Создание простейшего текстового редактора в Delphi

Преподаватель информатики

ГБОУ СПО «Сибайский педагогический колледж»

Гаврилова Екатерина Федоровна

Необходимые компоненты:

TMemo и TMainMenu

Установим

- свойство **Align** в **alClient** - компонент растянется на все свободное пространство формы.

-свойство **ScrollBars** в **ssBoth**, а **WordWrap** в **False**.

-уберем введенный в Memo1 текст по умолчанию, удалив этот текст из свойства **Lines**.

Меню "Файл".

Щелкнем по MainMenu1 два раза - появится окно.
Теперь создадим меню "Файл". В нем элементы: "Создать", "Открыть", "Сохранить", "Сохранить как...", разделительную полоску (символ "-") и "Выход"
Установим горячие клавиши как показанно на **рис 1**. Горячие клавиши устанавливаются в свойстве **ShortCut**

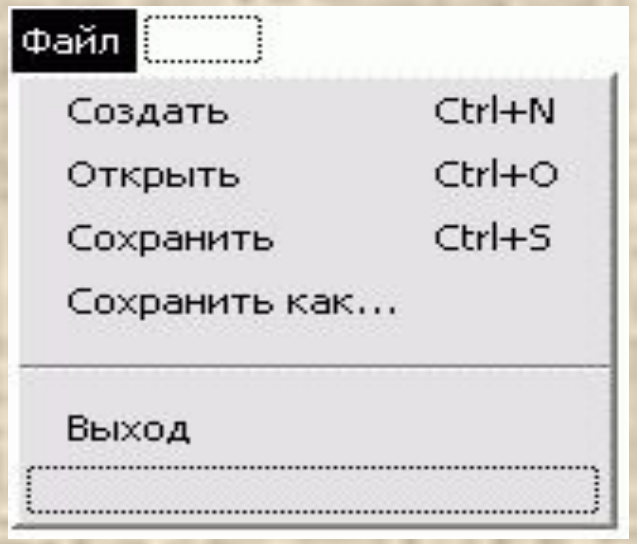
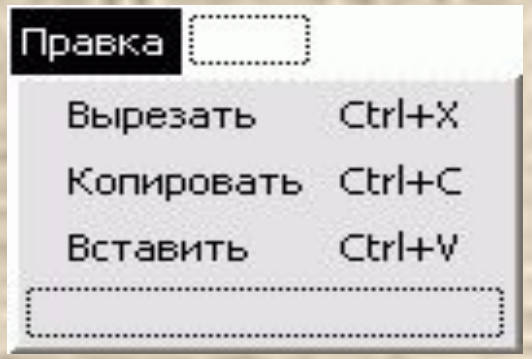


Рис. 1 - Меню "Файл"

Аналогично создаем меню "Правка":

Аналогично создаем меню "Формат":



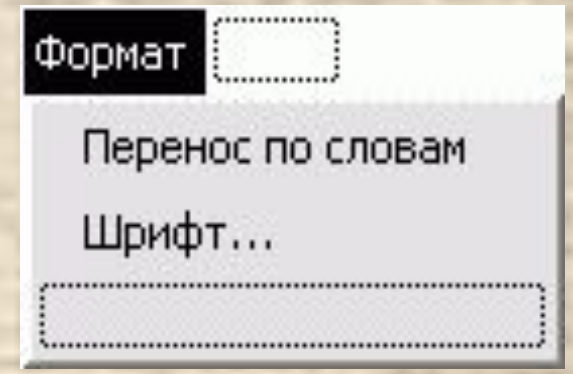


Рис. 2 - Меню "Правка" **Рис. 3 - Меню "Формат"**

Напишем код, который будет выполняться при выборе чего-либо из меню. Щелкнем по меню "Файл", а там выберем "Выход".

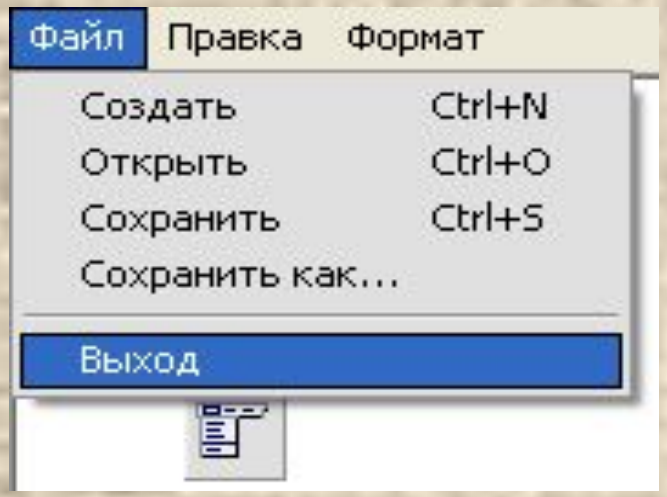


Рис. 4 - Создание события onClick

После нажатия по меню "Выход" появляется окно редактора кода с уже готовой процедурой. Пишем:

```
Close;
```

Реализация переноса по словам

В меню "Формат" выберем "Перенос по словам":

```
procedure TForm1.N14Click(Sender: TObject);
begin
 If Memo1.WordWrap then
 begin
 Memo1.WordWrap:=false;
 Memo1.ScrollBars:=ssBoth;
 N14.Checked:=False;
 end
 else
 begin
 Memo1.WordWrap:=True;
 Memo1.ScrollBars:=ssVertical;
 N14.Checked:=True;
 end;
end;
```

Реализация функции "Сохранить как..."

Воспользуемся компонентом **TSaveDialog**

на вкладке **Dialogs**. Нам нужно *.txt, т.е. текстовые документы. Чтобы это сделать нужно найти свойство **Filter** и нажать на кнопку

Появится окно. Заполните его как показанно на **рис. 5**.

Filter Name	Filter
Text File (*.txt)	*.txt
Все файлы	*.*

Рис. 5 - Редактор фильтра

измените свойство **DefaultExt** на ***.txt**. Это расширение которое будет добавляться к файлу.

Создадим процедуру сохранения файла. Напишем этот код сразу после **{$R *.dfm}**.

```
Procedure SaveAs;
begin
 If Form1.SaveDialog1.Execute then
 begin
 Form1.Memo1.Lines.SaveToFile(Form1.SaveDialog1.FileName);
 FileWork:=Form1.SaveDialog1.FileName;
 end;
end;
```

Далее выберем в меню "Файл" —> "Сохранить как...".

```
SaveAs;
```

Запишите в тетрадь как работает эта процедура? Самостоятельно.

В переменной **FileWork** будет храниться путь к файлу с которым работает пользователь. Это нужно для реализации функции "Файл" —> "Сохранить". Не забудьте ее объявить в разделе **Var** перед **implementation**.

```
var
 Form1: TForm1;
 FileWork: String;
```

Реализация функции "Сохранить"

Выберем "Файл" —> "Сохранить". И пишем:

```
If FileWork=" then SaveAs else
Memo1.Lines.SaveToFile(FileWork);
```

Реализация функции "Открыть"

Помимо **TSaveDialog** существует **TOpenDialog**

этот компонент предназначен для открытия файлов. Выложите этот компонент на форму и свойство Filter измените так же как у SaveDialog1.
Вообще компоненты TSaveDialog и TOpenDialog очень похожи. Диалог открытия вызывается так же — функцией **Execute**, которая возвращает **True** если пользователь желает открыть файл.
Жмем "Файл" —> "Открыть". Пишем:

```
if OpenDialog1.Execute then
begin
 Memo1.Lines.LoadFromFile(OpenDialog1.FileName);
 FileWork:=OpenDialog1.FileName;
end;
```

Реализация функции "Создать"

Жмем "Файл" —> "Создать". Пишем:

```
FileWork:='';
Memo1.Clear;
```

Реализация функций в меню "Правка"

По клику на меню "Вырезать":

```
Memo1.CutToClipboard;
```

По клику на меню "Копировать":

```
Memo1.CopyToClipboard;
```

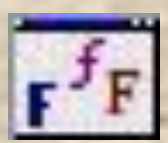
По клику на меню "Вставить":

```
Memo1.PasteFromClipboard;
```

Реализация функций в меню "Шрифт..."

Нам понадобится еще один компонент - **TFontDialog**

Создаем событие по клику по меню "Шрифт...".



if FontDialog1.Execute **then** Memo1.Font:=FontDialog1.Font

Завершаем...

Изменим заголовок окна: в свойстве **Caption** напишите "Блокнот". Свойство **Position** измените на **poScreenCenter** для того, чтобы форма появлялась по центру экрана.

Ну и напоследок добавим модуль XPMan в список модулей (секция uses):

```
uses
 Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
 Dialogs, Menus, StdCtrls, XPMan;
```

Этот модуль подключает XMManifest к приложению, т.е. если у Вас установленна XP, то все компоненты приобретают XP-шный вид, а если Vista - то становятся стиля Висты. Можете поэкспериментировать.

Реализуйте программу в среде программирования Delphi

Использованные источники:

- Валеева, Ю.А. Объектно-ориентированное программирование в среде Delphi [Текст] / Ю.А. Валеева. - Новокузнецк, 2003.
- Фаронов В.В. Delphi. Программирование на языке высокого уровня: учебник для вузов–СПб: Питер, 2006.–640с.