

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Киямова Гульназ Ильдушевна  
Должность: документовед  
Дата подписания: 20.03.2024 11:10:50  
Уникальный программный ключ:  
10c4b36bd0c879864f7a9841653c86c88b767329

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Казанский национальный исследовательский технический университет  
им. А.Н. Туполева-КАИ»  
(КНИТУ-КАИ)  
Чистопольский филиал «Восток»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ**  
по дисциплине  
**БАЗЫ ДАННЫХ**

Индекс по учебному плану: **Б1.О.24**  
Направление подготовки: **38.03.05 Бизнес-информатика**  
Квалификация: **Бакалавр**  
Профиль подготовки: **Информационные технологии в бизнесе**  
Типы задач профессиональной деятельности: **проектный,  
аналитический**

Рекомендовано УМК ЧФ КНИТУ-КАИ

Чистополь  
2023 г.

# Работа в среде Microsoft ACCESS

## Лабораторный практикум

Microsoft Access (MA) – это система, предназначенная для создания реляционных баз данных. Позволяет пользователю разнести данные по множеству таблиц и производить хранение и поиск данных. СУБД МА позволяет управлять базой данных, создавая оптимальную структуру для хранения и поиска информации. МА обеспечивает множество автоматизированных процедур, а также встроенную справочную систему, которые помогут в создании и модификации каждого элемента вашей базы данных.

Фирма Microsoft поставляет с Access три базы данных с программами. Поэтому во время работы с Access можно постоянно использовать примеры в качестве источника идей. Файлы с примерами находятся в каталоге ...\\Samrapps. Учебной базой данных является NWIND.MDB. СУБД Access сохраняет базы данных с расширением MDB. При этом все основные элементы базы данных - таблицы, поля, формы, запросы и отчеты хранятся в едином файле базы данных.

### Лабораторная работа N 1

#### Создание базы данных в среде Microsoft Access

##### Порядок выполнения лабораторной работы

1. Создайте базу данных из основного меню (ОМ) с именем БД\_КАФ.  
**ОМ/Файл/Создать.**
2. Создайте таблицы Сотрудники (СОТР) и Кафедры (КАФ)  
СОТР (НС, ФИО, Должн, Каф, Зарпл)  
КАФ(КАФ, НС, Спец, Кол, Тлф)

Таблица Сотрудники

НС	ФИО	Должность	Кафедра	Зарплата
101	Петров К.	Профессор	САПР	7000
102	Морев С.	Ассистент	САПР	3000
103	Григорьев А.	Преподаватель	САПР	3500
104	Ильина Е.	Доцент	ВМ	5000
105	Шакиров Т.	Ассистент	ВМ	3000
106	Рахимова Г	Ассистент	ЭВМ	3000
107	Литвин Н.	Доцент	ЭВМ	5000
108	Никитин В.	Профессор	ЭВМ	8000

## Таблица Кафедры

Кафедра	НС	Специальность	Кол.сотр.	Телефон
САПР	101	221200	40	36-87-90
ВМ	104	10500	68	32-77-50
ЭВМ	111	220100	32	38-60-33

Таблицы могут быть созданы в различных режимах: **Режим таблицы, Конструктор, Мастер таблиц.** Для этого в окне базы данных выберите **Таблица /Создать.** В открывшемся окне **Новая таблица** выберите методы: для создания СОТР – **Мастер таблиц,** для КАФ – **Конструктор.** В режиме конструктора на пустом бланке укажите имена полей, типы данных и размеры.

### Структура таблицы Кафедра

Имя поля	Ключевое поле	Тип данных	Размер	Подпись
КАФ	Да	Текстовый	5	Шифр кафедры
НС	Нет	Текстовый	5	Зав.кафедрой
СПЕЦ	Нет	Текстовый	20	Специальность
КОЛ	Нет	Числовой	2	Колич.сотр-ов
ТЛФ	Нет	Текстовый	10	Телефон

### Структура таблицы - Сотрудники

Имя поля	Ключевое поле	Тип данных	Размер	Подпись
НС	Да	Текстовый	5	Ном.отдела
ФИО	Нет	Текстовый	20	Фамилия И
Должн	Нет	Текстовый	20	Должность
КАФ	Нет	Текстовый	5	Кафедра
Зарплата	Нет	Денежный		Зарплата

3. Введите данные. Данные можно вводить или в **Режиме таблиц** или в **Режиме форм.** Введите данные в режиме **Мастер таблиц.** Из списка **Образцы таблицы** выберите подходящую таблицу. Из списка **Образцы полей** выделите подходящие поля и поместите в список. Поля новой таблицы выберите с помощью кнопки >. После нажатия кнопки >> в новую таблицу помещаются все предлагаемые по умолчанию поля данных. Далее осуществляется подтверждение выбора полей. В следующем окне задайте имя таблицы и установите первичный ключ, щелкнув на ярлыке ключ в панели инструментов. Для изменения структуры таблицы необходимо нажать кнопку **Конструктор.** После окончания формирования таблицы следует вызвать команду **Файл / Заккрыть.**

4.Выполните действия в режиме таблиц. Для выхода в режим таблиц выберите имя таблицы и нажмите кнопку **Открыть.** При необходимости настройте таблицу. Откорректируйте ширину столбцов, размер шрифта и т.д.

Произведите добавление, удаление и корректировку данных. Произведите сортировку данных по различным полям таблицы.

5. Произведите фильтрацию данных. Фильтрация – это процесс отбора данных по тем или иным критериям. Например, необходимо отобразить в таблице **Сотрудники** преподавателей кафедры САПР. Для этого щелкните правой кнопкой мыши и из контекстного меню в поле **Фильтр** для. Наберите САПР, затем щелкните на кнопке **Фильтр по выделению**. Для восстановления таблицы нажмите кнопку **Удалить фильтр**. Кроме того, фильтр можно включить с помощью кнопки на панели инструментов или выбрать в основном меню команды **Записи /Фильтр**.

6. Выполните запросы. Запросы позволяют извлекать данные из базы данных. С помощью запросов можно извлекать данные не только из отдельных таблиц, но и из множества таблиц. Запросы могут быть на выборку, на создание новой таблицы, запрос на изменение, на обновление, на добавление, на удаление и т.д. В Access существуют два способа создания запросов: в режиме **Конструктор** и с помощью мастеров запросов. С помощью мастера запросов можно создать **Запрос на выборку** из одной или нескольких таблиц.

Произведите выборку по следующим запросам:

- выделите всех доцентов (доцентов и ассистентов);
- выделите всех сотрудников, оклад у которых меньше 5000 руб.;
- выделите доцентов, работающих на кафедре САПР.

Чтобы создать запрос в режиме **Конструктор**, выделите в диалоговом окне **Новый запрос** опцию **Конструктор**. Далее выделите таблицы, на которых должен основываться запрос. Нижняя часть окна **Конструктора** запроса называется **Бланк запроса по образцу (QBE)**. Чтобы включить поля из таблицы в **Бланк запроса**, выделите соответствующие поля и перетащите их в бланк или введите из меню.

7. Выполните запросы с параметрами. Эти запросы позволяет более гибко задавать условия. Откройте окно запроса и добавьте в него таблицу (СОТР). Создайте запрос, отбуксировав необходимые поля в бланк запроса QBE и задав условие выбора, например: <[ЗАРПЛ]. Далее из меню выберите команды **Запрос / Параметры**. Откроется окно диалога **Параметры запроса**. Введите параметр **ЗАРПЛ** и в качестве типа выберите **Денежный**. Сохраните. Далее можно многократно запускать данный **Запрос**, подставляя различные условия. Например, 500,800,1000.

8. Произведите вычисление данных. При запросах можно выполнять групповые операции. Для этого выберите в меню команд **Вид / Групповые операции**. На **Бланке Конструктора** запросов в строке **Групповые операции** щелкните правой кн. мыши и из всплывающего меню выберите необходимую функцию: **SUM, AVG, MAX, MIN** и т.д.

Используя групповые операции, вычислите: Общую сумму стоимости книг, приведенных в таблице; Стоимость книг по городам издательств; по авторам, Среднюю стоимость: в целом, по издательствам, авторам. Количество книг заказанных тем или иным издательством.

Произведите сортировку итоговых данных. Для этого укажите на **Бланке** в строке **Сортировка**: по возрастанию, по убыванию.

## 9. Связывание таблиц в Microsoft Access.

В базах данных для уменьшения избыточности, данные различных областей применения, хранятся в разных таблицах. Для получения сводной информации из разных таблиц их необходимо связать. Различают связи 1:1, 1:М. При втором типе связи каждой записи главной таблицы могут быть поставлены в соответствие одна или несколько записей подчиненной таблицы. Но при этом должны быть определены первичные ключи.

9.1.Создайте базу данных ПОСТАВКИ, состоящую из 3-х таблиц:

Поставщики (КП, Имя, Статус, Город), где КП – код поставщика  
Товары (КТ, Наимен., Вес, Цвет, Цена, Город), где КТ - код товара  
Поставки (КП, КТ, Колич.)

Значения таблиц представлены в приложении 2.

9.2. Установите связь между таблицами. Для этого в основном меню щелкните на пиктограмме **Схема данных**. Добавьте все три таблицы в окно **Схемы данных**. Далее буксируйте поле КП из таблицы **Поставщики** на одноименное поле таблицы **Поставки**. Затем поле КТ из таблицы **Товары** – на одноименное поле таблицы **Поставки**. Сохраните созданную схему.

9.3. Создайте запросы к связанным таблицам

Откройте окно **Конструктора запроса** и добавьте необходимые таблицы для каждого запроса.

9.3.1. Выборка из одной таблицы:

– всех поставщиков в алфавитном порядке; – поставщиков, находящихся в г.Самаре; – поставщиков, чей статус выше 20.

9.3.2. Выборка из двух таблиц – Поставщики и Поставки:

– выдайте сведения о поставщиках, которые поставляют товар с кодом 2;  
– выдайте сведения о поставщиках, которые поставляют товары в Колич>200.

Выборка из всех трех таблиц: Поставщики, Товары и Поставки:

– сведения о поставщиках, которые поставляют товар Винт;  
– поставщиков, поставляющих товары, в наименованиях которых есть буква «а»;  
– поставщиков, поставляющих товары ценой менее (или более) 100р.

## 10. Отчеты

Отчеты предназначены для вывода данных на экран или печать в соответствии с требованиями пользователя. Отчет содержит ту же

информацию, что таблица или запрос, но в нем имеются и дополнительные элементы, в частности, заголовок отчета, верхний и нижний колонтитулы. Простейший путь создания отчета состоит в использовании мастеров отчетов.

10.1. В окне базы данных выберите команду отчеты и нажмите кнопку **Создать**. В окне **Новый отчет** выберите таблицу **Поставщики** и нажмите кнопку **Мастер**. Выберите нужные поля и вид отчета, например, **Табличный**. **Мастер** создаст отчет, и Access откроет его в режиме предварительного просмотра.

10.2. Если отчет должен содержать фильтр или сортировку, откройте таблицу в режиме **Таблица** и убедитесь в установке **фильтра** или **сортировки**. Если это так, они будут автоматически включены в отчет.

10.3. Если для создания отчета желательно использовать существующий запрос, щелкните на вкладке **Запросы** и выберите нужный запрос.

10.4. Отчет с группировкой данных

Группировка позволяет сгруппировать отчет по дополнительным уровням. Например: Наличие товаров по городам.

Выберите таблицу **Товары**. Из нее выберите поля **Город**, **Наимен.**, **Цена**. В окне **Создание отчета** укажите в качестве уровня группировки **Город**. Для получения итоговых данных по полю **Цена** нажмите кнопку **Итоги**. Выберите подходящий стиль отчета. Выберите заголовок. Сохраните отчет. Просмотрите отчет.

10.5. Конструктор отчетов

Пользователь может улучшить созданный ранее отчет или создать новый в режиме конструктора отчетов. По умолчанию Access различает в отчете пять областей: **Заголовок**, **Верхний колонтитул**, **Область данных**, **Нижний колонтитул**, **Область примечаний**. **Заголовок** и **примечание** отчета печатаются один раз в начале и в конце отчета. **Верхний колонтитул** содержит заголовок столбцов таблицы. **Область данных** содержит записи запроса. Отчет в режиме конструктора формируется так же, как и форма.

Для получения отчета из связанных таблиц необходимо в начале создать соответствующий запрос, в котором объединены данные из разных таблиц. После сохранения запроса можно приступить к созданию отчета прямо из окна запроса. Для этого необходимо выбрать пиктограмму **Новый отчет**. Воспользуйтесь услугами мастера **Группировка данных** и вычисление итогов и выберите для отчета все поля исходного запроса.

## Конструирование форм в среде Microsoft Access

Формы являются удобным интерфейсом и помогают пользователям осуществлять ввод и редактирование данных. В режиме формы возможно более полно воспользоваться средствами, предоставляемые графической средой Windows (различные шрифты, цвет, графика и т.д.). С помощью формы можно отобразить всю информацию, содержащуюся в каждой записи, в то время как в режиме таблицы часть полей может выходить за границу экрана. Формы могут быть созданы в режиме **Автоформы**, **Мастера** или **Конструктора**.

### Порядок выполнения лабораторной работы

1. Запустите Microsoft Access. Откройте базу данных БД\_КАФ. Создайте базу данных в режиме **Автоформа**. Для этого на вкладке **Формы** окна базы данных выберите **Создать**. В диалоговом окне **Новая форма** выделите опцию **Автоформа: в столбец**. Укажите таблицу СОТР. Access сгенерирует автоформу. **Автоформы в столбец** удобны для быстрого ввода данных. Введите несколько записей. Сохраните форму. Произведите поиск записей с помощью пиктограммы **Найти** или путем кнопки прокрутки (внизу).

2. Аналогично создайте формы типа **Ленточная** и **Табличная**. Сравните удобство заполнения форм.

3. Создайте форму в режиме **Мастера форм**. Выберите для формы фон. Задайте имя формы. **Мастер форм** создает формы быстро и более гибко.

4. Используйте фильтры, чтобы отобразить в форме только определенные записи. Для определения фильтра необходимо открыть соответствующую форму, затем следует выбрать пиктограмму **Изменить фильтр**. Далее, указывая курсором на то или иное поле, задайте условия выборки. Например, поиск по фамилии, выборка записей по должностям, по кафедрам и т. д.

### Диаграммы

В Access можно анализировать записи базы данных с помощью диаграмм. Для этого используется **Мастер диаграмм**. Для запуска **Мастера диаграмм** выполните:

1. На вкладке **Формы окна** базы данных щелкните кнопкой **Создать**.
2. В окне **Новая форма** выделите опцию **Диаграмма**, а также таблицу или запрос. Выделите таблицу СОТР.
3. Выберите поля, значения которых необходимо откладывать по осям диаграмм, например, поля ФИО и ЗАРПЛ.
4. Выберите тип диаграммы, присвойте имя и запустите.

5. Постройте диаграммы:

- Зависимости: зарплаты по сотрудникам, зарплаты по должностям.
- Зависимости: количество сотрудников по кафедрам, суммарная зарплата по кафедрам и т.д.

### Конструктор форм

Сложные формы удобнее создавать в режиме **Конструктора форм**. Окно формы может содержать следующие элементы: подписи, поля, поля со списком, списки, выключатели, переключатели, флажки и кнопки. Кроме того, окно формы можно дополнить иллюстрацией и линиями различного типа.

1. На вкладке **Формы окна** базы данных щелкните кнопкой **Создать**.

2. В окне **Новая форма** выделите опцию **Конструктор**, а также таблицу или запрос на основании которых должны создаваться формы: например, СОТР. В **Конструкторе** можно отредактировать и существующие формы.

3. Задайте свойства формы. Для этого щелкните правой кнопкой мыши.

В появившемся контекстном меню выберите команду **Свойства**. Установите свойства формы: размеры, цвет и т.д.

4. Добавьте к форме элементы управления. К числу элементов управления относятся поля, кнопки, флажки и надписи, размещенные на форме. Вызов **Панели элементов** производится щелчком на соответствующей кнопке на панели инструментов. Расположение элементов формы можно менять, используя технику буксировки.

– Введите в форму поля таблицы СОТР, сделайте к ним соответствующие надписи. Дайте общий заголовок формы.

– Установите размеры и шрифты.

- Введите в форму поле **Системная дата**. Для этого создайте поле. В левой части укажите **Дата**. Далее выделите мышкой правую часть, и из контекстного меню выберите **Свойства**. В окне **Поле** выберите опцию **Данные** и наберите выражение: **=Date()**. Сохраните и запустите форму.

- В форме преобразуйте поле **Кафедра** в поле со списком. Для этого выберите команду из меню **Вид - Список полей**. Затем выберите пиктограмму на панели элементов **Поле со списком** и буксируйте в форму. В открывшемся окне выберите пункт «Поиск записи в форме на основе значения, которое содержит поле со списком». Задайте имена. С помощью пиктограммы **Рисунок** введите в форму рисунок. Сохраните и запустите форму .

– Расположите на форме элемент **Кнопка**. В левом списке **Мастера кнопок** выберите категорию действия: **Работа с формой**. В правом списке **Действия** выберите **Закрытие формы**. Введите необходимый текст или выберите подходящий рисунок. Введите имя кнопки и щелкните кнопкой **Готово**. Сохраните и запустите форму .

### Вызов подчиненных форм из главной формы

1. Создайте три формы для работы с таблицей СОТР:



Форма1 - с полем ФИО и кнопкой закрытия формы( создавать с помощью конструктора);

Форма2 и Форма3 - содержат все поля таблицы СОТР ( создавать с помощью мастера форм, выбрав внешний вид форм в виде одного столбца или выравненным , соответственно).

2. Откройте Форму1 в режиме конструктора и создайте кнопку, отбуксировав ее с панели инструментов. В открывшемся 1- м окне выберите **Категорию - Работа с формой** а **Действие - Открыть форму**. Нажмите кнопку **Далее**.

3. Во 2-м окне выберите имя формы: **Форма2**.

4. В 3-м окне выберите пункт **Открыть форму для отображенных записей**. Это означает, что при открытии подчиненной **Формы2** из основной **Формы1** будут отображаться только те записи **Формы2** , которые связаны с текущей записью формы1.

5. В 4-м окне установите соответствие: **Фамилия <> Фамилия**.

6. Задайте текст и имя кнопки, сохраните изменения в форме и запустите ее.

7. Откройте Форму1 в режиме конструктора , создайте еще одну кнопку и повторите пункты 2-6. Но при выполнении п.4 выберите **Открыть форму и показать все ее записи**.

### Рисунки и другие объекты в среде Microsoft Access

В формах и отчетах для наглядности можно использовать различные объекты: рисунки, графики, фотографии и т.д. Когда вы вставляете объект в форму или отчет, он размещается в элементе управления, который называется **рамкой объекта**. Microsoft Access обеспечивает два типа рамок объектов: связанные и несвязанные. **Связанная рамка объекта** дает возможность представить рисунок, диаграмму или любой другой объект, хранящийся в таблице базы данных. **Несвязанная рамка объекта** не привязана к таблице.

Для проведения экспериментов по вводу объектов в форму выполните следующие действия:

1. Создайте базу данных или откройте существующую: Сотрудники (НС, ФИО, Должность, Кафедра, Зарплата, Фотография). В существующую таблицу добавьте поле Фотография и укажите тип **Поле объекта OLE**.

2. Введите данные

НС	ФИО	Должность	Кафедра	Зарплата	Фотография
1	Борисов	Доцент	САПР	2000	
2	Ярина	Доцент	АСОИУ	2000	
3	Хитров	Доцент	ПМ	2000	
4	Замалеев	Ст.препода	АСОИУ	1500	
5	Володин	Ассистент	САПР	1000	

3. С помощью мастера создайте форму **Визитки**.

4. В форму введите заголовок **Визитная карточка**.

5. Вставьте в форму несвязанный рисунок. Для этого войдите в режим конструктора и на панели элементов выберите **квадрат (слева) Несвязанный рисунок**. Перетащите его в форму справа от заголовка. В появившемся окне **Вставка объекта** выберите подходящий рисунок и перенесите его в форму. При желании можете создать в **Paint** свой рисунок, для того чтобы вставить в форму.

6. Таблица Сотрудники содержит поле Фотография. Для того, чтобы они были внесены в форму **Визитки**, в область данных формы включите элемент управления. Для этого выполните следующие действия:

7. Откройте форму **Визитки** в режиме конструктора. Перетащите из панели элементов квадрат **Связанная рамка объекта**.

8. Откройте свойства рамки объекта и во вкладке **Данные** выберите поле Фотография.

9. Перейдите в режим формы. Для этого в основном меню выполните **Вид / Режим формы**. В форме войдите в квадрат и щелкните мышкой. Затем в основном меню выполните команды **Вставка / Объект**. В появившемся окне **Вставка объекта** выберите режим: **Создать из файла**. Войдите в режим **Обзор**. Найдите папку с фотографиями. Из этой папки выберите подходящую фотографию (файл с расширением **.jpg**) и вставьте в форму. Установите размер фотографии в соответствии с указанной рамкой. Затем вставьте вторую фотографию, затем третью и т.д.

10. Откорректируйте форму и запустите на выполнение.

### Лабораторная работа N 3

#### Конструирование подчиненных форм в среде Microsoft Access

Необходимо создать форму для закупки книг в магазине. Форма должна содержать информацию о заказе, такую как номер заказа, дата заказа, фамилия заказчика, и информацию о выбранных книгах. Выбор книги должен осуществляться по выпадающему списку книг.

1. Создайте базу данных БД\_КНИГИ.

2. Создайте три таблицы:

Книги (**Шифр\_книги**, Название, Автор, Цена)

Заказ (**Номер\_заказа**, Дата\_заказа, Фамилия\_заказчика)

Книги\_в\_заказе (**Номер\_заказа**, **Шифр\_книги**, Количество)

Ключи в таблицах выделены жирным. Типы полей текстовые (размеры подберите по собственному усмотрению, но не по 50 символов), кроме полей *Цена* – денежный тип, *Количество* – числовой, а *Дата\_заказа* – тип дата/время.

3. Создайте схему данных, приведенную на рис.1.

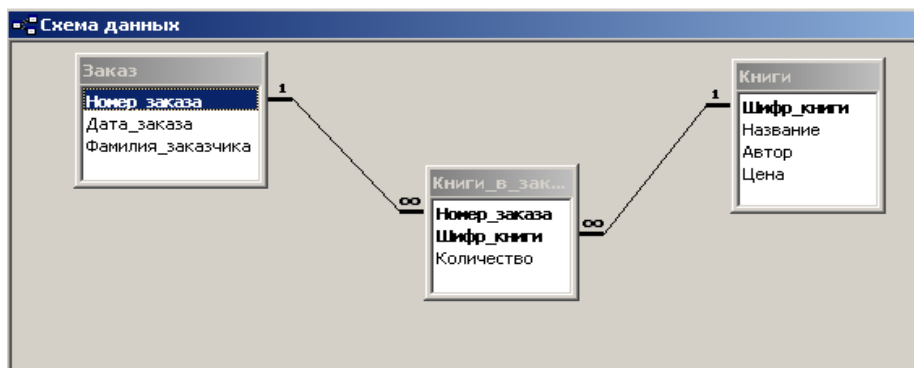


Рис.1. Схема данных базы данных БД\_КНИГИ.

Заполните только таблицу *Книги*, введите информацию о десятке книг.

4. Создайте форму для таблицы *Заказ* в режиме Мастер форм в один столбец. Сохраните форму *Заказ*.

Откройте форму *Заказ* в режиме конструктора. Увеличьте размер формы, как показано на рис.2.

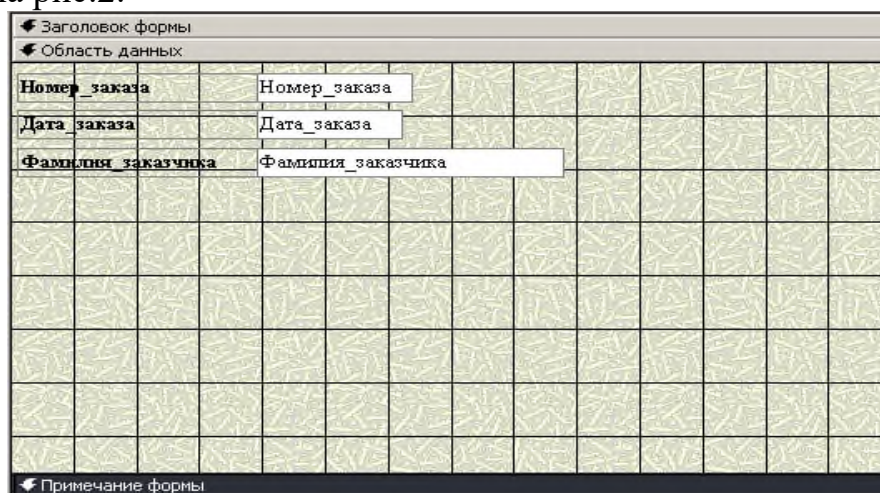


Рис.2. Форма Заказ в режиме конструктора

С помощью инструмента подчиненная форма добавьте подчиненную форму на свободное место формы *Заказ*:



Инструмент подчиненная форма

Запустится мастер подчиненных форм. В ходе работы мастера добавьте в подчиненную форму из таблицы *Книги* поля *Автор* и *Цена*, а из таблицы *Книги\_в\_заказе* поля *Шифр\_книги* и *Количество* (рис.3). Сохраните форму *Заказ*.

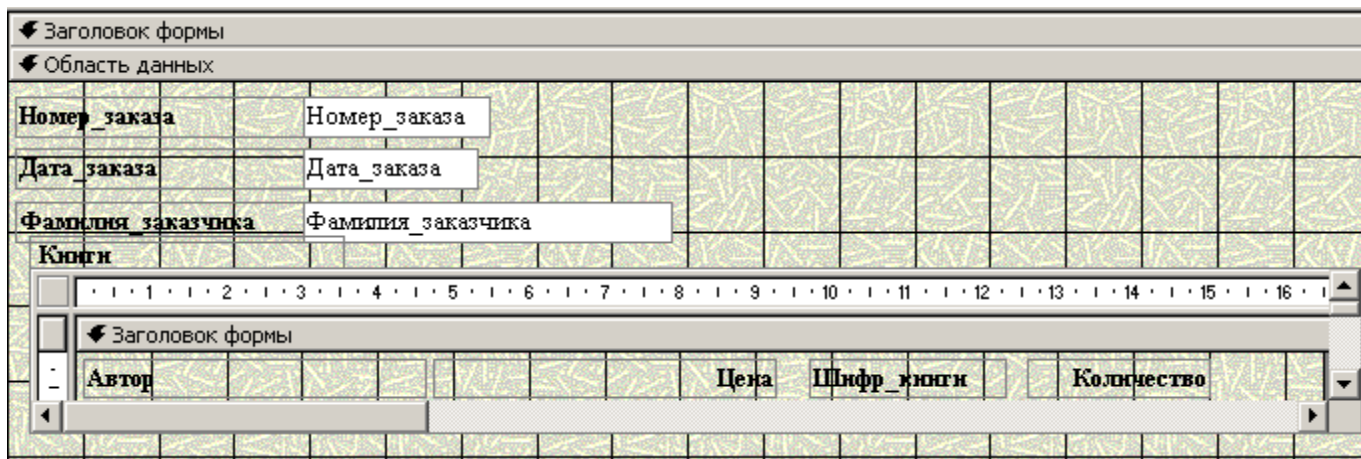
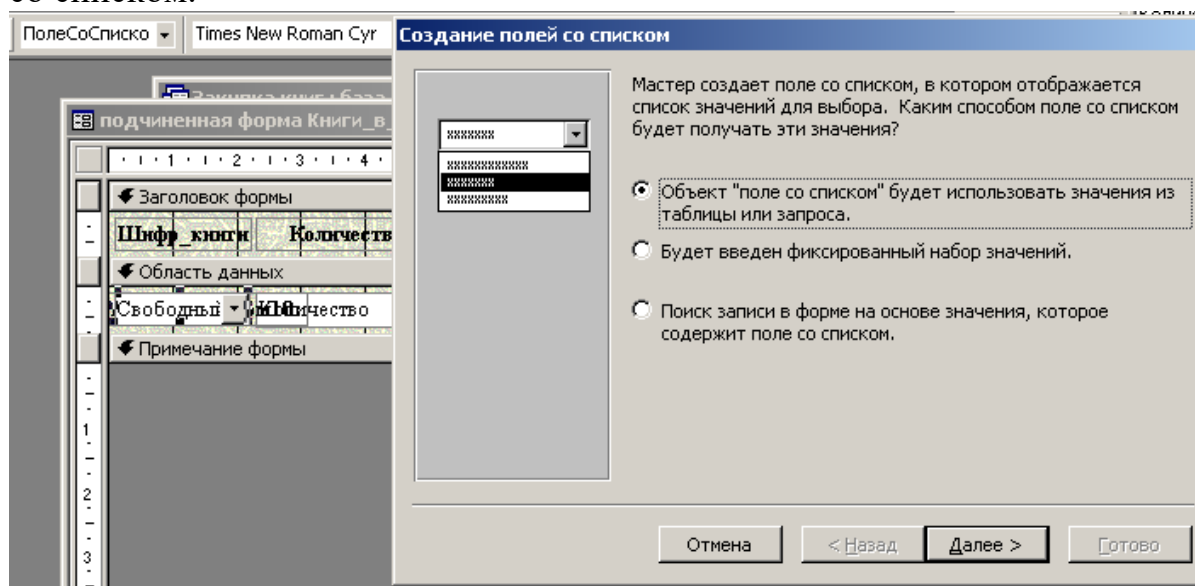


Рис.3. Форма Заказ с подчиненной формой Книги

Откройте *подчиненную форму Книги в заказе* в режиме конструктора. Измените ее следующим образом:

а) Удалите поле *Шифр книги* и на это место поставьте поле со списком с помощью соответствующего инструмента. Запустится мастер создания полей со списком.



б) В ходе работы мастера для поля со списком выберите *из таблицы Книги поле Название*, далее работайте по мастеру, в пункте *Сохранить в поле*, выберите поле *Шифр книги*. Задайте *подпись* для поля со списком: *Выбери книги*.

в) Вид подчиненной формы в режиме конструктора на рис.4. Сохраните форму.

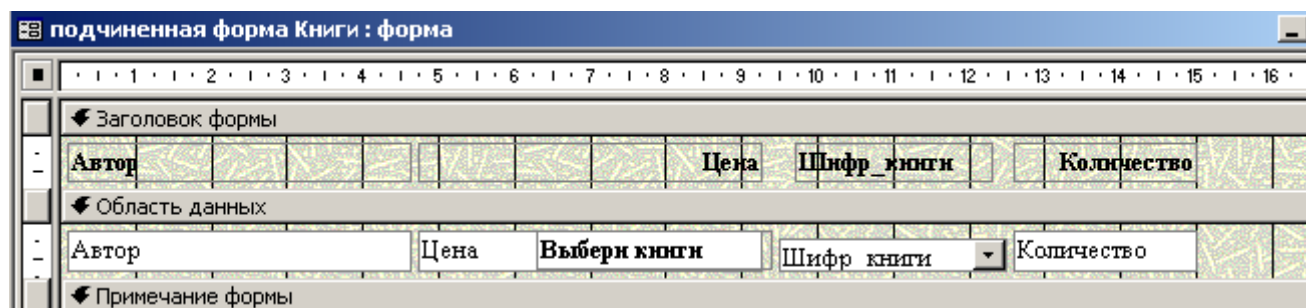


Рис.4. Подчиненная форма Книги с полем со списком.

5. Откройте форму Заказ. Заполните заказ номер 1 и выберите несколько книг из списка, укажите количество книг.

6. Измените подчиненную форму в режиме конструктора.

6.1. Добавьте вычисляемое поле Количество \* Цена, используя построитель выражений, задайте имя поля – Итого.

6.2. В части Примечание подчиненной формы добавьте поле для подсчета итоговой суммы заказа. Затем щелкните правой кнопкой мыши в новом поле и выберите опцию свойства, на вкладке Данные щелкните элемент «...» и запустится Построитель выражений. Постройте выражение  $=\text{sum}(\text{Количество} * \text{Цена})$ , где поля Количество и Цена выберите из загруженных форм. Запомните название (номер) этого поля, например Поле20. Измените свойства добавленных полей: на вкладке Макет выберите формат поля Денежный. Сохраните форму.

7. Откройте форму Заказ в режиме конструктора и добавьте поле для подсчета итоговой суммы: в свойствах этого поля с помощью построителя выражений (см. пункт 6.2.) укажите Поле20. Задайте имя поля – Всего к оплате. Также как в пункте 6.2. измените формат поля на формат Денежный. Сохраните форму.

8. Откройте форму Заказ и проведите закупки.

9. Откройте форму Заказ в режиме конструктора и добавьте кнопку для выполнения нового заказа. В процессе работы мастера создания кнопок укажите опции Обработка записей/ Добавить запись и задайте на кнопке текст **НОВЫЙ ЗАКАЗ**.

## Лабораторная работа №4

### МАКРОСЫ

Макрос – это небольшие программы, в которых содержатся макрокоманды, предназначенные для выполнения одного или нескольких действий, - например, открыть форму, распечатать отчет и т.д. Макрос может состоять из одной или нескольких макрокоманд, для каждой из которых можно задавать условие выполнения.

1. Создание макросов.

Процедура создания макросов состоит в выборе макрокоманд и задания для них параметров. Например, если создается макрос для открытия формы, то следует открыть окно конструктора макросов, выбрать из представляемого СУБД ACCESS списка, макрокоманду «Открыть форму» и ввести в качестве параметра имя открываемой формы (рис.12.1).

2. Вставка макрокоманды в существующий макрос.

Курсор устанавливается на строчку макроса перед которой необходимо вставить новую макрокоманду. нажимается кнопка добавления строки на панели инструментов

Затем в появившейся пустой строке вставляется новая макрокоманда.

### 3. Удаление макрокоманды.

Курсор устанавливается на удаляемую строку макроса и нажимается кнопка удаления строки на панели инструментов

Сохранение макроса.

Когда макрос уже составлен, его нужно сохранить под уникальным именем, для чего закрывается окно конструктора макросов и в появившееся окно вводится имя макроса.

Запуск макроса.

В окне макросов выделяется нужный макрос и запускается двойным щелчком мыши.

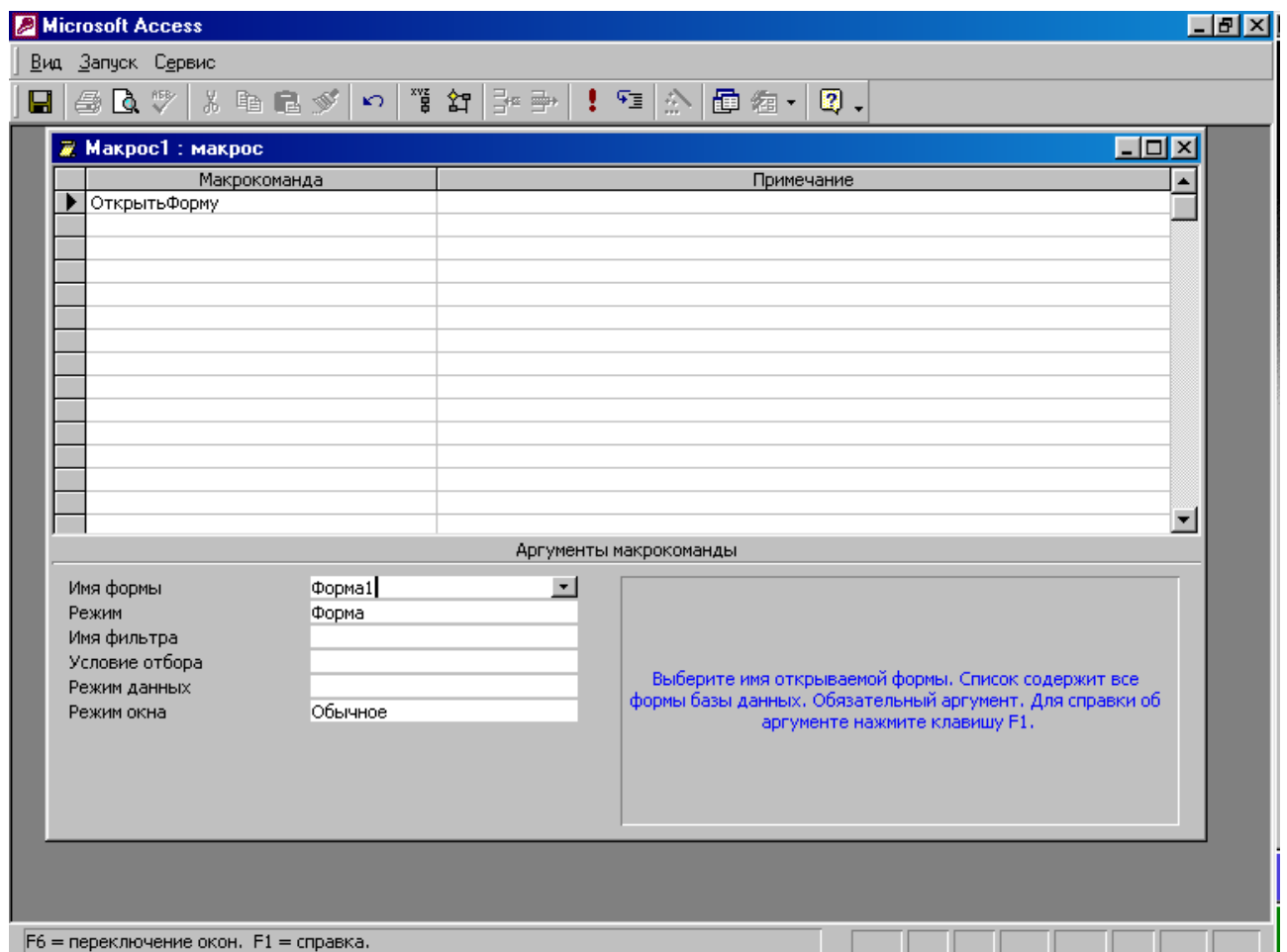


Рис. 12.1

### 6. Использование условий в макросах.

Для выполнения некоторых или каждой макрокоманды можно задать условия их выполнения. Если для макрокоманды существует условие, Access

выполнит ее лишь в случае когда это условие будет удовлетворено. В противном случае макрокоманда пропускается и выполняется переход к другой макрокоманде. Например, пусть создан макрос, состоящий из следующих макрокоманд :

- Открыть Форму 1
- Открыть Форму 2
- Открыть Форму 3
- Открыть Форму 6
- Открыть Отчет 1
- Открыть Отчет 2

Если не проставлены никакие условия для создания пауз между макрокомандами, они мгновенно сработают друг за другом и на экране останется только результат последнего запроса. Чтобы избежать такой ситуации, введем условия выполнения макрокоманд. Для этого в режиме конструктора макроса в верхнем меню выбирается пункт ВИД → УСЛОВИЯ. На экране появляется еще один столбец «Условия» (рис. 12.2) в который перед каждой макрокомандой вводится условие типа:

MsgBox(" Открыть Форму 1"; 4+32)=6

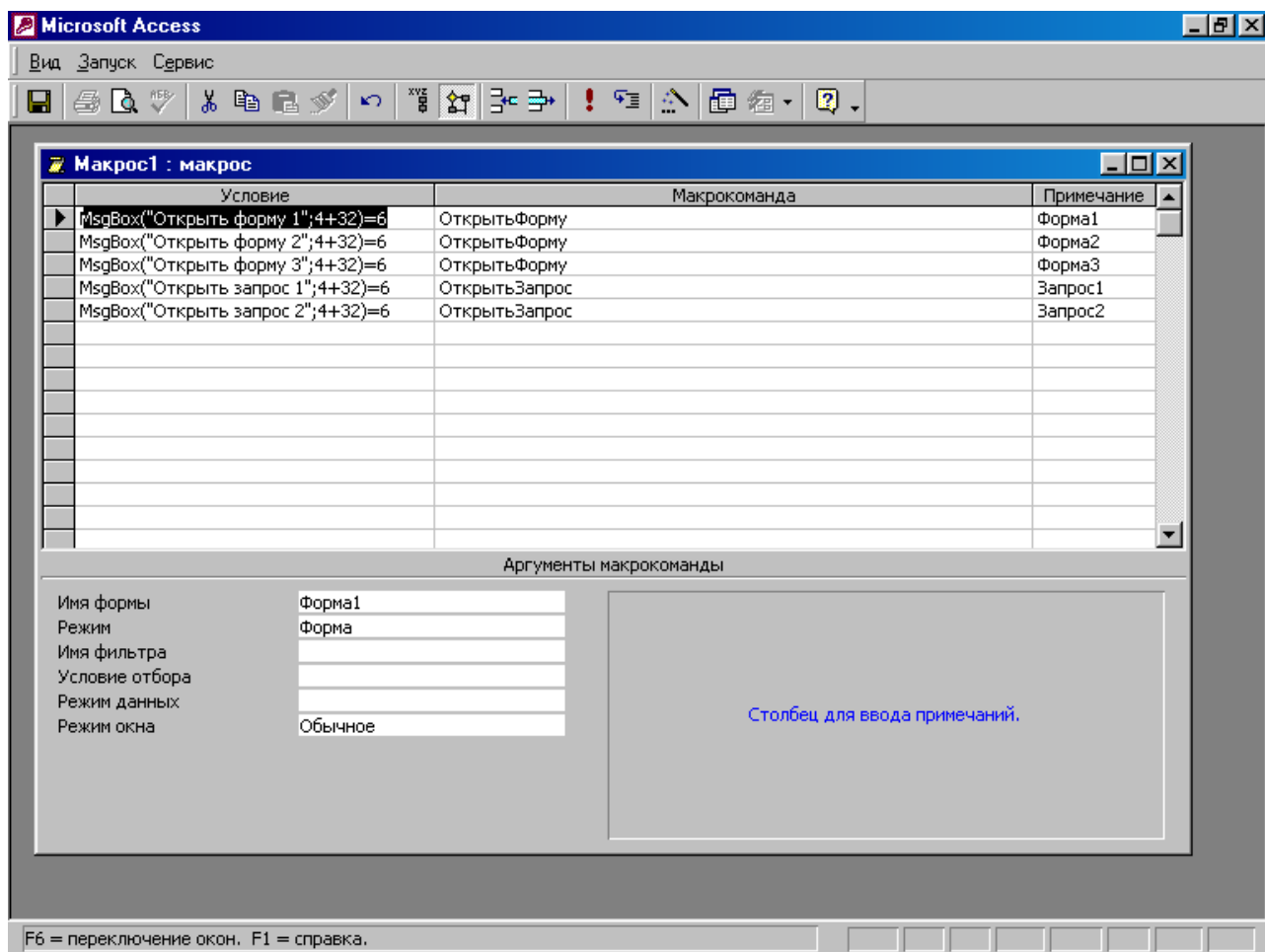


Рис.12.2

И теперь перед выполнением каждой макрокоманды система будет запрашивать, хотите вы ее выполнить или нет.

## 7. Связывание макроса с кнопкой.

Запуск макроса можно связать с нажатием кнопки в какой-либо форме.

Для этого надо открыть нужную форму (или создать ее заново), создать в ней кнопку выше упомянутым способом. Затем щелкнуть на ней правой клавишей мыши и выбрать пункт «Свойства». На следующем шаге раскрыть вкладку «События» и щелкнуть в строке «Нажатие кнопки». В раскрывающемся списке выбрать имя макроса. При необходимости можно изменить надпись на кнопке раскрыв вкладку «Макет» и изменив свойство «Надпись».

## З А Д А Н И Е

1. Создайте красочную форму с помощью конструктора.
2. Создайте три макроса: раскрытие форм, раскрытие запросов, раскрытие отчетов, созданных в предыдущих лабораторных работах.
3. Создайте три кнопки в новой форме и свяжите их нажатие с запуском ваших макросов.

## П р и л о ж е н и е 1

### Задачи проектирования баз данных.

1. Спроектировать базу данных для получения сведений о студентах данного вуза. Возможные атрибуты: ФИО студента, номер зачет.книжки, номер группы, наимен.специальности, размер стипендии.

2. Составить базу данных посетителей учреждения. Возможные атрибуты: ФИО посетителя, должность, организация, адрес, номер телефона.

3. Спроектировать базу данных получения сведений о поставщиках изделий. Возможные атрибуты: шифр поставщика, наименование поставщика, адрес, телефон, наименование изделия.

4. Спроектировать базу данных для получения сведений о профессиональной деятельности служащих. Возможные атрибуты: ФИО сотрудника, номер отдела, номер профессии, наименование профессии, дата начала работы по этой профессии.

5. Спроектировать базу данных о товарах, поставляемых на склад. Возможные атрибуты: номер товара, наименование товара, дата поставки, количество, цена единицы товара.

6. Спроектировать базу данных для хранения библиографических данных по литературе, поступающей на абонемент библиотеки. Возможные атрибуты: ФИО автора, название, наименование издательства, год издания, количество страниц, УДК, регистрационный номер.

7. Спроектировать базу данных маршрутов городского транспорта. Возможные атрибуты: номер маршрута, вид транспорта, начальный пункт, конечный пункт маршрута, протяженность маршрута, количество остановок, время в пути.



Приложение 2

База данных поставок.

Поставщики (S)

Таблица 1.

<b>NS</b>	<b>NameS</b>	<b>Status</b>	<b>Gorod</b>
S1	Саша	20	Самара
S2	Дима	10	Пермь
S3	Боря	30	Пермь
S4	Коля	20	Самара
S5	Аля	30	Уфа

Детали (P)

Таблица 2.

<b>NP</b>	<b>NameP</b>	<b>Color</b>	<b>Ves</b>	<b>Gorod</b>
P1	Гайка	Кр	12	Самара
P2	Болт	Зел	17	Пермь
P3	Винт	Гол	17	Казань
P4	Винт	Кр	14	Самара
P5	Кулачок	Гол	12	Пермь
P6	Звезда	Кр	19	Самара

Поставки (SP)

Таблица 3

<b>NS</b>	<b>NP</b>	<b>Kol</b>
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

## 1. Работа в среде FoxPro

FoxPro – реляционная система управления базами данных. FoxPro поддерживает файлы данных в формате dBASE III+ и на 100% совместима с FoxBase+, а также совместима по языку с dBASE IV. FoxPro работает более, чем в 8 раз быстрее dBASE IV и почти в 16 раз быстрее, чем dBASE III+. FoxPro имеет в высшей степени изящный интерфейс, уникальный среди продуктов DOS, обеспечивающий полную поддержку мыши.

### Лабораторная работа N 1

#### Создание, загрузка, редактирование и просмотр базы данных

Найдите файл Visual Foxpro1 и запустите на выполнение. На экране в верхней кромке высветится строка меню. Строка меню отображает имена (заголовки) всплывающих меню: SYSTEM, FILE, EDIT, DATABASE, RECORD, PROGRAM, WINDOW. Во всплывающих меню содержатся команды.

В FoxPro весьма развита оконная среда, включающая следующие окна: VIEW – основная панель отображения информации; BROWSE/CHANGE – окно просмотра, редактирования и ввода данных; COMMAND – окно команд; окно редактирования программы; окно трассировки и отладки.

Окно COMMAND является в FoxPro системным окном, в котором вводятся различные команды. Например, команды создания базы данных, создания программ, просмотра данных и т.д.

**Пример.** Создание базы данных для отношения:

КАДРЫ (Фамилия, Должность, Отдел, Оклад)

Для создания базы данных необходимо набрать команду CREATE KADRS и ввести следующие поля:

Имя базы данных – KADRS

имя поля	тип	длина
1. FIO	character	15
2. DOLGN	character	10
3. OTDL	character	5
4. OKL	numeric	7.0

Тип поля выбирается в момент, когда курсор находится в колонке TYPE, нажатием первой буквы указания типа (C, N, F, D, L, M) или клавиши SPACE/ENTER (Пробел/Ввод). Созданную структуру сохраните нажатием клавиш CTRL – W или CTRL – END. Модификация структуры базы данных может быть произведена по команде MODIFY STRUCTURE.

После создания структуры система может запросить: ввести данные? В случае согласия необходимо ответить YES, иначе NO. Если вы вновь вводите данные после запуска FOXPRO, то сначала откройте базу данных по команде USE KADRS, а затем введите команду для ввода данных - APPEND.

## Окно редактирования

При выдаче команд APPEND, INSERT, EDIT, CHANGE, BROWSE пользователь попадает в окно редактирования. В этом окне можно произвести изменение данных, а также дополнение и удаление записей. В окне редактирования для первых четырех команд все поля базы данных располагаются вертикально, (форма такого окна называется CHANGE-окном), а при использовании команды BROWSE (BROWSE-окно) поля располагаются горизонтально - колонками.

Перемещение внутри базы данных осуществляется с помощью клавиш перемещения курсора, мыши или набором клавиш:

- HOME/END – переход к началу/концу поля.
- TAB/SHIFT-TAB – переход к следующему/предыдущему полю.
- ENTER – переход к следующему полю.
- CTRL-T – пометка записей к удалению.
- CTRL-Y – удаление поля.
- CTRL-N – дополнение базы данных новой записью.
- CTRL-W/End – выход с сохранением измененных данных.
- CTRL-Home – вход в мемо-поле.
- ESC – выход без сохранения.

Для перемещения внутри базы данных по записям используются следующие команды:

- GO TOP – переход к самой первой записи файла.
- GO BOTTOM – переход к самой последней записи.
- GO n – переход к n-й записи.
- SKIP n – продвижение на n записей от текущей записи.

## Команды СУБД FOXPRO

Для просмотра записей базы данных существуют команды: DISPLAY, LIST, BROWSE, CHANGE. Полный формат команды:

ИМЯ КОМАНДЫ [<диапазон>][<поля>][FOR/WHILE<условие>]  
диапазоны: ALL, NEXT n, REST, RECORD n.

### **Пример.**

DISPLAY ALL fio,dolgn FOR otdl="САПР" WHILE okl<200000

Наберите команды DISPLAY, LIST с различными условиями.

Команда BROWSE является мощным и удобным средством доступа к данным в FoxPro. Команда BROWSE позволяет в полноэкранном режиме редактировать данные, вводить новые записи, вставлять и помечать к удалению.

Формат отображения полей в BROWSE-окне может настраиваться пользователем. Формат команды может включать, например, следующие опции:

## BROWSE

[FIELDS<поля>][FOR/WHEN<условия>][TITLE<выражение>]  
[LEDIT/REDIT] [WINDOW<окно>]

Опция FIELDS<список полей> может сопровождаться ключами: R,N,V,P,H,B,W. Формат имеет вид: FIELDS <поле>: ключ. R – разрешен только просмотр, N – видимый размер поля, V – контроль выхода из поля, P – задание формата отображения данных, H – указание собственного заголовка поля, B – указание границ чисел или дат, W – контроль входа в поле.

Опция TITLE задает заголовок BROWSE-окна. LEDIT/REDIT – опции, действующие в разделенном окне, так что в одном окне поля располагаются вертикально, а в другом горизонтально или наоборот; WINDOW<окно> – позволяет открыть BROWSE-окно внутри другого <окна>, которое предварительно определено командой DEFINE WINDOW.

Удаление данных в FOXPRO производится с помощью следующих команд: ERASE<файл> – удаление любого не открытого в данный момент файла; ZAP – удаление всех записей в активном файле базы данных; DELETE[<границы>][FOR/WHILE<условие>] – пометка к удалению записей в указанных границах; PACK – физическое удаление помеченных ранее записей и сжатие файла; RECALL – снятие пометок к удалению.

Просмотр и отбор записей может осуществляться с помощью команд DISPLAY и LIST с заданными условиями. Но последовательный поиск одной записи производится командой

LOCATE FOR<условие>[<границы>][WHILE<условие>].

**Пример:** LOCATE FOR fio='Иванов'. Для продолжения поиска с заданным условием служит команда CONTINUE.

Для ускорения поиска данных производится индексирование базы данных. Один файл базы данных может быть проиндексирован по нескольким полям. Формат команды обычного индексного файла имеет вид:

INDEX ON <выр> TO <IDX-файл>[FOR<условие>][COMPACT]

**Пример:**

USE kadr

INDEX ON fio TO kadr fio.idx COMPACT

LIST fio,dolgn

Опция COMPACT используется для создания компактного IDX-файла. Если индексный файл уже существует, то при открытии базы данных необходимо открыть и индексный файл: USE kadr INDEX kadr fio или произвести установку индексного файла командой SET INDEX TO kadr fio. Индексный файл не только упорядочивает базу данных для просмотра, но и ускоряет поиск в ней по ключу, заданному в индексе с помощью команды: SEEK<выражение>

**Пример:**

USE kadr INDEX kadr fio

SEEK 'Петров'

DISPLAY

## Режим командных файлов

Режим командных файлов служит для прямого программирования процессов обработки данных. При этом программы создаются с помощью внутреннего встроенного текстового редактора FOXPRO по команде MODIFY COMMAND <имя файла> или кратко MODI COMM. Созданный файл считается командным (программным) и ему присваивается расширение .PRG. Созданная в редакторе программа запоминается в указанном файле и в дальнейшем запускается на выполнение командой DO <имя файла>

Войдите в редактор по команде - MODI COMM prim1  
Наберите команду ?" Мы знакомимся с языком FOXPRO "  
Сохраните CTRL-W  
Запустите DO prim1

## Команды ввода/вывода

??/? – самая простая команда вывода, \ и \\ – вывод строки текста;  
TEXT –ENDTEXT – удобная для вывода значительных объемов текста;  
@ n,m SAY <выражение> [PICTURE<шаблон>]- форматированный вывод;  
где n=0-24 –номер строки; m=0-79 - номер колонки.  
INPUT, ACCEPT - операторы ввода;  
WAIT – оператор ввода с ожиданием;  
@ n,m GET <переменная> [PICTURE<шаблон>];  
READ – форматированный ввод.

Основные шаблоны опции PICTURE: А – допускается ввод только букв; N – допускается ввод только букв и цифр; X – допускается ввод любых символов; 9 – допускает ввод цифр; ! – преобразует строчные буквы в прописные.

### **Пример:**

```
@ 5,8 SAY "Введите оплату -" GET opl PICTURE '9999.99'  
READ
```

### **Пример:**

\* Программа вывода "шапки" таблицы

```
SET TALK OFF
```

```
CLEAR
```

```
@ 3,1 SAY "          СВЕДЕНИЯ О РАБОТНИКАХ          "
```

```
@ 4,1 SAY "-----"
```

```
@ 5,1 SAY " N   Фамилия И.О.  Должность  Отдел  Оклад  "
```

```
@ 6,1 SAY "-----"
```

Программа набирается с помощью встроенного редактора FoxPro. В редакторе действуют традиционные клавиши управления курсором. Для удаления и перемещения фрагментов текста используются следующие клавиши:

SHIFT + <стрелка вправо/влево> - выделение символа справа/слева.

SHIFT + <стрелка вверх/вниз> - выделение строки.

CTRL + A – выделение всего текста,  
CTRL + X – удаление выделенного фрагмента в буфер,  
CTRL + C – взятие фрагмента в буфер без удаления,  
CTRL + V – восстановление выделенного и удаленного фрагмента текста.

Выделение текста можно произвести и мышью. Для этого надо поставить ее маркер на начало выделяемого участка, нажать кнопку мыши и, не отпуская ее, переместить курсор в конец участка.

### Команды управления

Команды управления служат для изменения хода выполнения программ. К командам управления относятся следующие операторы:

Команда IF:

```
IF <условие>  
  <команды>  
ELSE  
  <команды>  
ENDIF
```

Команда DO CASE:

```
DO CASE  
  CASE <условие 1>  
    <команды>  
  CASE <условие 2>  
    <команды>  
  . . . .  
  [OTHERWISE  
    <команды>  
  ENDCASE
```

Цикл с условием:

```
DO WHILE<условие>  
  <команды>  
ENDDO
```

Цикл с параметром:

```
FOR i=n TO m [STEP 1]  
  <команды>  
ENDFOR
```

Цикл сканирования базы данных:

```
SCAN [<границы>][FOR/WHILE<условие>]  
  <команды>  
ENDSCAN
```

**Пример:** Фрагменты программ поиска записей с именами, начинающимися с буквы 'С'.

- |  |   |
|--|---|
| 1. USE kadrS<br>LOCATE FOR fio='C'<br>DO WHILE .NOT.EOF()<br>?fio,dolgn<br>CONTINUE<br>ENDDO           | 2. USE kadrS<br>SCAN FOR fio='C'<br>?fio,dolgn<br>ENDSCAN                                       |
| 3. USE kadrS<br>INDEX ON fio TO kdrfio<br>SEEK 'C'<br>DO WHILE fio='C'<br>?fio,dolgn<br>SKEEP<br>ENDDO | 4. USE kadrS<br>INDEX ON fio TO kdrfio<br>SEEK 'C'<br>SCAN FOR fio='C'<br>?fio,dolgn<br>ENDSCAN |

### Порядок выполнения лабораторной работы

1. Поработайте с системным интерфейсом. Для этого нажатием клавишу F10 и войдите в главное меню. Просмотрите пункты меню (SYSTEM, FILE, EDIT, DATABASE, RECORD, PROGRAM, WINDOW).

2. Войдите в редактор системы FoxPro. Для этого в окне COMMAND наберите команду MODI COMM prim1. В редакторе наберите текст, например:

ТЕХТ

Мы все учились понемногу  
Чему-нибудь и как-нибудь,  
Так воспитаньем, слава богу,  
У нас не мудрено блеснуть.

А.С.Пушкин

ENDТЕХТ

Отредактируйте текст. Для этого выделите слово (строчку) и удалите, восстановите. Поменяйте местами слова, строчки. Сохраните текст CTRL-W. Запустите на выполнение DO prim1

3. Составьте программу на выполнение вычислительной задачи

$$Y = \frac{(a + \text{sqrt}(b))}{a \cdot b + b} + \log(a),$$

где a, b – произвольные числа, корень и логарифм представлены в виде функций sqrt(x), log(x).

4. Создайте базу данных в соответствии с выданным заданием (см. приложение 1).

5. Произведите модификацию структуры базы данных. Введите произвольное дополнительное поле. Поменяйте имя, тип и размер. Удалите введенное поле.

6. Введите в базу данных порядка 7-10 записей.

7. Просмотрите записи по командам DISPLAY, LIST, BROWSE, CHANGE. Организуйте просмотр по различным условиям.

**Пример:** USE kadr  
LIST ALL fio, dolgn FOR otd="САПР"

С помощью команды LOCATE найдите отдельные записи.

8. Произведите индексацию базы данных по одному какому-либо полю или по двум полям. Просмотрите базу данных.

**Пример:** USE kadr  
INDEX ON otdl+fio TO idxfile COMPACT  
LIST

Произведите ускоренный поиск, например: SEEK "АСУ".

9. Произведите редактирование данных в режиме BROWSE. Поменяйте значения данных. Пометьте 2-3 записи на удаление. Вставьте и добавьте новые записи.

10. В режиме BROWSE-окна для своей базы данных выведите данные со своими собственными русифицированными заголовками.

**Пример:** Настроим выходной список базы данных KADRS таким образом, чтобы он имел общий заголовок и собственные заголовки колонок.

```
SET TALK OFF  
CLEAR  
USE kadr  
BROWSE TITLE " СВЕДЕНИЯ О СОТРУДНИКАХ ";  
FIELDS fio :H='Фамилия',;  
        DOLGN :H='Должность',;  
        OTDL :H='Отдел',;  
        OKL :H='Оклад'
```

Кроме того, с помощью параметра LEDIT/REDIT произведите конфигурирование окна редактирования в виде двух окон. Перемещение курсора из одного окна в другое можно осуществлять мышью или клавишами CTRL-N.

11. Создайте экранную форму для ввода информации, например:

ДАННЫЕ О СОТРУДНИКЕ	
Фамилия И.О.	Дашкова И.П.
Должность	инженер
Отдел	САПР
Оклад	1200

Программа, реализующая эту задачу, приведена ниже.

\* Бланк ввода данных - файл FKADR.PRG

```
SET TALK OFF  
USE kadr
```



```

CLEAR
otv='Y'
DO WHILE otv='Y'
  APPEND BLANK      && ввод пустой записи
  @ 4,11 TO 14,55 DOUBL && вычерчивание прямоугольника
  @ 5,24 SAY 'ДАННЫЕ О СОТРУДНИКЕ'
  @ 7,13 SAY 'Фамилия, инициалы'
  @ 7,35 GET fio PICTURE '@S(20)'
  @ 9,13 SAY 'Должность'
  @ 9,35 GET dolgn
  @ 11,13 SAY 'О т д е л'
  @ 11,35 GET otdl
  @ 13,13 SAY 'О к л а д'
  @ 13,35 GET okl PICTURE '999999руб.'
  READ
  @ 6,12 TO 6,54      && вычерчивание линий
  @ 8,12 TO 8,54
  @ 10,12 TO 10,54
  @ 12,12 TO 12,54
  @ 7,34 TO 13,34
  @ 16,11 SAY 'Еще будете вводить (Y/N)? ' GET otv PICT 'X'
  READ
ENDDO
CLEAR ALL

```

12. Просмотрите содержимое базы данных, например:

```

SET TALK OFF
CLEAR
USE kadr INDEX kdrfio
s=5
@ 3,15 SAY 'Сведения о сотрудниках'
SCAN
  @ s, 1 SAY k PICT '999'
  @ s, 5 SAY fio
  @ s,30 SAY dolgn
  @ s,40 SAY otdl
  @ s,50 SAY okl
  k=k+1
  s=s+1
ENDSCAN
WAIT ' '
CLAR ALL

```

13. Оформите и сдайте отчет по лабораторной работе

## ПРОГРАММИРОВАНИЕ МЕНЮ В СРЕДЕ FOXPRO

Программа "меню" представляет собой командный файл, предназначенный для объединения других командных файлов в некую систему. Она служит для автоматического вызова других командных файлов при просмотре и выборе пункта "меню" пользователем. Вызов производится командой DO <имя файла>.

В FoxPro имеются различные технологии построения меню:

1. FOX-меню – технология, используемая в версии СУБД FoxBase.
2. dBASE-меню – технология, позаимствованная из СУБД dBASEIV.
3. Клавишное меню.

### FOX-меню

В FOX-меню возможно создание трех различных форм меню:

- 1.1. LIGHTBAR-меню ( меню с подсветкой).
- 1.2. POPUP-меню ( вертикальное "всплывающее" меню).
- 1.3. PULLDOWN-меню (двухуровневое меню).

### LIGHTBAR-меню

Для создания меню используются команды:

@ X,Y PROMPT <выражение>

MENU TO <переменная>

#### **Пример:**

```
SET TALK OFF    && начало программы
USE kadr        && открытие базы данных
DO WHILE.T.    && начало бесконечного (.T.- TRUE) цикла
CLEAR          && очистка экрана
@ 1,5 SAY "Система учета сотрудников"
@ 2,4 TO 8,35 DOUBLE && изображение прямоугольника
@ 3,5 PROMPT " Ввод данных          "
@ 4,5 PROMPT " Просмотр данных      "
@ 6,5 PROMPT " Сумма по зарплате    "
@ 5,5 PROMPT " Редактирование данных "
@ 7,5 PROMPT " Выход                "
MENU TO num    && num-переменная числового типа, фиксирующая
DO CASE        && позицию меню
CASE num=1
DO PR1        && вызов программы ввода данных
CASE num=2
DO PR2        && вызов программы просмотра данных
CASE num=3
BROWSE        && редактирование данных
CASE num=4
SUM TO s      && SUM - оператор расчета суммы числового поля
```

```

CLEAR
?"СУММА ПО ЗАПЛАТЕ – ",s
WAIT " "
CASE num=5
EXIT      && выход из цикла
ENDCASE
ENDDO
CLEAR ALL  && очистка переменных и закрытие базы данных

```

### Вертикальное POPUP-меню

Данное меню значительно проще по сравнению с предыдущим. В данном случае операции описания меню и активации разделены. Создается командами:

```
@ Y,X MENU <массив>,<выр1>[,<выр2>][TITLE<вырC>]
```

<массив> - имя одномерного массива

<выр1> - число элементов меню (до 128),

<выр2> - число одновременно изображаемых на экране элементов меню(до 17),

TITLE <вырC> - заголовок меню.

Активация меню осуществляется командой

```
READ MENU TO <переменная>[SAVE]
```

#### **Пример:**

CLEAR	
DIMENSION A(3)	<u>Результат</u>
A(1)="Ввод данных"	
A(2)="Редактирование"	М Е Н Ю
A(3)="Удаление"	Ввод данных
@ 5,10 MENU A,3 TITLE "М Е Н Ю"	Редактирование
READ MENU TO n	Удаление

### Двухуровневое PULLDOWN-меню

Такое меню состоит из главного горизонтального меню (меню заголовков) и нескольких вложенных в него вспомогательных вертикальных меню. Главное меню определяется командой:

```
MENU BAR <массив>,<вырN>
```

Здесь:

<вырN> – общее число пунктов меню;

<массив> – двумерный символьный массив вида  $M(k,2)$ . Первый столбец массива  $M(i,1)$  – содержит собственно пункты меню, второй столбец

$M(i,2)$  – комментарии к позициям меню, располагаемым в строке по команде

```
SET MESSAGE TO <ном.строки>.
```

Вспомогательное POPUP-меню определяется командой:

```
MENU<вырN1>,<массив>,<вырN2>[,<вырN3>]
```

Здесь

<вырN1> –номер элемента главного меню;

<массив> – одномерный символьный массив, содержащий пункты меню;  
 <вырN2> – число пунктов меню, обычно равно размерности массива;  
 <вырN3> – число одновременно показываемых на экране пунктов меню;  
 Активирует меню команда

READ MENU BAR TO <перем1>,<перем2>[SAVE],

где переменные 1 и 2 фиксируют выбор пользователя в меню:

<перем1> – номер пункта главного меню;

<перем2> – номер пункта вспомогательного меню.

SAVE – предназначено для сохранения меню на экране.

**Пример:** Программа генерации двухуровневого меню.

SET TALK OFF

DIMENSION A(4,2)

A(1,1)='ФАКУЛЬТЕТЫ'

A(2,1)='КАФЕДРЫ '

A(3,1)='ГРУППЫ '

A(4,1)='КОНЕЦ '

STORE " " TO A(1,2), A(2,2), A(3,2), A(4,2)

DIMENSION B(2)

B(1)='Сведения '

B(2)='Деканат '

DIMENSION C(3)

C(1)='Преподаватели'

C(2)='Дисциплины '

C(3)='Лаборатории '

DIMENSION D(2)

D(1)='Сведения '

D(2)='Студенты '

\*Формирование образа меню

MENU BAR A,4 && Главное меню

MENU 1,B,2 && Вспомогательное меню к пункту факультеты

MENU 2,B,3 && Вспомогательное меню к пункту кафедры

MENU 3,C,2 && Вспомогательное меню к пункту группы

n=2 && Исходное положение курсора в глав.меню

m=1 && Исходное положение курсора во вспомог.меню

DO WHILE.T.

READ MENU BAR TO n,m

DO CASE

...

ENDCASE

ENDDO

В результате выполнения программы на экране появится меню ( рис.1).



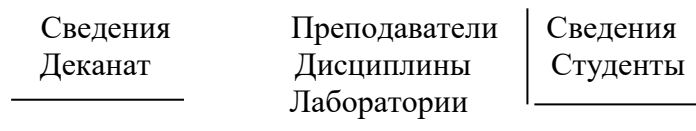


Рис.1. Двухуровневое меню

### DBASE – меню

dBASE-меню имеет два типа элементарных меню:

1. POPUP-меню, вертикальное или "всплывающее" меню;
2. BAR-меню, горизонтальное меню.

На основе данных элементарных меню можно строить иерархические меню любой сложности, связывая их так, как приведено на рис.2

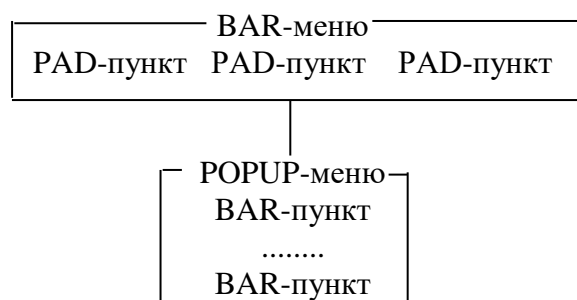


Рис.2. Иерархическое меню

### Вертикальное POPUP–меню

Описание меню производится командой

```
DEFINE POPUP<имя POPUP-меню>[FROM <Y1,X1>][TO <Y2,X2>]
[PROMPT FIELD<выр>/FILES][TITLE<выр>][MARGIN].
```

Здесь FROM <Y1,X1> TO <Y2,X2> – координаты углов прямоугольника.

Если параметры не заданы, ширина меню будет определяться по максимальному элементу, а высота по числу элементов меню. PROMPT FIELD<выр> – элементами меню будут записи из базы данных. <Выражение> может содержать одно или несколько полей, соединенных знаком '+'.  
PROMPT FILES [LIKE <маска>] – элементами будут названия файлов, возможно ограниченные маской.  
MARGIN – устанавливает пробелы справа и слева от приглашений меню.

#### **Пример 1:**

```
USE kadsr
```

```
DEFINE POPUP kdr FROM 5,8 TO 18,30 PROMPT FIELD kadsr.fio
```

#### **Пример 2:**

```
DEFINE POPUP text PROMPT FILES LIKE *.txt
```

В данном меню будут отображены все файлы с расширением .txt  
Описание элементов POPUP-меню производится командой

```
DEFINE BAR<вырN>OF<имя POPUP-меню>PROMPT<вырC>
```

Здесь `вырN` – номер `BAR`-пункта меню; `вырC` - содержание `BAR`-пункта на экране (приглашение).

Команда, определяющая реакцию меню, если выбор в меню сделан:

```
ON SELECTION POPUP<имя POPUP-меню>[<команда>].
```

Выбранная `<команда>` может быть любого типа, но обычно это команда `DO`, вызывающая процедуру. `<Команда>` выполняется при нажатии клавиши "Ввод". Для назначения пунктам `POPUP`-меню индивидуальных реакций используется команда

```
ON SELECTION BAR<вырN>OF<POPUP-меню>[<команда>]
```

Здесь `<команда>` производит обычно вызов процедуры.

Команда

```
ON BAR<вырN>OF<POPUP-меню1>
```

```
[ACTIVATE POPUP<POPUP-меню2>/ACTIVATE MENU<BAR-меню>] –
```

назначает вызов другого `POPUP/BAR`-меню на номер пункта из названного `<роруп-меню1>`. Использование команды без необязательного параметра отменяет закрепление.

Если за данным пунктом закреплен вызов другого меню, то справа от строки приглашения будет показана стрелка-треугольник. Чтобы для нее было выделено место в прямоугольнике меню, следует определять меню, из которого происходит вызов, с опцией `MARGIN`.

Активация меню производится командой

```
ACTIVATE POPUP<имя POPUP-меню>.
```

### Функции выбора из меню

Чтобы запомнить выбор в меню, сделанный пользователем, в процедуру можно передать параметры выбора с помощью следующих функций:

`BAR()` - функция возвращает номер выбранного `BAR`-пункта.

`POPUP()` - возвращает имя активного `POPUP`-меню.

`PROMPT()` - возвращает строку приглашение.

Отключение или деактивация меню может быть выполнена командами

`DEACTIVATE POPUP` (или `Esc`) – деактивирует меню и удаляет его с экрана.

`CLEAR POPUPS` – удаляет все деактивированные `POPUP`-меню с экрана и из памяти.

`RELEASE BAR <вырN>/ALL OF <POPUP-меню>` – команда удаляет перечисленные `BAR`-пункты из `POPUP`-меню.

`HIDE POPUP<список POPUP-меню>/ALL` – удаляет указанные или все (`ALL`) меню с экрана, но не из памяти, т.е. производит скрытие меню. Такое скрытое меню может быть восстановлено командами `ACTIVATE/SHOW POPUP`.

#### **Пример:**

```
CLEAR
```

```
DEFINE POPUP gmenu FROM 5,30
```

```
DEFINE BAR 1 OF gmenu PROMPT 'Дополнение'
```

```

DEFINE BAR 2 OF gmenu PROMPT 'Коррекция '
DEFINE BAR 1 OF gmenu PROMPT 'Удаление '
DEFINE BAR 1 OF gmenu PROMPT 'Выход '
ON SELECTION POPUP gmenu DO pmenu WITH BAR()
ACTIVATE POPUP gmenu

```

```

PROCEDURE pmenu
PARAMETR mbar
...
RETURN

```

### Горизонтальное BAR-меню

Описание меню производится командой  
**DEFINE MENU**<имя BAR-меню> [**AT LINE**<вырN>];  
 <вырN> определяет строку, на которой появится меню.

Описание PAD-пунктов меню:  
**DEFINE PAD**<имя PAD-элемента> **OF** <имя BAR-меню>  
**PROMPT**<вырC> [**AT**<Y,X>],  
 где <Y,X> - координаты пункта меню.

Реакция меню осуществляется командой  
**ON SELECTION MENU** <BAR-меню>[<команда>].

Индивидуальные назначения команд на каждый <PAD-пункт>  
 производится по команде **ON PAD** <PAD-пункт> **OF** <BAR-меню> [<команда>]  
 Обычно это <команды> **DO** вызова процедуры.

Команда **ON PAD** назначает на пункт меню из главного горизонтального  
 <BAR-меню> дополнительное меню следующего уровня  
 <POPUP-меню>/<BAR-меню>.

```

ON PAD<PAD-пункт>OF<BAR-меню>
[ACTIVATE POPUP<POPUP-меню>/ACTIVATE MENU<BAR-меню>]

```

Активация BAR-меню производится командой  
**ACTIVATE MENU**<имя BAR-меню>

**Пример:** Построим горизонтальное меню с именем MAINMENU и  
 содержанием: ФАКУЛЬТЕТЫ КАФЕДРЫ ГРУППЫ

```

CLEAR
DEFINE MENU mainmenu
DEFINE PAD fac OF mainmenu PROMPT 'ФАКУЛЬТЕТЫ' AT 1,10
DEFINE PAD kaf OF mainmenu PROMPT 'КАФЕДРЫ' AT 1,30
DEFINE PAD grup OF mainmenu PROMPT 'ГРУППЫ' AT 1,50
ON SELECTION PAD fac OF mainmenu DO P1
ON SELECTION PAD kaf OF mainmenu DO P2
ON SELECTION PAD grup OF mainmenu DO P3
ACTIVATE MENU mainmenu
*
PROCEDURE P1
...
RETURN
*
PROCEDURE P2

```

```
...  
RETURN  
*  
PROCEDURE P3  
...  
RETURN
```

### Двухуровневое PULLDOWN-меню

Создается с помощью уже указанных команд. Командами DEFINE MENU и DEFINE PAD описывается главное горизонтальное меню. Командой ON SELECTION PAD обычно описываются реакции тех пунктов меню, которые вызывают процедуры. Командой ON PAD с опцией ACTIVATE POPUP устанавливается связь горизонтального меню с вертикальными POPUP-меню, т.е. выход на второй уровень меню.

### Порядок выполнения лабораторной работы

1. Составьте произвольное FOX-меню с подсветкой.
2. Постройте двухуровневое меню по технологии FOX-меню.
3. Постройте вертикальное POPUP-меню по технологии dBASE-меню.
4. Постройте горизонтальное VAR-меню.
5. Постройте иерархическое dBASE-меню.
6. Оформите и сдайте отчет по лабораторной работе.

## **Лабораторная работа N 3**

### **Поиск информации в базах данных**

Для того чтобы начать работу с базой данных, ее нужно открыть. Это делает команда USE <база данных>

Одновременно можно открыть до десяти баз данных. При этом каждая база данных открывается в своей рабочей области. Рабочая область – это область оперативной памяти, в которой располагаются значения полей текущей записи базы данных, открытой в ней. Рабочие области идентифицируются номерами от 1 до 10 или буквами от A до J.

Одна из рабочих областей активная. Все операции производятся с базой данных, открытой в активной рабочей области.

Последовательность команд

```
SELECT A  
USE B1  
SELECT B  
USE B2
```

открывает в первой рабочей области базу B1, во второй - B2.



Активной считается последняя открытая база данных. Активная рабочая область устанавливается или переключается командой

```
SELECT <номер рабочей области>
```

или

```
SELECT <псевдоним>.
```

Псевдоним объявляется при открытии базы данных

```
USE <база> ALIAS <псевдоним>
```

Значения полей текущей записи активной базы данных доступны по своим именам. Значения полей текущей записи баз данных из других рабочих областей также доступны, но для этого нужно указать имя рабочей области.

```
<имя рабочей области> -> <поле>
```

или

```
<псевдоним> -> <поле>
```

**Пример:** "Аптечная сеть города" (Пример разработан Захаровой З.Х.).

Написать программу выдачи списка препаратов, имеющихся в заданной аптеке в количестве более 100 единиц.

Информационная база состоит из трех баз данных:

```
АРТЕКА.DBF
```

```
PREPARAT.DBF
```

```
NALICHIE.DBF
```

Структура базы данных АРТЕКА:

```
NOM1    символьное 4  – номер аптеки;
```

```
ADRESS  символьное 15 – адрес аптеки;
```

```
TELEF   символьное 8  – номер телефона.
```

Структура базы данных PREPARAT:

```
NOM2    символьное 4  – номер препарата;
```

```
NAME    символьное 15 – наименование препарата;
```

```
ZN      числовое 6.2 – цена препарата.
```

Структура базы данных NALICHIE:

```
NOM1    символьное 4  - номер аптеки;
```

```
NOM2    символьное 4  - номер препарата;
```

```
KOL     числовое 6   - количество.
```

\*\*\*\*\*Программа поиска информации\*\*\*\*\*

```
SET TALK OFF    * открываем базы данных
```

```
SELECT A
```

```
USE АРТЕКА
```

```
SELECT B
```

```
USE PREPARAT
```

```
SELECT C
```

```
USE NALICHIE
```

```
SELECT A    * запрашиваем номер аптеки
```

```
nomer=SPACE(4)
```

```
CLEAR
```

```
@ 10,10 SAY 'Введите номер аптеки ' GET nomer
```

```
READ
```

```
***** поиск заданной аптеки *****
```

```

LOCATE FOR nom1=nomer
IF FOUND()
CLEAR
@ 1,1 SAY 'Аптека номер'+nom1+' '+adress+'телефон '+telef
*поиск препаратов, имеющих в количестве больше 100 единиц
SELECT C
LOCATE FOR nom1=nomer.and.kol>100
*организация цикла по выборке всех таких препаратов
str=2
DO WHILE FOUND() * поиск наименования и цены препарата
SELECT B
LOCA FOR nom2=C->nom2
IF str>20
@ 2,0 CLEAR
str=2
ENDIF
str=str+1
IF FOUND()
@ str,1 SAY nom2
@ str,5 SAY name
@ str,25 SAY zn
@ str,35 SAY C->kol
ELSE
@ str,1 SAY 'препарат не найден'
ENDIF
SELECT C
*продолжение поиска
CONTINUE
ENDDO
WAIT 'Для продолжения работы нажмите любую клавишу'
ELSE
@ 1,1 SAY 'Аптека не найдена'
ENDIF
* закрываем все базы данных
CLOSE ALL
RETURN

```

### Задания к лабораторной работе

1. Пользуясь базами данных из рассмотренного примера выполните следующие задания:

1.1 Напишите программу выдачи перечня аптек, в которых имеется заданный препарат.

1.2. Напишите программу выдачи перечня аптек, в которых имеется заданный препарат в количестве больше 500 единиц.

1.3. Напишите программу выдачи перечня аптек, в которых заданный препарат имеется в недостаточном количестве (меньше 100 единиц).

1.4. Напишите программу выдачи списка препаратов, имеющих в заданной аптеке.

2. Спроектируйте базы данных для получения информации о наличии тех или иных товаров в магазинах города и выполните следующие задания:

2.1. Напишите программу выдачи перечня магазинов, в которых имеется заданный товар.

2.2. Напишите программу выдачи перечня магазинов, в которых имеется заданный товар в количестве больше 500 единиц.

2.3. Напишите программу выдачи перечня магазинов, в которых заданный товар имеется в недостаточном количестве (меньше 100 единиц).

2.4. Напишите программу выдачи списка товаров, имеющихся в заданном магазине

3. Спроектируйте систему учета штатов в медицинском учреждении, например в роддоме.

Пусть информационная база включает три файла:

OTDL.DBF - база данных отделений;

DOLG.DBF - база данных должностей;

SHTAT.DBF- база данных штатов.

Структура базы данных OTDL:

NOM символьное 2 – номер отделения;

NAME символьное 20 – наименование отделения.

Структура базы данных DOLG:

NOM символьное 2 – номер должности;

NAME символьное 20 – наименование должности.

Структура базы данных SHTAT:

NOM1 символьное 2 – номер отделения;

NOM2 символьное 2 – номер должности;

SHTAT числовое 5.2 – количество штатов.

3.1. Напишите программу выдачи сведений о штатах для заданного отделения.

3.2. Напишите программу выдачи наименований отделений, содержащих указанную должность.

3.3. Напишите программу получения итоговых сведений о штатах по всем отделениям.

3.4. Выделите отделения, заполненность штатами которых составляет больше 30, 20 единиц.

4. Программы и результаты выполнения программ представьте преподавателю.

## Лабораторная работа N 4

### Язык запросов SQL в FOXPRO

В язык FoxPro включен ряд команд из языка запросов SQL. Команды SQL могут непосредственно включаться в программы наряду с собственными командами FoxPro.

#### Создание баз данных

- CREATE DBF <DBF-файл>

( <имя поля>< тип> [( < размер>[ , < дробных разрядов>])[ , <имя поля> . .])

Команда создает новую базу данных <DBF-файл с указанным именем. Для каждого поля задаются его имя, тип (одной из букв C, N, D, M, F, L), длина и число десятичных разрядов. Длина и точность не задаются для типов дата (D), логический (L) и примечаний (M).

**Пример:** Создание базы KADR.DBF

```
CREATE DBF kadr (FAM C(15), TAB N(3),;  
DTR D, POL C(1), SEM C(1), DET N(1),;  
PODR C(15), SZAR N(9), PER M)
```

#### Дополнение базы данных

- INSERT INTO <файл БД> [( < поле1> [ , < поле2> ... ]  
VALUES ( <выр1> [ , < выр2> ... ] )

**Пример:**

```
INSERT INTO kadr (fam, tab, szar) VALUES ('Иванов', 350, 800000)
```

#### Формирование запросов

- SELECT < что выводится >  
FROM < откуда (источник)> INTO < куда (получатель)>  
WHERE < предикат>  
GROUP BY < колонки, по которым выполняется группирование>  
HAVING < условие группирования записей в одну строку>  
ORDER BY < в каком порядке выводить данные>

Каждый параметр команды SELECT имеет опции:

```
SELECT [DISTINCT] [ <псевдоним > . ] <выражение > [ AS <колонка > ]  
FROM <БД> [ <псевдоним > ] [ , <БД > [ <псевдоним > ] ... ]
```

Псевдоним используется в тех случаях, когда выборка делается из нескольких баз данных, где имена полей совпадают. Выражение – может быть полем записи из БД, константой, функцией. Если не устраивают имена,

формируемые по умолчанию, можно назначить свои, указав их после слова AS . Такое переименование имеет смысл, если выборка помещается во вновь создаваемую базу данных.

DISTINCT – исключает возможность вывода одинаковых строк в выборке.

INTO получатель

Получателем может быть

ARRAY <массив > – задается вновь создаваемый двумерный массив;

CURSOR <курсор > – задается имя Курсора;

DBF/TABLE <БД > - новая база данных с указанным именем.

Курсор - это временный набор данных, который может быть областью памяти или временным файлом. Кроме того, данные можно переслать в файл или на принтер.

TO FILE <файл > ADDITIVE / TO PRINTER – выборка посылается в текстовый файл или на принтер. Если используется слово ADDITIVE, то выборка будет добавлена в конец существующего файла без его перезаписи.

GROUP BY < колонка> – производит группирование выходных данных.

HAVING <предикат > – опция задает критерий отбора данных.

ORDER BY <колонка > [ASC/DESC] – опция задает упорядочение по заданной колонке. По умолчанию сортировка выполняется по возрастанию (ASC) или по убыванию (DESC).

### Функции

COUNT, SUM, AVG, MAX, MIN.

### Порядок выполнения лабораторной работы

1. Создайте обычным образом в FoxPro базу данных с таблицами

kadr(fam, tab, dtr, pol, sem, det, podr, szar, per);

brig1(nar, tab, vir);

brig2(nar, tab, vir).

Здесь fam – имя, tab – табельный номер, pol – пол, sem – семейное положение, det – наличие детей, podr – подразделение, szar – средняя зарплата, per – перемещения по службе, nar – номер наряда, vir – выработка в штуках.

2. Введите данные.

3. Сформируйте запросы на языке SQL.

### Примеры запросов

1. Выборка всех полей из базы данных kadr.dbf

SELECT \* FROM kadr

2. Вывод минимального, максимального и среднего значений поля SZAR (средняя зарплата).

SELECT MIN(szar), MAX(szar), AVG(szar) FROM kadr.

3. Вывод фамилий всех сотрудников, получающих от 4000 до 7000 рублей.

```
SELECT fam FROM kadr WHERE szar BETWEEN 4000 AND 7000
```

4. Выборка названий всех подразделений (поле PODR) предприятия.

```
SELECT DISTINCT podr FROM kadr
```

Опция DISTINCT предотвращает повторный вывод одних и тех же названий.

5. Выборка мужчин из всей базы данных

6. Список сотрудников с выработками по первой бригаде

```
SELECT s.fam, t.vir;
```

```
FROM kadr s, brig1 t WHERE s.tab=t.tab.
```

Здесь для сокращения записи введены псевдопеременные s и t.

7. Выборка фамилий всех родившихся в текущем месяце с указанием дня (числа) рождения, количества лет и премии по этому поводу – 50% от значения средней зарплаты.

```
SELECT fam, DAY(dtr), 'число', YEAR(DATE())-YEAR(dtr), 'лет',;  
'премия', 0.5*szar FROM kadr WHERE MONTH(dtr)=MONTH(DATE()).
```

8. Вывод полей FAM и TAB, отсортированных по полям POL (главное поле) FAM (подчиненное поле) в базу FAMTAB.DBF

```
SELECT fam, tab FROM kadr ORDER BY pol, fam INTO TABL famtab
```

Далее содержимое этой базы просмотрите по команде BROWSE.

9. Вывод для каждого работника выработок из баз BRIG1 и BRIG2, а также суммарной выработки работника в обеих бригадах.

```
SELECT kadr, tab, brig1.vir, brig2.vir, brig1.vir+brig2.vir;
```

```
FROM kadr, brig1, brig3 ;
```

```
WHERE kadr.tab=brig1.tab AND kadr.tab=brig2.tab.
```

10. Вывод фамилий всех работников, работавших ранее в конструкторском бюро (КБ). Поиск ведется в мемо – поле PER.

```
SELECT fam FROM kadr WHERE per LIKE "%КБ%".
```

11. Вывод табельных номеров и суммарной выработки каждого работника в бригаде номер 1. Вывод осуществляется в порядке увеличения табельных номеров

```
SELECT tab, SUM(vir) FROM brig1 GROUP BY tab ORDER BY tab
```

12. Вывод названий всех подразделений, количества сотрудников и значений суммарной заработной платы (фонда оплаты). Информация выводится только для подразделений, где количество сотрудников больше пяти.

```
SELECT podr, COUNT(*), SUM(szar);
```

```
FROM kadr GROUP BY podr HAVING COUNT(*)>5.
```

### Сформулируйте запросы самостоятельно

Пусть будет база данных поставок, состоящая из трех таблиц (приложение 2):

S (N\_P, FAM, SOST, Gorod) - таблица поставщиков;

P (N\_D, Name, Color, Ves, Gorod) - таблица деталей;

SP(N\_P, N\_D, Kol) - таблица поставок.

1. Произведите полные выборки из таблиц S, P, SP и просмотрите данные.
2. Выберите поставщиков, находящихся в городе Перми и имеющих состояние не более 20 тыс.руб.
3. Произведите выборку поставщиков в порядке убывания их состояния.
4. Выдайте сведения о поставщиках и деталях, находящихся в одинаковых городах.

### Подзапросы

5. Выдайте фамилии поставщиков, которые поставляют деталь P2. Используйте обычный запрос с подзапросом и коррелированный подзапрос.
6. Выдайте фамилии поставщиков, которые поставляют, по крайней мере, одну красную деталь.
7. Выдайте номера поставщиков, которые поставляют, по крайней мере, одну деталь, поставляемую поставщиком S2. Используйте псевдонимы (алиасы).
8. Выдайте сведения о поставщиках, которые поставляют деталь P3 с использованием квантора существования EXISTS.

### Запросы с использованием функций

9. Подсчитайте количество записей в SP.
10. Вычислите общий объем поставок для каждой детали.
11. Выдайте номера поставщиков со значением поля SOST меньшем чем текущее максимальное состояние.

Оформите и сдайте лабораторную работу.

## Лабораторная работа N 5

### Создание БД в среде Visual FoxPro

В Visual FoxPro имеется несколько уровней разработки приложений:

- Быстрый способ разработки с помощью многочисленных мастеров для создания форм, отчетов, почтовых этикеток и др.
- Конструкторы – для создания форм отчетов, меню, запросов для средних разработок.
- Для создания крупных проектов целесообразно создавать пользовательские классы.

Проект приложения. Служит для объединения отдельных элементов приложения. Выполняет функции:

- запоминает расположение элементов: базы данных, программ, экранных форм, отчетов;
- сохраняет объектный код в мемо-полях, что уменьшает количество отдельных файлов;
- осуществляет поиск и собирает файлы, на которые есть ссылки в проекте;
- все элементы приложения объединяет в один файл с расширением .APP или .EXE.

Создание нового проекта выполните в следующей последовательности:

1. Выполните команду File | New. В открывшемся окне диалога "New" выберите опцию Project. Нажмите кнопку New File.

2. В окне диалога "Create" задайте имя создаваемого проекта и сохраните, например, PRCT450501.

Создание базы данных осуществляется в интерактивном режиме с помощью конструктора базы данных. Конструктор базы данных позволяет создавать и модифицировать таблицы, входящие в базу данных, определять для них индексы, устанавливать отношения между таблицами.

База данных является частью проекта, поэтому ее целесообразно создавать в конструкторе проектов.

1. Выберите вкладку Data в верхней части окна конструктора проектов и нажмите кнопку New.

2. В окне диалога Create задайте имя создаваемой базы данных, например, BD450501.

3. Нажмите кнопку **Сохранить**, и на экране откроется окно конструктора базы данных, в котором вы можете создать новые таблицы.

Для создания новых таблиц и модификации существующих вы можете использовать: команды меню Database, команды контекстного меню, панель инструментов конструктора базы данных Database Designer.

При нажатии правой кнопки мыши на экране появляется контекстное меню конструктора базы данных, которое содержит наиболее часто используемые



команды.

Для создания новых таблиц и модификации, существующих можно использовать: команды меню Database, команды контекстного меню, панель инструментов конструктора БД.

### Создание таблиц

Существуют следующие способы создания таблиц:

- Database | New Table,
- Контекстное меню – New Table,
- На панели инструментов Database Designer – кнопка New Table.

В открывшемся окне диалога New Table можно воспользоваться мастером (Wizard) для создания таблиц или создать самостоятельно с помощью конструктора таблиц.

Создайте три таблицы:

Магазины(НМ, НаимМаг, Адрес, Телефон)

Товары(НТ, НаимТов, Вес, Цвет, Цена)

Наличие(НМ, НТ, Количество)

Для добавления нового поля в структуру таблицы или удаления используются кнопки Insert и Delete. Для модификации структуры таблицы в контекстном меню выберите команду Modify. Для связывания таблиц задайте в таблицах первичные ключи. Для этого в окне проекта установите курсор на модифицируемую таблицу и нажмите кнопку Modify. В окне диалога Table Designer выберите вкладку Indexes.

В открывшемся окне диалога в поле Name введите имя индекса, например, НМ для таблицы Магазины. Из списка возможных типов индекса в поле Type выберите Primary (создание уникального индекса). Перейдите на поле Expression и введите выражение для индекса НМ. То же самое сделайте и для таблицы Товары. Для таблицы Наличие введите в качестве индекса составное имя НМ+НТ.

Для установления отношений между таблицами необходимо открыть окно конструктора базы данных:

- выберите родительскую таблицу;
- установить курсор мыши на первичный ключ;
- нажмите клавишу мыши и, не отпуская ее, переместите курсор мыши на индекс дочерней таблицы. Возникнет линия связи.
- установив курсор на линии и дважды щелкнув, вызовите окно диалога Edit Relationship, в котором укажите тип отношения. Нажав ОК, сохраните связи;
- для удаления связи нажмите правую кнопку курсора и выберите из меню Remove.

В созданные таблицы введите данные.

## Порядок выполнения лабораторной работы

1. Найдите на диске C: в папке WORK папку своей группы и в ней создайте свой каталог. Все данные сохраняйте затем в своем личном каталоге.
2. Запустите Visual FoxPro, воспользовавшись кнопкой Пуск.
3. Познакомьтесь с системой команд и панелью инструментов главного окна.
4. Создайте проект.
5. Создайте базу данных.
6. Создайте таблицы Магазины, Товары, Наличие.
7. Используя команду Modify, проведите модификацию таблиц.
8. Установите отношения между таблицами.
9. Введите данные.

## Лабораторная работа № 6

### Конструирование форм в среде Visual FoxPro

В Visual FoxPro для создания форм используются следующие способы:

- Автоформат (AutoForm Wizard).
- Мастер форм (Form Wizard).
- Построитель (Builder).
- Конструктор форм (Form Designer).

#### Создание форм в конструкторе форм

Для создания форм откройте окно конструктора проектов. Нажмите ярлык Documents и выполните команды **Form | New | New Form**. Окно конструктора форм содержит панели инструментов: Color Palette (Цветовая палитра), Layout (Расположение объектов формы) и Form Controls (Объекты форм). Для отображения на экране панелей инструментов необходимо установить метки в соответствующих опциях меню View.

Процесс по созданию форм может включать в себя следующие процедуры:

- Настройка формы.
- Создание окружения.
- Размещение текста и полей.
- Создание управляющих кнопок.
- Размещение линий, прямоугольников и рисунков.
- Установка цвета объектов формы.

Любая форма в Visual FoxPro состоит из объектов формы, которые имеют характерные для них свойства. Процесс создания формы состоит в размещении объектов в форме и определении для них свойств, связанных с ними событий и

выполняемых действий. Для того чтобы получить доступ к свойствам и методам объекта, выделите нужный объект и вызовите окно **Properties**.

- **View | Properties**

В заголовке окна имеются кнопки: **V** – подтверждает ввод значения свойства, **X** – отказ от введенного значения.

### Последовательность действий по созданию формы

1. Откройте окно проекта.
2. Выберите вкладку **Documents**, перейдите в группу **Forms** и нажмите **New**.
3. В окне диалога **New Form** выберите опцию **New Form**. Откроется окно конструктора формы.
4. Откройте окно среды окружения (**Date Environment**), выполнив команду меню **View | Environment**.
5. Для добавления таблицы используйте пункт **Add** всплывающего меню или выполните команду меню **Data Environment | Add**.
6. В окне диалога **Add Table or View** выберите таблицу **Магазины**.
7. Определите свойства **View | Properties**.
8. В окне **Properties** скорректируйте свойство **Caption**, введя в текстовом поле заголовков формы: **Ввод списка магазинов**.
9. Скорректируйте свойство **BackColor**, выбрав в окне диалога **Color** цвет фона формы, отвечающий вашим требованиям.
10. Расположите форму в центре экрана **AutoCentr – True**. Измените мышкой размер формы. Установите свойства **FontName**, **FontSize** и выберите нужный шрифт.
11. Сохраните форму.

### Размещение текстовой информации

Под текстом понимается любая текстовая информация: заголовки, наименования полей и поясняющая информация.

1. Выберите инструмент **Label** на панели инструментов **Form Controls**, если панель отсутствует, то выполните **OM | View | Form Controls Toolbar**.
2. Мышью выделите рамку требуемого размера.
3. Откройте окно свойств **OM | View | Properties**.
4. Текстовую информацию задайте свойством **Caption**.
5. Определите вид и размер шрифта, а затем цвет, скорректировав свойства **FontName**, **FontSize** и **ForeColor**

### Размещение полей ввода

1. Выберите инструмент **Text Box** на панели **Form Controls**.
2. Установите указатель мыши на место, в котором предполагается разместить поле, и начертите рамку.
3. Откройте окно свойств.

4. Чтобы связать созданное поле с полем таблиц, выберите свойство ControlSource из вкладки Data. В поле ввода свойств воспользуйтесь кнопкой раскрытия списка и выберите поле, которое хотите добавить в форму.
5. Установите свойство Alignment для выравнивания информации (влево, вправо, по центру) и другие свойства по желанию.

### Создание кнопок управления

1. Для создания одиночной кнопки выберите инструмент Command Button на панели инструментов Form Controls и поместите кнопку в требуемом месте.
2. Откройте окно свойств созданного объекта Command и расположите текст или графическое изображение. Текст корректируется свойством Caption. Для расположения на кнопке графического изображения выберите свойства Picture и нажмите кнопку, расположенную правее поля ввода. В результате откроется окно диалога Open. В окне можно выбрать файл с графическим изображением и просмотреть его вид в области Picture. В качестве изображения может служить любой BMP-файл.
3. Для создания набора кнопок выберите инструмент Command Group на панели инструментов и поместите в форме.
4. Откройте окно свойств созданного объекта. Свойство Button Count объекта определяет количество кнопок. Задайте – 5.
5. Увеличьте мышью размер созданного объекта, чтобы разместить все 5 кнопок. Перейдите в режим редактирования (правая кнопка мыши + Edit)
6. Расположите кнопки горизонтально.
7. Откройте окно свойств объекта типа Command Group. Нажмите кнопку раскрытия списка в верхней части данного окна. Выбирая поочередно каждый элемент, можно изменить его свойства.
8. Скорректируйте для каждого элемента объекта свойство Caption, задав название кнопок: Первая, Следующая, Предыдущая, Последняя, Выход.
9. Далее определите команды, которые будут выполняться при нажатии кнопок. Для этого, для каждой кнопки, в окне процедур метода Click поместите следующие команды:

Для кн. “Первая”

```
if !bof()
```

```
  go top
```

```
endif
```

```
_Screen.ActiveForm.Refresh()
```

Для кн. “Следующая”

```
if !eof()
```

```
  skip
```

```
endif
```

```
_Screen.ActiveForm.Refresh()
```

Для кн. “Предыдущая”

```
if !bof()
```

```
  skip-1
```

```

endif
_Screen.ActiveForm.Refresh()
Для кн. “Последняя”
if !eof()
    go bottom
endif
_Screen.ActiveForm.Refresh()
Для кн. “Выход”
if messagebox(“Выходить из формы?”,4+32+256,”Выход”)=6
    _Screen.ActiveForm.Release()
else
    _Screen.ActiveForm.Refresh()
endif

```

9. Введите по отдельности еще две кнопки для ввода данных и удаления записей, используя команды: Append Blank и Delete.

### Размещение графических изображений

1. На панели инструментов Form Controls нажмите кнопку Image.
2. Мышью создайте рамку требуемого размера.
3. Откройте окно свойств созданного объекта **View | Properties**.
4. Установите курсор на свойство Picture и нажмите на кнопку правее поля ввода значения свойств, для вызова окна диалога Open. С помощью этого окна диалога можно не только выбрать файл, но и просмотреть содержащееся в нем графическое изображение.
5. Выберите необходимый файл с расширением .BMP, просмотрите его содержимое и нажмите кнопку ОК.
6. Для изменения размера картинки необходимо задаться свойством Stretch.
7. С помощью кнопки Label разместите все необходимые надписи.

### Порядок выполнения лабораторной работы

Создайте экранные формы для таблиц: Магазины, Товары, Наличие.

Требования:

- Формы должны позволять просматривать записи, листать вперед и назад, добавлять и удалять записи.
- Формы должны иметь улучшенные эстетические характеристики, для чего используются соответствующие цветовые эффекты.
- Формы должны быть дополнены графическими объектами.

## Лабораторная работа N 7

### Создание отчетов и запросов

В Visual FoxPro для создания отчетов используется:

- Мастер отчетов (Report Wizard), позволяющий достаточно быстро создать отчет.
- Стандартный отчет (Quick Report).
- Конструктор отчета.

#### Создание отчета с помощью конструктора отчета

Выполните команду меню **File | New**. Установите опцию Report и нажмите кнопку New File. Окно конструктора отчетов (ОКО) содержит три полосы. Каждая полоса может содержать текст, табличные поля, вычисляемые поля, линии, прямоугольники и рисунки. Размеры полос устанавливаются мышкой.

Первоначальное размещение полей данных можно облегчить, используя стандартный отчет. Для этого:

1. Находясь в ОКО, выполните команду меню **Report | Quick Report**.

2. В открывшемся окне диалога Open содержится список всех таблиц открытых в проекте базы данных. Выберите таблицу, для которой создается стандартный отчет.

3. Откроется окно диалога Quick Report. Выберите вариант расположения полей в отчете: в столбец или в строку. Для выбора полей отчета нажмите кнопку Fields и ОК.

4. Откорректируйте наименования полей и установите соответствующие шрифты. Для этого воспользуйтесь командами **Edit | Select All** и затем **Format | Font**.

#### Установка среды окружения

Для установки среды окружения выполните команду **View | Data Environment**. В открывшемся окне разместите все таблицы, используемые в отчете. Для добавления таблицы в окно используется команда Add. Все связи между таблицами сохраняются или изменяются. После закрытия данного окна среда окружения сохраняется.

#### Создание стандартного отчета

1. Откройте свой ранее созданный проект.
2. Перейдите на вкладку Документ, выберите Report и нажмите кнопку New.
3. Выберите New Report.
4. Выполните команду меню **View | Environment** и откройте окно диалога Data Environment. Для добавления таблицы выполните Add.

5. В открывшемся окне диалога Add Table or View выберите таблицу Магазины. Закройте окно Data Environment.
6. Для создания стандартного отчета выполните команду **Report | Quick Report**. В открывшемся окне диалога выберите вариант размещения полей по столбцам и нажмите кнопку Fields.
7. Появится окно диалога Field Picker. Выберите все поля таблицы и, используя кнопку Move, перенесите их в список Selected Fields и ОК.
8. Возвратившись в окно диалога Quick Report, нажмите кнопку ОК для завершения создания стандартного отчета.
9. Используя кнопку Label на панели инструментов Report Control, скорректируйте заголовки полей, присвоив им русские наименования.
10. Для того чтобы отчету придать законченный вид, добавьте область заголовка отчета, выполнив команду **Report | Title/Summary**.
11. В открывшемся окне диалога установите флажок Title Band и нажмите ОК. В отчете появится полоса Title. Разместите в ней текст заголовка отчета Список магазинов, используя кнопку Label. Скорректируйте текст, выполнив команду **Format | Font** и установив шрифт, поддерживающий кириллицу.
12. Просмотрите внешний вид отчета с помощью команды всплывающего меню Preview. Сохраните отчет.

#### Размещение итогового поля

Итоговое поле может быть размещено в полосе Detail для вывода нарастающих итогов. Для размещения поля выберите инструмент Field на панели инструментов Report Controls. Установите курсор в соответствующее место окна конструктора и установите требуемый размер поля. На экране появится окно диалога Report Expression. Для определения поля в качестве итогового нажмите кнопку Calculations. Выберите математическую операцию над значением выражения, определенного для данного поля.

Для определения выражения поля введите выражение непосредственно в поле ввода Expression или нажмите на кнопку вызова построителя, расположенную справа от поля ввода.

#### Размещение линий и прямоугольников

Линии и прямоугольники используются в отчетах для улучшения его внешнего вида и читабельности.

Для проведения вертикальной или горизонтальной линии в отчете выберите инструмент Line на панели Report Controls конструктора отчетов.

С помощью мыши проведите линию необходимой длины. Выбрав команду меню **Format | Pen**, можно установить толщину линий. Выбор панели Color Palette Toolbar позволяет определить цвет линии.

#### Выборка данных

В Visual FoxPro для формирования запросов используется конструктор запросов и команда SELECT. Результатом выборки всегда является таблица, которую можно сохранить в массиве, в создаваемой новой таблице, отобразить на экране в окне BROWSE или вывести в виде отчета.

Для вызова конструктора запросов в окне проекта выберите Data | Queries | New, затем New Query. Откроется окно диалога Add Table or View. Выберите таблицу и, используя кнопку Add, перенесите ее в окно конструктора запросов. Нажмите кнопку Add и закройте окно диалога Add Table or View. После выбора таблицы на экране появится окно конструктора запросов.

Для формирования запросов необходимо использовать меню Query и панель инструментов Query Designer. Для сохранения окна конструктора запросов выполните команду File | Save as основного меню.

Для формирования запросов используется команда SELECT:

```
SELECT [DISTINCT] <список выбираемых полей>  
FROM <список Таблиц>  
[WHERE <условие выборки> ]  
[GROUP BY <условие группировки> ]  
[ORDER BY <условие упорядочивания>]  
[[INTO <имя Таблицы> ]  
[TO FILE <имя файла> [ADDITIVE] | TO PRINT]]
```

Для выбора полей откройте вкладку Fields, и с помощью кнопки Add перенесите их в список Selected fields. Для просмотра результата запроса выполните OM | Query | Run Query или правую кнопку мыши и Run Query. Для просмотра сформированного запроса укажите правую кнопку мыши и SQL.

Для задания условия выборки необходимо ввести требуемые значения в соответствующие поля вкладки Filter. В столбце Field Name выберите поле и укажите условие сравнения, например, Наименование Магазина==Эльдорадо при точном совпадении. При точном несовпадении установите флажок Not. Для установки неточного совпадения укажите =. Для задания диапазона значений в окне конструктора запросов используются операторы <, > и Between.

Для формирования многотабличного запроса необходимо добавить в окно конструктора запросов все таблицы, участвующие в выборке, и определить условия их объединения. Для добавления таблицы укажите Query | Add Table или нажмите правую кнопку мыши и Add Table. Между таблицами будут установлены связи по совпадающим полям. Выберите нужные поля и укажите условия выборки во вкладке Filter.

При выполнении запроса можно вычислять итоговые (статистические) значения по одному или нескольким полям таблиц. Чтобы включить в запрос функцию или выражение, откройте вкладку Fields и нажмите кнопку открытия построителя выражения с правой стороны поля Functions and expression. В окне



диалога Expression Builder в поле ввода Expression сформируйте итоговое значение, используя поля списка Fields и функции области Functions.

### Порядок выполнения лабораторной работы

- 1.Создайте отчет для таблицы Магазины.
- 2.Произведите выборки данных по различным критериям для одной таблицы.
- 3.Произведите сортировку полей таблицы.
- 4.Сформируйте многотабличный запрос:
  - для двух таблиц;
  - для трех таблиц.
- 5.Для вычисления количества наличного товара воспользуйтесь функцией SUM(kol).
- 6.Для вычисления стоимости наличного товара введите выражение SUM(Товары.цена\*Наличие.количество).

## Лабораторная работа N 8

### Создание меню приложений

Обычно законченное приложение имеет свое собственное меню, которое содержит команды, предназначенные для выполнения конкретных задач.

Для создания строки меню выполните следующие действия:

- 1.Откройте окно конструктора меню.
- 2.Опишите пункты меню.
- 3.Отобразите строку меню на экране.
- 4.Определите действия, выполняемые при выборе пункта меню.

### Окно конструктора меню

Для открытия окна конструктора меню выполните последовательно действия **File | New | Menu | New File** или в окне конструктора проекта перейдите на вкладку Other и выберите Menus.

В открывшемся окне диалога New Menu предлагается два варианта создаваемого меню:

Menu – создание меню в виде строки меню;

Shortcut – создание всплывающего меню.

Выберите первый вариант

Описание меню состоит из двух частей. В первой части описывается вид меню, текст и типы пунктов меню. Во второй части описываются действия, выполняемые при выборе пункта меню. Внешний вид меню просматривается

с помощью кнопки Preview. Меню предварительно должно быть сгенерировано по команде Menu | Generate. Для определения текстов пунктов строки меню нажмите кнопку Insert и напечатайте текст в поле Prompt. Для определения типа пункта меню выберите требуемый элемент из списка Result.

### Последовательность создания меню

1. Откройте проект.
2. Перейдите на вкладку Other и выберите группу Menus, затем New.
3. В списке New Menu выберите тип меню Menu.
4. В поле Prompt введите тексты пунктов основного меню:

Магазины	тип	Submenu
Товары	тип	Submenu
Наличие	тип	Submenu
Выход	тип	Command
5. Нажмите кнопку Preview и просмотрите меню.
6. Сохраните **File | Save as** MenuPost.MPR
7. Для использования меню в приложениях его надо сгенерировать **Menu | Generate**.
8. Запустите программу меню **Program | Do**

Для восстановления на экране системного меню в командном окне введите команду

**SET SYSMENU TO DEFAULT**

### Создание подменю

1. Откройте файл MenuPost.mnx в окне конструктора меню.
2. Нажмите кнопку Create пункта меню Поставщики.
3. В поле Prompt введите пункт Ввод данных. Далее выберите тип элемента Command и в текстовой строке введите команду вызова экранной формы для ввода поставщиков:  
DO FORM <имя формы>
4. Аналогичным образом оформите остальные пункты меню: вызов отчетной формы, вызов запросов
5. Сгенерируйте меню.
6. Запустите созданное меню на выполнение.
7. Аналогично создайте подменю и для следующих пунктов строкового меню: Товары и Наличие.
8. Для пункта Выход введите команду Quit.

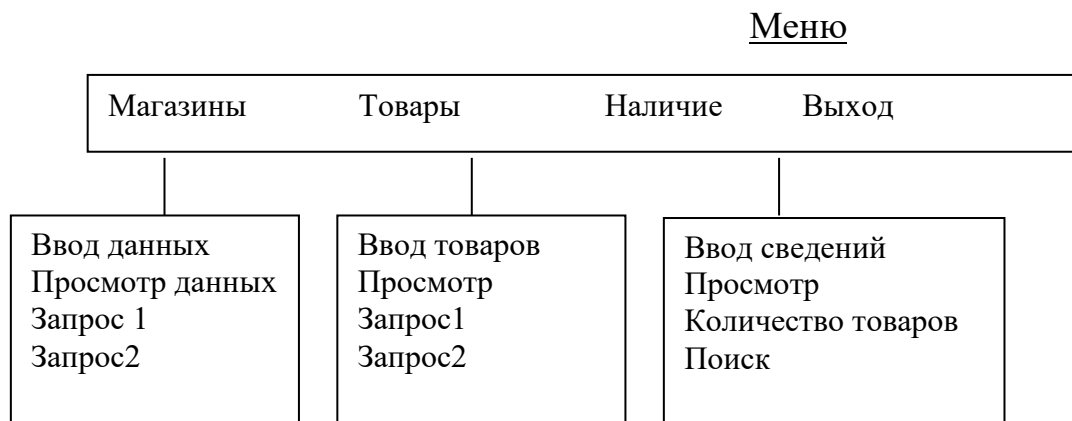
### Порядок выполнения лабораторной работы

1. Сконструируйте Меню.
2. Создайте меню и просмотрите.

3.Сгенерируйте.

4.Вызовите меню на выполнение.

5.Меню должно вызывать ранее созданные вами формы для ввода информации, вызова отчетов и запросов.



### П р и л о ж е н и е 1

#### Задачи проектирования баз данных.

1.Спроектировать базу данных для получения сведений о студентах данного вуза. Возможные атрибуты: ФИО студента, номер зачет.книжки, номер группы, наимен.специальности, размер стипендии.

2.Составить базу данных посетителей учреждения. Возможные атрибуты: ФИО посетителя, должность, организация, адрес, номер телефона.

3.Спроектировать базу данных получения сведений о поставщиках изделий. Возможные атрибуты: шифр поставщика, наименование поставщика, адрес, телефон, наименование изделия.

4. Спроектировать базу данных для получения сведений о профессиональной деятельности служащих. Возможные атрибуты: ФИО сотрудника, номер отдела, номер профессии, наименование профессии, дата начала работы по этой профессии.

5. Спроектировать базу данных о товарах, поставляемых на склад. Возможные атрибуты: номер товара, наименование товара, дата поставки, количество, цена единицы товара.

6. Спроектировать базу данных для хранения библиографических данных по литературе, поступающей на абонемент библиотеки. Возможные атрибуты: ФИО автора, название, наименование издательства, год издания, количество страниц, УДК, регистрационный номер.

7. Спроектировать базу данных маршрутов городского транспорта. Возможные атрибуты: номер маршрута, вид транспорта, начальный пункт, конечный пункт маршрута, протяженность маршрута, количество остановок, время в пути.

### П р и л о ж е н и е 2

База данных поставок.

Поставщики (S)

Таблица 1.

<b>NS</b>	<b>NameS</b>	<b>Status</b>	<b>Gorod</b>
S1	Саша	20	Самара
S2	Дима	10	Пермь
S3	Боря	30	Пермь
S4	Коля	20	Самара
S5	Аля	30	Уфа

Детали (P)

Таблица 2.

<b>NP</b>	<b>NameP</b>	<b>Color</b>	<b>Ves</b>	<b>Gorod</b>
P1	Гайка	Кр	12	Самара
P2	Болт	Зел	17	Пермь
P3	Винт	Гол	17	Казань
P4	Винт	Кр	14	Самара
P5	Кулачок	Гол	12	Пермь
P6	Звезда	Кр	19	Самара

Поставки (SP)

Таблица 3

<b>NS</b>	<b>NP</b>	<b>Kol</b>
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400