

Лабораторная работа №16 по дисциплине "Вычислительная техника и программирование"

ТЕМА: РАБОТА С ФАЙЛАМИ.


СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ПАПОК

1. В Вашей папке создайте папку с именем ВТП_лр16, совпадающим с именем файла с этим заданием, и затем скопируйте файл с заданием в эту папку.

2. При выполнении задания создавайте в папке ВТП_лр16 для каждой программы папку с именем, по смыслу соответствующим действиям, выполняемым составляемой программой, и сохраняйте все файлы, имеющие отношение к этой программе, в этой созданной папке (имена сохраняемых файлов должны отличаться от имен, присваиваемых по умолчанию, и должны по смыслу соответствовать действиям, выполняемым составляемой программой, причем рекомендуется, чтобы имя модуля отличалось от имени проекта цифрой, соответствующей номеру версии модуля).


3. ПРИ ВЫПОЛНЕНИИ ДРУГИХ ЛАБОРАТОРНЫХ РАБОТ ДЕЙСТВУЙТЕ АНАЛОГИЧНО ПРЕДЫДУЩЕМУ.

РАБОТА С ФАЙЛАМИ


4. Компонент TFileListBox  с вкладки Win3.1 окна Tool Palette предназначен для отображения списка файлов папки (каталога) или диска Windows. Наиболее характерные свойства компонента:

Свойство	Описание
FileType	Указывает типы включаемых и не включаемых в список файлов. Используются следующие подсвойства, принимающие, соответственно, значения True или False и указывающие на атрибуты файлов: ftReadOnly – только для чтения, ftHidden – невидимые, ftSystem – системные, ftVolumeID – обозначения дисков, ftDirectory – папки (каталоги), ftArchive – архивные, ftNormal – не имеющие особых атрибутов.
ShowGlyphs	Разрешает (True) или запрещает (False) показ пиктограмм файлов.
MultiSelect	Разрешает (True) или запрещает (False) выбор нескольких файлов.
FileEdit	Имя компонента TEdit (чаще всего Edit1), при наличии которого в нем отображается имя выбранного файла.
FileName	Свойство времени выполнения, значением которого является путь к выбранному файлу и его имя.

5. При наличии в окне приложения компонентов TFileListBox, TDirectoryListBox и TDriveComboBox рекомендуется установить между ними связи выбором для свойств FileList и DirList значений, совпадающих с конкретными именами соответствующих компонентов (обычно это имена FileListBox1 и DirectoryListBox1). Это необходимо для взаимосвязанного отображения изменений в компонентах.

6. Компонент TDirectoryListBox  с вкладки Win3.1 окна Tool Palette предназначен для отображения списка папок (каталогов) диска. Наиболее характерные свойства компонента:

Свойство	Описание
DirLabel	Имя компонента TLabel (чаще всего Label1), при наличии которого в нем отображается путь к текущей папке (каталогу) и ее имя. Если путь полностью не помещается в области компонента TLabel , то этот путь автоматически отображается в сокращенном виде.
FileList	Устанавливает связь с компонентом TFileListBox выбором значения, совпадающего с конкретным именем компонента (обычно FileListBox1), для отображения выбранным компонентом TFileListBox списка файлов папки (каталога).
Drive	Свойство времени выполнения, значением которого является имя диска, папки (каталога) которого отображаются компонентом.
Directory	Свойство времени выполнения, значением которого является путь к выбранной папке (каталогу) и ее имя.

7. Компонент **TDriveComboBox**  с вкладки Win3.1 окна **Tool Palette** предназначен для отображения раскрывающегося списка дисков компьютера. Наиболее характерные свойства компонента:

Свойство	Описание
TextCase	Указывает регистр буквы в имени диска: tcUpperCase – верхний или tcLowerCase – нижний.
DirList	Устанавливает связь с компонентом TDirectoryListBox выбором значения, совпадающего с конкретным именем компонента (обычно DirectoryListBox1), для отображения выбранным компонентом TDirectoryListBox списка папок (каталогов) диска.
Drive	Свойство времени выполнения, значением которого является имя выбранного диска.
Text	Свойство времени выполнения, значением которого является строка, содержащаяся в окне раскрывающегося списка.

ПРОГРАММА 16-1

Создайте приложение, при выполнении которого на экране появляется окно формы с заголовком **Проводник** со следующими компонентами с вкладок Win3.1 и **Standard** окна **Tool Palette**, расположенными следующим образом (соответствующим расположению аналогичных элементов в окне программы **Проводник**):

слева сверху непосредственно под заголовком окна формы – компонент **TDriveComboBox** с отображением имени диска на верхнем регистре;

непосредственно ниже компонента **TDriveComboBox** – компонент **TDirectoryListBox** с шириной, равной ширине компонента **TDriveComboBox**, и с высотой до нижней границы окна формы;

сверху непосредственно правее компонента **TDriveComboBox** – компонент **TLabel** с белым цветом фона, с высотой, равной высоте компонента **TDriveComboBox**, и с шириной до правой границы окна формы, предназначенный для отображения пути к текущей папке (каталогу) и ее имени (имя этого компонента **TLabel** должно быть значением свойства **DirLabel** компонента **TDirectoryListBox**);

на одном уровне с компонентом **TDirectoryListBox** и непосредственно правее его – компонент **TFileListBox** с отображением пиктограмм файлов, с возможностью выбора нескольких файлов, с высотой, равной высоте компонента **TDirectoryListBox**, и с шириной до правой границы окна формы.

При открытии диска и папки (каталога) в области компонентов **TDriveComboBox** и **TDirectoryListBox** ее содержимое должно отображаться в области компонента **TFileListBox**.

ПРОГРАММА 16-2

Создайте приложение с размещением в окне формы следующих компонентов из окна Tool Palette (см. также Рисунок 1):

MainMenu с вкладки Standard с командами в соответствии с Рисунком 1;

OpenDialog, SaveDialog, FontDialog, ColorDialog с вкладки Dialogs;

Memo с вкладки Standard.

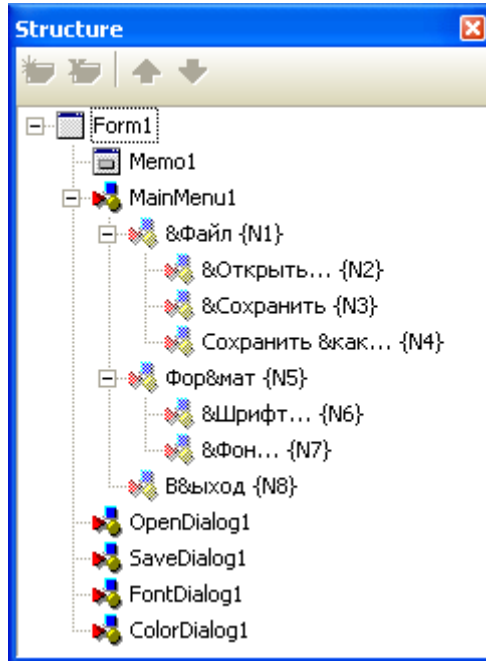


Рисунок 1

Для компонента Memo1 установите для свойства ScrollBars значение, соответствующее появлению у этого компонента горизонтальной и вертикальной полос прокрутки, для свойства Align – значение, при котором этот компонент занимает всю клиентскую область окна приложения.

Компонент Memo будет использоваться, в частности, для ввода в его строки (свойство Lines) текстовой информации из файла.

Окно приложения после сразу запуска должно иметь вид, как на Рисунке 2:

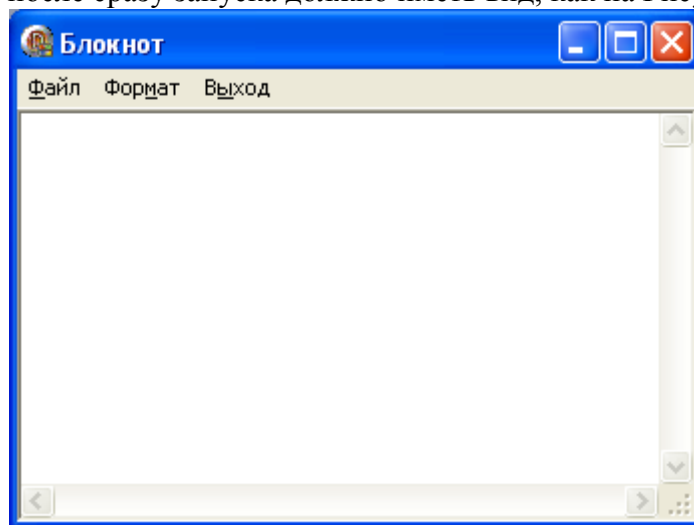


Рисунок 2

Так как для методов, используемых для открытия и сохранения файлов, необходимо в качестве аргумента использовать имя файла, то для хранения этого имени определите

переменную, назвав ее, например, FName:

```
var FName: string;
```

Для компонента `OpenDialog1` установите для свойства `Filter` значения как на Рисунке 3 в окне `Filter Editor` (это окно появляется на экране после нажатия на кнопку "...", в поле для значения свойства `Filter`).

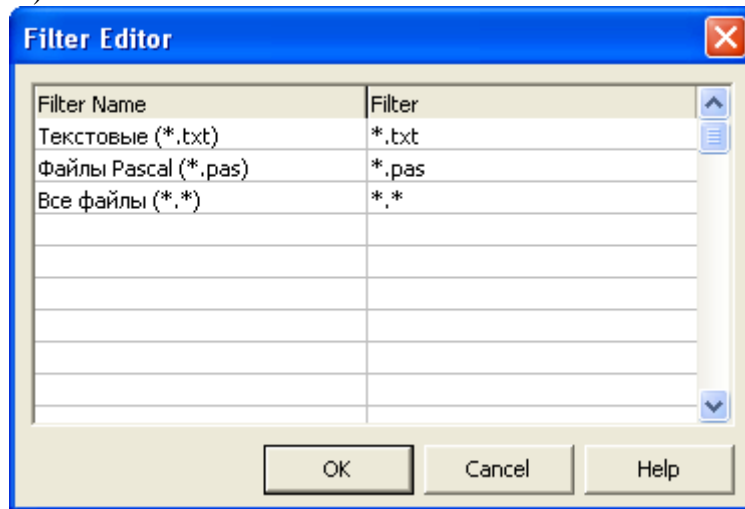


Рис.3

Процедура обработки выполнения команды меню Файл => Открыть... может состоять из следующих операторов:

```
if OpenDialog1.Execute then
begin
    FName:=OpenDialog1.FileName;
    Mem1.Lines.LoadFromFile(FName);
end;
```

Здесь **Execute** – стандартный метод для создания на экране диалогового окна, и если в этом окне был произведен какой-то выбор, то метод возвращает значение **True**. При этом в свойствах компонента-диалога запоминается выбор пользователя, который в дальнейшем можно использовать (в данном случае имя файла было запомнено в свойстве **FileName** и затем присвоено переменной **FName**); **LoadFromFile** – метод загрузки данных из файла (с именем **FName**).

Данные здесь загружаются из файла в строки поля **Mem1**.

Процедура обработки выполнения команды меню Файл => Сохранить может состоять из следующих операторов (с учетом возможности сохранения ранее не сохранявшегося файла):

```
if SaveDialog1.FileName<>' ' then
Mem1.Lines.SaveToFile(FName)
else
if SaveDialog1.Execute then
begin
    FName:=SaveDialog1.FileName;
    Mem1.Lines.SaveToFile(FName)
end;
```

Процедура обработки выполнения команды меню Файл => Сохранить как... может состоять из следующих операторов:

```
if SaveDialog1.Execute then
begin
    FName:=SaveDialog1.FileName;
    Mem1.Lines.SaveToFile(FName)
end;
```

Здесь **SaveToFile** – метод загрузки данных в файл.

Процедура обработки выполнения команды меню **Формат => Шрифт...** может состоять из оператора: **if** FontDialog1.Execute **then** Memo1.Font.Assign(FontDialog1.Font);

Здесь **Assign** – метод копирования данных из одного объекта в другой.

Для компонента **FontDialog** установите для свойства **Options.fApplyButton** значение **True** для возможности использования кнопки **Применить** в соответствующем диалоговом окне при выполнении в процедуре обработки события **OnApply** оператора

Memo1.Font.Assign(FontDialog1.Font)

Процедура обработки выполнения команды меню **Формат => Фон...** может состоять из оператора: **if** ColorDialog1.Execute **then** Memo1.Color:=ColorDialog1.Color

При выполнении команды меню **Выход** окно приложения должно закрываться.

Запустите созданное приложение и выполните действия, связанные с использованием диалоговых окон для открытия и сохранения текстового файла и замены цвета шрифта и фона в поле **Memo1**.

ОТЧЕТ О ВЫПОЛНЕНИИ РАБОТЫ

8. Продемонстрируйте проекты созданных приложений, а также их выполнение в среде программирования и независимо от нее.
