из проекта
Rename File (Переименовать файл)	Переименовывает файл, входящий в проект
Run File (Выполнить файл)	Запускает файл на выполнение
Exclude (Исключить)	Исключает файл из проекта
Include (Включить)	Включает файл в проект
Set Main (Основная программа)	Устанавливает файл в качестве основной программы проекта
Edit Description (Правка описания)	Открывает окно редактирования описания файла
Project Info (Информация проекта)	Отображает информацию о проекте
Errors (Ошибки)	Отображает ошибки, возникшие при построении проекта
Build (Построить)	Перестраивает проект
Refresh (Обновить)	Обновляет информацию в окне проекта
Clean Up Project (Упаковать)	Упаковывает проект, очищая его от удаленных файлов

Для работы с файлами проекта можно использовать контекстное меню, раскрываемое при щелчке по нему правой кнопкой мыши (рис. 8) и содержащее некоторые команды из меню Project (Проект), а также команду Expand All (Раскрыть все), раскрывающую все уровни вложенности данных в окне.

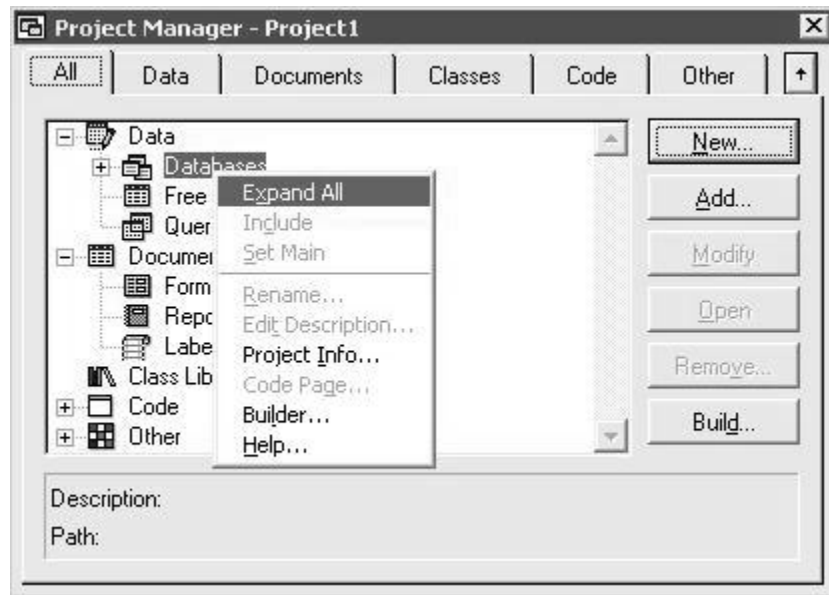


Рис. 8 — Контекстное меню для выбранного файла проекта

## 5.4 Создание базы данных проекта

База данных в Visual FoxPro — это совокупность таблиц, отношений между таблицами, индексов, триггеров и хранимых процедур.

Создание базы данных в Visual FoxPro осуществляется в интерактивном режиме с помощью конструктора базы данных, который позволяет:

- создавать и модифицировать таблицы, хранимые процедуры, представления данных;
- добавлять созданные ранее таблицы;
- определять для таблиц индексы;
- устанавливать отношения между таблицами, которые будут поддерживаться при создании форм и отчетов.

Описание структуры базы хранится в словаре базы данных, представляющем собой совокупность системных файлов.

### 5.4.1 Создание базы данных в окне проекта

База данных является частью проекта, поэтому ее целесообразно создавать в окне проекта. Для создания базы данных выполните следующие действия:

Откройте созданный проект.

Выберите в верхней части окна конструктора проектов вкладку Data(Данные). Курсор по умолчанию устанавливается в начале вкладки назначения Databases (Базы данных).

Нажмите кнопку New (Новый) в окне проекта.

В открывшемся диалоговом окне New Database (Новая база данных) нажмите кнопку New Database (Новая база данных).

В поле ввода Enter database (Введите имя базы данных) появившегося на экране диалогового окна Create (Создать) задайте имя создаваемой базы данных, убедившись, что в поле Тип файла установлен тип сохраняемого файла Database (База данных), а в раскрывающемся списке Папка правильно указана папка, в которой вы хотите расположить создаваемую базу данных.

Для сохранения созданной базы данных нажмите кнопку Сохранить. После этого откроется пустое окно базы данных Database Designer(Конструктор базы данных) (рис. 9).

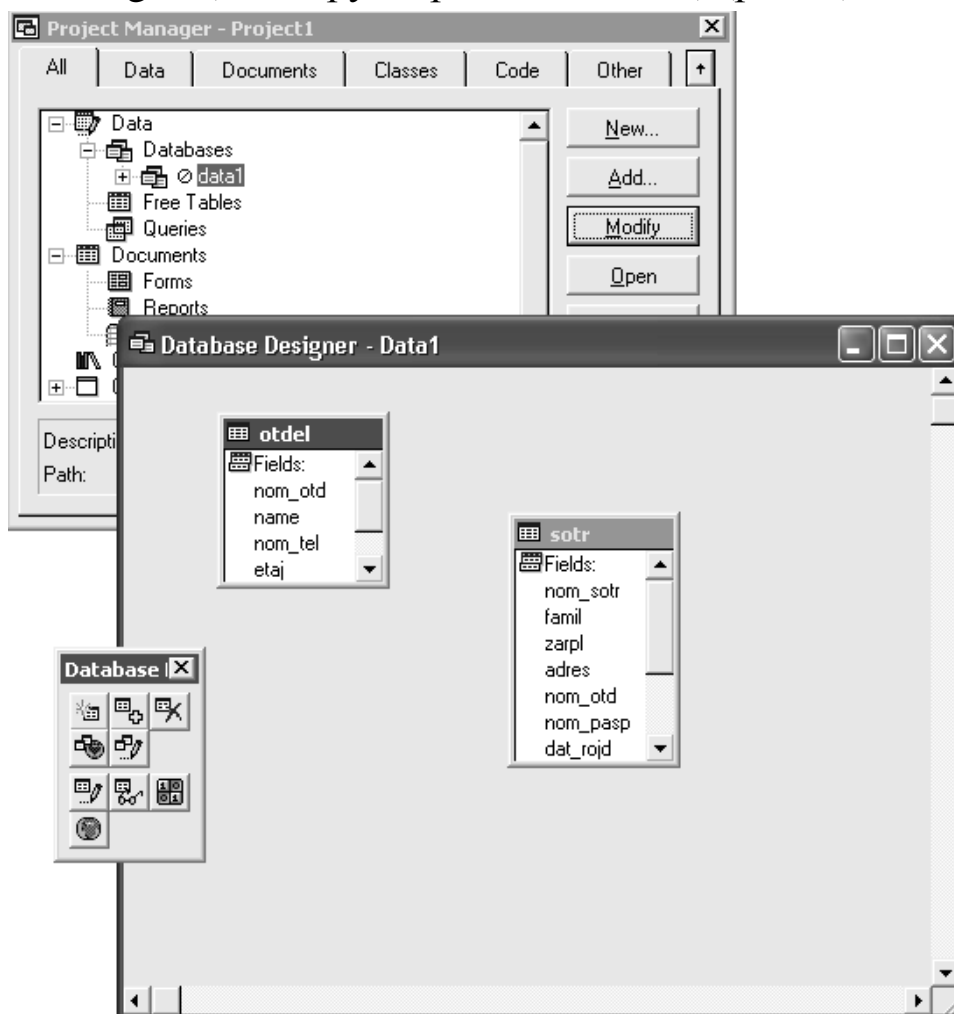


Рис. 9 — Окно конструктора базы данных Data1 с размещенными в нем таблицами

Используя панель инструментов Designer (Конструктор базы данных), команды меню Database (База данных) и контекстное меню, в окне конструктора базы данных вы можете создавать новые таблицы, модифицировать существующие, создавать для них индексы, устанавливая отношения между таблицами.

### Замечание

Если панель инструментов Database Designer (Конструктор базы данных) невидна на экране, в меню View (Вид) выберите команду Toolbars (Панели инструментов). Открывается диалоговое окно Toolbars (Панель инструментов), в котором установите флажок Database Designer (Конструктор базы данных).

В начале создания базы данных окно конструктора базы данных пусто.

Для создания в конструкторе базы данных новых таблиц и модификации существующих вы можете использовать:

- команды меню Database (База данных) (табл. 6);
- команды контекстного меню, появляющегося при нажатии правой кнопки мыши в окне конструктора базы данных;
- панель инструментов Database Designer (Конструктор базы данных).

Таблица 6 — Назначение команд меню Database

Команда	Назначение
New Table (Новая таблица)	Создает новую таблицу
Add Table (Добавить таблицу)	Добавляет созданную таблицу в базу данных
New Remote View (Новое удаленное представление)	Создает удаленное представление данных
New Local View (Новое локальное представление)	Создает локальное представление данных
Modify (Модифицировать)	Открывает таблицу в конструкторе таблицы
Browse (Обзор таблицы)	Показывает содержимое таблицы в режиме Browse
Remove (Удалить)	Удаляет таблицу из базы данных

Окончание табл. 6

<b>Команда</b>	<b>Назначение</b>
Find Object (Найти объект)	Находит указанный вами объект в окне конструктора базы данных
Rebuild Table Indexes (Перестроить индексы)	Перестраивает индексы
Remove Deleted Records (Удалить помеченные записи)	Физически удаляет из таблицы помеченные для удаления записи
Edit Relationship (Редактирование отношения)	Редактирует отношения между таблицами
Edit Referential Integrity (Редактирование условия целостности)	Определяет условия целостности данных
Edit Stored Procedures (Редактирование хранимых процедур)	Открывает окно редактирования хранимой процедуры
Connections (Соединения)	Выводит на экран диалоговое окно Connections(Соединения), в котором вы можете создавать или модифицировать соединения с удаленными данными
Arrange (Упорядочить)	Упорядочивает объекты по имени или типу и выравнивает их по горизонтали или вертикали
Refresh (Обновить)	Обновляет информацию в окне конструктора базы данных
Properties (Свойства)	Выводит на экран диалоговое окно Database Properties (Свойства базы данных)
Clean Up Database (Очистка базы данных)	Очищает базу данных от помеченных на удаление объектов

Для работы в окне конструктора базы данных можно использовать контекстное меню, содержащее наиболее часто используемые команды из меню Database (База данных), команду вызова справочной системы, а также команды Expand All (Развернуть все) и Collapse All (Свернуть все), предназначенные, соответственно, для раскрытия и свертывания уровней вложенности объектов в окне конструктора базы данных.

Панель инструментов Database Designer (Конструктор базы данных) содержит кнопки для выполнения наиболее часто используемых операций над базой данных. Вид панели инструментов

приведен на рис. 10, а описание кнопок панели инструментов — в табл. 7.



Рис. 10 — Панель инструментов Database Designer

Таблица 7 — Назначение кнопок панели инструментов Database Designer

Название	Назначение
New Table (Новая таблица)	Создает новую таблицу
Add Table (Добавить таблицу)	Добавляет ранее созданную таблицу в базу данных
Remove Table (Удалить таблицу)	Удаляет таблицу из базы данных
New Remote View (Новое удаленное представление)	Создает удаленное представление данных
New Local View (Новое локальное представление)	Создает локальное представление данных
Modify Table (Модифицировать таблицу)	Открывает таблицу в конструкторе таблицы
Browse Table (Обзор таблицы)	Показывает содержимое таблицы в режиме Browse
Edit Stored Procedures (Редактирование хранимых процедур)	Открывает окно для редактирования хранимых процедур
Connections (Соединения)	Создает связь с удаленными данными

## 5.5 Таблицы — основа базы данных

Таблицы составляют основу вашей базы данных. В них будет храниться вся необходимая информация. В дальнейшем данные в таблице будут дополняться новыми данными, редактироваться или исключаться из таблицы. Вы сможете просматривать данные таблиц с помощью форм или упорядочивать их по заданному критерию. Информация, содержащаяся в таблицах, может быть использована для составления отчетов. Кроме того, исполь-

зую диаграммы, вы можете графически представить информацию, содержащуюся в базе данных.

Таблица состоит из строк и столбцов и имеет уникальное имя в базе данных. В каждой из таблиц содержится информация о каких-либо объектах одного типа.

В Visual FoxPro вы можете создавать как таблицы, входящие в базу данных, так и отдельные таблицы, называемые свободными, аналогичные создаваемым в предыдущих версиях FoxPro. В данной главе мы остановимся на создании таблиц, входящих в состав базы данных, поскольку в этом случае сможете использовать все средства, предоставляемые Visual FoxPro для работы с таблицами.

### 5.5.1 Имена таблиц

Таблицы, как и любые другие объекты в Visual FoxPro, имеют имена. Как было сказано выше, таблицы бывают свободные и входящие в базу данных. Для таблиц, входящих в базу данных, вы можете задать два имени. Одно вводится в диалоговом окне Create (Создать), а второе — на вкладке Table(Таблица) окна конструктора таблицы. Имя, вводимое в диалоговом окне Create (Создать) при создании таблицы, является именем файла, в котором таблица сохраняется на диске. При задании этого имени необходимо придерживаться ограничений, накладываемых операционной системой на количество символов в имени файла. Второе имя таблицы является внутренним и хранится в базе данных. Внутреннее имя таблицы может содержать до 128 символов. Оно вводится в поле Name (Имя) вкладки Table (Таблица) окна конструктора таблицы.

Наименование таблицы может содержать буквы, цифры и знак подчеркивания. Создавая новую таблицу, необходимо помнить, что в базе данных не может быть двух таблиц, имеющих одинаковые имена. Если в базе данных уже имеется таблица с таким именем, на экране появляется запрос, заменить ли существующую таблицу новой.



### Замечание

При работе в окне конструктора базы данных внутреннее имя таблицы отображается в заголовке окна таблицы. В том случае, если таблица не имеет внутреннего имени, вместо него отображается имя файла, в котором данная таблица сохранена.

### Предупреждение

В наименовании таблицы недопустимо использование пробелов и специальных символов.

## 5.5.2 Окно конструктора таблицы

Конструктор таблицы предоставляет разработчику более широкие возможности при создании таблицы, чем мастер.

В конструктор таблицы вы можете перейти из мастера по созданию таблицы или непосредственно из диалогового окна New Table (Новая таблица) (см. рис. 11), нажав кнопку New Table (Новая таблица) и введя в диалоговом окне Create (Создать) имя создаваемой таблицы. В результате выполнения этих действий откроется окно конструктора таблицы Table Designer (Конструктор таблицы).

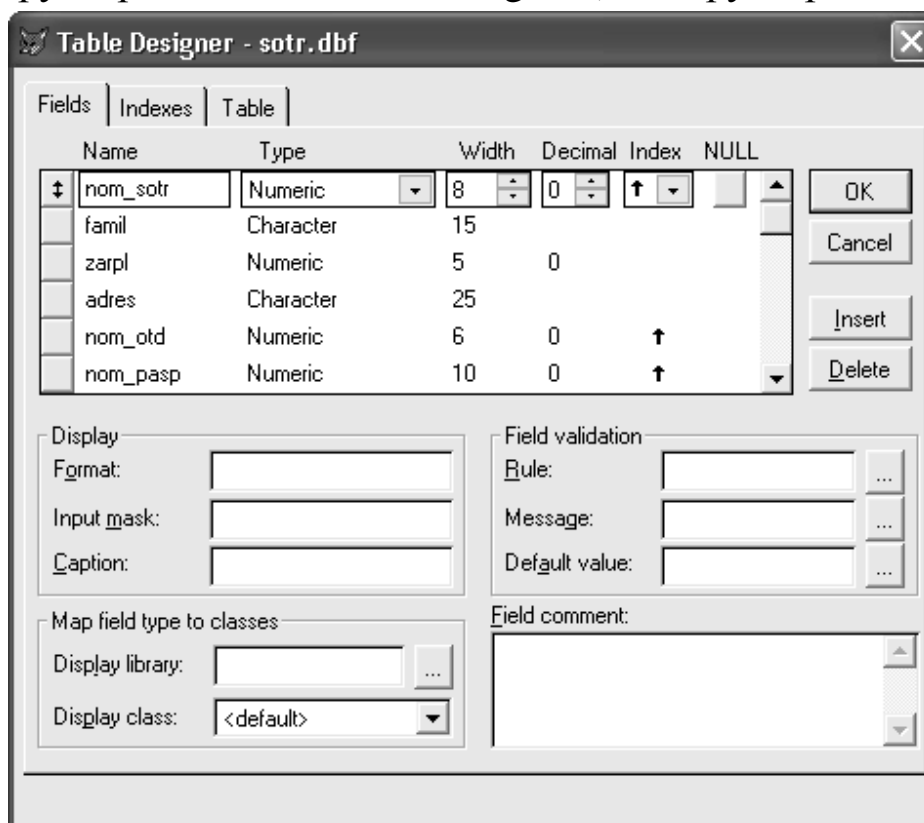


Рис. 11 — Окно конструктора таблицы Table Designer

Окно конструктора таблицы Table Designer (Конструктор таблицы) (рис. 12) содержит три вкладки, предназначенные для определения следующих параметров:

- Fields (Поля) — полей таблицы;
- Indexes (Индексы) — индексов;
- Table (Таблица) — условий достоверности вводимых данных, а также триггеров добавления, удаления и модификации.

### 5.5.3 Определение полей таблицы

При определении полей таблицы используется вкладка Fields (Поля), позволяющая ввести наименование поля, тип данных, размещаемых в поле, и ширину. При определении числовых полей необходимо также задать количество десятичных знаков.

Помимо основных параметров, в нижней части вкладки Fields (Поля) для каждого поля можно определить дополнительные параметры, которые будут определять условия ввода в него данных, а также краткое описание, которое поможет разработчику при модификации таблицы в процессе создания приложения или его сопровождения.

### 5.5.4 Ввод наименований полей

Наименования полей таблицы вводятся на вкладке Fields (Поля) в строке ввода столбца Name (Имя). При задании наименований полей вы можете использовать буквы, цифры и знак подчеркивания. Ваши попытки ввести специальные символы Visual FoxPro проигнорирует.

Длина наименования поля ограничивается 128-ю символами. Вряд ли вы воспользуетесь возможностью задавать такие длинные имена. Обычно для имен полей используют краткие названия, которые более наглядны и легко читаемы.

Visual FoxPro не позволяет иметь в таблице поля с одинаковыми именами. При вводе имени, уже имеющегося в таблице, программа выдает предупреждение, указывающее, что поле с таким именем в таблице уже существует.

### 5.5.5 Типы полей

Поля таблицы предназначены для хранения в них данных. Это могут быть числа, текстовая информация, даты, графические файлы и т.д. Для определения типа данных, размещаемых в поле, используются тип поля, его ширина и количество знаков после запятой. Для их ввода предназначены столбцы Type (Тип), Width (Ширина) и Decimal (Десятичные) вкладки Fields(Поля) конструктора таблицы.

В Visual FoxPro допустимыми являются типы полей, перечисленные в табл. 8.

Таблица 8 — Типы полей Visual FoxPro

Тип	Наименование	Отображаемые данные
Текстовый	Character, Character(binary)	Текстовые поля могут содержать буквы, цифры и специальные символы. Максимальная ширина поля составляет 254 символа. Тип Character (binary) используется в том случае, если не требуется учитывать кодовую страницу отображаемых данных
Числовой	Integer, Numeric, Float, Double	Integer отображает целые числа от $-10^{31}-1$ до $+10^{31}-1$ . Числовые поля типа Numeric и Float отображают данные с фиксированной точкой в диапазоне от $-0.9999999999 \cdot 10^{19}$ до $+0.9999999999 \cdot 10^{20}$ . Тип данных Double используется для хранения данных с высокой точностью в диапазоне от $\pm 4.94065648541247 \cdot 10^{-324}$ до $\pm 1.79769313486232 \cdot 10^{308}$
Денежный	Currency	В поле денежного типа могут содержаться числа от $-922\,337\,203\,685\,477.5807$ до $922\,337\,203\,685\,477.5807$
Дата	Date	В поле типа Date может содержаться любая дата от 01.01.0001 до 31.12.9999
Дата и время	DateTime	В поле типа DateTime может содержаться любая дата от 01.01.0001 до 31.12.9999 и время от 00:00:00 a.m. до 11:59:59 p.m.

Окончание табл. 8

Тип	Наименование	Отображаемые данные
Логический	Logical	Содержит логическое значение True (.T.) или False (.F.)
Текстовое поле произвольной длины	Memo, Memo(binary)	Мемо-поле содержит символьные данные большого объема
Двоичное поле произвольной длины	General	Поле данного типа предназначено для хранения в таблицах изображений и других двоичных данных

### 5.5.6 Задание свойств полей

В нижней части вкладки Fields (Поля) конструктора расположены поля, позволяющие задать для каждого поля таблицы свойства, которые будут использоваться при вводе в них данных.

Область Display (Отображение) содержит поля, позволяющие задать форматы ввода и отображения данных (табл. 9).

Таблица 9 — Назначение полей области Display вкладки Fields

Поле	Назначение
Format (Формат)	Задаёт формат отображения данных в формах, отчетах и окне Browse
Input mask (Маска ввода)	Задаёт формат ввода данных
Caption (Надпись)	Определяет заголовок поля

Область Map field type to classes (Используемые типы полей для классов) предназначена для указания библиотеки и имени класса, который будет использоваться для создания объектов при размещении данного поля таблицы в форме (табл. 10).

Таблица 10 — Назначение полей области Map field type to classes вкладки Fields

Поле	Назначение
Display library (Показывать библиотеку)	Задаёт местоположение и имя файла библиотеки классов
Display class (Показывать класс)	Задаёт имя класса из выбранной библиотеки

Область Field validation (Проверка правильности ввода) позволяет задать параметры, описанные в табл. 11.

Таблица 11 — Назначение полей области Field validation вкладки Fields

Поле	Назначение
Rule (Условие)	Условие правильности ввода данных
Message (Сообщение)	Сообщение, выводимое при неправильном вводе данных в поле
Default Value (Значение по умолчанию)	Значение, вводимое в поле по умолчанию

В текстовом поле Field comment (Комментарий) можно ввести краткое описание поля, которое может потребоваться при последующих модификациях структуры таблицы и сопровождении проекта.

### 5.5.7 Свойство NULL

Для каждого поля вы можете определить признак, разрешающий при вводе данных оставлять это поле пустым. Для этого используется свойство NULL в описании поля таблицы.

### 5.5.8 Назначение кнопок окна конструктора таблицы

Ввод полей в окне конструктора таблицы осуществляется последовательно. После определения всех необходимых параметров первого поля вы переходите на новую строку и вводите информацию для следующего поля таблицы. На вкладке Fields (Поля) справа расположены четыре кнопки. Кнопка ОК предназначена для закрытия окна конструктора таблицы и сохранения всех изменений, внесенных в структуру таблицы.

Если вы изменили структуру таблицы, но хотите от этого отказаться, вам необходимо воспользоваться кнопкой Cancel (Отмена).

Для добавления в таблицу нового поля установите курсор на поле, выше которого вы предполагаете разместить новое, и на-

жмите кнопку Insert(Вставить). Будет добавлена пустая строка, в которую вы можете ввести информацию о новом поле.

Для удаления поля таблицы перейдите на строку с описанием данного поля и нажмите кнопку Delete (Удалить).

Кнопки Insert (Вставить) и Delete (Удалить) на вкладке Indexes (Индексы) позволяют добавлять в таблицу новые индексы и удалять существующие.

### 5.5.9 Определение свойств таблицы

Для определения свойств самой таблицы предназначена вкладка Table (Таблица) конструктора (рис. 12).

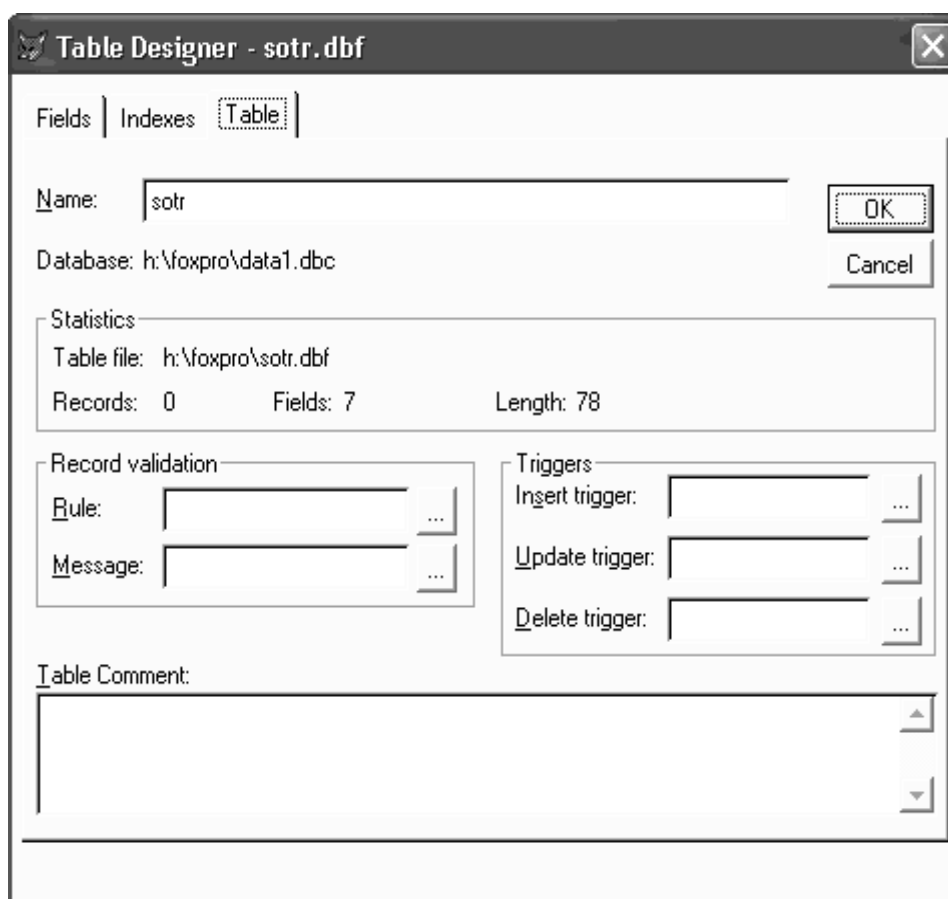


Рис 12

В ее верхней части расположено поле Name (Имя), в котором вы можете задать длинное имя таблицы. Это имя будет отображаться в окне проекта, а также использоваться при создании форм, запросов и отчетов. При создании несвободной таблицы в

строке Database (База данных) указывается имя базы данных, в которую будет входить создаваемая таблица. Это поле доступно только для чтения.

Используя поле Table Comment (Описание таблицы) вкладки Table(Таблица), вы можете ввести текстовое описание таблицы. Для определения условия проверки правильности ввода информации на уровне записей, гарантирующих достоверность вводимых в таблицу данных, и создания триггеров для добавления, изменения и удаления записей таблицы, предназначены поля областей Record validation (Достоверность записей) и Triggers (Триггеры).

### 5.5.10 Создание таблицы в конструкторе таблицы

Мы рассмотрели назначение основных параметров окна конструктора таблицы. Теперь можно приступать к созданию таблицы.

Рассмотрим последовательность действий при создании таблицы sotr (рис. 13), содержащей сведения о сотрудниках, из конструктора базы данных data1, предназначенной для хранения данных предприятия.

Откройте проект proj1.

В иерархической структуре конструктора проекта установите курсор на базу данных data1 и нажмите кнопку Modify (Модифицировать). На экране откроется окно конструктора базы данных.

Находясь в конструкторе базы данных, нажмите кнопку New Table (Новая таблица) на панели инструментов Database Designer (Конструктор базы данных).

В поле Enter table (Введите имя таблицы) диалогового окна Create (Создать) задайте имя таблицы customer и нажмите кнопку Сохранить. На экране появится окно конструктора таблицы Table Designer (Конструктор таблицы). Курсор находится в строке ввода столбца Name (Имя). Вы можете вводить имя, тип и ширину полей создаваемой таблицы.

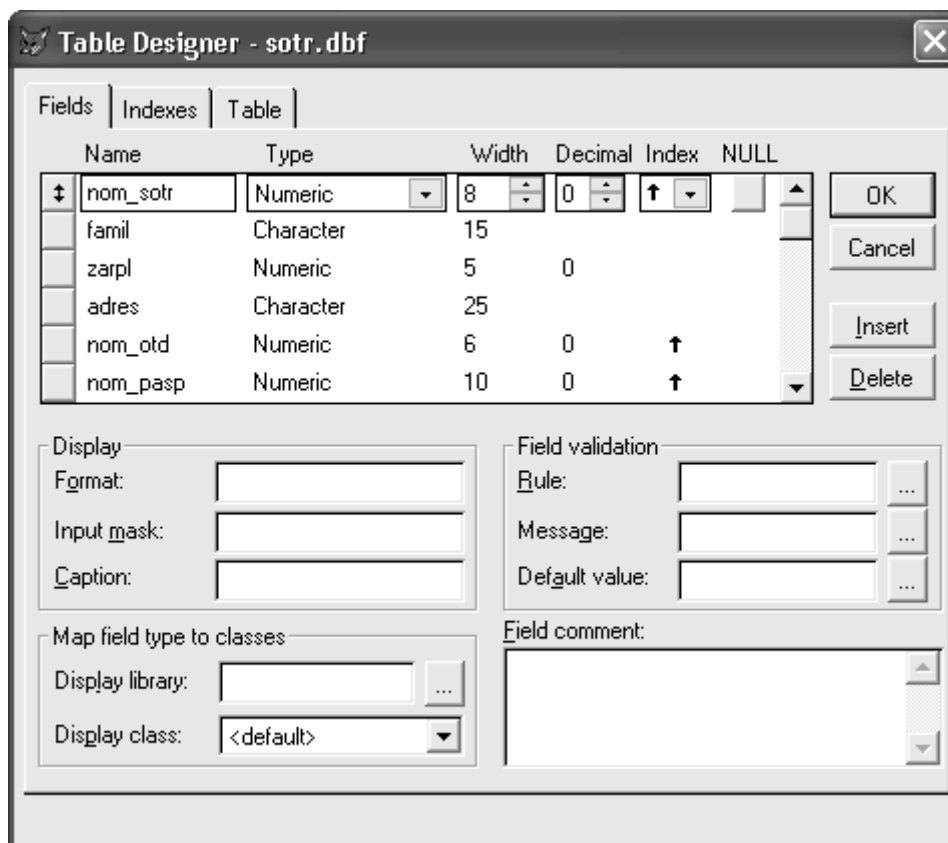


Рис. 13 — Структура таблицы Sotr

В строке ввода столбца Name (Имя) введите наименование поля, содержащего номер сотрудника клиента nom\_sotr. Для перехода в следующий столбец воспользуйтесь мышью или нажмите клавишу <Tab>. Нажмите кнопку раскрытия списка Type (Тип) и выберите значение Number и укажите разрядность 8. Кроме того, вы можете изменить тип поля с помощью клавиш.

Перейдите в поле Caption (Надпись), расположенное в области Display(Отображение), и укажите заголовок поля Номер сотрудника. В поле Fieldcomment (Комментарий) введите краткое описание поля.

Аналогичным образом определите остальные поля. Для перехода с одного поля на другое используйте клавишу <Tab >, клавиши-стрелки или мышь. Для изменения ширины поля используйте кнопки изменения ширины поля, расположенные в правой части поля, а также клавиши.



### Замечание

В процессе создания структуры вы можете изменить порядок следования полей в таблице. Для этого используйте кнопку с двунаправленной стрелкой, расположенную с левой стороны имени поля.

При описании поля Adres, указывающего адресом сотрудника, определите значение, вводимое в поле по умолчанию при добавлении новой записи. Для этого перейдите в поле Default value (Значение по умолчанию) и введите Кирова 12—22.

После завершения определения структуры таблицы нажмите кнопку ОК.

### Замечание

В окне конструктора вы можете отказаться от создания новой таблицы или введенных в ее структуру изменений. Для этого нажмите кнопку Cancel(Отмена). На экране появится запрос системы об отмене или принятии изменений (рис. 14). Выберите значение Yes (Да).

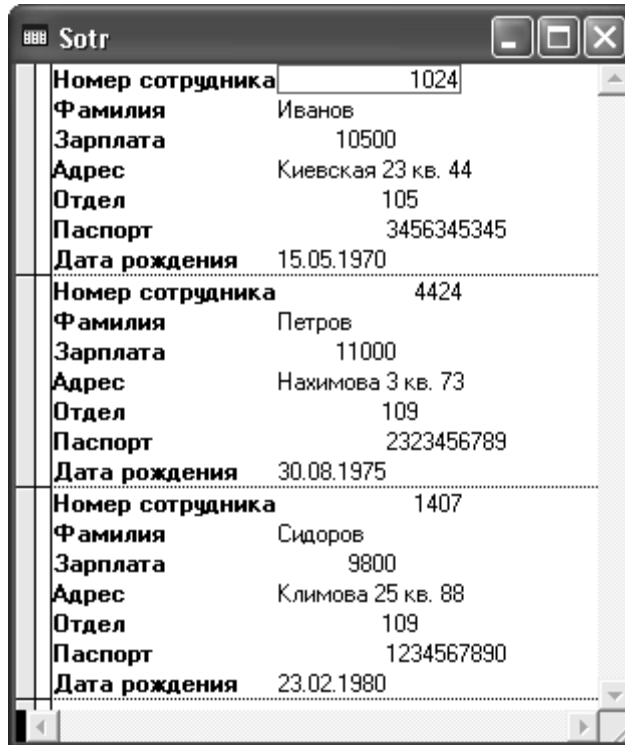


Рис. 14 — Запрос системы об отмене изменений структуры таблицы

На экране появляется запрос, будете ли вы сейчас вводить данные в созданную таблицу. Если вы готовы приступить к вводу данных, нажмите кнопку Yes (Да). Созданная таблица откроется в режиме Edit (Ввод), позволяя вводить данные. При нажатии кнопки No (Нет) окно конструктора таблицы закрывается.

### 5.5.11 Режимы просмотра таблицы

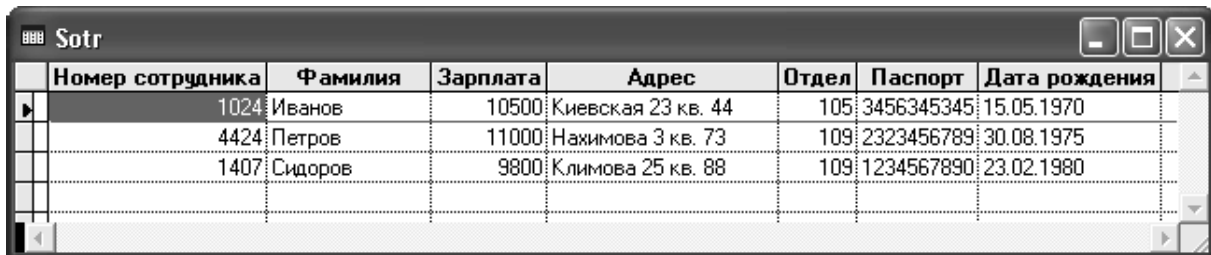
В Visual FoxPro существует два режима просмотра таблицы: Edit (Ввод) (рис. 15) и Browse (Обзор).



Номер сотрудника	1024
Фамилия	Иванов
Зарплата	10500
Адрес	Киевская 23 кв. 44
Отдел	105
Паспорт	3456345345
Дата рождения	15.05.1970
Номер сотрудника	4424
Фамилия	Петров
Зарплата	11000
Адрес	Нахимова 3 кв. 73
Отдел	109
Паспорт	2323456789
Дата рождения	30.08.1975
Номер сотрудника	1407
Фамилия	Сидоров
Зарплата	9800
Адрес	Климова 25 кв. 88
Отдел	109
Паспорт	1234567890
Дата рождения	23.02.1980

Рис. 15 — Просмотр таблицы в режиме Edit

В режиме Edit (Ввод) (рис. 16) поля таблицы располагаются в столбец друг под другом. Записи таблицы отделяются горизонтальными линиями.



	Номер сотрудника	Фамилия	Зарплата	Адрес	Отдел	Паспорт	Дата рождения
	1024	Иванов	10500	Киевская 23 кв. 44	105	3456345345	15.05.1970
	4424	Петров	11000	Нахимова 3 кв. 73	109	2323456789	30.08.1975
	1407	Сидоров	9800	Климова 25 кв. 88	109	1234567890	23.02.1980

Рис. 16 — Просмотр таблицы в режиме Browse

По умолчанию после создания структуры таблицы Visual FoxPro предлагает вам вводить данные в таблицу в режиме Edit (Ввод). В этом режиме пользователю нет необходимости после ввода информации во все поля одной записи заботиться о добавлении новой пустой записи. Visual FoxPro автоматически добавляет следующую запись, и пользователь продолжает вводить информацию.

В режиме Browse (Обзор) (рис. 16) таблица более наглядна. Поля расположены в один ряд. Одна строка соответствует одной записи таблицы, а записи размещены одна под другой.

При вводе данных в таблицу, представленную в режиме Browse (Обзор), после ввода одной записи пользователю необходимо самому добавлять новую пустую запись, используя для этого команду Append New Record (Добавить новую запись) из меню Table (Таблица) или комбинацию клавиш <Ctrl>+<Y>.

### **Замечание**

Для перехода в режимы просмотра таблицы Edit (Ввод) и Browse (Обзор) можно использовать одноименные команды из меню View (Вид).

## **5.5.12 Модификация таблицы**

Структуру таблицы, созданную с помощью мастера или конструктора таблицы, можно модифицировать, т.е. изменить наименование любого поля и его тип, вставить новое поле или удалить существующее, изменить порядок следования полей в таблице.

Чтобы модифицировать таблицу, откройте ее в конструкторе таблицы. Для этого выполните одно из следующих действий.

В окне проекта установите курсор на модифицируемую таблицу и нажмите кнопку Modify (Модифицировать).

В окне проекта установите курсор на модифицируемую таблицу, нажмите правую кнопку мыши и в появившемся на экране контекстном меню выберите команду Modify (Модифицировать).

При открытии конструктора таблицы из окна конструктора базы данных выделите модифицируемую таблицу и нажмите кнопку Modify Table (Модифицировать таблицу) на панели инструментов Database Designer (Конструктор базы данных) или выберите команду Modify (Модифицировать) в меню Database (База данных).

В результате на экране открывается диалоговое окно Table Designer (Конструктор таблицы), содержащее структуру модифицируемой таблицы.

### 5.5.13 Изменение наименований полей и их типов

Ошибки, допущенные при задании имени поля или его типа, легко устраняются. Установите курсор на имя поля, которое хотите изменить, и, используя клавишу <Backspace> или <Del>, удалите ошибочные символы. После этого введите правильное имя поля.

Для изменения типа поля установите курсор в столбец Type (Тип) и выберите из списка требуемое значение.

#### **Предупреждение**

Изменение типов полей таблицы, содержащей данные, может привести к потере информации.

### 5.5.14 Добавление и удаление полей

Для добавления нового поля в таблицу выполните следующие действия:

- С помощью клавиш-стрелок или мыши установите курсор на строку, перед которой вы хотите вставить пропущенное поле. При этом курсор может находиться в любом столбце Name (Имя), Type (Тип) или Width (Размер).
- Нажмите кнопку Insert (Вставить).
- Visual FoxPro вставляет пустую строку с именем NewField. Введите требуемое имя поля.
- Используя значения списка Type (Тип), задайте тип поля.
- В столбце Width (Размер) введите размер нового поля.
- Для удаления поля из структуры таблицы установите курсор на поле, которое требуется удалить, и нажмите кнопку Delete (Удалить) окна конструктора.

### 5.5.15 Изменение порядка расположения полей

Порядок расположения полей, заданный при создании структуры таблицы, можно изменить. Для этого выполните следующие действия:

- Установите курсор на поле, расположение которого хотите изменить. На кнопке появляется значок перемещения поля в виде двунаправленной стрелки.
- Установите курсор на значок перемещения.
- Нажмите кнопку мыши и, удерживая ее нажатой, переместите значок вверх или вниз на требуемое место в структуре.
- Отпустите кнопку мыши. Поле изменило свое местоположение.

### **5.5.16 Индексы**

Среди требований, предъявляемых к СУБД, основное место занимает возможность быстрого поиска необходимой информации. Прекрасным средством, позволяющим решить эту проблему, является использование индексов, которые, в зависимости от количества полей, применяемых при их создании, делятся на простые и составные.

#### **Замечание**

В отличие от многих реляционных баз данных, при определении составного индекса в Visual FoxPro можно использовать не просто список индексных полей, а индексное выражение, при формировании которого применяются наименования полей таблицы.

Индексы в Visual FoxPro предназначаются также для создания первичных ключей, используемых при определении отношений между таблицами и условий целостности данных. В этом случае индексы должны быть уникальными. Это означает, что если для создания уникального индекса применяется простой индекс, значения поля должны быть неповторяющимися, т.е. уникальными. Часто для создания уникального индекса используется составной индекс, т.е. он может состоять из нескольких полей. В этом случае каждое поле в отдельности может содержать повторяющиеся значения, но индексное выражение, в состав которого входят эти поля, должно быть уникальным.

Значения индексов хранятся в индексном файле, в котором для каждого значения индекса существует уникальная ссылка, указывающая на местонахождение в таблице записи, соответствующей индексу, что обеспечивает при поиске нужной информации прямой доступ к записи на основании упорядоченных значений индекса, а не последовательный просмотр всей таблицы. Это, естественно, ускоряет поиск. Для создания индекса таблицы используется вкладка Indexes (Индексы) (рис.17) окна конструктора таблицы Table Designer (Конструктор таблицы).

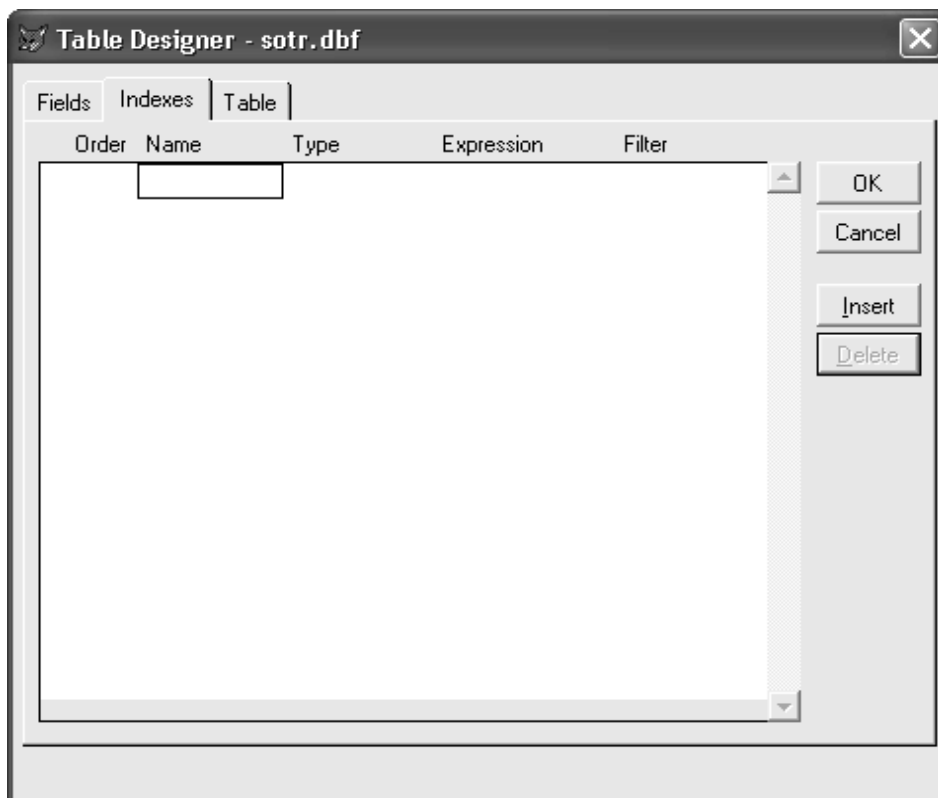


Рис. 17 — Вкладка Indexes конструктора таблицы

Все индексы в Visual FoxPro имеют имена, задаваемые в поле Name (Имя). Слева от имени индекса в столбце Order (Упорядочение) располагается переключатель, определяющий порядок, в котором будут выстраиваться значения индексного выражения. По умолчанию при создании индекса в данном поле появляется стрелка, направленная вверх. Это означает, что значения индексного выражения упорядочены по возрастанию. Если стрелка направлена вниз, это говорит о том, что значения упорядочены по

убыванию. Для изменения способа упорядочения можно нажать клавишу <Пробел>или щелкнуть кнопкой мыши.

Список Type (Тип) используется для задания типа создаваемого индекса и содержит значения, описанные в табл. 12.

Таблица 12 — Описание типов индекса

Тип индекса	Описание
Regular(Обычный)	Создается индекс, в котором для каждой записи таблицы хранится значение индексного выражения. Если несколько записей имеют одинаковое значение индексного выражения, то каждое значение хранится отдельно и содержит ссылку на связанную с ней запись. Этот индекс используется для создания внешнего ключа.
Unique (Уникальный)	Создается индекс, в котором хранятся только неповторяющиеся значения индексного выражения. Если две или более записей содержат одинаковое значение индексного выражения, то будет храниться только одно значение и ссылка на первую из записей с одинаковым значением индексного выражения. Таблица может иметь несколько уникальных индексов
Candidate (Кандидат)	Создается уникальный индекс, который не содержит полей с пустыми значениями. Этот индекс обладает всеми качествами первичного ключа и не является им только по той причине, что таблица не может содержать более одного первичного ключа
Primary (Первичный)	Создается уникальный индекс, который используется для связывания таблиц и определения условий целостности данных. Поля, входящие в первичный ключ, не должны допускать ввода пустых значений. В отличие от уникального индекса, таблица может иметь только один первичный ключ

Значение индекса или индексного выражения вводится в поле Expression(Выражение). Вы можете ввести индексное выражение непосредственно в поле ввода или для формирования выражения использовать диалоговое окно конструктора выражений Expression Builder (Построитель выражения), представленное на рис. 18. Для открытия данного окна нажмите кнопку, расположенную справа от поля Expression (Выражение).

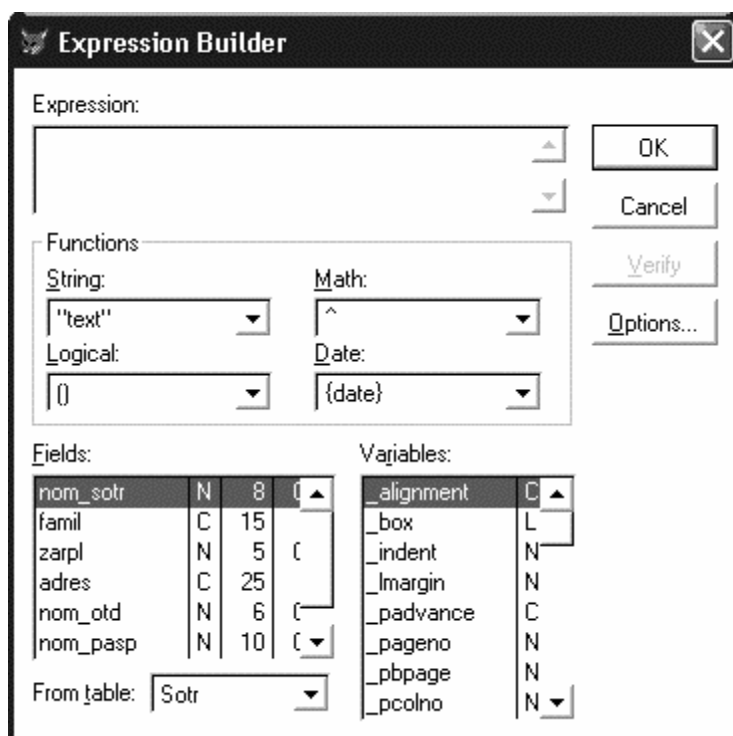


Рис. 18 — Диалоговое окно Expression Builder

При использовании в индексном выражении полей разных типов вы должны проверять допустимость введенного выражения. Например, для таблицы Sotr вы можете создать индекс для упорядочения данных по адресу и по номеру сотрудника: `adres+STR(nom_sotr)`.

### Замечание

Без индексов в Visual FoxPro не обойтись. Однако при их создании вы должны помнить, что если вы используете индекс, в состав которого входят поля большой ширины, индексный файл может оказаться очень больших размеров. Visual FoxPro не разрешает при создании индексов использовать Memo-поля и поля, содержащие графические изображения.

### 5.5.17 Создание первичного ключа для таблицы

Рассмотрим последовательность действий при создании первичного ключа для таблицы Sotrudnik. Так как первичный ключ таблицы должен быть уникальным, для его создания можно



использовать поле `nomer_sotr` таблицы, содержащее номера сотрудников.

Откройте окно конструктора таблицы для таблицы `Sotr`. Для этого в окне проекта установите курсор на модифицируемую таблицу и нажмите кнопку `Modify` (Модифицировать) конструктора проекта.

В окне `Table Designer` (Конструктор таблицы) перейдите на вкладку `Indexes` (Индексы).

В поле `Name` (Имя) введите имя индекса `nom_s`.

Из списка возможных типов индекса в поле `Type` (Тип) выберите значение `Primary` (Первичный).

Перейдите в поле `Expression` (Выражение) и введите выражение для индекса `nom_sotr`.

Установите переключатель столбца `Order` (Упорядочение) в значение по возрастанию. Результат создания первичного ключа представлен на рис. 19.

Для завершения создания индекса нажмите кнопку `OK`.

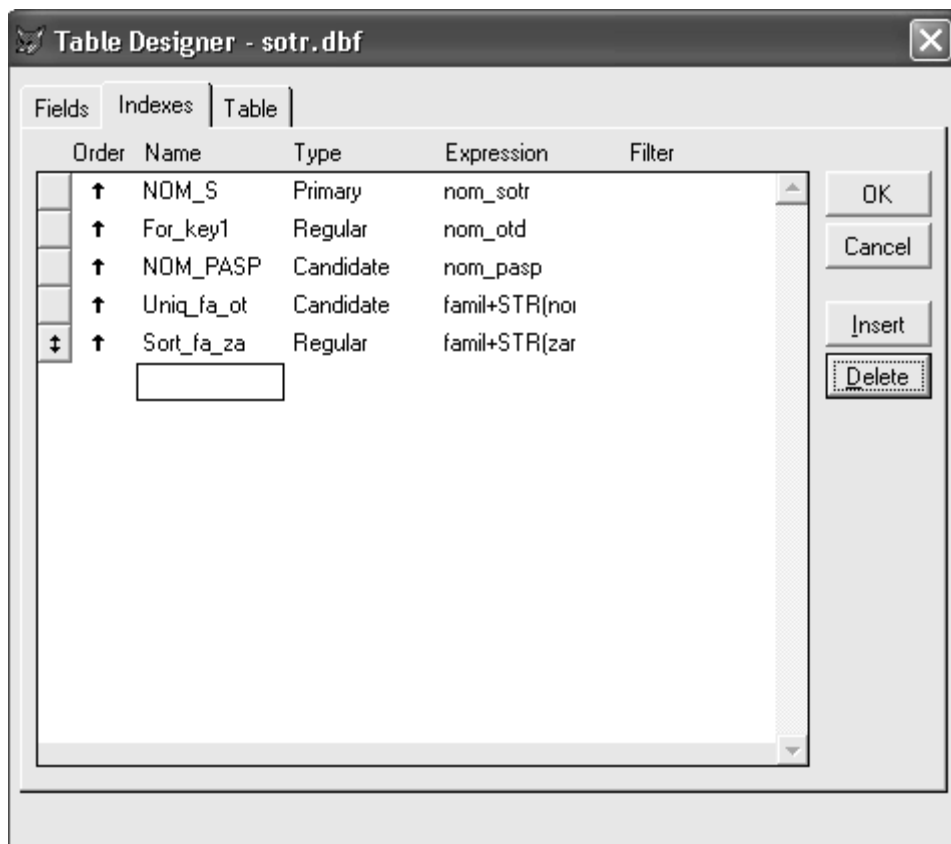


Рис. 19 — Первичный ключ таблицы `Sotr`

### 5.5.18 Создание составного индекса

Рассмотрим создание индекса для таблицы Sotr, который будет упорядочивать данные в таблице по фамилии и зарплате. Для создания такого индекса воспользуемся индексным выражением.

Откройте окно Table Designer (Конструктор таблицы) для таблицы Sotr и перейдите на вкладку Indexes (Индексы).

В поле Name (Имя) введите имя индекса Sort\_fa\_za.

Из списка возможных типов индекса в поле Type (Тип) выберите значение Regular (Обычный).

Для задания индекса нажмите кнопку открытия построителя выражений для поля Expression (Выражение). В окне построителя сформируйте следующее выражение:

Famil+STR(zarpl)

Проверьте синтаксис выражения, воспользовавшись кнопкой Verify (Проверить), и закройте окно построителя выражений.

Для завершения создания индекса нажмите кнопку ОК.

## 5.6 Форма как средство ввода и редактирования данных

В Visual FoxPro для просмотра, ввода и редактирования данных, хранящихся в таблицах, используются формы, являющиеся более наглядным средством представления информации. Рассмотрим, например, приложение, предназначенное для работы с бухгалтерскими документами, которые имеют стандартный вид. Естественно, что формы, предназначенные для ввода документов, должны выглядеть на экране монитора точно так же, как стандартные бланки этих документов. Другим важным преимуществом форм является то, что они позволяют работать не с одной, а с несколькими связанными таблицами, что, в свою очередь, также увеличивает наглядность.

Пользователю приложения нет необходимости знать, что такое VisualFoxPro, какие команды используются для добавления или удаления записей в таблицах. Он может даже вообще не знать, с использованием каких программных средств создавалось приложение. Для него главным является перемещение по табли-

це, добавление новых записей, редактирование и удаление имеющихся. Все эти возможности имеются в формах.

Форма предназначена для ввода и просмотра списка товаров магазина мелкооптовой торговли. В форме размещен иерархический список. В полях формы, находящихся справа, отображается краткое наименование товара и его описание. В верхней части экрана расположена панель, позволяющая управлять формой: перемещаться по записям таблицы, добавлять и удалять записи, осуществлять поиск необходимой информации, печатать отчет.

### **Замечание**

В Visual FoxPro (в отличие от FoxPro для Windows) форма существует как функционально полный объект проектирования, который не требует предварительной генерации программного кода. Поэтому теперь вы не можете редактировать сгенерированный код и должны все действия, связанные с данной формой, описать в ней самой.

При создании форм в Visual FoxPro разработчик может использовать следующие средства:

- Form Wizard — мастер форм;
- Form Builder — построитель формы;
- Builder — построитель объектов формы;
- AutoFormat Builder — построитель авто формата;
- Form Designer — конструктор форм.

Чтобы создать форму для одной или связанных таблиц с возможностью задания отображаемых в форме полей, стиля их отображения и указания типа кнопок управления, можно использовать мастер создания форм.

Для самостоятельной разработки формы с заданными свойствами или изменения формы, созданной с помощью мастера, необходимо использовать конструктор форм.

Для облегчения размещения в конструкторе форм полей и надписей, оформленных в соответствии с выбранным стилем, можно использовать построитель формы. Помимо этого, в конструкторе форм для большинства объектов (полей, списков, переключателей, таблиц и т.д.) существуют построители, позволяю-

щие размещать в форме заданные объекты и настраивать их свойства. Кроме того, в Visual FoxPro имеется построитель автоформата, позволяющий задать одинаковый стиль оформления выделенных элементов формы.

### 5.6.1 Запуск формы

Форма, созданная в Visual FoxPro, не требует генерации программных кодов, как это было в FoxPro для Windows. Ее можно сразу запустить на выполнение. Для запуска формы в Visual FoxPro существует много способов. Если вы находитесь в конструкторе форм, то выполните одно из следующих действий:

- Выберите в меню Form (Форма) команду Run Form (Запустить форму).
- Выполните команду Run Form (Запустить форму) контекстного меню.
- Нажмите кнопку Run (Запустить) на стандартной панели инструментов Visual FoxPro.
- Нажмите комбинацию клавиш <Ctrl>+<E>.
- При нахождении в окне конструктора проекта для запуска формы установите курсор на ее имя и нажмите кнопку Run (Запустить).

Вы можете также запустить форму как программу. Для этого выполните следующие действия:

- В меню Program (Программа) выберите команду Do (Выполнить).
- В открывшемся диалоговом окне Do (Выполнить) установите тип выполняемого файла (в данном случае — Form (Форма)) и выберите из списка файлов необходимую форму.
- Нажмите кнопку Do (Выполнить).

### 5.6.2 Объекты формы

Откройте созданную форму в конструкторе форм рис.. Для этого выделите форму в окне проекта и выполните одно из следующих действий:

- Нажмите кнопку Modify (Модифицировать) окна проекта.
- Дважды щелкните на ее наименовании.
- Выберите команду Modify File (Модифицировать файл) из меню Project(Проект).
- Нажмите кнопку Modify Form (Модифицировать форму) стандартной панели инструментов.

Все объекты, размещенные в форме (заголовок, поля, надписи, кнопки, линии), характеризуются свойствами, которые вы можете настроить в соответствии со своими требованиями. Помимо свойств, для объектов существуют встроенные методы, выполняемые при наступлении связанных с ними событий. Для просмотра свойств и методов объекта, размещенного в форме, выделите его, а затем выполните одно из следующих действий:

- Выберите команду Properties (Свойства) из меню View (Вид).
- Нажмите правую кнопку мыши и выберите пункт контекстного меню Properties (Свойства).
- Нажмите кнопку Properties Window (Окно свойств) на панели инструментов Form Designer (Конструктор форм).

В результате откроется окно Properties (Свойства) со свойствами и методами выбранного объекта (рис. 20).

Раскрывающийся список в верхней части окна Properties (Свойства) содержит перечень всех объектов формы. Его можно использовать для выбора объекта вместо выделения нужного объекта в форме. Вкладки, расположенные ниже списка, содержат названия сгруппированных по типам свойств и методов объектов формы (табл. 13).

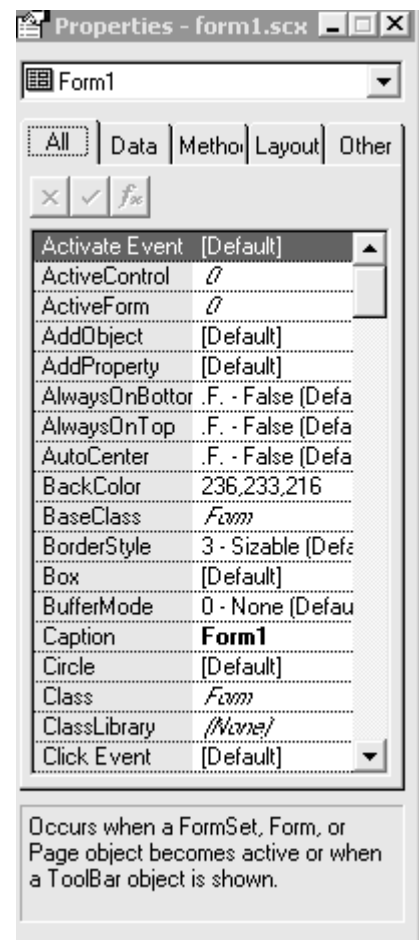


Рис. 20 — Окно Properties

Таблица 13 — Назначение вкладок окна Properties

<b>Вкладка</b>	<b>Назначение</b>
All (Все)	Содержит список всех свойств и методов формы в алфавитном порядке
Data (Данные)	Содержит свойства объекта, связанные с источником данных
Methods (Методы)	Содержит список всех методов объекта
Layout (Расположение)	Содержит свойства объекта, связанные с его оформлением
Other (Другие)	В данную вкладку собраны все свойства, не вошедшие во вкладки Data (Данные) и Layout (Расположение)

Для изменения какого-либо свойства объекта необходимо открыть окно Properties (Свойства) и перейти на строку, содержащую данное свойство. Значение свойства отобразится в столбце, расположенном правее наименования свойства, а для редактирования его значения используется поле ввода, расположенное над списком всех свойств объекта. При изменении значения свойства возможны варианты действий, описанные в табл. 14.

Таблица 14 — Действия пользователя в зависимости от типа свойства

<b>Тип свойства</b>	<b>Состояние поля ввода значения свойства</b>
Свойство доступно только для чтения.	Поле значения свойства недоступно для ввода. Информация в поле выделена курсивом. Примером такого свойства на рис. 5.11 является свойство BaseClass(Базовый класс). Данное свойство не редактируется
Свойство может редактироваться.	Поле значения свойства активно, в него можно ввести с клавиатуры требуемое значение. Примером такого свойства является Caption (Надпись)
Возможны два или более различных вариантов значений свойства.	При выборе такого свойства в поле ввода значения появляется кнопка раскрытия списка, позволяющая для ввода нового значения использовать элементы списка. На рис. 5.12 показан список для изменения значения свойства Alignment (Выравнивание)

Окончание табл. 14

Тип свойства	Состояние поля ввода значения свойства
Возможен выбор свойств с помощью окна настройки.	При выборе свойства рядом с полем значения свойства появляется кнопка открытия диалогового окна, из которого выбираются необходимые значения

Слева от поля ввода свойства находятся три кнопки, которые предназначены для выполнения функций, описанных в табл. 15.

Таблица 15 — Назначение кнопок окна Properties

Назначение
Вызывает построитель выражений для определения значения свойства
Нажатие этой кнопки подтверждает ввод значения свойства
Кнопка используется для отказа от введенного в поле ввода значения

В процессе создания формы вы можете перемещать, удалять объекты или изменять их размеры и свойства.

Чтобы управлять объектом, вы сначала должны его выделить. Для выделения одного объекта достаточно щелкнуть на нем. Для выделения нескольких объектов выполните одно из следующих действий.

Нажмите клавишу <Shift>. Удерживая ее в нажатом состоянии, щелкните мышью поочередно на всех выделяемых объектах.

Выберите кнопку Select Objects (Выбрать объекты) на панели инструментов Form Controls (Элементы управления формы). Установите указатель в форму. Не отпуская кнопку мыши, нарисуйте рамку выделения так, чтобы внутри нее оказались все необходимые объекты.

### Замечание

Второй способ применим в том случае, если все выделяемые объекты размещены в форме компактной группой.

Для выделения всех объектов в форме можно воспользоваться командой Select All (Выделить все) из меню Edit (Правка) или комбинацией клавиш <Ctrl>+<A>. После того как объекты выделены, вы можете управлять ими как единым целым.

Если вы хотите отменить выделение со всех объектов формы, щелкните мышью вне выделенных объектов. Для снятия выделения с отдельных объектов выполните следующие действия:

- Нажмите и удерживайте клавишу <Shift>.
- Щелкните мышью поочередно на объектах, с которых хотите снять выделение.

Visual FoxPro позволяет перемещать один или несколько объектов формы одновременно как мышью, так и клавишами-стрелками. Использование клавиш-стрелок применяется, когда требуется точное позиционирование, т.к. одно нажатие клавиши приводит к перемещению объекта на один пиксель. Если требуется переместить объекты на большое расстояние, можно комбинировать оба этих способа: сначала вы перемещаете объекты мышью, а затем с помощью клавиш-стрелок задаете их точное расположение.

Для удаления из формы выделенных объектов выполните одно из следующих действий:

- В меню Edit (Правка) выберите команду Cut (Вырезать).
- Нажмите клавишу <Backspace>.
- Нажмите клавишу <Del>.

Для изменения размера одного выделенного в форме объекта можно использовать маркеры управления, представляющие собой черные квадратики по углам и сторонам.

Если требуется установить точные размеры объекта, лучше использовать свойства Height и width, определяющие его высоту и ширину соответственно. Для этого откройте окно свойств объекта Properties (Свойства) и с помощью клавиатуры введите необходимые значения в поля свойств. Для улучшения внешнего вида размещенные в форме объекты выравнивают относительно друг друга и сетки формы. Для выравнивания можно использовать команды меню Format (Формат), а также панель инструментов Layout

### 5.6.3 Конструктор форм

Любая форма в Visual FoxPro состоит из объектов, каждый из которых имеет характерные свойства. Для любого объекта вы



можете указать действия, выполняемые программой при наступлении определенных событий. Процесс создания формы в конструкторе форм состоит в размещении в форме объектов и определении свойств, а также связанных с ними событий и выполняемых действий.

Для открытия окна конструктора форм при создании новой формы воспользуйтесь одним из следующих способов:

- Выполните команду New (Новый) из меню File (Файл). В открывшемся диалоговом окне New (Новый) выберите опцию Form (Форма) и нажмите кнопку New File (Новый файл).
- Нажмите кнопку на стандартной панели инструментов Visual FoxPro. В открывшемся диалоговом окне New (Новый) выберите опцию Form(Форма) и нажмите кнопку New File (Новый файл).
- Для размещения создаваемой формы в проекте выберите вкладку Documents (Документы), перейдите в группу Forms (Формы) и нажмите кнопку New окна проекта. Затем в открывшемся окне New Form (Новая форма) нажмите одноименную кнопку.

На рис. 21 показано окно конструктора форм, содержащее панели инструментов Color Palette (Цветовая палитра), Layout (Расположение), Form Designer (Конструктор форм) и Form Controls (Элементы управления формы), используемые при работе в конструкторе. В окне конструктора размещена новая форма, с которой вы можете работать.

В том случае, когда необходимые панели инструментов отсутствуют, для их отображения на экране выберите соответствующие команды в меню View(Вид) (рис. 22) или установите флажки выбора панелей инструментов в диалоговом окне Toolbars (Панели инструментов).

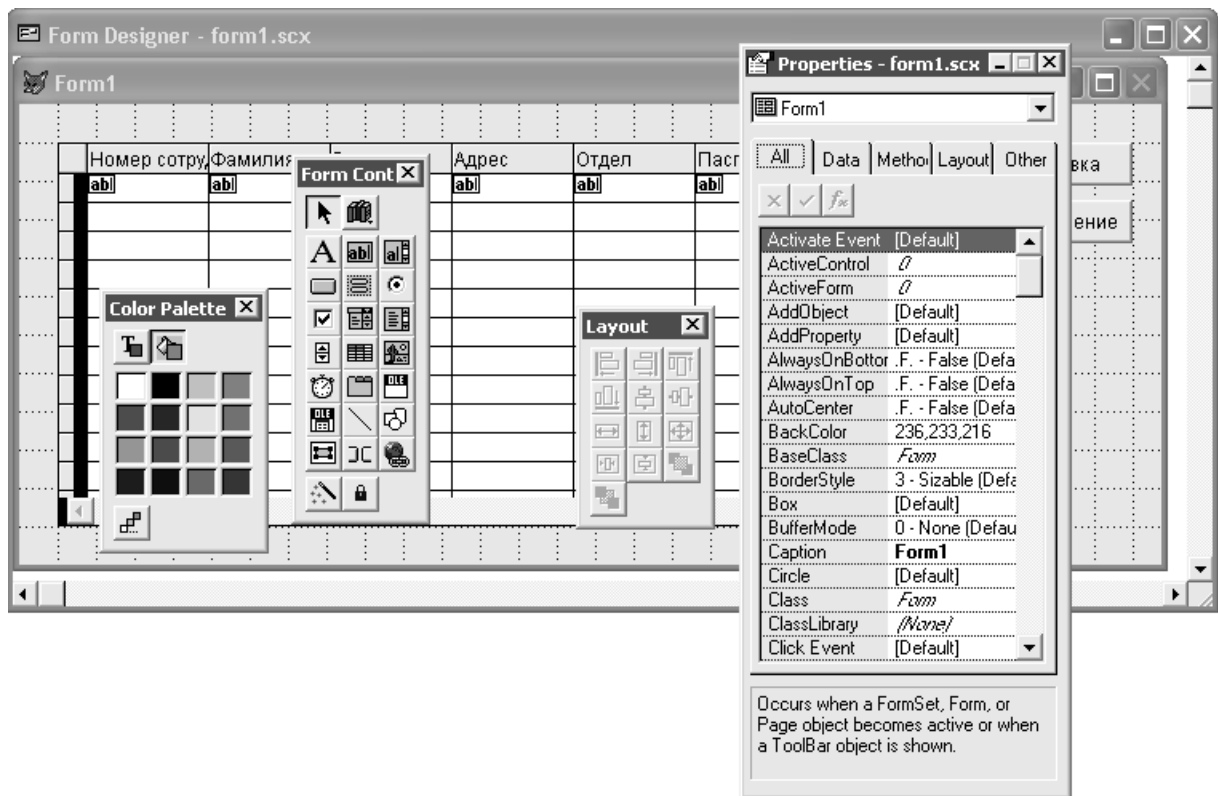


Рис. 21 — Окно конструктора форм с панелями инструментов



Рис. 22 — Установленные метки в опциях меню View для отображения на экране панелей инструментов

Панель инструментов Form Controls (Элементы управления формы) используется для размещения в форме объектов. Краткое описание кнопок этой панели приведено в табл. 16. Более подробно назначение этих кнопок будет рассмотрено ниже.

Таблица 16 — Кнопки панели инструментов Form Controls

Наименование	Назначение
Select Objects (Выбор объектов)	Указатель выделения. Позволяет выбирать в форме объекты
View Classes (Просмотр классов)	Позволяет выбрать класс для создаваемых в форме объектов
Label (Метка)	Создает в форме текстовый объект
Text Box (Поле ввода)	Создает в форме поле ввода
Edit Box (Поле редактирования)	Создает в форме поле редактирования
Command Button (Кнопка)	Создает в форме кнопку управления
Option Group (Переключатель)	Создает в форме переключатель
Check Box (Флажок)	Создает в форме флажок
Grid (Таблица)	Создает в форме для размещения полей объект в виде таблицы
Combo Box (Раскрывающийся список)	Создает в форме раскрывающийся список
List Box (Список)	Создает в форме список
Spinner (Счетчик)	Создает в форме поле ввода значения в виде счетчика
Line (Линия)	Создает в форме линию
Shape (Контур)	Создает в форме контур
Container (Контейнер)	Создает в форме контейнер
Image (Изображение)	Размещает в форме рисунок
Command Group (Группа кнопок)	Размещает в форме группу кнопок
Timer (Таймер)	Создает в форме объект типа таймера
Page Frame (Вкладка)	Размещает в форме страницы с вкладками
ActivX Bound Control(OleBoundControl) (ActivX-объект)	Отображает содержимое OLE-объекта, хранящегося в поле типа General
ActivX Control(OleControl) (OLE-объект)	Создает OLE-объект
HyperLink (Гиперссылка)	Создает объект для работы с Интернетом

Окончание табл. 16

Наименование	Назначение
Separator (Разделитель)	Размещает на панели инструментов разделитель кнопок
Builder Lock (Закрепитель построителя)	Закрепляет выбор построителя
Button Lock (Закрепитель кнопки)	Закрепляет выбранную кнопку на панели инструментов

Панель инструментов Form Designer (Конструктор форм) содержит кнопки вызова панелей инструментов Form Controls (Элементы управления формы), Color Palette (Цветовая палитра), Layout (Расположение). С помощью этой панели можно выполнять и некоторые дополнительные действия по управлению формой. Краткое назначение кнопок данной панели инструментов приведено в табл. 17.

Таблица 17 — Кнопки панели инструментов Form Designer

Наименование	Назначение
Set Tab Order (Порядок объектов)	Переключает конструктор форм в режим установления порядка обхода объектов формы
Data Environment (Окружение данных)	Открывает окно определения среды окружения формы
Code Window (Окно кода)	Открывает окно просмотра исходного кода формы
Color Palette Toolbar (Панель инструментов Цветовая палитра)	Отображает на экране панель инструментов Color Palette (Цветовая палитра)
Form Builder (Построитель формы)	Вызывает построитель формы
Properties Window (Окно свойств)	Открывает на экране окно свойств объектов формы
Form Controls Toolbar (Панель инструментов Элементы управления формы)	Вызывает на экран панель инструментов FormControls (Элементы управления формы)
Layout (Расположение)	Вызывает на экран панель инструментов Layout(Расположение)
Auto Format (Авто формат)	Вызывает построитель авто формата для выбранных объектов формы

Для выравнивания объектов, размещенных в форме, удобно использовать панель инструментов Layout (Расположение). Краткое назначение кнопок данной панели приведено в табл. 18.

Таблица 18 — Кнопки панели инструментов Layout

<b>Наименование</b>	<b>Назначение</b>
Align Left Sides (По левому краю)	Выравнивает выбранные объекты по левому краю самого левого объекта
Align Top Edges (По верхнему краю)	Выравнивает выбранные объекты по верхнему краю самого верхнего объекта
Align Right Sides (По правому краю)	Выравнивает выбранные объекты по правому краю самого правого объекта
Align Bottom Edges (По нижнему краю)	Выравнивает выбранные объекты по нижнему краю самого нижнего объекта
Align Vertical Centers (По вертикали)	Выравнивает выбранные объекты по вертикали
Align Horizontal Centers (По горизонтали)	Выравнивает выбранные объекты по горизонтали
Center Vertically (По вертикальному центру)	Центрирует выбранные объекты относительно вертикальной средней линии формы
Center Horizontally (По горизонтальному центру)	Центрирует выбранные объекты относительно горизонтальной средней линии
Same Width (Одинаковая ширина)	Устанавливает одинаковую ширину для выбранных объектов формы
Same Size (Одинаковый размер)	Устанавливает одинаковую ширину и высоту для выбранных объектов формы
Same Height (Одинаковая высота)	Устанавливает одинаковую высоту для выбранных объектов формы
Send to Back (Позади)	Направляет выбранный объект на самый нижний слой формы
Bring to Front (Поверх)	Направляет выбранный объект на самый верхний слой формы

#### 5.6.4 Настройка параметров формы

Форма, как и все располагаемые в ней объекты, имеет свойства, используя которые можно задать ее размер, координаты верхнего левого угла, стиль рамки обрамления, заголовок, цвет и т.д.

Настройка параметров формы осуществляется в окне свойств Properties (Свойства), для открытия которого установите курсор на свободную от объектов поверхность формы, нажмите правую кнопку мыши и выберите из контекстного меню команду Properties (Свойства).

### 5.6.5 Расположение формы

Новая форма по умолчанию располагается в верхнем левом углу основного окна Visual FoxPro. Для изменения ее положения можно использовать свойства Left (Левый) и top (Верхний), указывающие расстояние в пикселях от левого и верхнего края, соответственно, а также мышь. При использовании мыши для изменения положения формы установите курсор на заголовок формы, нажмите кнопку мыши и, удерживая ее, переместите форму в окне конструктора в место ее предполагаемого расположения.

Для изменения размера формы можно использовать свойства Height (Высота) и width (Ширина), определяющие высоту и ширину формы, а также мышь. При использовании мыши для изменения размера формы установите курсор в ее нижний правый угол. Когда он примет вид двунаправленной стрелки, нажмите кнопку мыши и, удерживая ее, измените размер формы. Установив необходимый размер, отпустите кнопку мыши.

#### Совет

Для размещения формы в центре главного окна Visual FoxPro необходимо в окне свойств установить для свойства AutoCenter (Авто центр) значение True (Истина).

### 5.6.6 Заголовок формы

Для задания текста заголовка формы, располагающегося в строке заголовка, предназначено свойство Caption (Надпись) окна свойств. Чтобы отредактировать заголовок, откройте окно Properties (Свойства), выделите свойство Caption (Надпись) и в поле ввода, ставшее активным, введите заголовок формы.

Если вы хотите, чтобы форма вообще не содержала заголовок, установителя свойства TitleBar (Строка заголовка) значение off (Выкл.)

### **5.6.7 Среда окружения формы**

При создании формы, предназначенной для редактирования или просмотра информации, содержащейся в таблицах, в конструкторе форм необходимо определить среду окружения, т.е. задать таблицы, используемые в форме, и установить связи между ними.

#### **Замечание**

При создании форм с помощью мастера и размещении объектов в форме с помощью построителя среда окружения создается Visual FoxPro без участия разработчика.

При определении среды окружения вы выполняете следующие действия:

- добавляете все таблицы, используемые в форме;
- устанавливаете для таблиц индексы;
- устанавливаете между таблицами отношения, необходимые для создания формы.

Вся эта информация, относящаяся к среде окружения, хранится в файле описания формы.

Для создания среды окружения формы предназначено диалоговое окно DataEnvironment (Среда окружения), открыть которое можно одним из следующих способов:

- выбрать команду Data Environment (Среда окружения) из меню View(Вид);
- нажать кнопку Data Environment (Среда окружения) на панели инструментов Form Designer (Конструктор форм);
- выбрать команду контекстного меню формы Data Environment (Среда окружения).

#### **Замечание**

При открытии окна среды окружения Data Environment (Среда окружения) в основное меню добавляется соответствующий пункт.

Для работы в окне Data Environment (Среда окружения) можно использовать команды из меню Data Environment (Среда окружения) или контекстное меню (рис. 23), позволяющие добавить в окружение таблицы, просмотреть их в режиме Browse (Обзор), открыть окно свойств окружения для задания различных параметров.

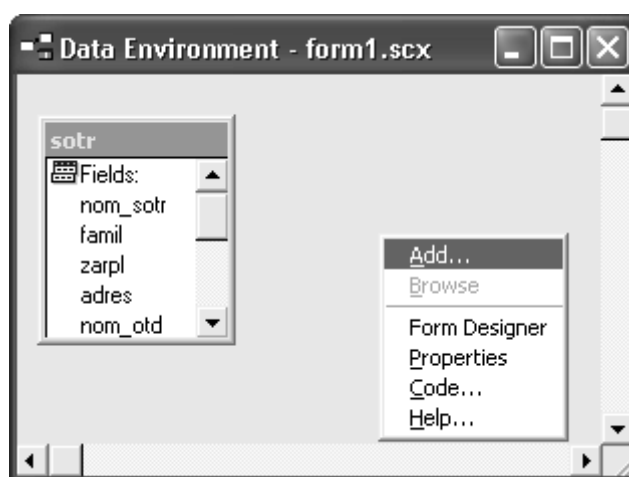


Рис. 23 — Диалоговое окно Data Environment с контекстным меню

Для добавления новой таблицы в среду окружения можно выполнить одно из следующих действий:

- выбрать команду контекстного меню Add (Добавить);
- выбрать команду Add (Добавить) из меню Data Environment (Среда окружения).

При этом открывается диалоговое окно Add Table or View (Добавить таблицу или представление данных), содержащее список таблиц открытой базы данных. Опция Views (Представления данных) области Select (Выбор) позволяет разместить в среде окружения созданные в базе данных представления данных.

После размещения таблиц в среде окружения формы вам необходимо упорядочить данные, находящиеся в таблицах. Для этого выполните следующие действия:

Выделите таблицу, в которой хотите упорядочить данные.

Откройте окно свойств таблицы. Для этого установите на нее курсор, нажмите правую кнопку мыши и выберите из контекстного меню команду Properties (Свойства).



Выделите свойство Order (Упорядочение).

В поле коррекции свойства нажмите кнопку раскрытия списка. Из списка индексов таблицы выберите тот, по которому хотите упорядочить данные в форме.

При размещении в форме связанных таблиц они переносятся в окружение формы вместе со связями, установленными в базе данных. Вам необходимо убедиться, что связи установлены должным образом. Для этого выполните следующие действия:

Выделите линию, соединяющую таблицы. При этом в окне Properties (Свойства) будут отображаться свойства, характеризующие установленную между таблицами связь.

Проверьте, какая из таблиц является родительской, а какая дочерней по отношению к ней. Для этого просмотрите свойства ChildAlias (Дочерняя таблица) и ParentAlias (Родительская таблица).

Посмотрите выражение, используемое для связи таблиц, воспользовавшись свойством RelationalExpr (Выражение отношения). Оно должно совпадать с выражением, используемым при создании индекса, по которому таблицы связаны.

На рис. 24 представлено диалоговое окно Data Environment (Среда окружения) с размещенными в нем таблицами Otdel и Sotr и открытым окном свойств для установленной между таблицами связи.

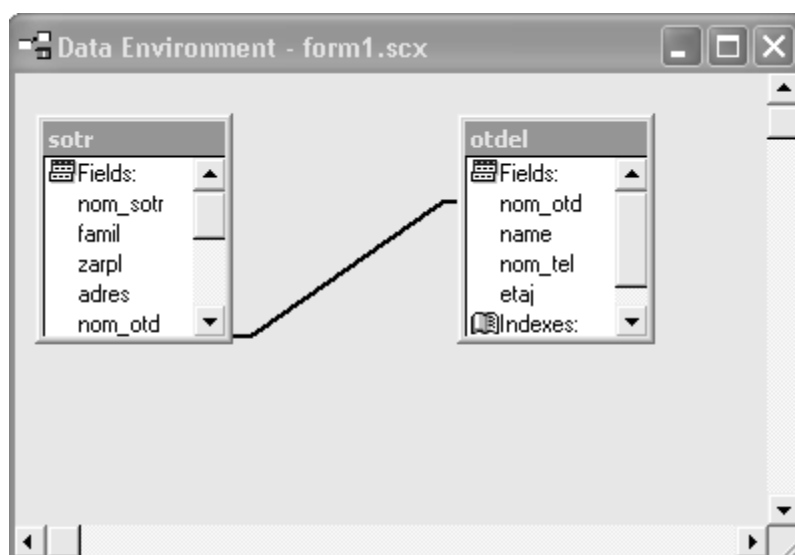


Рис. 24 — Диалоговое окно Data Environment с размещенными таблицами

После того как вы определили параметры формы, разместили в окружении используемые таблицы, можно приступать к размещению объектов в форме.

### 5.6.8 Размещение полей таблиц и надписей к ним

Для размещения в форме полей таблицы и надписей к ним в конструкторе форм можно использовать построитель формы.

Чтобы запустить построитель форм, выберите команду Builder (Построитель) контекстного меню формы или нажмите кнопку Form Builder (Построитель формы) на панели инструментов Form Designer (Конструктор форм). Откроется диалоговое окно Form Builder (Построитель формы), содержащее две вкладки (табл. 19).

Таблица 19 — Назначение вкладок окна Form Builder

Вкладка	Назначение
Field Selection (Выбор поля)	Выбор полей, которые будут размещены в форме (рис. 25)
Style (Стиль)	Задание стиля отображения объектов формы

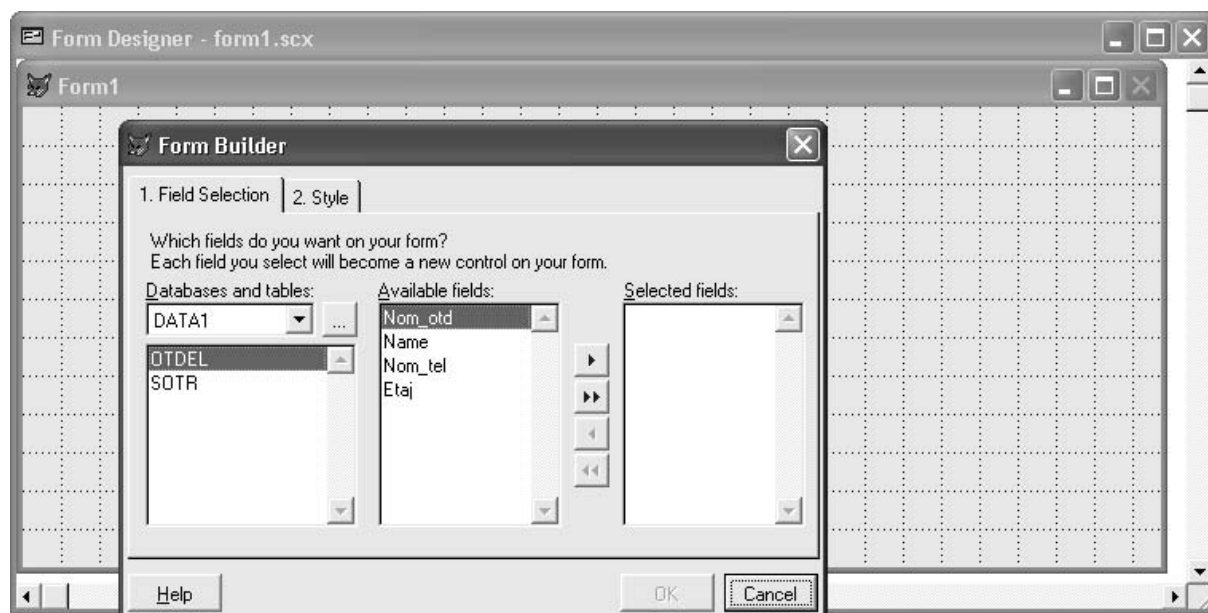


Рис. 25 — Вкладка для выбора полей, размещаемых в форме с помощью построителя

Используя вкладку Field Selection (Выбор поля) диалогового окна FormBuilder (Построитель формы), сформируйте список полей таблицы, размещаемых в форме. Для этого из верхнего списка области Databases and tables (Базы данных и таблицы) выберите необходимую базу данных, а из нижнего — таблицу, поля которой размещаете в форме. Затем из списка Availablefields (Имеющиеся поля) перенесите в Selected fields (Выбранные поля) поля, которые вы хотите разместить в создаваемой форме. Для переноса полей используйте кнопки, расположенные между списками.

### **Замечание**

При размещении в форме объектов с помощью построителя нет необходимости размещать таблицы в окне окружения, т.к. их разместит там сам построитель.

### **5.6.9 Текстовая информация**

Размещение в форме текста (заголовков, надписей к полям, поясняющей информации) осуществляется с помощью кнопки Label (Метка), находящейся на панели инструментов Form Controls (Элементы управления формы).

Для размещения текста выполните следующие действия:

Нажмите кнопку Label (Метка) на панели инструментов Form Controls (Элементы управления формы). Если данная панель отсутствует на экране, для ее отображения выберите в меню View (Вид) команду Form Controls Toolbar (Панель элементов управления формы).

Установите указатель мыши на место предполагаемого расположения текстового объекта и нажмите кнопку мыши. В форме появляется объект, в названии которого содержится слово Label1.

Для открытия окна свойств созданного объекта выделите его и выберите из контекстного меню команду Properties (Свойства). Откроется окно Properties (Свойства).

Фон текстового объекта определяется свойством Backstyle (Стиль фона). Если вы хотите, чтобы фон текста не отличался от фона формы, установите для свойства BackStyle (Стиль фона) значение Transparent (Прозрачный).

Текст задается свойством caption (Надпись). Выделите данное свойство, после чего в поле ввода свойства введите нужную текстовую информацию и нажмите клавишу <Enter>.

Задайте с помощью свойств `FontName` (Наименование шрифта) и `Fonsize` (Размер шрифта) вид и размер шрифта.

Используя свойство `ForeColor` (Цвет надписи), задайте цвет текстовой информации.

Скорректируйте размер объекта, чтобы в нем помещалась вся надпись. Для этого используйте маркеры выделения или установите значение `True` (Истина) в поле свойства `AutoSize` (Авто размер).

### 5.6.10 Поле ввода

Для отображения информации из таблиц в форме используются поля различных типов. Наиболее простым типом поля является поле ввода. Для размещения поля ввода в форме выполните следующие действия:

Нажмите кнопку `Text Box` (Поле ввода) на панели инструментов `Form Controls` (Элементы управления формы).

Щелкните в том месте формы, в котором вы предполагаете расположить поле ввода.

Откройте окно свойств для созданного объекта.

Чтобы связать созданное поле с полем таблицы, выберите на вкладке `Data` (Данные) свойство `ControlSource` (Источник данных). В поле ввода свойства воспользуйтесь кнопкой раскрытия списка и из списка всех полей открытой таблицы выберите поле, которое хотите добавить в форму (рис. 26).

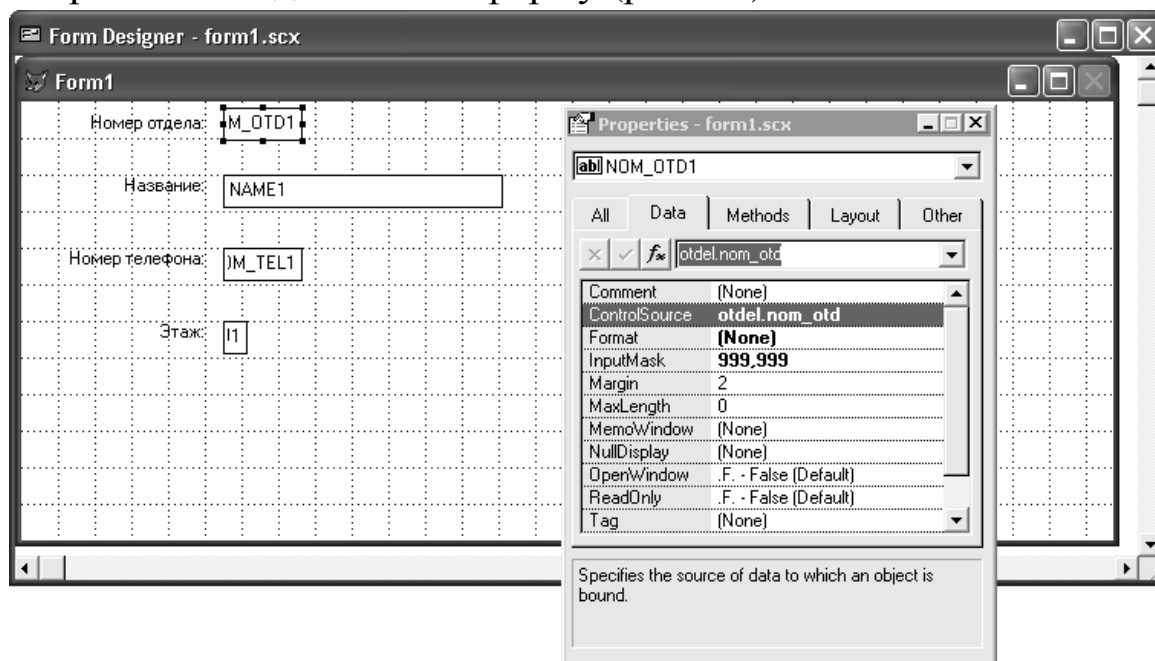


Рис. 26 — Связывание поля формы с полем таблицы

Используя свойство `Alignment` (Выравнивание), задайте вариант выравнивания информации в поле: по центру, по левому или правому краю поля.

Для задания стиля и цвета рамки поля используйте свойства `BorderStyle` (Стиль рамки) и `BorderColor` (Цвет рамки) соответственно.

С помощью свойства `DisabledBackColor` (Цвет фона неактивного поля) задайте цвет фона неактивного поля.

Применяя свойство `comment` (Описание), вы можете задать краткое описание назначения размещенного объекта. Это описание будет полезно при разработке приложения и его сопровождении.

Используя свойства `FontName` (Наименование Шрифта) и `FontSize` (Размер шрифта), задайте используемый при отображении информации шрифт и его размер.

С помощью свойства `ForeColor` (Цвет надписи) задайте цвет, которым будет отображаться информация в поле ввода.

Для отображения полей ввода в заданном формате используйте свойство `Format` (Формат).

Допустимые форматы и их назначения приведены в табл. 20.

Таблица 20 — Форматы данных

Код	Назначение
!	Преобразует буквы к верхнему регистру
\$	Выводит на экран текущий денежный символ
^	Отображает числа в экспоненциальном виде
A	Разрешает ввод только текстовых символов
D	Для редактирования данных типа даты использует текущий формат SET DATE (например, BRITISH, GERMAN и т.д.)
E	Редактирует данные, рассматривая их как европейские (BRITISH) даты
K	Выделяет все поле целиком, когда курсор перемещается в данное поле
L	Отображает ведущие нули при их выводе в поле ввода
M	Задаёт несколько predetermined значений на выбор
R	Содержит маску формата для поля ввода. Символы маски отображаются на экране, но не сохраняются в источнике данных
T	Удаляет начальные и конечные пробелы в форме

Окончание табл. 20

Код	Назначение
YS	Отображает дату в кратком формате, используя установки Windows
YL	Отображает дату в полном формате, используя установки Windows
Z	Отображает 0 как пробел, за исключением случая, когда фокус установлен на объекте

Свойство `inputMask` (Маска ввода) позволяет задать шаблон. Символы, которые могут быть использованы в шаблоне, представлены в табл. 21.

Таблица 21 — Коды шаблонов PICTURE

Код	Назначение
X	Допускает ввод любых символов
9	В случае символьных данных позволяет вводить только цифры. В случае числовых данных позволяет вводить цифру и знак
#	Позволяет вводить цифры, пробелы и знак
\$	Выводит на экран текущий денежный символ (определяемый при помощи команды <code>SET CURRENCY</code> ). По умолчанию этот символ помещается непосредственно до или после поля. Однако денежный символ и его позиция ( <code>SET CURRENCY</code> ), разделительный символ ( <code>SET SEPARATOR</code> ) и символ десятичного знака ( <code>SET POINT</code> ) могут быть переопределены
\$\$	Отображает плавающий денежный символ, положение которого определяется расположением цифр в поле ввода или счетчике
*	Перед числовыми значениями выводятся звездочки. Используется со знаком доллара \$
.	Задаёт позицию десятичной точки
,	Используется для отделения цифр, стоящих слева от десятичной точки

Если вы создаете поле, информация которого должна быть доступна только для чтения, необходимо установить значение свойства `ReadOnly` (Только чтение) равным `True` (Истина).

Используя свойство `specialEffect` (Специальный эффект), задайте стиль отображения поля из трех предложенных вариантов:

обычный, с эффектом объемности или принимающий объемный вид при перемещении курсора мыши над ним.

Для поля можно задать длинные поясняющие надписи, выводимые в строку состояния при установке на поле курсора мыши. Для этого используется свойство `statusBarText` (Текст строки состояния). На рис. 6.11 представлена форма с полем код клиента, при определении которого было использовано данное свойство.

Visual FoxPro позволяет с помощью свойства `ToolTipText` (Текст подсказки) создавать текст краткого пояснения, появляющийся ниже курсора, когда он установлен на поле. Чтобы текст заданного пояснения отображался, установите для свойства `ShowTips` (Показывать подсказки) формы значение `True` (Истина).

При добавлении новой записи в поле по умолчанию можно вводить наиболее часто встречающиеся значения. Для этого используется свойство `value` (Значение).

### 5.6.11 Поле редактирования

Поля `Edit Box` (Поле редактирования) очень удобны для редактирования символьных полей большого размера и Метополей. Чтобы разместить поле редактирования в форме, выполните следующие действия:

Нажмите кнопку `Edit Box` (Поле редактирования) на панели инструментов `Form Controls` (Элементы управления формы).

Щелкните в том месте формы, где вы предполагаете разместить поле редактирования.

Откройте окно свойств `Properties` (Свойства) для размещенного объекта.

При использовании поля данного типа для просмотра и редактирования полей большого размера, в его правой части можно расположить вертикальную полосу прокрутки, предназначенную для просмотра информации, не поместившейся в окне просмотра. Для этого необходимо в свойстве `scrollbars` (Полоса прокрутки) задать значение `vertical` (Вертикальная). При установке значения `None` (Нет) полоса прокрутки у поля будет отсутствовать.

Остальные свойства поля редактирования аналогичны свойствам поля ввода, которые описаны выше.

### 5.6.12 Кнопки управления

Как правило, в полях формы отображается информация об одной записи. Для просмотра всех записей таблицы необходимо добавить в форму средства перемещения по записям. В Visual FoxPro имеются удобные инструменты, предназначенные для этих целей, — кнопки.

Для размещения кнопок управления в форме можно использовать две кнопки панели инструментов Form Controls (Элементы управления формы) (табл. 22).

Таблица 22 — Кнопки панели инструментов Form Controls

Наименование	Назначение
Command Button (Кнопка)	Создание одиночной кнопки
Command Group (Группа кнопок)	Создание набора кнопок

### 5.6.13 Размещение одиночных кнопок

Для размещения в форме одной кнопки выполните следующие действия:

Нажмите кнопку Command Button (Кнопка) на панели инструментов Form Controls (Элементы управления формы) и щелкните мышью в месте предполагаемого размещения создаваемой кнопки.

Откройте окно свойств созданного объекта.

Кнопка может содержать текст или графическое изображение. При создании кнопки, содержащей текст, скорректируйте свойство Caption(Надпись), разместив в поле ввода значения текст, который будет отображаться на кнопке. Например, при создании кнопки для закрытия формы введите Выход.

Кнопка размещена в форме. Теперь необходимо, используя автоматически вызываемый при нажатии на кнопку метод объекта click (Нажатие), определить действия, выполняемые при нажатии на эту кнопку. Отобразите в окне свойств объекта список всех методов. Для этого в окне Properties (Свойства) выберите вкладку Methods (Методы).



Установите курсор на метод click (Нажатие) и щелкните мышью. На экране откроется окно процедур (рис. 27).

Введите команды, которые должны выполняться при нажатии на данную кнопку. Например, если вы создаете кнопку для выхода из формы, это могут быть следующие команды, использующие функцию MESSAGEBOX ():

```
* Запрос для выхода из формы
IF MESSAGEBOX("Выход из формы?", 4+32+256, "Выход")=6
_screen.ActiveForm.Release()
ELSE
_screen.ActiveForm.Refresh()
ENDIF
```

Закройте окно процедур. Кнопка создана.

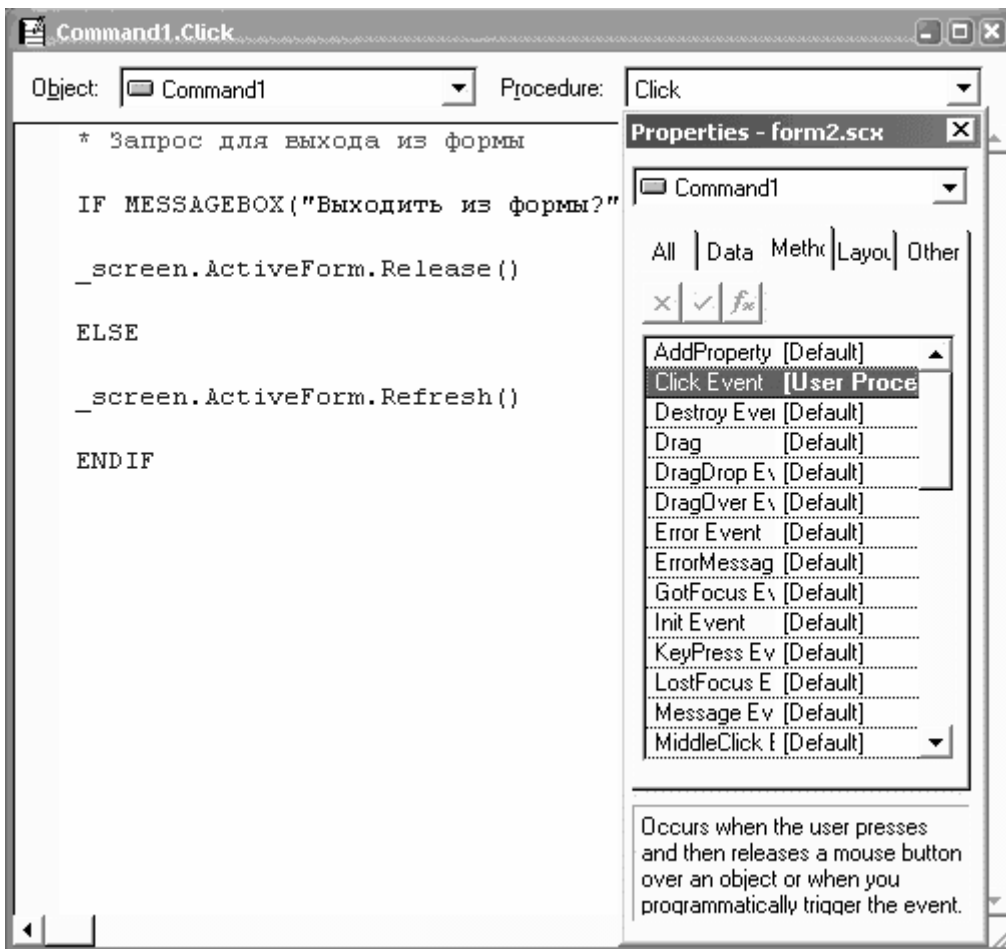


Рис. 27 — Окно процедур для метода Click

### 5.6.14 Редактирование составных объектов

Некоторые объекты Visual FoxPro, такие как `commandGroup` (Группа кнопок) и `optionGroup` (Переключатель), являются составными объектами, т.к. они содержат несколько объектов, имеющих свои собственные свойства. Для работы с составными объектами Visual FoxPro предоставляет в распоряжение разработчика контекстное меню, содержащее команду `Edit` (Правка), переводящую объект в режим редактирования и позволяющую управлять входящими в его состав простыми объектами: перемещать их внутри рамки, изменять размеры, цвет, настраивать другие свойства. В режиме редактирования вокруг составного объекта появляется заштрихованная рамка (рис. 28).

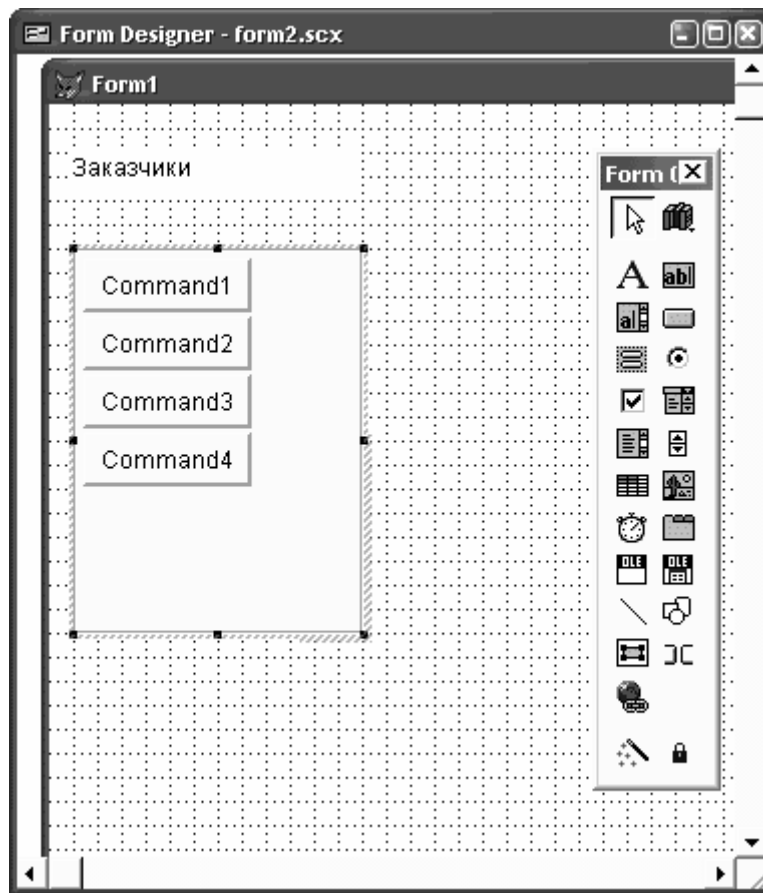


Рис. 28 — Объект `CommandGroup` в режиме редактирования

Для выхода из режима редактирования необходимо щелкнуть вне области составного объекта.

### 5.6.15 Создание группы кнопок

Для размещения в форме группы из нескольких кнопок можно использовать инструмент Command Group (Группа кнопок) на панели инструментов Form Controls (Элементы управления формы). Создаваемый при этом объект является составным и обладает свойством ButtonCount (Количество кнопок), определяющим количество входящих в его состав кнопок.

Воспользуемся данным инструментом для размещения в форме кнопок перемещения по записям таблицы и кнопки выхода из формы.

Для создания набора кнопок нажмите кнопку Command Group (Группа кнопок) на панели инструментов Form Controls (Элементы управления формы) и щелкните в месте их предполагаемого размещения в форме.

Откройте окно свойств для размещенного составного объекта.

Свойство ButtonCount (Количество кнопок) объекта определяет количество кнопок, размещаемых в объекте (по умолчанию 2). Скорректируйте его, задав необходимое количество кнопок, например 5.

Увеличьте с помощью мыши размеры рамки, окружающей созданный объект, чтобы в ней можно было расположить горизонтально все пять кнопок.

Переведите объект в режим редактирования. Для этого установите на него курсор, нажмите правую кнопку мыши и выберите из контекстного меню команду Edit (Редактировать).

Выделяя поочередно кнопки, переместите их, расположив горизонтально в одну линию.

Выйдите из режима редактирования, щелкнув вне области объекта CommandGroup (Группа кнопок).

Скорректируйте размер рамки, окружающей составной объект.

#### **Совет**

Для более точного задания размера рамки составного объекта выделите свойство AutoSize (Авто размер) и установите для него значение True (Истина).

Откройте окно свойств объекта типа CommandGroup (Группа кнопок). Нажмите кнопку раскрытия списка в верхней части

данного окна (рис. 29). Отметьте, что этот список содержит все объекты, размещенные в форме, а также все элементы, входящие в составной объект, под именами Command1, Command2, Command3, Command4, Command5. Выбирая поочередно элементы в этом списке, можно изменить свойства каждой кнопки.

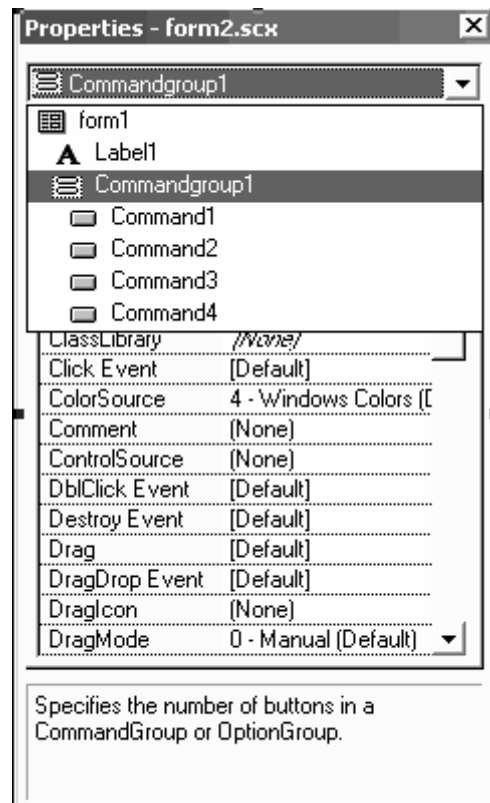


Рис. 29 — Окно свойств объекта типа CommandGroup

Используя свойство Caption (Надпись) каждого элемента составного объекта, задайте названия кнопок: Первая, Следующая, Предыдущая, Последняя и Выход.

Для задания цвета фона, на котором располагаются кнопки, используйте свойство Backcolor (Цвет фона) объекта CommandGroup (Группа кнопок). Если вы хотите, чтобы он совпадал с цветом фона формы, установите для свойства Backstyle (Стиль фона) значение Transparent (Прозрачный).

Теперь необходимо определить команды, которые будут выполняться при нажатии на данные кнопки. Для этого, открывая поочередно окно процедур метода click (Нажатие) каждого элемента, входящего в составной объект, введите следующие команды:

для кнопки Первая:

```
* Переходим на первую запись и обновляем информацию в форме
IF !BOF()
GO TOP
ENDIF
_screen.ActiveForm.Refresh()
```

для кнопки Следующая:

```
* Переходим на следующую запись и обновляем информацию в форме
IF !EOF()
SKIP
ENDIF
_screen.ActiveForm.Refresh()
```

для кнопки Предыдущая:

```
* Переходим на предыдущую запись и обновляем информацию в форме
IF !BOF ( )
SKIP - 1
ENDIF
_screen. ActiveForm. Refresh ( )
```

для кнопки Последняя:

```
* Переходим на последнюю запись и обновляем информацию в форме
IF !EOF()
GO BOTTOM
ENDIF
_screen.ActiveForm.Refresh( )
```

для кнопки Выход:

```
* Запрашиваем и выходим, если Да
IF MESSAGEBOX("Выходить из формы?",4+32+256, "Выход")=6
_screen.ActiveForm.Release()
ELSE
_screen.ActiveForm.Refresh()
ENDIF
```

После ввода команд закройте окна процедур.

Набор кнопок для перемещения по записям таблицы и выхода из формы создан. Запустите форму на выполнение по команде Run Form(Запустить форму) из меню Form (Форма). Для перемещения по записям таблицы и закрытия формы используйте кнопки, находящиеся в нижней части окна (рис. 30).

The screenshot shows a window titled 'SOTR' with a form titled 'СОТРУДНИКИ'. The form contains several input fields for employee information:

- Номер сотрудника: 1 024
- Фамилия: Иванов
- Зарплата: 10 500
- Адрес: Киевская 23 кв. 44
- Отдел: 105
- Паспорт: 3,456,345,345
- Дата рождения: 15.05.1970

At the bottom of the form, there is a group of five buttons: 'Первая', 'Следующая', 'Предыдущая', 'Последняя', and 'Выход'.

Рис. 30 — Форма с размещенной группой кнопок

### 5.6.16 Объект Grid

Объект Grid очень удобен для просмотра и редактирования данных, поскольку размещает данные в табличной форме. Поместите объект на форму и, выделив его, щелкните правой кнопкой мыши. Откроется контекстное меню, в котором выберите пункт Builder... в котором на вкладке Grid Items можно выбрать таблицу и ее поля для размещения на объект Grid. На вкладке Style можно выбрать стиль отображения объекта, а на вкладке Relationship выбрать первичный ключ родительской таблицы и внешний ключ дочерней таблицы для корректной навигации связанных таблиц (в дочерней таблице будут отображаться только порожденные строки от родительской таблицы). Пример показан на рис. 31

The screenshot shows a window titled 'Form1' containing two data grids. The left grid displays department data, and the right grid displays employee data.

Номер отдел	Название	Ном телефон	Этаж
101	Комплектац	456574	4
105	Бухгалтерия	234345	2
103	Снабжения	876543	1
109	Поставок	121212	5

Номер сотру	Фамилия	Зарплата	Адрес	Отдел	Паспорт
4424	Петров	11000	Нахимова 3 кв.	109	3456789
1407	Сидоров	9800	Климова 25 кв.	109	4567890

Рис. 31