

# Работа с текстовыми и типизированными файлами

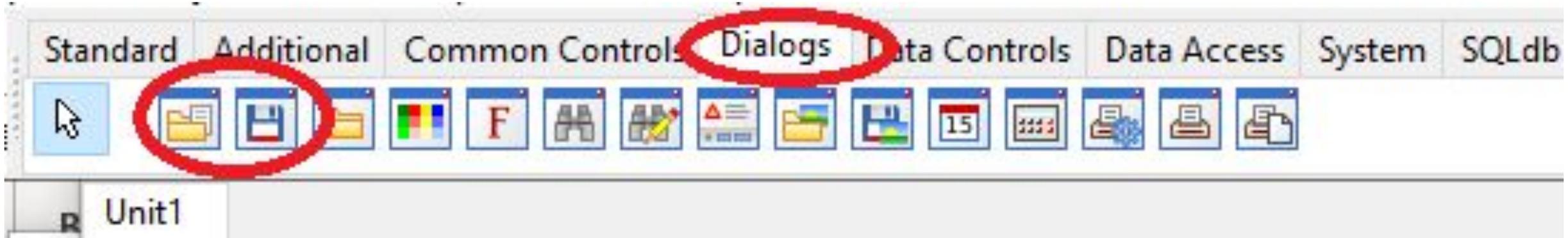
# tOpenDialog, tSaveDialog

Для визуальной работы с файлами существуют такие компоненты как tOpenDialog и tSaveDialog. Компоненты располагаются на панели во вкладке «Dialogs»

Работа с компонентами аналогична, поэтому разбираем на примере работы с tOpenDialog

Компоненты дают возможность использовать стандартные для Windows диалогам открытия и сохранения файлов.

Компоненты являются не визуальными. На этапе проектирования это квадратик, при работе приложения их не видно.



# Некоторые свойства и методы диалоговых компонент

- Метод **Execute** – функция, вызывающая стандартный диалог открытия файлов. Возвращаемое значение типа `boolean`. Принимает значения **True**, если пользователем был выбран файл и нажата кнопка **ОК**, **False** – в противном случае.
- Свойство **FileName** – содержит имя и полный путь выбранного файла.

Пример работы:

```
If not OpenFileDialog1.Execute then
```

```
Begin      ShowMessage ( 'Файл не выбран!' );
```

```
            exit;
```

```
end;
```

```
fn:=OpenDialog1.FileName; //считали имя и путь выбранного  
файла
```

```
AssignFile (f, fn); //связали это имя и файловую переменную
```

# Стандартные процедуры и функции для работы с файлами.

Основные процедуры и функции для работы с файлами были изучены в прошлом семестре. Вспомним.

При работе всегда используется файловая переменная. Не забываем описывать.

- **AssignFile(*f*, <ИмяФайла>)** – процедура, связывающая объявленную файловую переменную с физическим файлом на диске.

Параметры:

- ✓ *f* - файловая переменная
- ✓ <Имя файла> - полное (вместе с путем) имя файла. В случае отсутствия пути, файл будет записан/ищется в каталоге проекта.
- **CloseFile(*f*)** – процедура, закрывающая файл. Связь файла и файловой переменной при этом не меняется. Параметр:
  - ✓ *f* – имя файловой переменной.

# I Процедуры открытия

## файлов

- **Reset(*f*)** – процедура, открывающая **текстовый** файл на чтение, **типизированный** – на чтение и запись.
  - ✓ Параметром является имя файловой переменной.
  - ✓ Курсор при открытии ставится на начало файла
- **Rewrite(*f*)** – процедура, открывающая файл на запись.
  - ✓ Параметр – имя файловой переменной.
  - ✓ Если файл ранее не существовал, создает его.
  - ✓ Если файл был не пустой, очищает все содержимое.
  - ✓ Курсор на начале файла.
- **Append(*f*)** – процедура, открывающая **текстовый** файл на дозапись.
  - ✓ Параметр – имя файловой переменной.
  - ✓ Курсор при открытии устанавливается в конец файла.

# Процедуры чтения из и записи в файл

Параметрами всех процедур является имя файловой переменной *f* и **список переменных**, откуда/куда будут записаны данные в файл/из файла.

- **Write(*f*, <переменные>)** – записывает **данные переменных** в текущую позицию файла. Указатель перемещается на следующий элемент.
- **WriteLn(*f*, <переменные>)** – записывает **данные переменных** в текущую позицию файла. На текущую строку ставится указатель ее конца. Указатель перемещается на первую позицию следующей строки.
- **Read(*f*, <переменные>)** – считывает в **переменные** очередной(ые) элементы файла с текущей позиции. Указатель перемещается на следующий элемент.
- **ReadLn(*f*, <переменные>)** – считывает в **переменные** очередной(ые) элементы файла с текущей позиции. Указатель перемещается на

# Индикаторы конца файла / строки. Проверка существования файла

Это функции, возвращающие значения *True* или *False*.

- **EoF(*f*)** – “*End of File*” принимает значение *True*, если курсор достиг конца файла, *False* – в противном случае. Параметр – файловая переменная.
- **EoLn(*f*)** – “*End of Line*” принимает значение *True*, если курсор достиг конца строки, *False* – в противном случае. Параметр – файловая переменная.
- **FileExists(<ИмяФайла>)** – принимает значение *True*, если файл существует и найден, *False* – в противном случае.

# Пример

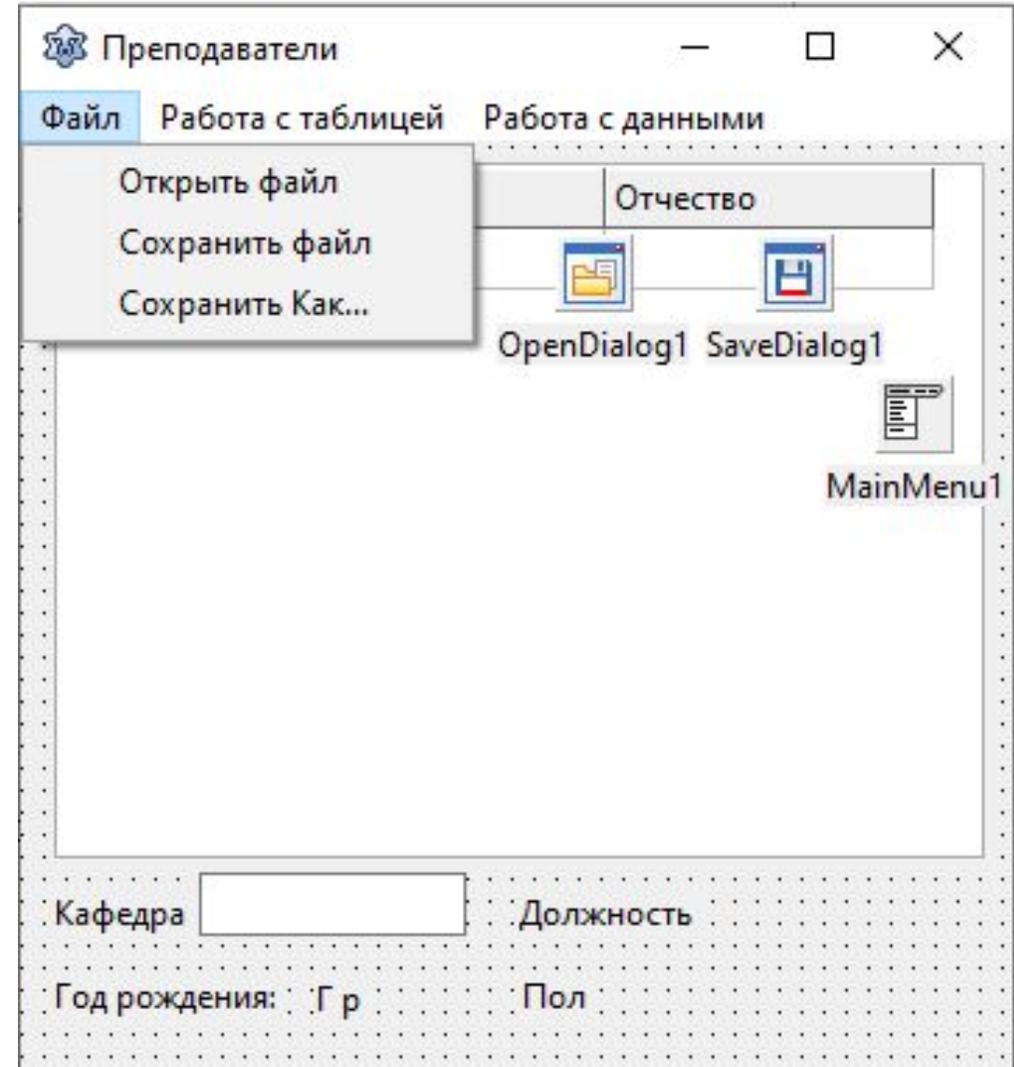
## ИСПОЛЬЗОВАНИЯ

Воспользуемся программой из прошлого семинара – база данных преподавателей.

Наше приложение обрабатывает некоторые данные. Но каждый раз при закрытии программы, данные теряются и надо снова их водить. Чтобы этого избежать, запишем наши данные в файл.

Модифицируем форму как показано на рисунке:

- Добавляем на форму компоненты `tOpenDialog` и `tSaveDialog`
- Добавляем пункты меню для



Также нам понадобятся новые глобальные типы и переменные:

**type**

```
tfile = file of prep;
```

**var**

```
Form1: TForm1; f:tfile; fn:string;
```

Описание типов и переменных должно быть в разделе **interface**

Как видно из описания, *f* – файловая переменная для хранения элементов типа *prep*, переменная *fn* вводится для хранения имени файла с путем к нему.

# Обработка выбора пункта меню «Сохранить Как...»

```
procedure TForm1.MenuItem8Click(Sender: TObject);  
var j:byte;  
begin  
  If not SaveDialog1.Execute then exit; //Если файл не выбран, выход  
  fn:=SaveDialog1.FileName; //запоминаем имя файла на диске  
  AssignFile(f,fn); //связываем файл и файловую переменную  
  Rewrite(f); //открываем файл на перезапись  
  for j:=1 to n do Write(f,t[j]);//цикл записи в файл каждого эл-та таблицы  
  CloseFile(f); //закрытие файла.  
end;
```

# Обработка выбора пункта меню «Сохранить файл»

В отличие от предыдущего пункта, тут есть возможность сохранить информацию в тот же файл, если он был выбран ранее.

```
var j:byte;
```

```
begin
```

```
if not FileExists(fn) then begin //проверка, что нет ранее выбранного файла
```

```
  If not SaveDialog1.Execute then exit; //тогда надо его выбрать,
```

```
  fn:=SaveDialog1.FileName; //запомнить имя и путь
```

```
  AssignFile(f,fn); //связать с файловой переменной
```

```
end;
```

```
Rewrite(f);
```

```
for j:=1 to n do Write(f,t[j]);
```

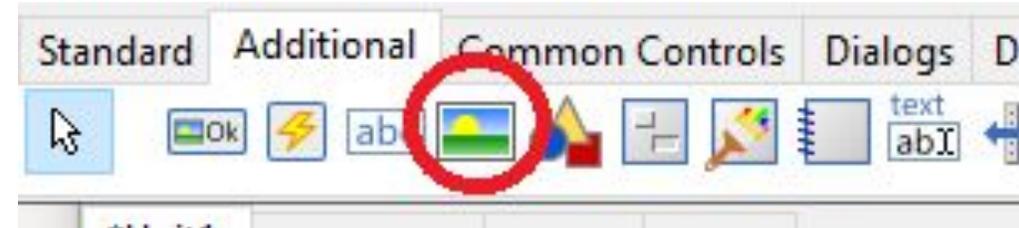
```
CloseFile(f);
```

# Обработка выбора пункта меню «Открыть файл»

```
begin
  If not OpenFileDialog1.Execute then exit; //если файл не выбран, выход
  fn:=OpenDialog1.FileName; //запомнить имя и путь файла на диске
  AssignFile(f,fn); //связать физический файл с файловой переменной
  Reset(f); //открыть файл на чтение. Курсор в начале
  n:=0; //количество элементов таблицы обнуляем
  While not EOF(f) Do begin //пока не дойдем до конца файла
    n:=n+1; //увеличить на 1 количество элементов таблицы
    read(f,t[n]); //записать очередной элемент типизированного файла в эл-т
табл
  end;
  CloseFile(f); //Закрывать файл
  UpdateTab(); //Обновить таблицу на форме. (Процедура из прошлого
семинара)
end;
```

# Вставка картинки на форму

Для вставки картинки/рисунка используется компонента **tImage**. Она расположена на палитре во вкладке **Additional**



Некоторые свойства и методы **tImage**:

- Свойство **Picture** – собственно изображение
- Метод **LoadFromFile(<ИмяФайла>)** позволяет загрузить картинку из файла
- Свойство **Stretch** – признак автоматического масштабирования. Если установлен в **True**, рисунок впишется в размер компоненты.
- **Proportional** – признак пропорциональности. Рисунок впишется в компоненту без искажения. Работает при выключенном **AutoSize**
- **AutoSize** – изменяет размер компоненты по размеру рисунка, если **True**

# Пример работы

Добавим в наше приложение возможность вставки картинки.

Например, фото преподавателя. Вставим на формы 1 и 2

Преподаватели

Файл Работа с таблицей Работа с данными

№	Фамилия	Имя	Отчество

OpenDialog1 SaveDialog1

MainMenu1

Кафедра  Должность

Год рождения:  Г р Пол

Ввод элемента

Фамилия

Имя

Отчество

Кафедра

Должность

Год рождения

Пол  Муж  Жен

Выбрать

Файл изображения

Отмена ОК

OpenDialog1

Устанавливаем свойства компоненты 

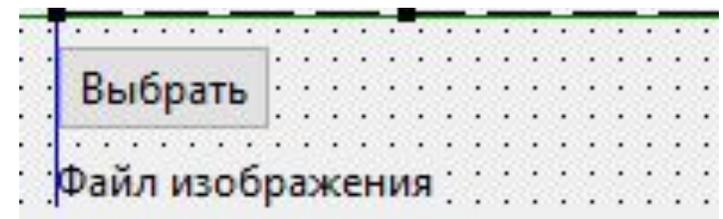
Добавляем в тип еще одно поле «FileOfImage»,  
в котором будет храниться имя рисунка:

Proportional	<input checked="" type="checkbox"/> (True)
ShowHint	<input type="checkbox"/> (False)
Stretch	<input checked="" type="checkbox"/> (True)
Stretchable	<input checked="" type="checkbox"/> (True)

```
prep = Record
  fam,name,otch: string[50];
  kaf,dol:string[30];
  brn:longint;
  gen:byte;
  FileOfImage:string[250];
end;
```

На форму 2 также необходимо добавить:

- кнопку «Выбрать», для выбора файла, содержащего рисунок,
- метку (Label) для отображения имени файла с рисунком
- Компонент OpenFileDialog для интерактивного выбора рисунка



# Обработка кнопки «Выбрать»

```
procedure TForm2.Button3Click(Sender: TObject);
```

```
Begin
```

```
//если после открытия диалога файл не найден, то выход из  
процедуры
```

```
If not OpenFileDialog1.Execute Then exit;
```

```
//присваиваем метке имя выбранного файла с рисунком
```

```
Label7.Caption:=OpenDialog1.FileName;
```

```
//отображаем содержимое файла в компоненте
```

```
Image1.Picture.LoadFromFile(OpenDialog1.FileName);
```

```
end;
```

# Изменение в обработке отображения элемента (Форма 1)

```
procedure TForm1.WhenSelecting(i:integer);
Begin
  Edit1.Text:=t[i].kaf;
  Label3.Caption:=t[i].dol;
  Label4.Caption:=IntToStr(t[i].brn);
  if t[i].gen=0 then Label5.Caption:='Мужчина' else
  Label5.Caption:='Женщина';
  if FileExists(t[i].FileOfImage) then
    Image1.Picture.LoadFromFile(t[i].FileOfImage);
end;
```

**Если существует файл, находящийся в поле элемента, то отображаем его в компоненте**

# Изменение в обработке меню «Добавить запись»

```
begin
  Form2.Edit1.Text:="";
  Form2.Edit2.Text:="";
  Form2.Edit3.Text:="";
  Form2.Edit4.Text:="";
  Form2.Edit5.Text:="";
  Form2.ComboBox1.Text:="";
  Form2.RadioGroup1.ItemIndex:=0;
  Form2.Label7.Caption:='Файл
изображения';
  Form2.Image1.Picture.Clear;
  Form2.ShowModal;
```

```
if flag then begin
  n:=n+1;
  t[n].fam:=Form2.Edit1.Text;
  t[n].name:=Form2.Edit2.Text;
  t[n].otch:=Form2.Edit3.Text;
  t[n].kaf:=Form2.Edit4.Text;
  TryStrToInt(Form2.Edit5.Text,t[n].brn);
  t[n].dol:=Form2.ComboBox1.Text;
  t[n].gen:=Form2.RadioGroup1.ItemIndex;
  t[n].FileOfImage:=Form2.Label7.Caption;
end;
UpdateTab();
end;
```

# Изменения в обработке меню «Изменить запись»

```
var k:byte;
begin
  k:=StringGrid1.Row;
  Form2.Edit1.Text:=t[k].fam;
  Form2.Edit2.Text:=t[k].name;
  Form2.Edit3.Text:=t[k].otch;
  Form2.Edit4.Text:=t[k].kaf;
  Form2.Edit5.Text:=IntToStr(t[k].brn);
  Form2.RadioGroup1.ItemIndex:=t[k].gen;
  Form2.ComboBox1.Text:=t[k].dol;
  Form2.Label7.Caption:=t[k].FileOfImage;
  Form2.image1.Picture.LoadFromFile
    (t[k].FileOfImage);
```

```
Form2.ShowModal;
if flag then begin
  t[k].fam:=Form2.Edit1.Text;
  t[k].name:=Form2.Edit2.Text;
  t[k].otch:=Form2.Edit3.Text;
  t[k].kaf:=Form2.Edit4.Text;
  TryStrToInt(Form2.Edit5.Text,t[k].brn);
  t[k].dol:=Form2.ComboBox1.Text;
  t[k].gen:=Form2.RadioGroup1.ItemIndex;
  t[k].FileOfImage:=Form2.Label7.Caption;
end;
UpdateTab();
end;
```

# Работа с текстовым

## файлом.

Модифицируем наше приложение таким образом, чтобы кафедры и должности можно было загружать из текстового файла и выбирать из предложенного списка.

Заменяем компонент ввода кафедры Edit на компонент ComboBox. При этом в остальных частях программы необходимо сделать следующие изменения в Unit1:

- Всюду запись `Form2.Edit(N).Text` меняем на `Form2.ComboBox(K).Text` (N, K – номера компонент в проекте).

Далее необходимо создать текстовые файлы, в которых будут записаны должности и кафедры (`kaf.txt`, `dolz.txt`). Их можно создавать средствами ОС. При этом необходимо подчиняться следующим условиям.

- Файлы должны находиться в директории проекта
- На одной строке одно наименование

# Программирование.

Очевидно, для того, чтобы пользователь мог выбирать, необходимо создать этот список.

Удобнее всего загружать список в момент открытия формы. Итак нам необходима обработка открытия формы. Двойной щелчок по форме.

```
If FileExists('kaf.txt') then begin
```

```
    ComboBox2.Clear;
```

```
    ComboBox2.Items.LoadFromFile('kaf.txt'); //метод загрузки данных из файла
```

```
end;
```

```
If FileExists('dolz.txt') then begin
```

```
    ComboBox1.Clear;
```

```
    ComboBox1.Items.LoadFromFile('dolz.txt');
```

```
end;
```

- Необходимо предусмотреть ситуацию, когда в списке не оказалось нужной нам кафедры или должности.
- Компонент дает возможность вручную ввести информацию в поле Text.
- После этого необходимо сохранить то, что было введено вручную.
- Произойти это должно при сохранении данных формы. То есть при нажатии на кнопку «ОК».
- Добавлять в список нужно ТОЛЬКО тот элемент, которого ранее не было в списке.
- Элементов ComboBox на форме два. Значит сохранять нужно для двух компонент. Процессы идентичные, поэтому удобно написать одну процедуру, к которой можно обращаться при необходимости.

Описать новую процедуру, назовем ее «SvComboBox», надо вне класса TForm2, но в разделе interface:

```
procedure SvComboBox(Fnam:string; cb:tComboBox);
```

Параметры – имя файла Fnam, объект ComboBox;

Тело процедуры в разделе implementation

```
procedure SvComboBox(Fnam:string; cb:tComboBox);
```

```
var j:byte; check:boolean;
```

```
begin
```

```
    check:=False;           // флаг, что нового значения нет в списке
```

```
    For j:=0 to cb.Items.Count-1 Do      //сравнить каждый элемент списка
```

```
        if cb.Text = cb.Items[j] Then begin //со значением в поле Text
```

```
            Check:=True;           //если в списке уже существует значение
```

```
            Break;                 //переключить флаг, выйти из цикла
```

```
        end;
```

```
    if check=False Then begin //в случае, если такого значения не обнаружено
```

```
        cb.Items.Add(cb.Text);      //добавить значение в список
```

```
        cb.Items.SaveToFile(Fnam);  //сохранить список в файл
```

```
    end;
```

```
end;
```