

## **Лабораторная работа**

*На тему: «Обработка файлов в среде Delphi»*

## Процедура AssignFile

Перед тем, как осуществлять ввод-вывод, файловая переменная должна быть связана с конкретным внешним файлом с помощью процедуры AssignFile.

Вызов	Параметры
AssignFile(F, Name)	F: TextFile; F - файловая переменная, связанная с конкретным физическим файлом. Name: String; Name – имя внешнего файла. или Name: PChar;

Затем файл должен быть открыт для чтения и/или записи (соответственно Reset – для чтения, Rewrite – для записи). После этого можно осуществлять организацию ввода-вывода.

## Процедура Reset

Открывает существующий файл, с которым связана файловая переменная F. Указатель текущей компоненты файла настраивается на начало файла.

Вызов	Параметры
Reset(F)	F: TextFile; F - файловая переменная, связанная с конкретным физическим файлом.

## Процедура Rewrite

Открывает новый пустой файл, присваивает ему имя, заданное процедурой AssignFile. Если файл с таким именем уже существует, то он уничтожается.

Вызов	Параметры
Rewrite(F)	F: TextFile; F - файловая переменная, связанная с конкретным физическим файлом.

Основополагающий момент – с каждым файлом в конкретный момент времени можно работать или в режиме чтения или в режиме записи. Одновременно нельзя, т.е. либо можно читать, либо записывать – другого не дано.

После работы с файлом он, как правило, должен быть закрыт процедурой Close. Это требование обязательно должно соблюдаться для файла, в который производилась запись.

## Процедура Close

Закрывает внешний файл, с которым связана файловая переменная F. При этом в случае необходимости в содержимое файла вносятся все произведенные изменения.

Вызов	Параметры
CloseFile(F)	F: TextFile; F - файловая переменная, связанная с конкретным физическим файлом.

Текстовый файл представляет собой совокупность символов, разделенных на строки, причем в конце каждой строки стоит признак конца строки.

Особенностью работы с текстовыми файлами является то, что параметры, значения которых вводятся и выводятся с помощью процедур Read или Write, могут быть не обязательно типа Char или string, а и других простых типов (целых, вещественных типов при вводе; целых, вещественных типов, типа Boolean - при выводе). Эти процедуры могут также работать и с ASCII-строками.

### Функция Eof

Принимает значение True, если указатель текущей компоненты файла находится за последним символом и False - в противном случае.

Значение функции типа Boolean.

Вызов	Параметры
Eof(F)	F: TextFile; F - файловая переменная, связанная с конкретным физическим файлом.

### Функция EoLn

Принимает значение True, если текущей компонентой файла является признак конца строки или если функция Eof(F) принимает значение True. В остальных случаях функция принимает значение False.

Значение функции типа Boolean.

Вызов	Параметры
EoLn(F)	F: TextFile; F - файловая переменная, связанная с конкретным физическим файлом.

Задание 1: Создайте приложение, которое создает текстовый файл text1.txt и записывает в него текст, введенный пользователем в окно Edit, после чего закрывает файл.

Решение:

**Создайте** форму и задайте для ее свойства Caption значение «Создание файла и вывод в него текста». Разместите на форме компоненты Edit1, Label1, Button1, как показано на рис. 6.1. Задайте значения для свойства Label1. Caption – «Введите текст», Button1. Caption – «Сохранить». Выровняйте компоненты и зафиксируйте их положение на форме.

**Сохраните** файлы модуля под именем **main** и проекта под именем **TextEditFile** в папке Обработка текстовых файлов.



Рис. 6.1 Пример формы проекта

**Создайте процедуру** обработки события кнопки «Сохранить», введите в окне Редактора кода следующий текст:

```
procedure TForm1. Button1Click (Sender: TObject);  
var  
f: TextFile; {описание файловой переменной}  
begin  
AssignFile (f, 'text1.txt'); {связь файловой переменной с файлом}  
Rewrite(f); {создать новый файл}
```

**Writeln (f, Edit1. Text);** {записать в файл}

**CloseFile(f); end;** {закрыть файл}

**Запустите** приложение и введите в окно Edit следующее предложение – «Мой первый пример текста». Щелкните мышкой на кнопке «Сохранить» и закройте окно приложения.

**Откройте** окно Проводника Windows папку Обработка текстовых файлов, в которой сохранены файлы проекта. В списке файлов этой папки находится вновь созданный файл text1.txt. Дважды щелкните левой кнопкой мыши на имени файла text1.txt. Убедитесь, что это – тот самый текст, который введен в окне приложения. Откроется окно редактора Блокнот с этим файлом. Закройте окно редактора Блокнот и Проводник.

Задание 2: Создайте приложение, открывающее текстовый файл для чтения и считывающие из него текст в окно Мемо. Перед открытием файла следует проверить его наличие, в случае его отсутствия должно выводиться соответствующее сообщение.

Решение:

**Создайте новое приложение** (проект). Создайте форму «Чтение текста из файла в окно». На форме разместите компоненты Memo1, Label1, Button1, как показано на рис. 6.2. Задайте значения свойств Label1. Caption – «Текст из файла», Button1. Caption – «Прочитать текст из файла».

**Для удаления** текста Memo1 из компонента выберите в окне Инспектора объектов объект Memo1, затем на странице Свойства выберите свойство Lines и в поле со значением Strings произведите двойной щелчок. После этого в окне StringListEditor удалите текст и щелкните мышью на кнопке ОК.

**Для обеспечения** возможности просмотра в окне Memo1 длинных текстов с использованием вертикальной полосы прокрутки в окне Инспектора объектов выберите свойство ScrollBars значение ssVertical. Выровняйте компоненты и зафиксируйте их положение на форме.

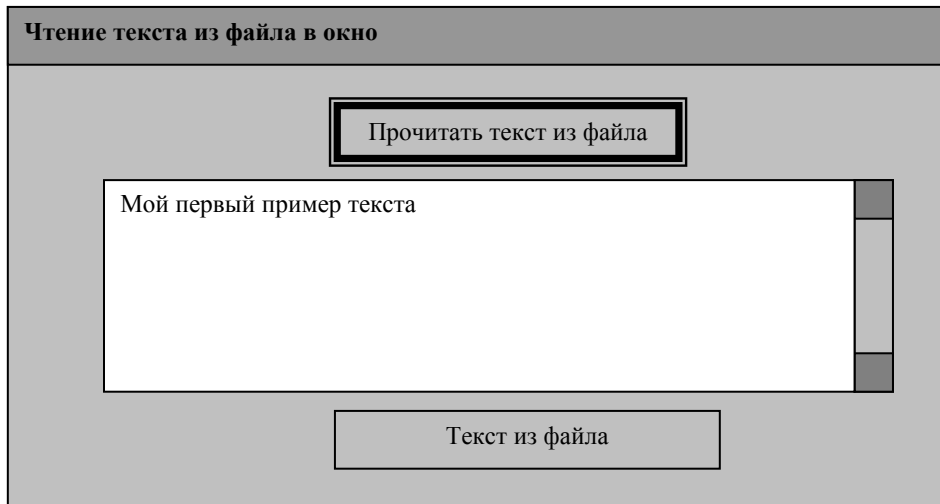


Рис. 6.2 Пример формы приложения

Создайте процедуру обработки события кнопки «Прочитать текст из файла», отредактируйте текст процедуры следующим образом:

```
procedure TForm1. Button1Click (Sender: TObject);  
var f: TextFile;  
ch: Char;  
begin  
AssignFile (f, 'text1.txt');  
{SI-}  
Reset(f);  
{SI+}  
if IOResult=0 then begin  
while not Eof(f) do  
begin  
Read (f, ch);  
Memo1. Text:=Memo1. Text+ch;  
end;  
CloseFile(f);  
end else  
ShowMessage ('Нет такого файла');
```

**end;**

**Сохраните** файл модуля под именем main1, а файл проекта – под именем TextMemoFile1 в папке Обработка текстовых файлов. Откомпилируйте и запустите приложение, проверьте его работу.

Задание 3: Создайте приложение, открывающее текстовый файл для дополнения и затем добавляющее в него введенный текст.

Решение:

**Создайте** новый проект, задайте название формы «Добавление текста в файл». На форме разместите компоненты Button1,2, Memo1, Label1,2, Edit1 как показано на рис. 6.3. Присвойте значения свойствам Label1. Caption – «Текст из файла», Button1. Caption – «Прочитать текст из файла», Label2. Caption – «Добавляемый текст», Button2. Caption – «Добавить текст в файл». Удалите текст из компонентов Memo1, Edit1. Установите линейку вертикальной прокрутки для обеспечения возможности просмотра длинных текстов в компоненте Memo1. Выровняйте компоненты и зафиксируйте их положение на форме.

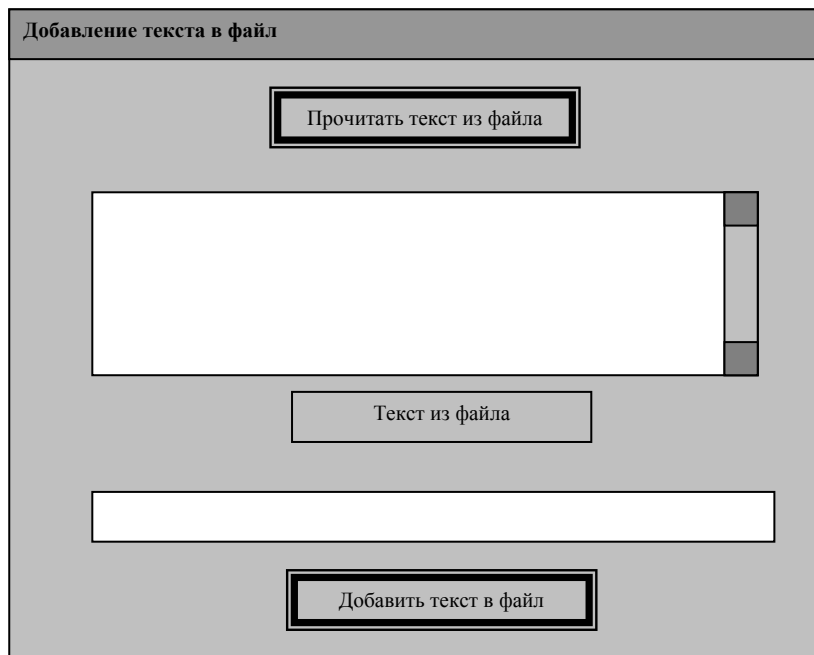


Рис. 6.3 Пример формы приложения

**Создайте** обработчик нажатия кнопки «Прочитать текст из файла» самостоятельно. Для кнопки «Добавить текст в файл» запишите следующий код события:

**Procedure TForm1. Button2Click (Sender: TObject);**

**Var**

**F: TextFile;**

**Begin**

**AssingFile (f, 'text1.txt');**

**Append (f);**

**Writeln (f, Edit1. Text);**

**CloseFile (f);**

**End;**

**Сохраните файл** модуля под именем Main2, а файл проекта – под именем TextMemoFile2 в папке Обработка текстовых файлов.

**Запустите** и проверьте работу приложения.



Задание 4: Создайте приложение, которое открывает текстовый файл с использованием метода `OpenDialog`, считывает текст из него в объект `Memo`, затем сохраняет измененный текст в файл с использованием метода `SaveDialog` и выводит текст на печать, используя метод `PrintDialog`.

Решение:

**Создайте** новый проект и сохраните в папке «Диалоговая панель». На форме разместите компоненты `Memo1`, `Button1,2,3`. Кнопки назовите «Сохранить», «Открыть», «Печать» соответственно. Задайте компоненту `Memo1` вертикальную полосу прокрутки и удалите текст. Выровняйте и зафиксируйте компоненты на форме (см. рис. 6.4).

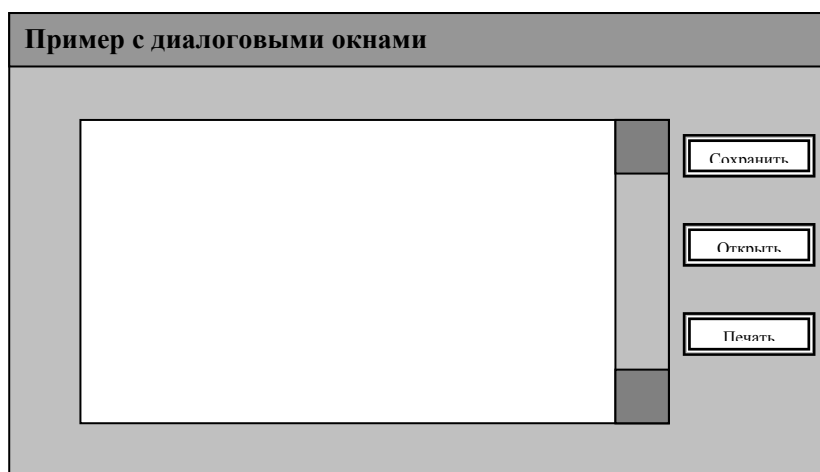


Рис. 6.4 Форма проекта «Диалоговая панель»

**Выберите** в палитре компонентов страницу `Dialog` и поместите на форму компоненты `OpenDialog`, `SaveDialog`, `PrintDialog`. Так как они **не являются визуальными** компонентами, то их можно поместить в любое место формы.

**Задайте** для свойства `SaveDialog`. `Title` значение «Сохранить текстовый файл», которое будет отображаться в заголовке диалогового окна сохранения файла. Чтобы при сохранении файла в окне диалога обеспечить выбор типа файла, выберите свойство `Filter` и произведите двойной щелчок в списке

значений. Откроется окно FilterEditor. Задайте фильтры для выбора типа и расширения файла:

Filter Name	Filter
Текстовый (*.txt)	*.txt
Текстовый (*.doc)	*.doc

И щелкните по кнопке **ОК**, затем установите расширение **\*.txt** по умолчанию – задайте свойству SaveDialog1.FilterIndex значение 1.

**Задайте** для свойства OpenFileDialog.Title – «Открыть текстовый файл». Чтобы реализовать выбор типа файла при открытии файла в окне диалога, на странице Свойства произведите двойной щелчок по списку значений свойства Filter. В окне Filter Editor задайте фильтры для выбора типа и расширения файла:

Filter Name	Filter
Текстовый (*.txt)	*.txt
Все файлы (*.*)	*.*

И щелкните по кнопке **ОК**, затем установите расширение **\*.txt** по умолчанию – задайте свойству OpenFileDialog.FilterIndex значение 1.

**Чтобы в диалоговом окне Печать** включить возможность выбора диапазона печатаемых страниц и печати выделенного фрагмента, задайте для свойств PrintDialog1.Options.poPageNums и PrintDialog1.Options.poSelection значение True.

После этого в раздел описания модулей добавьте в список **USES** модуль **PRINTERS**, чтобы не вводить свою переменную. Этот модуль позволяет управлять процессом печати.

**Отредактируйте** раздел описания переменных:

**Var**

**Form1: TForm1;**

**FName: string;**

**F: TextFile;**

**S: string;**

Создайте процедуры обработки событий трех кнопок, следующим образом:

1) сохранение:

```
procedure TForm1. Button1Click (Sender: TObject);
```

```
begin
```

```
fName:='Text1';
```

```
SaveDialog1. FileName:=FName;
```

```
if SaveDialog1. Execute then begin
```

```
fName:=SaveDialog1. FileName;
```

```
case SaveDialog1. FilterIndex of
```

```
1: fName:=fName+'.txt';
```

```
2: fName:=fName+'.doc';
```

```
end;
```

```
Memo1. Lines. SaveToFile(fName);
```

```
end;
```

```
end;
```

2) открытие:

```
procedure TForm1. Button2Click (Sender: TObject);
```

```
begin
```

```
if OpenFileDialog1. Execute then
```

```
begin
```

```
AssignFile (f, OpenFileDialog1. FileName);
```

```
fName:=OpenFileDialog1. FileName;
```

```
Reset(F);
```

```
Readln (F, s);
```

**Memo1. Text:=s;**

**CloseFile(F);**

**end;**

**end;**

3) печать:

**procedure TForm1. Button3Click (Sender: TObject);**

**begin**

**if PrintDialog1. Execute then**

**begin**

**AssignPrn(f);**

**Rewrite(F);**

**Writeln (f, Memo1. Text);**

**System. CloseFile(f);**

**end;**

**end;**