

## Лабораторная работа № 12

1. **Тема:** Создание интерфейса (входной формы).
2. **Цель работы:** 1. Освоить команды для создания входной формы.  
2. Изучить свойства элементов управления.

### 3. Литература:

1. Омельченко Л. ДШевякова, VisualFoxPro 9.0 Санкт-Петербург, 2009
2. ТВМусина, VisualFoxPro 9.0 Санкт-Петербург, 2009
3. ВКлепинин, Т.Агафонова, VisualFoxPro 9.0 Санкт-Петербург, 2008
4. Голицына О.Л., Максимов Н.В., Попов И.И. Базы данных Учебное пособие. М.Форум -Инфра-М,2009
5. Фуфаев Э.В., Фуфаев Д.Э. Базы данных.Учебное пособие.М. Издат.центр «Академия», 2006

### 4. Теоретические сведения

Входные формы создаются для просмотра и редактирования данных как в одной таблице, так и во взаимосвязанных таблицах. Для создания формы можно воспользоваться одним из инструментов:

- **FormWizard - Мастер форм**, который предназначен для быстрого построения «чернового» варианта формы и объектам, расположенным на форме присваиваются значение по умолчанию;
  - **Builder - Построитель**, который позволяет создать объекты с определенными свойствами;
  - **FormDesigner - Конструктор форм**, с помощью которого программист самостоятельно разрабатывает экранные формы любой степени сложности и наделяет объекты любыми свойствами.
- **Создание входной формы с помощью Мастера форм.**

1. Для открытия мастера форм в менеджере проекта необходимо выбрать вкладку **DOCUMENTS - >FORMS - >NEW - >FORMWIZARD**. Откроется диалоговое окно **WIZARDSELECTION** (рис.1) (выбор мастера) в котором **FORMWIZARD** используется для создания однотабличной формы, а **One-to-ManyFormWizard** - для многотабличной. Выбрать FormWizard.

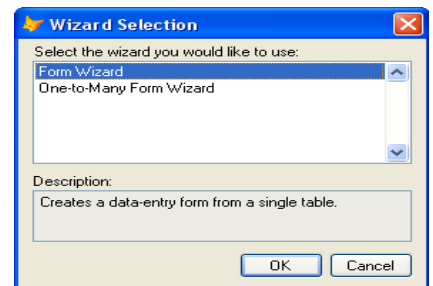
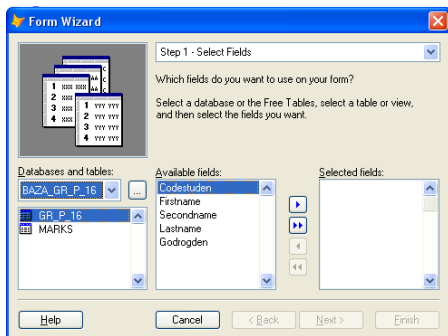


рис.1



- 2) Появляется первое диалоговое окно мастера (рис.2), в котором необходимо **указать таблицу**, для которой создается форма, и **поля таблицы**. На вкладке **Databaseandtables** необходимо выбрать базу данных и таблицу. В окне **Availablefields** выбрать необходимые поля таблицы.

рис.2

- 3) В следующем диалоговом окне (рис.3) следует установить **стиль**

**отображения объектов и типы кнопок управления**. В окне **STYLE** выбрать любой стиль. В области просмотра можно просмотреть, как будут выглядеть объекты формы. В разделе **Buttontype** выбрать стиль кнопок (например, **TextButtons** - кнопки с надписями; **PictureButtons** - кнопки с рисунками; **NoButtons** - кнопки отсутствуют).

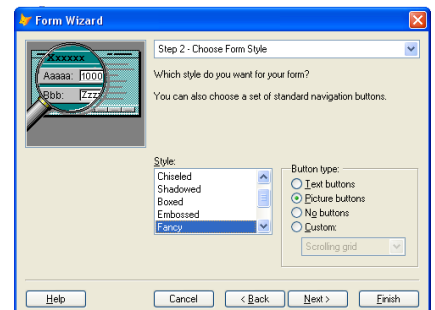


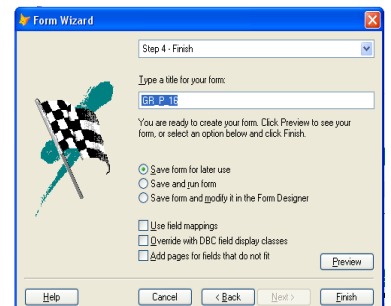
рис.3

- 4) На следующем шаге задается **критерий сортировки** данных, отображаемых в форме. (рис.4) Для переноса полей из списка **Availablefieldsorindextag** в список **Selectedfields** можно использовать кнопку **Add** или просто перетянуть их мышью, кнопка **Remove** удаляет выбранные поля. Опции **Ascending** (по возрастанию) и **Descending** (по убыванию) определяют вариант сортировки.

Рис.4

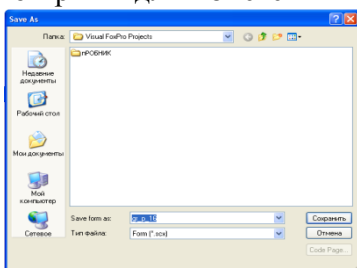
- 5) На заключительном этапе можно задать **заголовок формы** в поле

**Type a title for your form.** (рис.5) Далее необходимо выбрать одну из опций сохранения: **Save form for later use** (Сохранить форму), **Save and run form** (сохранить и запустить на выполнение) и **Save form and modify it in the Form Designer** (сохранить и открыть для изменения в конструкторе форм). С помощью кнопки **Preview** (просмотр), можно просмотреть, как будет выглядеть форма. Рис. 5



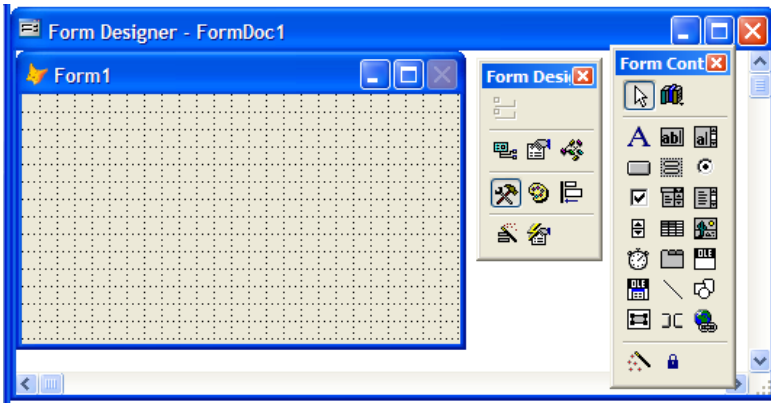
После нажатия кнопки **Finish**, откроется окно **Save As** (рис.6) в котором необходимо указать имя и папку, в которой будет храниться форма.

Рис.6



## Создание формы с помощью конструктора.

Предварительно открывается файл проекта.. Затем из главного меню подают команду:



File → New → в радиогруппе «File Type» выбирают кнопку «Form» и для запуска Конструктора форм нажимают кнопку «NewFile». На экран выводится диалоговая панель «FormDesigner».

Процесс создания интерфейса состоит из этапов :

- размещение элемента управления на форме;
- задание свойств элемента управления;
- задание обработчиков событий, если это необходимо.

1. Для **размещения элемента управления** на форме надо установить курсор мыши на нужный элемент управления на палитре инструментов и щелкнуть левой кнопкой. Затем перевести курсор мыши на форму, зажать **левую кнопку** и выделить **прямоугольную область**. После того как кнопка мыши будет отпущена, в выделенной прямоугольной области **будет размещен выбранный элемент управления**. Затем, размещенный на форме элемент управления, может быть **перемещен в другое место формы** и **изменены его размеры**.

2. **Задать значения свойств элемента управления** можно либо при создании интерфейса на диалоговой панели «Properties»( вкладка «All» ), либо **программным способом в обработчиках других элементов управления**.

Для написания программного кода обработчика надо вывести на экран окно специального редактора ( рис. 2 ) одним из способов:

- дважды щелкнуть левой кнопкой мыши на выбранном элементе управления;
- в диалоговой панели «Properties» **выбрать вкладку «Methods»**, затем справа от нужного имени метода ( в светлом поле ) дважды щелкнуть левой кнопкой мыши.

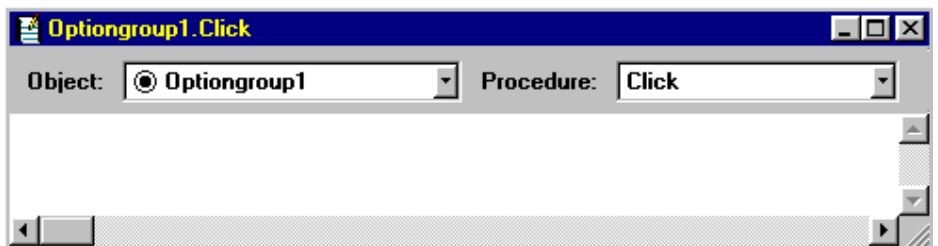


Рис. 2. Окно редактора для написания программного кода обработчика события.

3. Окно редактора содержит рабочее поле, где пишется программный текст обработчика и два раскрывающихся списка :

- **Object** - выбирается имя элемента управления, для которого пишется обработчик события.
- **Procedure** - определяется имя события, для которого пишется обработчик

## Рассмотрим подробнее создание формы с помощью конструктора.

Итак, в менеджере проектов выбрать вкладку **Documents**, перейти в группу **Forms**, и нажать кнопку **New**. Затем в открывшемся окне **NewForm** нажать одноименную кнопку.

### Окно конструктора.

Для задания **заголовка** формы предназначено свойство **Caption** окна **Properties** (свойства) (рис.1). Свойство **BackColor** позволяет задать цвет фона формы. Задайте заголовок и цвет на

свое усмотрение.

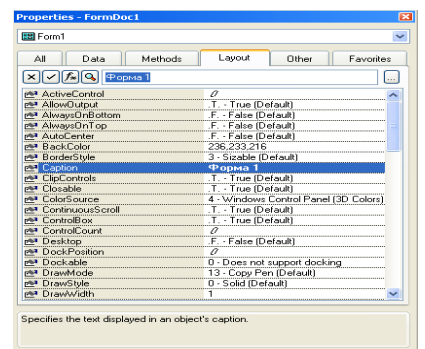
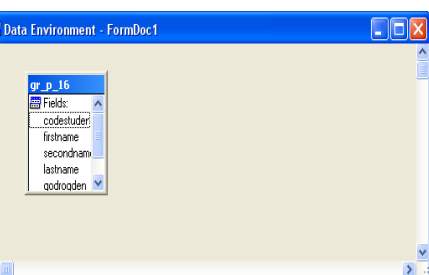


рис.1



1. Следующим шагом является определение **среды окружения** формы. Для этого необходимо **из контекстного меню** выбрать команду **DataEnvironment**. Откроется окно DataEnvironment (рис.2) в котором с помощью команды **Add** из контекстного меню добавить необходимую таблицу. Если нужной таблицы нет в списке, нажмите кнопку **Other** и, используя открывшееся окно **Open**, найдите требуемую таблицу.

рис.2

2. После того, как таблица была добавлена в среду окружения необходимо произвести **порядочение** данных, отображаемых в форме. Для этого в окне свойств таблицы (рис.3) (Properties) в опции **Order** выбрать название **индекса**.

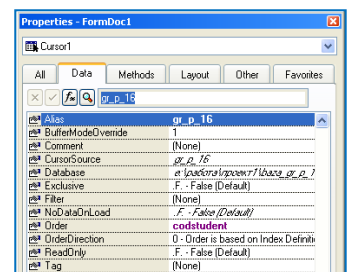
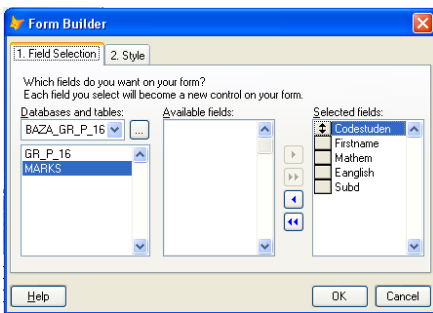
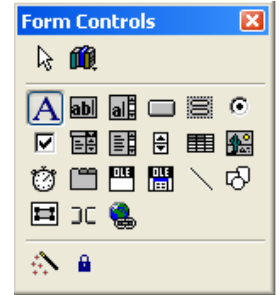


рис.3



3. Следующим шагом является **размещение объектов** в форме. Для начала разместим на форме **поля** нашей таблицы и **надписи** к ним. Для этого необходимо запустить **построитель форм**. Для этого из **контекстного** меню формы подать команду **Builder**, после этого откроется окно **FormBuilder** (рис.4). В котором, на вкладке **FieldSelection** нужно выбрать поля, отображаемые в форме, и на вкладке **Style** выбрать стиль их отображения.

рис.4



4. Далее необходимо разместить на форме **заголовок** формы. Для этого на панели инструментов **FormControls** (рис.5) необходимо выбрать кнопку **Label**, установить курсор на место расположения текста и нажать **ЛКМ**. Далее в окне **Properties** для данного объекта необходимо отредактировать его **свойства**. В опции **Caption** задать текст, который необходимо разместить на форме. С помощью опций **FontName**, **FontSize**, **ForeColor** – редактируются **вид, размер и цвет** текста соответственно. Отредактируйте их на свое усмотрение. Рис 5

5. Следующим элементом управления являются **кнопки**. Для размещения группы кнопок необходимо на панели инструментов **FormControls** выбрать инструмент **CommandGroup**, установить курсор в то место, где предполагается разместить кнопки, и нажать **ЛКМ**. На нашей форме будут размещены **кнопки перемещения** по записям таблицы и **кнопки выхода**.

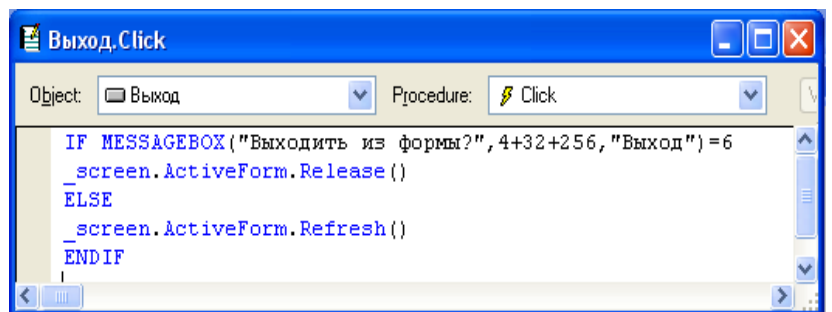
В окне свойств для объекта **CommandGroup** отредактируйте свойство **ButtonCount**, задав количество кнопок **равное 5**. Растяните рамку объекта так, чтобы в ней поместились все 5 кнопок.

Переведите объект в режим **редактирования**. Для этого установите на него курсор, нажмите **правую кнопку мыши** и из **контекстного меню** выберите команду **Edit**. Выделяя поочередно кнопки, переместите их, расположив горизонтально в одну линию.

Откройте окно свойств объекта типа **CommandGroup**. Нажмите кнопку раскрытия **списка**. 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**.

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

Теперь необходимо **определить команды**, которые будут выполняться при нажатии на данные кнопки. Для этого, открывая поочередно окно процедур метода **Click** (рис.6) каждого элемента, входящего в составной объект, введите следующие команды:



Для кнопки **Первая:**

```
IF !BOF()
GO TOP
ENDIF
_screen.ActiveForm.Refresh()
```

Для кнопки **Следующая:**

```
IF !EOF()
SKIP
ENDIF
_screen.ActiveForm.Refresh()
```

Для кнопки **Предыдущая:**

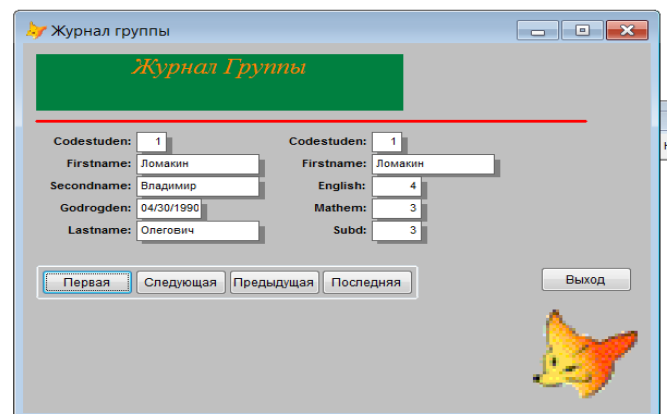
```
IF !BOF()
SKIP-1
ENDIF
_screen.ActiveForm.Refresh()
```

Для кнопки **Последняя:**

```
IF !EOF()
GO bottom
ENDIF
_screen.ActiveForm.Refresh()
```

Для кнопки **Выход:**

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



## Вопросы для самопроверки.

1. Укажите назначение входной формы.
2. Укажите способы создания входной формы.
3. Расскажите об особенностях Мастера форм.
4. Расскажите об особенностях Конструктора форм.
5. Расскажите о приемах размещения на форме элементов управления.
6. Расскажите о приемах задания свойств элементов управления.
7. Расскажите о приемах задания методов элементов управления

## Задание

Создайте различные входные формы для одной из таблиц проекта, используя образцы

The image displays three examples of Visual Basic forms and their corresponding code in the IDE.

**Example 1: Комбинированный список (combo\_prg.prg)**

```
oForm = CREATEOBJECT("Form")
* Добавляем в форму список
oForm.AddObject("combo1", "ComboBox")
WITH oForm
    .WindowState = 1
    .AutoCenter = .T.
    .Width = 300
    .Height = 100
    .Caption = "Комбинированный список"
ENDWITH
WITH oForm.combo1    && Определяем свойства списка
    .Style = 2
    .Left = 50
    .Top = 30
    .Width = 200
    .RowSourceType = 1
    .RowSource = "Москва, Новосибирск, Курган, Алушта"
ENDWITH
oForm.combo1.Visible = .T.
* Выводим форму со списком на экран
oForm.Show
```

**Example 2: Виды CheckBox (checkbox1\_progr.prg)**

```
oForm = CREATEOBJECT("Form")
* Добавляем в нее объекты CheckBox
oForm.AddObject("oChk1", "CheckBox")
oForm.AddObject("oChk2", "CheckBox")
oForm.AddObject("oChk3", "CheckBox")
* Устанавливаем для всех объектов нужные значения свойств
WITH oForm
    .WindowState = 1
    .AutoCenter = .T.
    .Width = 300
    .Height = 100
    .Caption = "Виды CheckBox"
ENDWITH
WITH oForm.oChk1
    .Caption = "Value = 0"
    .Value = 0
    .Left = 55
    .Top = 20
    .AutoSize = .T.
ENDWITH
WITH oForm.oChk2
    .Caption = "Value = 1"
    .Value = 1
    .Left = 55
    .Top = 40
    .AutoSize = .T.
ENDWITH
WITH oForm.oChk3
    .Caption = "Value = 2"
    .Value = 2
    .Left = 55
    .Top = 60
    .AutoSize = .T.
ENDWITH
oForm.oChk1.Visible = .T.
oForm.oChk2.Visible = .T.
oForm.oChk3.Visible = .T.
```

**Example 3: Form (primer.prg)**

```
LOCAL loForm
loForm = CREATEOBJECT("Form")
loForm.Width = 200
loForm.AddObject("myButton1", "myCommand")
loForm.AddObject("myButton2", "myCommand")
WITH loForm.myButton1
    .left = 10
    .top = 20
    .height = 26
    .width = 100
    .Caption = "Команда 1"
    .Tag = "Wait window [Команда кнопки 1] NOWAIT"
ENDWITH
WITH loForm.myButton2
    .left = 10
    .top = 80
    .height = 26
    .Caption = "Команда 2"
    .Tag = "Wait window [Команда кнопки 2] NOWAIT"
    .width = 100
ENDWITH
STORE .t. TO loForm.myButton1.visible, loForm.myButton2.visible
loForm.Show(1)

define class myCommand as commandbutton
proc Click
local cTemp
if not empty(this.tag) and type("this.Tag")="C"
cTemp = this.Tag
&cTemp
endif
endproc
ENDDEFINE
```