Практическая работа 7
Обработка файлов. Системные диалоги

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 папку Задание 1.

Создайте форму и задайте для ее свойства Caption значение «Создание файла и вывод в него текста». Разместите на форме компоненты Edit1, Labbel1, Button1, как показано на рис.



Задайте значения для свойства Label1.Caption – «Введите текст», Button1.Caption – «Сохранить». Выровняйте компоненты и зафиксируйте их положение на форме.

Сохраните файлы модуля и проекта в папке **Задание 1**.

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 введите произвольный текст. Щелкните мышкой на кнопке «Сохранить» и закройте окно приложения.

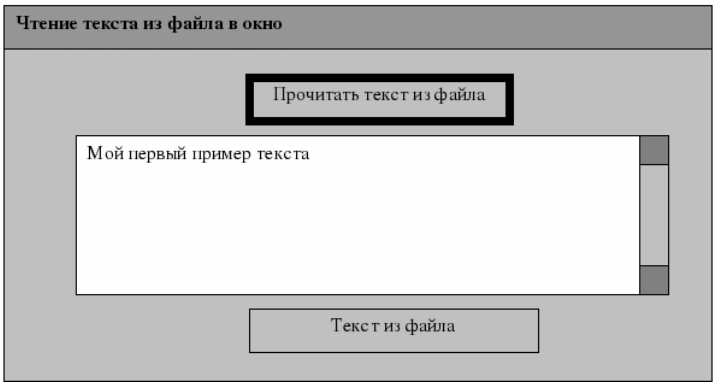
Откройте окно Проводника Windows папку Обработка текстовых файлов, в которой сохранены файлы проекта. В списке файлов этой папки находится вновь созданный файл text1.txt. Дважды щелкните левой кнопкой мыши на имени файла text1.txt. Убедитесь, что это – тот самый текст, который введен в окне приложения. 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 Memo. Перед открытием файла следует проверить его наличие, в случае его отсутствия должно выводиться соответствующее сообщение.

Решение:

Создайте папку Задание 2 и скопируйте в нее текстовый файл **`text1.txt`** из папки Задание 1.

Создайте новое приложение (проект). Создайте форму «Чтение текста из файла в окно». На форме разместите компоненты Memo1, Label1, Button1, как показано на рис.



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 свойсво 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.

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');
 {$I-}
Reset(f);
{$I+}
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 файлы модуля и проекта в папке **Задание2**. Откомпилируйте и запустите приложение, проверьте его работу.

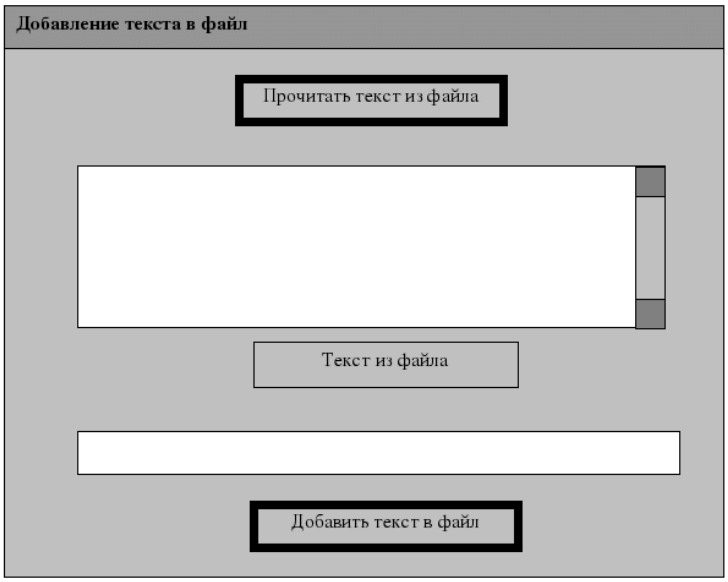
Задание 3: Создайте приложение, открывающее текстовый файл для дополнения и затем добавляющее в него введенный текст.

Решение:

Создайте папку Задание3 и скопируйте в нее текстовый файл **`text1.txt`** из папки Задание 1.

Создайте новый проект, задайте название формы «Добавление текста в файл». На форме разместите компонентыButton1,2, Memo1, Label1,2, Edit1 как показано на рис. Присвойте значения свойствам Label1.Caption – «Текст из файла», Button1.Capton – «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.



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

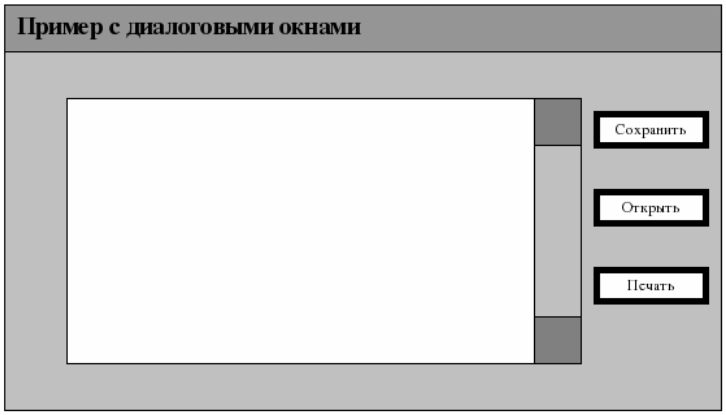
```
Procedure TForm1.Button2Click(Sender: TObject);
Var
F : TextFile ;
Begin
AssingFile (f, 'text1.txt') ;
Append ( f ) ;
Writeln ( f , Edit1.Text ) ;
CloseFile ( f ) ;
End ;
```

Сохраните файлы модуля и проекта в папке Задание 3.
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.



Выберите в палитре компонентов страницу Dialog и поместите на форму компоненты OpenDialog, SaveDialog. 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 OpenDialog1.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 OpenDialog1.FilterIndex значение 1.

Отредактируйте раздел описания переменных:
Создайте процедуры обработки событий трех кнопок, следующим образом:

```
Var
Form1 : TForm1;
FName : string ;
F : TextFile ;
S : string ;
```

Создайте обработчик события on`Click` для кнопки **Сохранить.**

```
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;
```

Создайте обработчик события on`Click` для кнопки **Открыть.**

```
procedure TForm1.Button2Click(Sender: TObject);
begin
if OpenDialog1.Execute then
begin
AssignFile(f, OpenDialog1.FileName);
fName:=OpenDialog1.FileName;
Reset(F);
Readln(F, s);
Memo1.Text:=s;
CloseFile(F);
end;
end;
```

Сохраните и запустите на выполнение данное приложение. Проверьте работу кнопок Открыть и Сохранить.

Контрольные вопросы:

1. Опишите назначение процедур и функций файлового ввода-вывода: Append, AssignFile, CloseFile, Read, Reset, Rewrite, Write?
2. Опишите назначение компонентов OpenDialog, SaveDialog?
3. Каковы особенности их размещения на форме приложения?
4. Каково назначение свойства SaveDialog1.Filter? Какими способами можно изменять значения этого свойства?