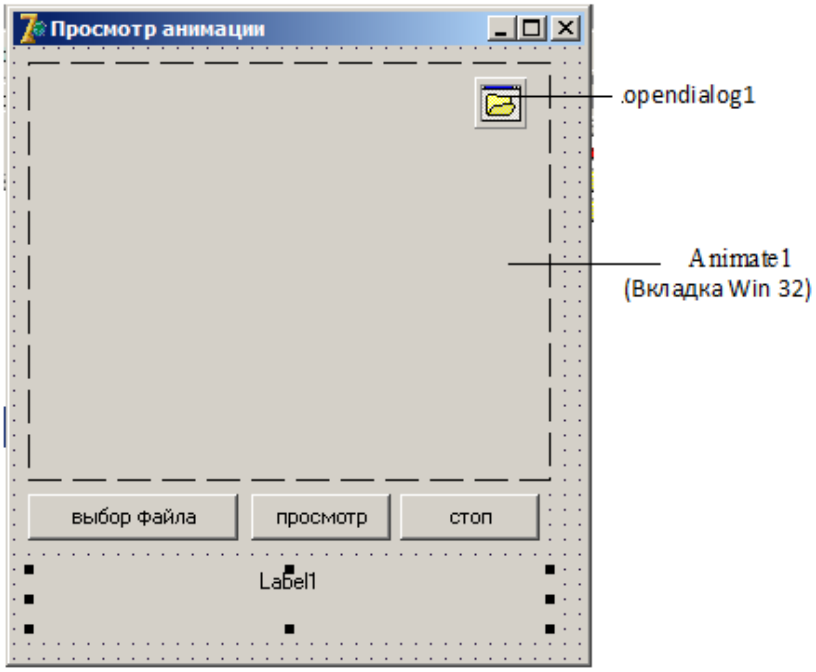
Практическая работа 9
Компоненты мультимедиа

Задание 1. Написать программу, используя которую можно просматривать анимацию без звука при помощи компонента Animate.

Описание хода работы

Разместите на форме компоненты, в соответствии с рисунком.



Создайте обработчик события OnClick для кнопки «выбор файла», которая используется для выбора загружаемого файла.

```
procedure TForm1.Button1Click(Sender: TObject);
begin
OpenDialog1.Title := 'Выбрать AVI-файл';
OpenDialog1.InitialDir :='';
if OpenDialog1.Execute then
Animate1.FileName := opendialog1.FileName;
label1.Caption:='Загружен файл '+opendialog1.FileName;
end
```

Создайте обработчик события OnClick для кнопки «просмотр», которая используется для выбора загружаемого файла.

```
procedure TForm1.Button2Click(Sender: TObject);
begin
Animate1.Active:=true;
end;
```

Создайте обработчик события OnClick для кнопки «просмотр», которая используется для выбора загружаемого файла.

```
procedure TForm1.Button7Click(Sender: TObject);
begin
Animate1.Active:=false;
end;
```

Запустите приложение. Откройте видеофайл без звука (например, **bart.avi**), протестируйте работу приложения.

Если формат загружаемого файла не поддерживается компонентом, то вам будет выдано сообщение об ошибке с текстом «cannot open AVI»

Задание 2. Создание собственного видеопроигрывателя

В среде Delphi имеется очень полезный компонент (палитра System), представляющий собой компактную мини-студию и благодаря которому можно прослушивать звуковые файлы, например в форматах *.wav или *.mp3, а также просматривать видеофайлы различных известных форматов, например *avi.

Сам компонент состоит из управляющих кнопок, напоминает вид обычного видеомагнитофона с возможностью проигрыша, паузы, остановки, записи, и т.д.

Определим общую схему проекта программы «Мой Проигрыватель», позволяющей просматривать видео файлы с расширением *wmv, *avi.

Создайте новый проект. Сбросьте на форму следующие компоненты:

1. компонент MediaPlayer1, установите значение его свойства AutoEnable равным false. Данное свойство отвечает за доступность кнопок компонента MediaPlayer1. Если оно установлено в false, то все кнопки всегда доступны. Выберите свойство VisibleButtons. Установите для btPlay, btPause, btStop значение true, для остальных копок: btPrev, btStep, btBack, btRecord, btEject значение false;
2. OpenDialog1
3. Timer1, установите значение его свойства Enabled равным false, а Interval:=100
4. TrackBar1
5. Button1, установите значение ее свойства Caption равным OPEN



Здесь кнопка «OPEN» будет содержать код, который, во-первых, активизирует диалоговое окно в виде компонента OpenDialog1, с помощью которого будет выбираться имя видео-файла, который затем будет проигрываться компонентом MediaPlayer1, а во-вторых, непосредственно запустит MediaPlayer1 на выполнение.

Указатель компонента TrackBar1 будет перемещаться в момент проигрывания файла, то есть информировать пользователя о том, какая часть видеофайла в данный момент проигрывается, а также установкой этого указателя можно будет перемещаться на любую позицию проигрываемого файла.

В данной программе запрограммируем произвольное перемещение пользователем указателя на любую позицию только в момент «Паузы» или «Остановки» проигрывателя.

Определим соотношения между кнопками «Play», «Pause» и «Stop» и кратко рассмотрим тип TButtonSet

В момент нажатия кнопки «Play» должно начаться воспроизведение, указатель компонента TrackBar1 должен перемещаться по мере проигрывания, данная кнопка «Play» должна перестать быть доступной, а должны быть доступны только кнопки «Stop» и «Pause».

В момент нажатия кнопки «Pause» проигрывание должно остановиться, указатель компонента TrackBar1 должен остановиться в той позиции, где он находился, данная кнопка «Pause» должна перестать быть доступной, и только кнопки «Stop» и «Play» должны быть доступны.

В момент нажатия кнопки «Stop» проигрывание должно завершиться, указатель компонента TrackBar1 должен установиться в начальную позицию, а кнопка «Stop» вместе с кнопкой «Pause» должны перестать быть доступными, зато должна быть доступна кнопка «Play».

Доступность всех кнопок компонента MediaPlayer1 определяет его свойство

EnabledButtons, которое имеет тип – множество. В этом множестве перечисляются идентификаторы кнопок компонента MediaPlayer1. Если они находятся во множестве, то данные кнопки доступны, иначе нет.

Описание данного множества имеет следующий вид:

```
    property EnabledButtons: TButtonSet read FEnabledButtons
write SetEnabledButtons
    default [btPlay, btPause, btStop, btNext, btPrev, btStep,
btBack, btRecord, btEject];

    TMPBtnType = (btPlay, btPause, btStop, btNext, btPrev,
btStep, btBack, btRecord, btEject);
    TButtonSet = set of TMPBtnType;
```

Из описания следует, что по умолчанию (default) доступны все имеющиеся кнопки, однако нас интересуют только первые 3 кнопки btPlay, btPause и btStop из множества типа TButtonSet. Остальные кнопки мы ради простоты программировать не будем.

Поэтому, чтобы изменять доступность той или иной кнопки, требуется просто уметь работать со множеством.

В данном случае, если множество пустое, то есть имеет вид [] , значит недоступна ни одна кнопка, если, например, множество имеет вид [btPlay, btPause], то доступны только две кнопки «Play» и «Pause».

Чтобы, например, создать множество [btPlay, btPause, btStop], можно применить простое сложение множеств [btPlay, btPause] + [btStop], аналогично, чтобы получить множество [btPlay, btPause], можно применить операцию разности множеств [btPlay, btPause, btStop] – [btStop].

Создадим глобальную переменную типа TbuttonSet, хранящую множество идентификаторов кнопок компонента MediaPlayer1 по умолчанию.

В окне редактора кода объявите глобальную переменную T типа TButtonSet.

```
var
  Form1: TForm1;
  T:TButtonSet;
```

Затем в обработчике события onCreate формы напишите следующее:

```
    T:=MediaPlayer1.EnabledButtons;
    {Инициализируем глобальную переменную T множеством всех
кнопок MediaPlayer1.EnabledButtons в начальный момент, то есть
все кнопки доступны}

    MediaPlayer1.EnabledButtons:=[];
    {Устанавливаем множество пустым, то есть ни одна кнопка не
доступна в MediaPlayer1 в начальный момент запуск программы}
```

В обработчике события onClick кнопки Button1 напишите следующее:

```
    MediaPlayer1.EnabledButtons:=T-[btPlay];
    {Отключаем доступность кнопки «Play», просто вычитая ее из
множества T, где доступны все кнопки, и присваиваем данный
результат свойству типа множество EnabledButton компонента
MediaPlayer1}

    if OpenDialog1.Execute then

    begin
    MediaPlayer1.FileName:=OpenDialog1.FileName; {Указываем имя
проигрываемого файла компоненту MediaPlayer1}

    MediaPlayer1.Frames:=1; {Устанавливаем количество кадров, на
которое будет сдвигаться видеофайл при щелчке на кнопке «Step»}

    MediaPlayer1.Open; {Загружаем видеофайл в оперативную память}

    MediaPlayer1.Play; {Начать воспроизведение видеофайла}
    end;

    MediaPlayer1.Notify:=true;{Устанавливаем уведомление о
событии мультимедиа}

    TrackBar1.Min:=0; {Минимально возможный номер,
соответствующий позиции указателя TrackBar1}
    TrackBar1.Max:=MediaPlayer1.Length; {Присваиваем
максимальному значению позиции указателя TrackBar1 число, равное
количеству кадров видеофайла}
    Timer1.Enabled:=true; {активируем таймер для перемещения
указателя}
```

Здесь все ясно из комментариев, метод Open загружает видеофайл в оперативную память, а Play воспроизводит видеофайл. У компонента MediaPlayer1 имеется событие onNotify – уведомление о событии мультимедиа, которое происходит всякий раз, когда нажали какую- либо кнопку компонента MediaPlayer1,или когда началось воспроизведение или прекратилось и т.д., то есть когда произошло любое мультимедийное событие.

Существует булевский флаг Notify, отвечающий за то, будет ли обрабатываться такое событие. Если он установлен в True, то всякий раз, когда генерируется мультимедийное событие, будет вызываться обработчик данного события, иначе он вызываться не будет. В описываемой будем всегда устанавливать его значение равным True.

Затем устанавливаем в минимальное и максимальное значение позиции указателя TrackBar1. Минимальное ноль, а максимальное равно полному числу кадров видеофайла.

И чтобы в реальном времени видеть, какая часть файла проигрывается, перемещаем указатель TrackBar1 через определенные промежутки времени, используя обработку события onTime таймера Timer1.

Создадим обработчик события onTime компонента Timer1. Создайте данный обработчик события и напишите там всего одно выражение:

```
    TrackBar1.Position:=MediaPlayer1.Position;
```

Устанавливаем указатель TrackBar1 в позицию равную номеру кадра видеофайла.

Создадим обработчик события onClick компонента MediaPlayer1. Создайте данный обработчик события и напишите там следующее:

```
    MediaPlayer1.Notify:=true; {Генерировать уведомление о
событии мультимедиа}
    if Button=btPlay then begin {Если нажали кнопку Play}
    MediaPlayer1.EnabledButtons:=T-[btPlay]; {то она становится
недоступной}
    MediaPlayer1.Position:=TrackBar1.Position; {Устанавливаем
номер текущего кадра MediaPlayer1 в позицию определяемую
указателем TrackBar1}
    timer1.Enabled:=true; {активируем таймер}
    end;
    if Button=btPause then MediaPlayer1.EnabledButtons:=T-
[btPause]; {Если нажали кнопку Pause, то она становится
недоступной }
    if Button=btStop then MediaPlayer1.EnabledButtons:=T-
[btPause,btStop]; {Если нажали кнопку Stop, то она становится
недоступной вместе с кнопкой Pause}
```

И наконец осталось описать обработчик события мультимедиа.

Создадим обработчик события onNotify компонента MediaPlayer1

```
    if (MediaPlayer1.Mode=mpPaused) {Если режим воспроизведении
находится в состоянии Паузы, тогда}
    then
    timer1.Enabled:=false; {останавливаем перемещение указателя
TrackBar1 путем перевода таймера в неактивное состояние}
    if (MediaPlayer1.Mode=mpStopped) {Если режим воспроизведении
находится в состоянии Остановки, тогда}
    then
    begin
    timer1.Enabled:=false; {останавливаем перемещение указателя
TrackBar1 }
    TrackBar1.Position:=0; {устанавливаем его позицию в ноль}
    MediaPlayer1.Rewind; {переходим в начало видеофайла }
    end;
    if (MediaPlayer1.Mode=mpPlaying) {Если режим воспроизведении
находится в состоянии Проигрывания, тогда}
    then timer1.Enabled:=true; {активируем перемещение указателя
TrackBar1 путем перевода таймера в активное состояние}
```

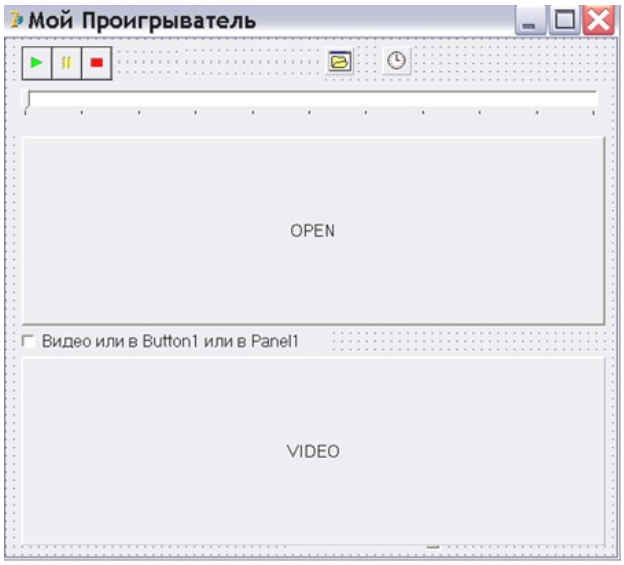
Запустите программу щелкните на кнопке Button1, затем с помощью диалогового окна выберите нужный видеофайл и нажмите OK. Начнется проигрывание. Щелкайте на кнопках Play, Pause и Stop. В состоянии паузы переместите указатель и снова нажмите Play. Программа должна работать правильно.

Вывод видео на стандартные компоненты Delphi.

В этом же проекте перенаправим видеопоток либо на Button1 либо на Panel1.

Растяните немножко кнопку Button1. Сбросьте на форму следующие компоненты:

Компонент	Свойство	Значение свойства
CheckBox1	Caption	Видео или в Button1 или в Panel1
Panel1	Caption	VIDEO



Создайте обработчик события onClick компонента CheckBox1 и напишите там всего лишь одно выражение:

```
if CheckBox1.Checked then MediaPlayer1.Display:=Button1
                     else MediaPlayer1.Display:=Panel1;
```

То есть, если CheckBox1 выбран, то видеоизображение будет направлено на кнопку Button1, а иначе будет выводиться на поверхность компонента Panel1.

Здесь используется свойство Display класса TwinControl, определяющее компонент. на который будет выводиться изображение. Таким образом, любой компонент, являющийся наследником класса TwinControl, может стать «экраном» для отображения видео. Чтобы в этом убедиться, в дальнейшем попробуйте самостоятельно добавить любой компонент этого класса и присвойте его имя свойству Display.

Запустите программу. Щелкайте на CheckBox1, при каждом щелчке изображение должно перепрыгивать с одного компонента на другой.