

Коробицин Владимир Михайлович

ЗНАКОМСТВО С МУЛЬТИМЕДИА В VISUAL BASIC. УРОКИ 4-5

УРОК 4:

Запуск анимационных и видео-файлов.

Использование таймера.

Извлечение текущего времени.

Экранная заставка для Windows.

Назначение клавиш.

Создайте папку Lesson41. Поместите в нее файлы Mci.vbx, 00.avi, 01.avi и 41.wmf. Три последних файла можно заменить любыми другими, имеющими расширения AVI, WMF (или BMP). Запустите Visual Basic. Окно формы Form1 «окрасьте», например, в серый цвет. Расположите это окно в правой нижней части экрана. Добавьте в проект Вашего приложения файл поддержки мультимедийных объектов Mci.vbx. Для этого в меню File (Файл) выберите команду Add File ... (Добавить файл ...) и через появившееся окно загрузите файл Mci.vbx. В окне объектов появится значок мультимедийных объектов. Щелкните мышью по кнопке с этим значком и поместите объект MMControl1 на форму Вашего приложения в нижней ее части. Подберите подходящие размеры этого объекта.

Сохраните проект Вашего приложения в папке Lesson41. Выйдите из среды Visual Basic. Из папки Lesson41 запустите



файл Project1.mak, который автоматически загрузит Visual Basic. В правой части формы Form1 поместите две командные кнопки. Для командной кнопки Command1 в процедуру Command1_Click впишите программный код:

```
MMControl1.Command = "Close"
MMControl1.FileName = "00.avi"
MMControl1.Command = "Open"
```

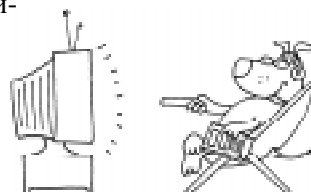
Вторая строка устанавливает имя файла для объекта MMControl1. Третья строка открывает этот файл. Первая строка необходима для закрытия файла при многократном обращении к объекту MMControl1.

Для командной кнопки Command2 в процедуру Command2_Click впишите программный код:

```
MMControl1.Command = "Close"
MMControl1.FileName = "01.avi"
MMControl1.Command = "Open"
```

Для командной кнопки Command1 свойству Caption придайте значение «Видео 1», а для кнопки Command2 свойству Caption придайте значение «Видео 2» (рисунок 1).

Запустите приложение и проверьте его в работе. Неудачно выбранные клавиши проигрывателя лучше скрыть от пользователя, придав их свойству



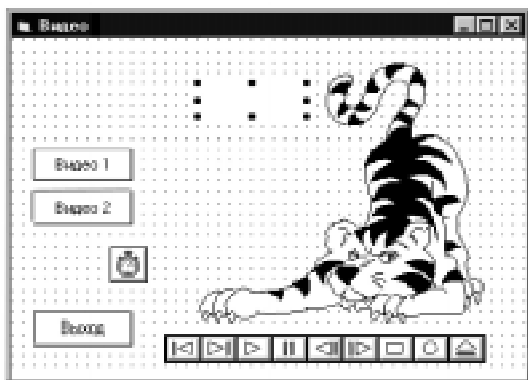


Рисунок 1.

Visible (Видимость) значение False (Ложь). При желании Вы можете поместить на форму еще несколько командных кнопок, с помощью которых будете загружать другие видео и анимационные файлы. В окно приложения можете поместить выбранный на Ваше усмотрение рисунок, например, из метафайла 41.wmf. Если Вы хотите, чтобы сразу после нажатия на клавиши «Видео 1» или «Видео 2» автоматически начиналась демонстрация видео-файла, то в процедуру Command1_Click и Command2_Click необходимо будет дописать четвертую строку:

```
MMControl1.Command = "Play"
```

Эта строка объекту MMControl1 подаст команду на воспроизведение загруженного в него файла. Ее выполнение равносильно нажатию на кнопку «Воспроизведение» объекта MMControl1.

А теперь познакомимся еще с одним полезным объектом типа таймер. Во время работы приложения данный объект остается «невидимым». Таймер в приложении служит для точного отсчета промежутков времени. Минимальный интервал времени может составить одну миллисекунду. По истечении заданных промежутков времени можно заставить приложение выполнять определенные действия. Мы используем таймер для вывода показаний часов компьютера. Для размещения таймера нажмите на кнопку с его изображением в окне объектов и поместите таймер в любой части фор-

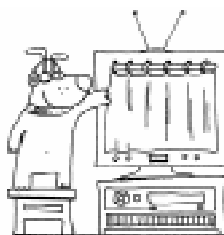


мы Form1, свободной от других объектов. В окне свойств этого объекта для свойства Interval (Интервал) установите значение 1000, что будет соответствовать одной секунде. На форме создайте этикетку Label1. Ее свойству Caption (Заголовок) придайте пустое значение. Измените белый цвет фона этикетки – пусть он будет серым. В процедуру Timer1_Timer запишите программный код:

```
Label1.Caption = Time  
    \ показать текущее время
```

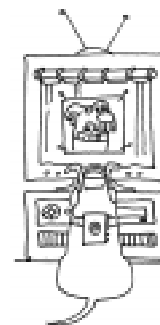
Запустив приложение, вы увидите показания часов, которые будут обновляться каждую секунду. Закройте приложение. Откомпилируйте программу и не забудьте сохранить ее проект.

А теперь изготовьте экранную заставку для Windows. Создайте папку Lesson42. Сюда поместите файлы 42.bmp, 43.mid и Mci.vbx. Заставка должна быть темной, и ее изображение должно постоянно изменяться



или передвигаться по экрану. Ведь основное назначение заставки – защита экрана дисплея от излишней яркости. Форму Form1 «выкрасите» в черный цвет. Ее свойству WindowState (Статус

окна) придайте значение 2-Maximized (Максимальное), а свойству BorderStyle (Граничный стиль) значение 0-None (Нет), для того чтобы приложение-заставка заняло всю площадь экрана и исчезла рамка окна вместе со стандартными кнопками управления. На форму Form1 поместите объект Image1 (Образ). Используя свойство Picture (Рисунок) этого объекта, поместите на форму фотографию Альберта Эйнштейна из файла 42.bmp. Для объекта Image1 свойству Stretch (Изменение размера) придайте значение True (Истина). В процедуры прерывания Form_Click для формы Form1 и Image1_Click для объекта Image1 (Образ) запишите одинаковую строку:



End \ закрытие работы приложения

Она необходима, для того чтобы в любой момент пользователь смог нажатием на мышью убрать с экрана заставку. Ведь кнопка закрытия приложения после удаления рамки окна стала недоступной. Будет нелишним, если Вы предусмотрите выход из заставки, назначив для этой цели клавиши «Esc», «Enter» и «Пробел» на тот случай, если компьютер не оснащен мышью. В процедуру прерывания Form_KeyPress впишите программный код:

```
If KeyAscii = 27 Then End
If KeyAscii = 13 Then End
If KeyAscii = 32 Then End
```

Первая строка при условии нажатия на клавишу «Esc» завершает работу приложения. Две следующие строки работают аналогично для клавиш «Enter» и «Пробел».

Если Вы захотите в своих приложениях задействовать другие клавиши, то напишите программу «Коды клавиш», описание которой можно найти в разделе «После всех уроков». Для назначения клавиш используйте, наряду с событием KeyPress (Нажатие на клавишу), такие события, как KeyDown (Клавиша вниз) и KeyUp (Клавиша вверх).

Для объекта Image1 свойству Visible (Видимость) придайте значение False (Ложь), для того, чтобы в первое время работы приложения изображение Эйнштейна было невидимым. Для управления объектом Image1 на форме разместите таймер Timer1 и для его свойства Interval (Интервал) установите значение 2000 (две секунды). В процедуру прерывания Timer1_Timer впишите программный код:

```
m = m + 1 \ счетчик
Image1.Visible = False
\ скрывает фотографию
Image1.Height = Image1.Height * 1.5
\ увеличение высоты
Image1.Top = 5000 - Image1.Height / 2
\ смещение вверх
Image1.Width = Image1.Width * 1.5
\ увеличение ширины
```

```
Image1.Left = 5000 - Image1.Width / 2
\ смещение влево
Image1.Visible = True
\ показывает фотографию
If m = 9 Then \ условие
Image1.Visible = False
\ скрывает фотографию
Image1.Height = 1000
\ устанавливает начальное
\ значение высоты
Image1.Width = 1200
\ устанавливает начальное
\ значение ширины
m = 0 \ обнуляет счетчик
End If \ конец условия
```

В окне для ввода программного кода для формы Form1 в общем разделе (general) запишите строку:

```
Dim m \ резервирование памяти
\ для переменной
```



Эта строка необходима для того, чтобы зарезервировать область памяти для переменной *m*. Переменная *m* служит для организации счетчика $m = m + 1$. Переменная *m* наращивается на единицу через каждые две секунды. В зависимости от значения переменной *m* работают операторы в процедуре прерывания Timer1_Timer.

В процедуру прерывания Form_Load запишите строки, устанавливающие начальные высоту и ширину фотографии при загрузке приложения:

```
Image1.Height = 1000
Image1.Width = 1200
```

Если хотите, чтобы работа заставки сопровождалась мелодией, подберите подходящий для этого случая файл с расширением MID. Например, файл 43.mid. Добавьте в проект Вашего приложения файл Mci.vbx, так как Вы это сделали в самом начале данного урока для первого приложения. На форме Form1 разместите объект MMControl1. Превратите его в «невидимку», придав его свойству Visible (Видимость) значение False (Ложь). В процедуру прерывания Form_Load, которая срабатывает во время загрузки приложения, допишите строки, смысл которых Вы уже знаете:

```
MMControll.FileName = "43.mid"
MMControll.Command = "Open"
MMControll.Command = "Play"
```

Для того чтобы файл мелодии воспроизводился «бесконечно», в процедуру прерывания MMControll _Done, которая срабатывает в момент



окончания воспроизведения файла мелодии, запишите строки:

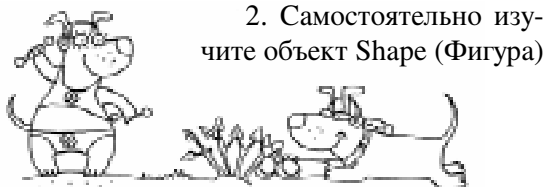
```
MMControll.Command = "Prev"
MMControll.Command = "Play"
```

Первая строка объекту MMControll подаст команду «перемотка назад», с командой второй строки Вы уже знакомы.

Создайте исполняемый EXE-файл, а затем переименуйте его, изменив расширение EXE на расширение SCR. Полученный файл с расширением SCR и библиотеку динамических связей Vbrun300.dll скопируйте в папку Windows. Для того чтобы заставка шла в сопровождении мелодии, ее файл с расширением MID и файл его поддержки Mci.vbx необходимо будет также скопировать в папку Windows.

После урока

1. Создайте приложение, в котором будут присутствовать пять видео-файлов, так чтобы при запуске этого приложения автоматически началась демонстрация первого файла и через две секунды после ее завершения запускался следующий файл. При завершении демонстрации последнего файла приложение также автоматически останавливало бы свою работу.



2. Самостоятельно изучите объект Shape (Фигура)

и его свойства. Подумайте, как применить этот объект в ранее созданных Вами приложениях.

3. Используйте свойство Stretch (Изменение размера) объек-



та Image (Образ) для приложения, в котором автомобиль с прицепом, доехав до края окна формы, останавливается и начинает деформироваться в горизонтальном направлении, а затем, восстановив прежние размеры, откатывается назад.

4. Озвучьте в выше указанном приложении столкновение автомобиля подходящим файлом с расширением WAV. Мультимедийный объект MMControl должен быть невидимым.

УРОК 5:

Меню в Вашей программе.

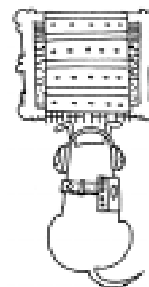
Управление принтером.

Мышь в роли карандаша.



Создайте папку Lesson51. Поместите в нее пять произвольных файлов рисунков с расширениями BMP и WMF. Например, файлы 51.wmf, 52.wmf, 51.bmp, 52.bmp, 53.bmp. Желательно, чтобы размеры рисунков были небольшими, около 200x150 точек. Запустите Visual Basic. Увеличьте размеры формы Form1 примерно до половины общей площади экрана. Цвет формы должен быть белым. Разместите на форме все выбранные рисунки. Рисунки можно накладывать друг на друга, лишь бы они уместились в окне приложения. В окне Properties (Свойства) для всех рисунков свойству Visible (Видимость) придайте значение False (Ложь).

Чтобы построить систему меню в приложении, откройте окно разработки меню (Menu Design Window) (рисунок 2). Для этого на панели инструментов выберите Window (Окно), а затем Menu Design (Разработка меню) или нажмите сочетание клавиш <Ctrl + M>. В появившемся окне в текстовом блоке Caption (Надпись) запишите название для меню Вашего приложения, например, «Рисунки». А в текстовом блоке Name (Имя)



запишите имя меню в программе, например, Pictures. Это и есть основные установки при создании системы меню. Затем щелкните мышью по кнопке Next (Далее) для создания следующего пункта меню. Впишите для него название «1» и имя в программе Pict1. Щелкните мышью по кнопке «стрелка вправо», для того чтобы эту строку превратить в команду для меню «Рисунки». Аналогичным образом создайте следующие команды меню. Для утверждения построенного меню щелкните мышью по кнопке «Ok».



А теперь необходимо провести обработку выборов меню. Если Вы запустите приложение, то команды меню работать не будут, так как они пока являются пустыми. Наполним их программным кодом. Для чего на форме Вашего приложения нажмите на меню «Рисунки». Тут же выпадет группа команд. Нажмите на команду «1». Появится окно программного кода для процедуры Pict1_Click. В эту процедуру прерывания впишите программный код:

```
Picture1.Visible = True
Picture2.Visible = False
Picture3.Visible = False
Picture4.Visible = False
Picture5.Visible = False
```

Нажмите на команду «2» и в процедуру прерывания Pict2_Click впишите программный код:

```
Picture1.Visible = False
Picture2.Visible = True
Picture3.Visible = False
Picture4.Visible = False
Picture5.Visible = False
```

Аналогичным образом заполните программным кодом процедуры прерывания для оставшихся трех команд меню. Запустите приложение. Посмотрите, как оно работает, поочередно вызывая посредством меню каждый из пяти рисунков. Закройте приложение.

Расширим возможности меню данного приложения. Вызовите окно Menu Design (Разработка меню). Нажмите на свободную

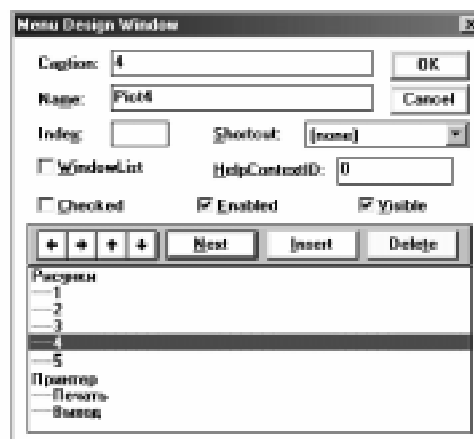
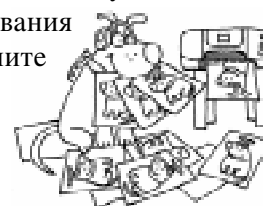


Рисунок 2. Окно разработки меню.

строку, следующую за пунктом меню «5». В текстовом блоке Caption (Надпись) запишите название для следующего меню приложения, например, «Принтер». А в текстовом блоке Name (Имя) запишите имя меню в программе, например, Printer. Щелкните по кнопке Next (Далее) и по кнопке «стрелка вправо». В текстовом блоке Caption (Надпись) запишите название для команды второго меню «Печать». А в текстовом блоке Name (Имя) запишите имя меню в программе, например, Printer. Создайте следующую команду второго меню, для которой в текстовом блоке Caption (Надпись) запишите название для команды «Выход», а в текстовом блоке Name (Имя) запишите имя меню в программе, например, Exit. Для утверждения построенного меню щелкните мышью по кнопке «Ok». В меню «Принтер» нажмите на команду «Печать» и в процедуру прерывания Printer_Click запишите программный код, который будет печатать на бумаге содержимое окна формы:



Form1.PrintForm
 \ печать формы на бумагу



В меню «Принтер» нажмите на команду «Выход» и в процедуру прерывания Exit_Click запишите программный код:

```
End \ закрытие работы приложения
```



Рисунок 3.

Запустите приложение. Выберите один из рисунков и распечатайте его на бумаге. Насладившись работой в приложении, закройте его с помощью команды «Выход» (рисунок 3). Сохраните проект приложения и создайте исполняемый EXE-файл.

А теперь изготовьте приложение для своих домочадцев, в котором они смогли бы порисовать. Создайте папку Lesson52. Запустите Visual Basic. В процедуру прерывания Form_MouseDown запишите строку:

```
Line - (x, y) ' рисование линии
```

Запустите приложение. Попробуйте рисовать. Остановите работу приложения. Замените программный код процедуры Form_MouseDown строками:

```
If Button And 1 Then Line - (x, y) ' рисование линии
If Button And 2 Then PSet (x, y) ' рисование точки
```



Теперь правой кнопкой мыши можно рисовать точки, а левой – линии, а главное при таком не-

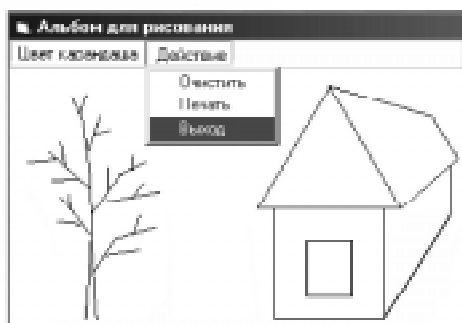
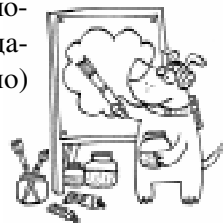


Рисунок 4.

большом усложнении программы можно начать рисовать из любой точки окна приложения. Впрочем, попробуйте изобразить дом и дерево на некотором расстоянии друг от друга (рисунок 4).

Для вызова в Вашем приложении набора цветных карандашей постройте меню «Цвет карандаша». В меню Window (Окно) выберите Menu Design (Разработка меню). В появившемся окне Menu Design Window (Окно разработки меню) запишите в текстовом блоке Caption (Надпись) название будущего меню «Цвет карандаша». Текстовый блок Name (Имя) заполните названием будущей процедуры Color. Указателем мыши щелкните по кнопке Next (Далее) и по стрелке «Вправо».



В текстовый блок Caption впишите «Синий», а в текстовый блок Name запишите внутреннее имя меню – Blue. Аналогичным образом дополните еще два цвета: Red – «Красный» и Green – «Зеленый». Для утверждения построенного меню щелкните по кнопке «Ok». Для оживления меню наполните его программным кодом. В окне формы Form1 нажмите на меню «Цвет карандаша», а затем на команду «Синий». В появившемся окне программных утверждений в процедуру Blue_Click впишите строку:

```
ForeColor = RGB(0, 0, 255) ' синий цвет
```

В процедуру Red_Click поместите строку:

```
ForeColor = RGB(255, 0, 0) ' красный цвет
```

А в процедуру Green_Click поместите строку:

```
ForeColor = RGB(0, 255, 0) ' зеленый цвет
```



Свойство ForeColor (Передний цвет) принимает различные значения путем смешивания трех основных цветов – красный, зеленый, синий. Каждый из основных цветов задается целым числом в пределах от 0 до 255.

Запустите приложение. Порисуйте в нем, используя различные цвета.

Расширим возможности меню. Вызовите окно разработки меню. Нажмите указателем мыши на свободную строку, следующую за строкой «Зеленый». В текстовом блоке Caption (Надпись) запишите название для следующего меню приложения, например, «Действие». А в текстовом блоке Name (Имя) запишите имя меню в программе, например, Action. Щелкните мышью по кнопке Next (Далее) и по кнопке «стрелка вправо». В текстовом блоке Caption (Надпись) запишите название для команды второго меню «Очистить». А в текстовом блоке Name (Имя) запишите имя меню в программе, например, CleSc. Утвердите меню, щелкнув мышью по



кнопке «Ок». На форме приложения нажмите на меню «Действие» и «Очистить». Появится окно программного кода для процедуры CleSc_Click. В эту процедуру впишите строку, состоящую из одного оператора отвечающего за очистку окна приложения:

```
Cls          \ очистка экрана
```

Поместите в меню команду, которая позволит пользователю распечатывать его рисунки на бумагу, так как это Вы сделали в предыдущем приложении. Отличие будет заключаться в том, что для свойства формы AutoRedraw (Автоматическое перерисовывание) Вам необходимо будет установить значение True (Истина). В противном случае рисунок не появится на бумаге! Завершите построение меню командой «Выход». Свойству формы Caption (Надпись) придайте значение «Альбом для рисования». Проверьте работу приложения. Сохраните вновь созданный проект и создайте для него исполняемый EXE-файл. Предложите кому-нибудь порисовать в Вашем приложении.

После урока

1. Создайте приложение, из которого можно будет запускать несколько видео-файлов с помощью меню.

2. Продолжите работу над расширением меню вышеуказанного приложения:

сделайте так, чтобы из меню можно было вызвать показания часов компьютера, вывести текущую дату.



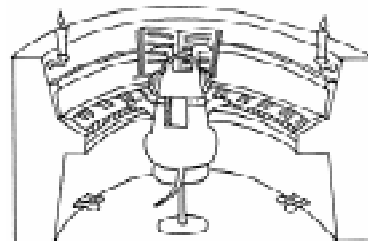
3. Расширьте цветовую палитру в приложении «Альбом для рисования».

4. Сделайте так, чтобы работу в «Альбоме для рисования» сопровождала спокойная мелодия. Мультимедийный объект MMControl должен быть невидимым.

5. В меню «Альбома для рисования» поместите команду «О программе», которая будет вызывать отдельное окно, содержащее сведения об авторе.

ПОСЛЕ ВСЕХ УРОКОВ

Полезная для программиста программа «Коды клавиш»



Если Вы хотите в своих приложениях шире использовать возможности клавиатуры, то создайте простейшее приложение для определения кода клавиши. Для этого процедуру прерывания Form_KeyPress заполните программным кодом, состоящим всего из одной строки:

```
Print KeyAscii
\ печать на экран
\ кода нажатой клавиши
```

Запустите приложение. При нажатии, например, на клавишу «Enter» получите значение кода этой клавиши – 13, а при нажатии на клавишу «d» – 100. При одновременном нажатии на клавиши «Shift» и «d» получите 68.

В указанную выше процедуру прерывания на первое место желательно записать строку, состоящую из оператора «Очистка экрана». Как назначать клавиши в приложениях, Вы узнали из урока № 4.

Если Вы решили заняться программированием

Напишите приложение «Универсальный прослушивающий просмотрщик». Эта программа позволит пользователю:



1. Просматривать файлы рисунков в форматах BMP, WMF.
2. Просматривать видео-файлы в формате AVI.
3. Прослушивать музыкальные файлы в форматах MID, RMI.
4. Прослушивать звуковые файлы в формате WAV.

Создайте папку Browser, в которую позднее поместите проект этого приложения. В проект добавьте файл Mci.vbx (поддержка мультимедийных объектов) и файл Cmdialog.vbx (поддержка окна общих диалогов). На форме Form1 расположите объекты Image1 (Образ), MMControl1 (Мультимедийные объекты) и CMDialog1 (Окно общих диалогов). Объект MMControl1 сделайте «невидимым».

Объект CMDialog1 служит для вызова во время работы программы стандартных диалоговых окон:

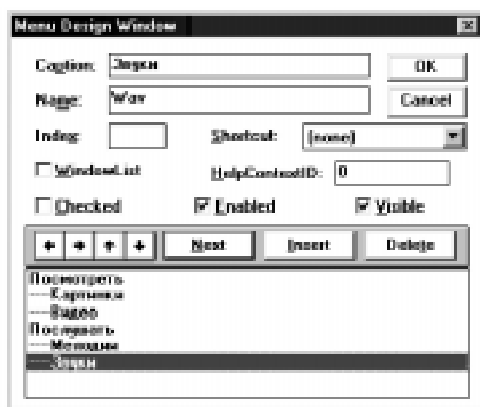


Рисунок 5.

1. Открытие файла.
2. Сохранение файла.
3. Выбор цвета.
4. Выбор шрифта.
5. Настройка принтера.

Например, для вызова окна общего диалога «Выбор цвета» необходимо использовать строку программного кода:

```
CMDialog1.Action = 3
  \ активизация окна ВЫБОР ЦВЕТА
```

В данном приложении объект CMDialog1 Вы будете использовать для активизации окна «Открытие файла».

Сначала постройте в приложении систему меню (рисунок 5) по следующей схеме:

Caption (Надпись)	Name (Имя в программе)
Посмотреть	Look
---Картинки	Pictures
---Видео	Video
Послушать	Listen
---Мелодии	Mid
---Звуки	Wav

Вы помните, что надписи в меню и соответствующие им имена процедур прерывания в программе могут быть произвольными. Во вновь созданную процедуру прерывания Pictures_Click поместите программный код:

```
CMDialog1.FileName = "*.bmp;*.wmf"
  \ указание типов открываемых файлов
CMDialog1.Action = 1
  \ активизация окна диалога
  \ ОТКРЫТИЕ ФАЙЛА

Image1 =
  LoadPicture(CMDialog1.FileName)
  \ загрузка рисунка в объект Image1

Form1.Width = Image1.Width + 120
  \ установление ширины окна формы
Form1.Height = Image1.Height + 720
  \ установление высоты окна формы

Form1.Caption =
  FileLen(CMDialog1.FileName) & " байт "
  \ информация о длине файла
Form1.Caption =
  Form1.Caption + CMDialog1.FileName
  \ добавление пути и имени файла

MMControl1.Visible = False
```


Данная процедура прерывания вызовется тогда, когда пользователь в меню «Посмотреть» выберет команду «Картинки». Пустые строки в программе совершенно необязательны. В данном случае они используются для наглядности и лучшей удобочитаемости текста программного кода. Первая строка указывает объекту общего диалога CMDialog1 разрешенные форматы для открываемых файлов – BMP и WMF. Знак «*» указывает на то, что имя файла может быть любым. Вторая строка вызывает стандарт-

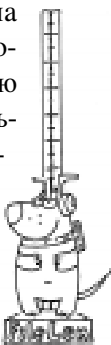


ное окно общего диалога «Открытие файла», посредством которого пользователь отыскивает на нужном диске и в нужной папке необходимый файл. Третья строка загружает файл рисунка, выбранный пользователем, и помещает его в объект Image1 (Образ), расположенный на форме Form1. Это возможно с помощью оператора LoadPicture (Загрузка рисунка).

Следующие две строки приводят в соответствие размеры окна приложения и размеры помещаемого в это окно рисунка.

Шестая строка программного кода посредством оператора FileLen (Длина файла) устанавливает размер выбранного файла и помещает эту информацию в строку заголовка формы Form1. Седьмая строка добавляет в строку заголовка информацию о пути и имени этого файла. Как работают последние две строки, Вы уже знаете.

Желательно в стандартную процедуру прерывания Image1_Click полностью скопировать вышеприведенный программный код. Тогда пользователю при последующих обращениях к окну общего диалога «Открытие файла» будет необязательно прибегать к услугам меню. Достаточно будет щелкнуть мышью по полю уже вызванного рисунка, как вновь появится окно общего диалога «Открытие файла».



Продолжим работу по обработке выборов меню. Процедуру прерывания Video_Click заполните программным кодом:

```

CMDialog1.FileName = «*.avi»
  \ указание типа открываемых файлов
CMDialog1.Action = 1
  \ активизация окна диалога
  \ ОТКРЫТИЕ ФАЙЛА
MMControll1.FileName =
      CMDialog1.FileName
  \ передача имени файла
MMControll1.Command = "Close"
MMControll1.Command = "Open"
MMControll1.Command = "Play"
MMControll1.Visible = True

Form1.Caption =
  FileLen(CMDialog1.FileName) & " байт "
Form1.Caption = Form1.Caption +
  CMDialog1.FileName

```

```

Form1.Width = 5805
Form1.Height = 3780

```

Третья строка имя файла, выбранное пользователем в общем окне диалога «Открытие файла», передает объекту MMControll1.



Смысл остальных строк Вам уже должен быть понятен. Данная процедура прерывания Video_Click сработает при выборе пользователем в меню «Посмотреть» команду «Видео».

Завершите обработку выборов меню. Для двух оставшихся процедур прерывания Mid_Click и Wav_Click потребуется такой же программный код, как и для процедуры Video_Click. Изменения коснутся только первой строки. Для процедуры Mid_Click строка примет вид:

```

CMDialog1.FileName = "*.mid"
  \ указание типа открываемых файлов
      А для процедуры Wav_Click:
CMDialog1.FileName = "*.wav"
  \ указание типа открываемых файлов

```

Свойству Caption (Надпись) объекта Form1 придайте значение «Универсальный прослушивающий просмотрщик». Создайте исполняемый EXE-файл. Для успешной ра-

боты приложения в его папку наряду с исполняемым файлом должны присутствовать файлы Vbrun300.dll, CMDialog.vbx и Mcl.vbx. Опробуйте приложение в работе.

Самостоятельно доработайте приложение:

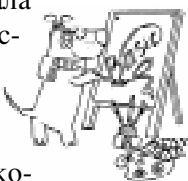
1. В систему меню поместите команду «Выход».

2. Добавьте в приложение второе окно, содержащее помощь по работе с приложением. Сделайте так, чтобы это окно появилось при выборе в меню команды «Помощь».

3. Добавьте в приложение третье окно, содержащее сведения о Вашем авторстве на программу. Это окно пользователь должен вызывать из меню командой «О программе».

4. Сделайте так, чтобы сразу после загрузки приложения начинала звучать выбранная на Ваше усмотрение мелодия и в окне Form1 появлялся какой-либо рисунок или фотография.

5. В меню поместите команду «Печать» с помощью, которой пользователь смог бы распечатывать просматриваемые рисунки на бумагу.



Если Вы хотите расширить возможности палитры приложения «Альбом для рисования»



Запустите файл проекта приложения «Альбом для рисования» с расширением MAK. Вызовите окно разработки меню (Menu Design Window). Воспользовавшись кнопкой Delete, уда-

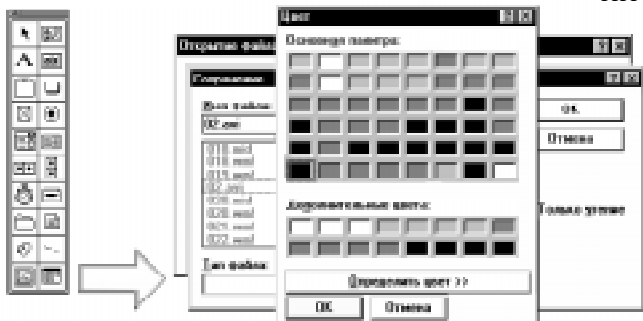


Рисунок 6.

лите из системы меню приложения три ранее созданные Вами процедуры прерывания: Blue_Click, Red_Click и Green_Click. В процедуру прерывания Color_Click впишите программный код:

```
CMDialog1.Action = 3
    \ активизация окна диалога
    \ ВЫБОР ЦВЕТА
ForeColor = CMDialog1.Color
    \ передача выбранного цвета
```

Во второй строке свойству ForeColor (Передний цвет) для объекта Form1 присваивается значение выбранного цвета в окне общего диалога «Выбор цвета» (рисунок 6). Теперь количество цветов для рисования в данном приложении будет ограничено только возможностями видеосистемы Вашего компьютера. Как правило, это более шестнадцати миллионов системных цветов.

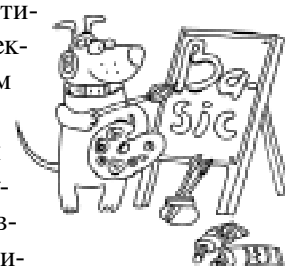


Откомпилируйте проект приложения. Проверьте полученный EXE-файл в работе.

Самостоятельно изготовьте в данном приложении пункт меню с названием «Цвет бумаги» для изменения цвета фона рисования. Для этой цели используйте свойство BackColor (Цвет фона) объекта Form1 (Форма).

Чтение, создание, запись и печать текстовых файлов

Создайте папку с произвольным названием, например, Write. Запустите Visual Basic. В проект будущего приложения добавьте файл Cmdialog.vbx (поддержка окна общих диалогов). На форме Form1 разместите объекты Text1 (Текстовое поле), CMDialog1 (Окно общего диалога) (рисунок 7). Для объекта Text1 свойству MultiLine (Много строк) придайте значение True (Истина) – теперь в тексте, вводимом пользователем, по достижении конца строки будет осуществляться автоматичес-



ческий перенос по словам. Свойству Text (Текст) этого объекта придайте значение пустого множества, а свойству ScrollBars (Полоса прокрутки) значение 2-Vertical (Вертикально). Вызовите окно разработки меню (Menu Design Window) и постройте систему меню по следующей схеме:

Caption (Надпись)	Name (Имя в программе)
Файл	File
---Новый	New
---Открыть	Open
---Сохранить	Save
---Печать	Print
---Выход	End

Проведите обработку выборов меню – вновь созданные процедуры наполните программным кодом. В процедуру прерывания New_Click впишите строку:

```
Form1.Text1.Text = ""
    \ очистка текстового поля
```

В этой строке очистка текстового поля от ранее введенного текста происходит путем присвоения свойству Text (Текст) объекта Text1 значения, соответствующего пустому множеству.

В процедуру прерывания Open_Click запишите программный код:

```
CMDialog1.FileName = "*.txt"
    \ указание типа открываемых файлов
CMDialog1.Action = 1
    \ активизация окна диалога
    \ ОТКРЫТИЕ ФАЙЛА
```

```
Form1.Caption = CMDialog1.FileName
    \ показ пути и имени файла
```

```
Open CMDialog1.FileName For Input As #1
    \ открытие файла для чтения
Form1.Text1.Text = Input$(LOF(1), 1)
    \ чтение файла в текстовое поле
Close #1
    \ закрытие файла
```

В четвертой строке происходит открытие выбранного файла для чтения под номером «Первый». Пятая строка считывает содержимое открытого файла в текстовое поле

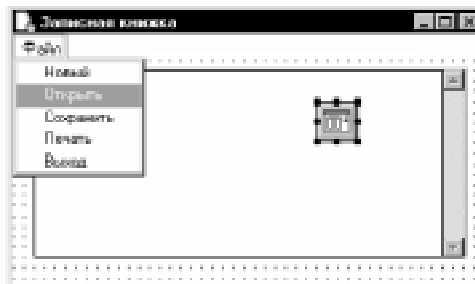
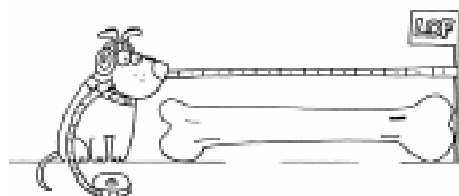


Рисунок 7.

объекта Text1. Здесь инструкция LOF определяет длину файла, открытого под номером



«Первый». Шестая строка необходима для закрытия файла под номером «Первый».

В процедуру прерывания Save_Click впишите строки:

```
CMDialog1.Action = 2
    \ активизация окна диалога
    \ СОХРАНЕНИЕ ФАЙЛА
If CMDialog1.FileName = ""
Then Exit Sub
    \ если имя файла не указано,
    \ то выход
Form1.Caption =
CMDialog1.FileName & ".txt"
    \ показ пути и имени файла
Open CMDialog1.FileName & ".txt"
    For Output As #1
        \ открытие файла для записи
Print #1, Form1.Text1.Text
        \ запись текста в файл
Close #1
    \ закрытие файла
```

Во второй строке программа защищена от возможной ошибки в четвертой строке. Такая ошибка может возникнуть в том случае, если пользователь не указал имя сохраняемого файла. Вторая строка осуществляет выход из процедуры Save_Click, если именем файла является пустое множество. Третья строка помещает информацию о пути

и имени сохраняемого файла в строку заголовка окна формы Form1. В четвертой строке открывается файл для записи под номером «Первый». Пятая строка записывает в открытый файл содержимое текстового поля. Последняя строка закрывает файл под номером «Первый».

В процедуру прерывания Print_Click впишите программный код:

```
Printer.FontName = "Arial Cyr"  
    \ выбор вида шрифта для печати  
Printer.FontSize = 12  
    \ установка размера шрифта  
    \ для печати  
Printer.Print ""  
    \ инициализация объекта принтер
```

```
Printer.Print Form1.Text1.Text  
    \ вывод текста на бумагу  
Printer.EndDoc  
    \ остановка печати документа
```

Самостоятельно завершите обработку выборов меню для знакомой Вам команды «Выход». Сохраните проект приложения. Создайте исполняемый EXE-файл.

При желании в меню добавьте команды, позволяющие пользователю менять размер шрифта как для набора текста, так и для печати.



Наши авторы, 2003.
Our authors, 2003.

*Коробицин Владимир Михайлович,
учитель физики и информатики
Чернолученской средней школы,
Омская область.*