СЦЕНАРИИ УРОКОВ

Коробицин Владимир Михайлович

ЗНАКОМСТВО С МУЛЬТИМЕДИА В VISUAL BASIC. УРОКИ 1–3

Автор предлагает вниманию читателей небольшое электронное пособие для знакомства с языком программирования Visual Basic версии 3.0. Данное пособие может быть использовано как для самостоятельного изучения Visual Basic, так и в качестве помощника учителя на уроках информатики или факультативных занятиях.

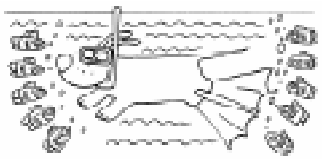
Принимая в расчет богатый иллюстративный материал и несложные языковые конструкции рассмотренных в данном пособии приложений, это пособие можно рекомендовать и для работы с младшими школьниками.

Читатели же, знакомые с циклом журнальных статей по Visual Basic в № 1–5 за 2001 год, смогут отобрать для себя дополнительный материал под условным названием «Знакомство с мультимедиа в Visual Basic-приложениях». Автору хорошо удалось объединить в своем пособии звук, музыку, графику, компьютерную анимацию и видео, а также продемонстрировать их несомненно важное значение в развитии информационных технологий. Данная мультимедиа-тема, будучи привлекательной для школьников любого возраста, безусловно поможет закрепить интерес к дальнейшему изучению Visual Basic.

Н.Н. Паньгина

ВВЕДЕНИЕ

Это пособие для тех, кого завораживает простота и элегантность Windows-приложений и кто захотел попробовать себя в искусстве программирования, а также для тех, кто писал программы под DOS в каком-либо из Бейсиков и, наконец, решился создать приложение для Windows. Тогда Вам пора окунуться с головой в среду объектно-ориентированного языка Microsoft Visual Basic. Остановимся на третьей, не самой поздней его версии, которая отличается относительной простотой и обладает достаточно широкими возможностями. Кроме того, эта версия не потребует от компьютера больших ресурсов. После того как Вы освоите данную версию, можно смело начинать работу с последующими версиями Visual Basic. Там Вы столкнетесь с более развитым интерфейсом и с еще более широкими возможностями языка. Мы поставили перед собой скромную задачу: помочь не утонуть Вам в мощной и глубоководной среде Visual Basic. После пяти уроков Вы освоите простейшие и самые распространенные приемы создания пусть небольших, но зато построенных по Вашему усмотрению и вкусу приложений для Windows. Вы будете использовать некоторые объекты языка и заниматься написанием простейшего программного кода для управления выбранными объектами. Предлагаемые приложения одинаково уютно себя чувствуют при создании и использовании как в Windows 95, 98, так и в Windows 2000.



Установите на Ваш компьютер Visual Basic третьей версии. Если у Вас нет такой возможности, то скопируйте на жесткий диск минимум файлов из этой версии: Vb.exe, Vbrun300.dll, Mci.vbx, Cmdialog.vbx и Vb.lic. Этих файлов будет достаточно для работы с данным пособием.

Итак, включайте компьютер, открывайте урок первый и за дело!

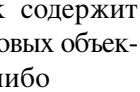
ДО УРОКОВ

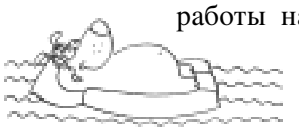
Не судите строго. Перед Вами не учебник по основам Visual Basic, а практическое руководство для самого первого шага в его среду.

Нетерпеливые могут этот начальный кусочек информации перед самым первым уроком пропустить. Но помните, что все равно после какого из уроков Вы непременно должны вернуться к данной части пособия, которая поможет Вам осмыслить основные принципы работы Visual Basic и понять, о каком ничтожном круге вопросов идет речь в данном пособии.

Среда Visual Basic обладает свойством визуальности за счет своего добротного графического интерфейса. Это значит, что процесс программирования в этой среде является предельно наглядным. Даже те объекты, которые Вы не видите в работе приложения, отображаются в среде Visual Basic на экране дисплея в процессе создания приложения.

В отличие от алгоритмических языков программирования, интегрированная среда программирования Microsoft Visual Basic является объектно-ориентированной, так как содержит большое число разнообразных готовых объектов. Задавая свойства какого-либо объекта, Вы сможете изменить его до неузнаваемости, а если это нужно, даже заставить данный объект исчезнуть с экрана дисплея в любой момент времени. Свойства объекта можно задавать до запуска программы и изменять во время ее работы. Благодаря использованию объектов, программист может очень быстро и без больших усилий создавать эффектные и много делающие программы, причем объем программного кода может уменьшиться в десятки раз. В это же время можно использовать бесконечные возможности разнообразных конструкций алгоритмического языка, особенно когда Ваша программа нуждается в сложной логике. Таким образом, Visual Basic, как, впрочем, и другие объектно-ориентированные языки программирования, освобождают программиста от рутинной работы над созданием удобного для пользователя интерфейса, хорошего рисунка или звукового фрагмента. В результате высвобождается масса времени для творческой работы над приложением. На рисунке 1 Вы видите некоторые объекты Visual Basic.





Объект Visual Basic – это кирпичик пользовательского интерфейса, который можно расположить на форме. Форма же, в свою очередь, тоже является объектом, причем самым главным, если приложение отображается на экране. Визуально форма выглядит как правило в виде стандартного окна Microsoft Windows. Любой из объектов создается с помощью средств управления. Программист в Visual Basic уподобляется строителю, которому не нужно перед строительством замка изготовлять кирпичи различной формы и различного цвета, так как они уже кем-то заранее

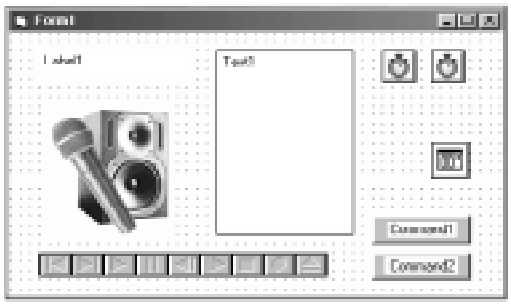


Рисунок 1.

заготовлены и хранятся в среде Visual Basic. Программист при помощи специальных инструментов и программного кода выкладывает здание замка. Такие кирпичики никогда не иссякнут, сколько бы замков Вы ни построили, причем самых разных по очертаниям и красоте. Удивительно, но объект – это часть Вашей программы – часть, которая вылеплена до Вас при помощи того же программного кода.

Рисунок 2. Окно инструментов.

В основе работы приложения для Windows лежит принцип управления по прерываниям. Каждый объект в среде Visual Basic имеет фиксированный набор событий. Для любого из этих событий можно написать процедуру обработки прерывания. В процедуру записываются одна или несколько команд, подчиняясь которым компьютер должен выполнить строго определенные действия. Если какое-либо событие, вызванное пользователем или временем, происходит, то Visual Basic тут же выполняет вполне определенную процедуру обработки прерывания, которая содержит написанный Вами программный код.

УРОК 1:

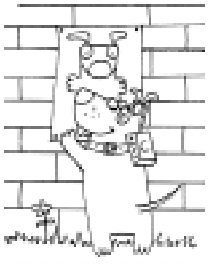
Размещение картинки в программе

Оживление картинок

Изготовление электронной кнопки

Создание исполняемого EXE-файла

На одном из дисков вашего компьютера, например, на диске C, создайте папку с произвольным названием, допустим Lesson11. Запустите Visual Basic. В средней части экрана Вы увидите окно формы с именем Form1. Это и есть тот самый фундамент Вашего будущего приложения для Windows.



Для начала на форму поместите рисунок. В левой части экрана расположено окно инструментов (рисунок 2). Если Вы его не видите, выберите в меню Window (Окно) команду Toolbox (Окно инструментов). Наведите указатель мыши на значок «Рисунок», который расположен в окне инструментов, и щелкните по нему левой кнопкой мыши. Затем поместите указатель мыши на свободное поле окна формы Form1 в его верхнем левом углу. Нажмите на левую кнопку мыши и, не отпуская ее, переместите указатель в сторону правого нижнего угла, расширяя появившийся прямоугольник до произвольных размеров. Отпустите кнопку мыши.

Таким образом на форму Вы поместили объект PictureBox (Окно рисунка) с именем Picture1 (Рисунок первый). Имена объектов можно изменять. Мы на протяжении всей книги будем использовать те имена объектов, которые будет предлагать для них Visual Basic по умолчанию. В дальнейшем для простоты описания объектов будем называть их по именам, не упоминая общего названия объекта.

В правой части экрана расположено окно Properties (Свойства) (рисунок 3). Если в данный момент окно скрыто, выберите в меню Window (Окно) команду Properties (Свойства) или нажмите клавишу <F4>. В окне Properties (Свойства) щелкните два раза левой кнопкой мыши по надписи Picture (Рисунок). Во вновь появившемся окне Load Picture (Загрузка рисунка) выделите файл рисунка 01.wmf. Щелкните по имени этого файла два раза. Изображение яблока окажется на форме Form1, выделенное маркерами – черными квадратиками небольших размеров. Откорректируйте размеры рисунка на Ваше усмотрение. Для чего ухватите любой из маркеров окна рисунка и перемещайте мышь в нажа-

том состоянии в необходимом направлении. Для перетаскивания рисунка по форме целиком нажмите на поле картинки и, не отпуская кнопку мыши, переместите картинку. Освободите кнопку мыши. Желательно, чтобы изображение яблока расположилось в левой половине формы Вашего приложения. В окне Properties (Свойства) выберите свойство BorderStyle (Граничный стиль) и два

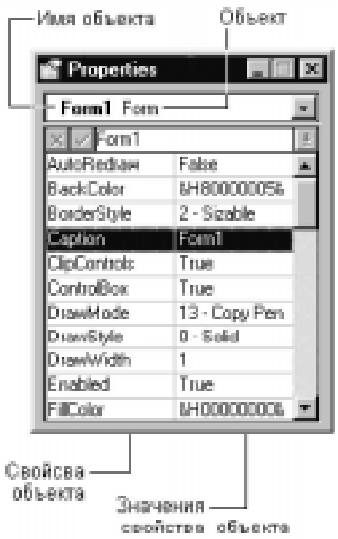


Рисунок 3. Окно свойства объектов.

раза щелкните по его надписи. Данное свойство объекта Picture1 (Рисунок) примет значение 0-None (Нет), при этом граница рисунка исчезнет.

Запустите Вашу программу на выполнение, для чего щелкните один раз левой кнопкой мыши по значку «Запуск программы», расположенному на панели инструментов (рисунок 4) или нажмите на клавишу <F5>. Можно также на панели инструментов выбрать в меню Run (Пуск) команду Start (Старт). Если на панели инструментов не видны кнопки со значками, то в меню View (Просмотр) выберите команду Toolbar (Набор средств). Остановите работу программы, нажав в заголовке приложения на кнопку «Закрыть» или сочетание клавиш <Alt + F4>. Аналогичным образом на форму Вашего приложения поместите второй рисунок, используя теперь файл 02.wmf. Рисунок вишни расположите в правой половине формы. Наведите указатель мыши на

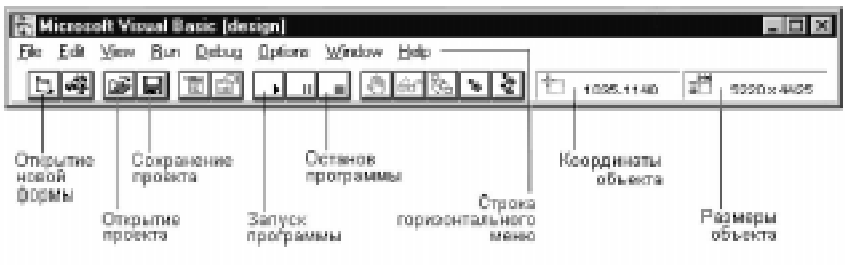
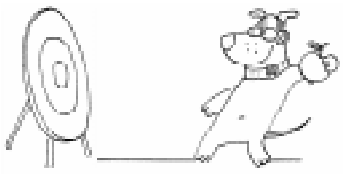


Рисунок 4. Панель инструментов.



изображение яблока и щелкните два раза левой кнопкой мыши – появится окно программного кода с процедурой Picture1_Click. Имя процедуры обычно состоит из имени объекта и имени возможного для данного объекта события. Данная процедура будет вызвана во время работы программы при свершении такого события, как одиночный щелчок мыши по изображению яблока. В эту процедуру прерывания запишите две строки:

```
Sub Picture1_Click ( )
' начало работы процедуры прерывания
Picture1.Visible = False
' скрыть первый рисунок
Picture2.Visible = True
' показать второй рисунок
End Sub
' завершение прерывания
```

Для немедленного вызова окна программных утверждений в меню View (Просмотр) нажмите команду Code (Программный код) (рисунок 5) или клавишу <F7>.

Первая строка для объекта Picture1 (Рисунок) устанавливает его свойству Visible (Видимость) значение False (Ложь). Это программное утверждение сделает яблоко «невидимым». Вторая

Рисунок 5. Око программного кода.

строка покажет вишню. Для объекта Picture2 в процедуру Picture2_Click поместите две строки противоположного действия:

```
Sub Picture2_Click ( )
Picture1.Visible = True
Picture2.Visible = False
End Sub
```

Запустите программу. Поочередно понажимайте левой кнопкой мыши на изображения яблока и вишни. Остановите программу.

Таким образом, управление объектами в процессе работы программы Visual Basic происходит путем изменения их свойств. Для изменения свойства объекта этому свойству присваиваются различные значения в окне для ввода программного кода по формуле: Объект.свойство=Значение

Сначала записывается имя объекта, затем его свойство, причем разделяются они точкой. После указания свойства объекта ставится знак «присвоить значение» и на последнем месте записывается значение свойства объекта. Как видите знак «=» в языках программирования несет совершенно другую смысловую нагрузку, в отличие от его назначения в математике.

Нажмите один раз на свободное от рисунков поле формы Form1 и в окне свойств объекта выберите свойство Caption (Заголовок) и измените надпись строки заголовка окна Form1 на произвольную, например: «Мое первое приложение для Windows». Сохраните Вашу программу в папке Lesson11, для этого в меню File (Файл) выберите команду Save Project As... (Сохранить проект как...). Подтвердите предлагаемые имена файлов Form1.frm (Первое окно приложения) и Project1.mak (Проект приложения). Можно дать другие имена, придерживаясь правил DOS. Типы данных файлов изменять недопустимо.

Откомпилируйте программу, то есть создайте исполняемый EXE-файл. Для этого на панели инструментов выберите меню File, а затем команду Make EXE File... . В процессе компиляции Visual Basic сливает отдельные файлы-компоненты проекта приложения в единый файл с расширением EXE. Компиляция проекта приложения – это процесс рождения самостоятельного EXE-файла. С этого момента дитя выходит из подчинения своего родителя по имени Visual Basic.

Завершите работу в Visual Basic, нажав в его заголовке кнопку «Закрыть» или сочетание клавиш <Alt + F4>. Откройте папку Lesson11. В ней Вы обнаружите файлы: Project1.mak, Form1.frm и Project1.exe. Запустите свое приложение Project1.exe. Поработайте в нем. Закройте приложение. При желании Вы можете переименовать этот файл, но расширение должно остаться прежним – EXE.

А теперь рассмотрим некоторые приемы простейшей анимации. Создайте папку Lesson12 для следующего приложения. Запустите Visual Basic. Окно формы Form1 увеличьте по ширине почти до полной ширины экрана Вашего дисплея, а по высоте уменьшите примерно в два раза. В правую половину формы поместите рисунок из фай-

ла 01.bmp, так чтобы передняя часть автомобиля виднелась из-за правого края формы. Вызовите окно программного кода для этого рисунка и в процедуру Picture1_Click впишите строки:

```
For n = 1 To 500
        ' начало цикла
Picture1.Left = Picture1.Left - 15
        ' передвижение вперед
Next     ' завершение цикла
```

Цикл For-To-Next выполнит строку «Picture1.Left = Picture1.Left – 15» пятьсот раз, что приведет к постепенному передвижению картинки влево, так как координата его по горизонтали каждый раз будет уменьшаться на 15 единиц (или на один пиксель экрана).

Запустите программу. При одиночном щелчке мышью по автомобилю, последний будет перемещаться вперед. Закройте приложение. В процедуру Form_Click запишите:

```
Picture1.Left = 7920
    ' возврат на стартовую позицию
```

Left – это одно из многочисленных свойств объекта Picture1, задающее значение координаты левого края этого объекта. Для изменения расположения по вертикали существует свойство Top, отвечающее за значение верхнего края объекта. Этими свойствами обладают и другие объекты в Visual Basic.

Запустите приложение. Заставьте автомобиль исчезнуть за пределами окна Вашего приложения. Щелкните по свободному полю окна – автомобиль немедленно займет стартовую позицию. Подумайте, как это осуществляет строка в процедуре Form_Click, которая обрабатывается при одиночном щелчке указателем мыши на форму. Вновь приведите автомобиль в движение и верните его в исходное состояние. Закройте приложение. Сохраните проект Вашей программы в папке Lesson12. Создайте исполняемый файл и сохраните его в той же папке. Приложение готово. На следующем уроке Вы озвучите это приложение.

Для вызова окна проекта в меню Window (Окно) выполните команду Project (Проект). В этом окне Вы получите сведения о составе проекта приложения, а при необходимости с помощью кнопок «View Form» (Просмотр формы) и «View Code» (Просмотр программного кода) можете вызвать на экран форму приложения и ее программный код.

Рассмотрим еще один анимационный прием – масштабирование изображения. Для «резиновых» метафайлов с расширением WMF (Электронные картинки) можно использовать объект Picture (Рисунок) или объект Image (Образ), а для файлов точечных рисунков с расширением BMP масштабирование возможно лишь посредством объекта Image (Образ).

Создайте папку Lesson13 для Вашего следующего приложения. Запустите Visual Basic. В окне свойств формы Form1 выберите свойство BackColor (Цвет формы) и установите для него значение цвета – серый. Для этого щелкните два раза по свойству BackColor и в появившемся окне палитры цветов выберите серый цвет.

Для размещения объекта Image1 (Образ) на форме приложения щелкните по значку объектов типа Image (Образ) в окне объектов. Затем нажмите мышкой на поле Вашей формы и, не отпуская кнопки, переместите указатель в произвольном направлении по диагонали. Отпустите мышку. В окне свойства объекта выберите свойство Stretch (Изменение размера) и установите для него значение True (Истина). Здесь же выберите свойство Picture (Рисунок) и щелкните по нему два раза. Через появившееся окно загрузите рисунок из файла 02.bmp в форму Form1.

Рисунок 6.

В окне объектов щелкните по значку «Командная кнопка», затем установите электронную командную кнопку Command1 на форму и подберите подходящие для нее размеры. Данные операции похожи на действия над объектами Picture (Рисунок) и Image (Образ). Щелкните указателем мыши по вновь созданному объекту два раза и в процедуру Command1_Click впишите программные утверждения:

```
Image1.Width = Image1.Width * 1.25
          ' увеличение ширины
Image1.Height = Image1.Height * 1.25
          ' увеличение высоты
```

Первая строка при одиночном щелчке по командной кнопке увеличит значение свойства Width (Ширина) объекта Image1 в 1,25 раза. Вторая строка сработает таким же образом для свойства Height (Высота). В результате размеры изображения увеличатся на 25%.

В окне свойств объекта Command1 выберите свойство Caption (Заголовок) и придайте ему значение «Увеличить», а свойству FontBold (Жирный шрифт) значение False (Ложь). Аналогичным образом создайте еще две командные кнопки: «Уменьшить» и «Выход». Для кнопки Command2 в процедуру Command2_Click впишите строки, которые будут уменьшать размеры фотографии на 25%:

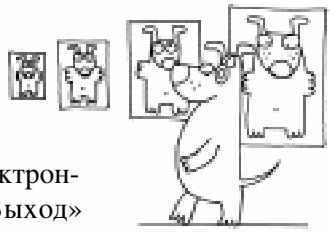
```
Image1.Width = Image1.Width / 1.25
          ' уменьшение ширины
Image1.Height = Image1.Height / 1.25
          ' уменьшение высоты
```

Для кнопки Command3 в процедуру Command3_Click запишите программное утверждение:

```
End    ' завершение работы приложения
```

Это утверждение при одиночном щелчке по командной кнопке «Выход» будет завершать работу Вашего приложения.

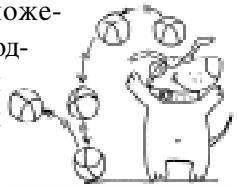
Запустите программу. Поочередно, по нескольку раз, понажимайте на кнопки «Увеличить» и «Уменьшить», наблюдая за изменением размеров фотографии. Для окончания работы приложения щелкните по созданной Вами электронной кнопке «Выход» (рисунок 6).



Сохраните проект приложения и создайте для него исполняемый EXE-файл. Проверьте этот файл в работе.

После урока

1. Изготовьте приложение, в котором мяч будет подпрыгивать вверх и затем возвращаться обратно при нажатии на электронную кнопку.

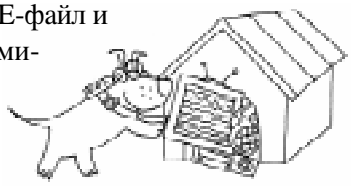


2. Создайте приложение, в котором какой-либо рисунок будет уменьшаться в размерах при нажатии на него указателем мыши и увеличиваться при нажатии на свободное от этого рисунка поле окна формы.

3. У объекта Form1 (Форма) самостоятельно исследуйте его свойство BorderStyle (Граничный стиль), которое может принимать четыре значения от 0 до 3.

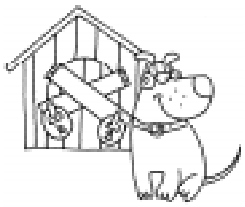
Подготовка приложения для установки на другом компьютере

Создайте папку с произвольным названием, используя только латинские буквы и цифры. Скопируйте в эту папку со-



зданный Вами EXE-файл и библиотеку динамических связей Vbrun300.dll. Если Ваше приложение содержит элементы мультимедиа (звук, музыка, видео, анимация), то в эту же папку необходимо скопировать файл Mci.vbx и мультимедийные файлы с расширениями WAV, MID, RMI, AVI, которые Вы задействовали в программе. Если приложение использует окно общего диалога, необходим будет файл Cmdialog.vbx. Помните, что файлы рисунков с расширениями BMP и WMF копировать не надо, так как Visual Basic в процессе компиляции автоматически помещает их в исполняемый EXE-файл. Если объем папки по размеру окажется больше емкости дискеты, то воспользуйтесь каким-либо архиватором, например WinRAR. Этот архиватор позволит Вам сжать приложение, при необходимости разбить архив на дискеты, а при желании файл архива можно сделать саморазворачивающимся.

Сохранение исходных текстов Ваших программ



Все исходные тексты программ Ваших приложений для Windows необходимо самым тщательным образом сохранять. Они могут понадобиться Вам в тех случаях, если появится желание изменить что-либо в ранее написанном приложении или Вы захотите использовать части этого приложения в своих новых работах. Кроме того, дискета с исходными текстами программ станет своеобразной копилкой Ваших знаний в Visual Basic. Придет время, когда понадобится освежить в памяти Ваши прежние достижения, и тогда не нужно будет начинать все сначала. Для сохранения текстов программ достаточно скопировать на надежный носитель файл проекта Вашего приложения с расширением MAK и файлы форм этого приложения с расширением FRM. Если проект приложения содержит файлы модулей с расширением BAS, то их тоже нужно будет скопировать. При сохранении приложений создавайте отдельную папку для каждого из них. Если Ваше приложение является мультимедийным, то задействованные в нем файлы музыки, видео, звуков поместите в эту же папку.

УРОК 2: Озвучивание Ваших программ Реакция приложения на левую и правую кнопки мыши

Создайте папку Lesson21. Для удобства в работе поместите в нее файлы 05.bmp, 06.bmp, 01.mid, 01.wav, 02.wav, 03.wav, 04.wav, Mci.vbx. Запустите Visual Basic.

Окно формы Form1 «выкрасите» в серый цвет и поместите в него два рисунка в виде объектов Picture (Рисунок) из файлов 05.bmp, 06.bmp. Для обоих рисунков в окне Properties (Свойства) для свойства AutoSize (Авто размер) установите значение True (Истина), а для свойства BorderStyle (Граничный стиль) значение 0-None (Нет). Чтобы озвучить Ваше приложение, необходимо в его проект добавить файл Mci.vbx. Для этого в меню File выберите команду Add File ... (Добавить файл ...) и через появившееся окно загрузите файл Mci.vbx. В окне объектов появится значок мультимедийных объектов MMControl (Microsoft Multimedia Control) (рисунок 7). Файл Mci.vbx необходим в тех случаях, когда Вы хотите, чтобы Ваше приложение не было «немым». Это файл поддержки мультимедийных объектов. Кроме того, компьютер должен быть оснащен музыкальной картой. В окне объектов щелк-

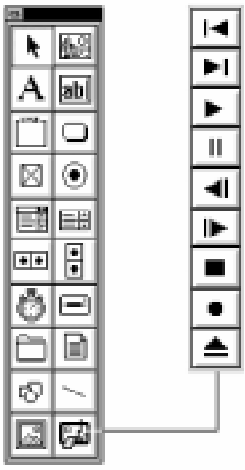


Рисунок 7. Microsoft Multimedia Control.

ните по кнопке со значком мультимедийных объектов и поместите объект MMControl1 на форму Вашего приложения. Для свойства Visible (Видимость) объекта MMControl1 установите значение False (Ложь), так как в данном приложении лучше этот объект сделать «невидимым». Для свойства FileName (Имя файла) запишите значение 01.wav. Щелкните два раза по изображению телефона для вызова процедуры события Picture1_Click. Процедура события Picture1_Click будет вызываться в работе программы всякий раз, когда пользователь щелкнет мышкой по рисунку Picture1. В данную процедуру впишите программный код:

```
MMControl1.Command = "Close"
MMControl1.Command = "Open"
MMControl1.Command = "Play"
```

Вторая строка объекту MMControl1 подаст команду открытия файла 1.wav, третья строка заставит объект MMControl1 воспроизвести звуковой файл 1.wav. Первая строка необходима для закрытия проигрывателя, в случае если пользователь пожелает прослушать данный звук еще раз.

Сохраните проект приложения в папке Lesson21 и закройте Visual Basic. Из выше указанной папки запустите проект Вашего приложения Project1.mak, который автоматически вызовет Visual Basic и загрузит приложение. Запустите приложение. Во время работы приложения понажимайте мышью на изображение телефона несколько раз. Вы должны услышать звук телефонного вызова.

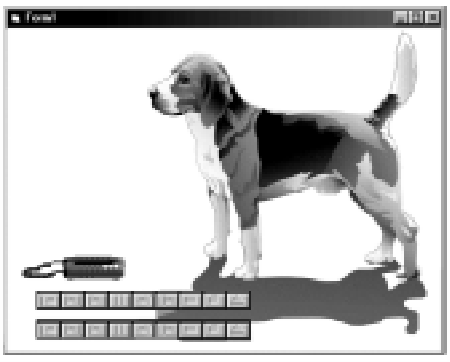


Рисунок 8.

В окно приложения поместите объект MMControl2. Не забудьте сделать его «невидимым». Для его свойства FileName (Имя файла) установите значение 02.wav. В процедуру события Picture2_Click впишите аналогичный программный код:

```
MMControl2.Command = "Close"
MMControl2.Command = "Open"
MMControl2.Command = "Play"
```

Откомпилируйте программу. Завершите работу в Visual Basic. Запустите приложение Project1.exe. Поочередно понажимайте мышью на изображения телефона и собаки (рисунок 8). Закройте приложение.

Продолжим работу над приложением «Движение автомобиля» из первого урока. Создайте папку Lesson22 и скопируйте в нее файлы Project1.mak и Form1.frm из папки Lesson12. Добавьте в папку Lesson22 файлы Mci.vbx, 01.mid, 03.wav, 04.wav.

Из папки Lesson22 запустите приложение Project.mak. 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 MMControl (Microsoft Multimedia Control). Поместите на форму Form1 два объекта: MMControl1 и MMControl2. Для свойства Visible (Видимость) этих объектов установите значение False (Ложь). Для того чтобы сразу после запуска приложения зазвучала музыка, воспользуйтесь про-

цедурой Form_Load, которая сработает при загрузке формы Form1. В данную процедуру Form_Load поместите строки программного кода:

```
MMControl1.FileName = "01.mid"
MMControl1.Command = "Close"
MMControl1.Command = "Open"
MMControl1.Command = "Play"
```

Первая строка для объекта MMControl1 присваивает его свойству FileName (Имя файла) значение «01.mid». Как работают остальные строки, Вы уже знаете.

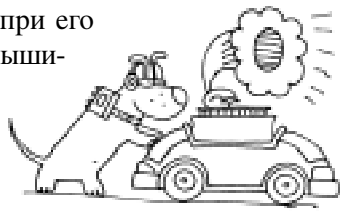
В процедуру прерывания Picture1_Click добавьте строки:

```
MMControl2.FileName = "03.wav"
MMControl2.Command = "Close"
MMControl2.Command = "Open"
MMControl2.Command = "Play"
```

В процедуре прерывания Form_Click допишите строки:

```
MMControl2.FileName = "04.wav"
MMControl2.Command = "Close"
MMControl2.Command = "Open"
MMControl2.Command = "Play"
```

Запустите программу на выполнение в среде Visual Basic. Работа приложения будет проходить на фоне мелодии. При нажатии мышью на автомобиль, последний начнет двигаться, а при его остановке Вы услышите скрип тормозов. При нажатии на свободное от рисунка поле окна приложения автомобиль будет возвращен на стартовую позицию в сопровождении сигнала. Поработав в приложении, закройте его.



А теперь заставьте приложение отличать щелчки левой и правой кнопок мыши. Для начала удалите все программные утверждения из процедуры Picture1_Click. Для управления объектом Picture1 (Рисунок) воспользуемся процедурой прерывания Picture1_MouseDown, которая вызывается всякий раз, когда кнопка мыши оказывается по приказу пользователя внизу, а указатель мыши в поле рисунка Picture1. Процедуру Picture1_MouseDown заполните программным кодом как показано в листинге 1.

Запустите приложение. Наводя указатель мыши на изображение автомобиля, щелкайте поочередно левой и правой кнопками мышки. Автомобиль будет передвигаться вперед и задним ходом, причем с различной скоростью. Объясните, почему это воз-

Листинг 1.

```
Sub Picture1_MouseDown (Button As Integer, Shift As Integer, X As Single, _
Y As Single)

If Button And 1 Then                ' условие щелчка левой кнопкой мыши
For n = 1 To 250
Picture1.Left = Picture1.Left - 30  ' передвижение вперед
Next
End If                              ' конец условия

If Button And 2 Then                ' условие щелчка правой кнопкой мыши
For n = 1 To 250
Picture1.Left = Picture1.Left + 15  ' передвижение назад
Next
End If                              ' конец условия

MMControl2.FileName = «03.wav»
MMControl2.Command = «Close»
MMControl2.Command = «Open»
MMControl2.Command = «Play»

End Sub
```

можно, внимательно разобрав работу программных утверждений процедуры Picture1_MouseDown.

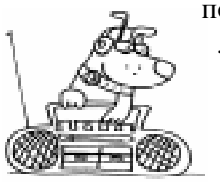
Если у Вас трехкнопочная мышка, то Вы можете задействовать и среднюю кнопку, изменив или добавив условие:

```
If Button And 4 Then
' условие щелчка средней кнопкой мыши
```

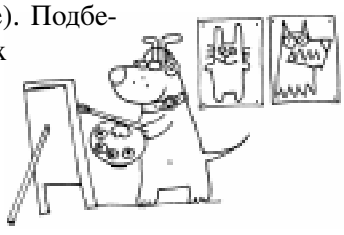
Сохраните проект в папке Lesson22. Создайте исполняемый EXE-файл. Проверьте работоспособность этого файла.

После урока

1. Изготовьте приложение, в окне которого будет размещено пять командных кнопок. При щелчке мышью по любой из них должна звучать одна из пяти различных мелодий. Для этой цели используйте любые файлы с расширением MID. Не забудьте Ваше приложение украсить подходящим для этого случая рисунком или фотографией.



2. Создайте приложение, в котором будут присутствовать волк и заяц (или любые другие животные). Подберите два звуковых файла с расширением WAV. Сделайте так, чтобы при щелчке левой кнопкой мыши по изображениям животных издавались различные звуки, а при щелчке правой кнопкой – они исчезали. Поместите на форму приложения кнопку, которая позволяла бы увидеть исчезнувших животных. Если Ваш компьютер снабжен микрофоном, то без большого труда можно создавать для своих приложений звуковые файлы с расширением WAV. Для записи звука можно использовать, например, стандартное приложение в Windows «Звукозапись» (Рисунок 9).



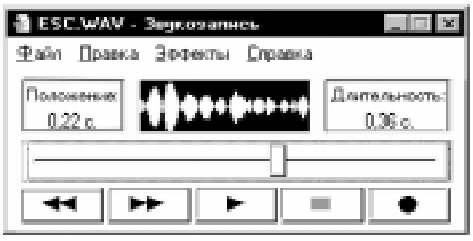


Рисунок 9.

3. Для объекта Form (Форма) самостоятельно изучите его свойство WindowState (Статус окна), которое может принимать три значения от 0 до 2.

УРОК 3:

Проигрыватель музыкальных дисков

Вставка этикеток

Извлечение текущей даты из компьютера

Управление приложением с клавиатуры

Многооконность приложения

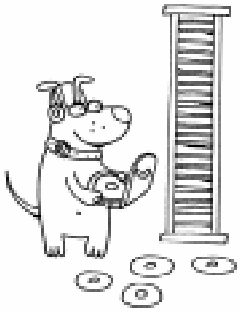
Если Вы хотите изготовить приложение собственного проигрывателя музыкальных компакт-дисков, то Ваш компьютер должен быть оснащен устройством чтения оптических CD-дисков и музыкальной картой. Если музыкальная карта отсутствует, то прослушивать записи придется посредством головных телефонов.

Создайте папку Lesson31. Поместите в нее файл поддержки мультимедийных объектов Mci.vbx. Запустите Visual Basic. Окно формы Form1 окрасьте, например, в темно-серый цвет. Добавьте в проект Вашего приложения файл Mci.vbx. Для этого в меню File (Файл) выберите команду Add File ... (Добавить файл ...) и через появившееся окно загрузите файл поддержки Mci.vbx. В окне объектов появится значок мультимедийных объектов. Щелкните по кнопке с этим значком и поместите объект MMControl1 на форму Вашего приложения. В окне свойств объекта MMControl1 для свойств BackVisible, StepVisible, RecordVisible установите значе-

ние False (Ложь), так как эти кнопки в проигрывателе использоваться не будут. В процедуру Form_Load, которая вызывается сразу после загрузки приложения, впишите программный код:

```
Sub Form_Load ( )
MMControl1.DeviceType = "CDAudio"
MMControl1.Command = "Open"
MMControl1.Command = "Play"
End Sub
```

Первая строка программного кода активизирует устройство чтения компакт-дисков, следующая строка открывает первую музыкальную запись, и последняя строка заставляет воспроизводить мелодии с диска по порядку.



В устройство для CD-дисков поместите музыкальный компакт-диск. Запустите приложение. Проигрыватель начнет воспроизводить первую запись на компакт-диске. Если этого не произойдет, вызовите в Windows стандартную панель управления звуком Volume Control и меню «Параметры», выберите команду «Свойства», где включите CD Audio. Теперь должен появиться звук, в противном случае проверьте правильность написания программного кода.

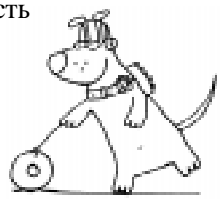
Во время работы приложения изучите назначение каждой клавиши вновь созданного проигрывателя (рисунок 10). Для этого поочередно щелкайте мышкой по клавишам проигрывателя.

В окно приложения можете поместить, выбранный на Ваше усмотрение рисунок.

В окне Properties (Свойства) для формы Form1 выберите свойство Caption (Надпись) и измените его значение на «Проигрыватель музыкальных компакт-дисков». Для полного останова проигрывателя и выхода из приложения создайте командную кнопку Command1. В процедуру прерывания Command1_Click запишите программный код:

```
Sub Command1_Click ( )
MMControl1.Command = "Stop"
MMControl1.Command = "Close"
End
End Sub
```

Первая строка останавливает воспроизведение любой записи компакт-диска, следующая строка закрывает файл мелодии для объекта MMControl1, и последняя строка полностью завершает работу приложения. Надпись на командной кнопке измените на надпись «Выход». Если Вы хотите, чтобы мелодии продолжали звучать после завершения работы приложения, смонтируйте еще одну командную кнопку Command2 и для нее в процедуру прерывания Command2_Click впишите только последнюю строку из процедуры Command1_Click, опустив две первые строки. Если через некоторое время все-таки возникнет необходимость в останове воспроизведения музыкального диска, то это можно сделать вручную, нажав на кнопку «Выезд диска» устройства для воспроизведения оптических дисков.



Для объекта MMControl1 самостоятельно изучите свойство Orientation (Расположение), которое может принимать два значения: 0-Horizontal (Горизонтальное) и 1-Vertical (Вертикальное).

А теперь рассмотрим операцию размещения этикеток в приложении. На пане-

Рисунок 10. Панель управления проигрывателем компакт-дисков.

ли объектов щелкните по значку «Этикетка» и поместите объект Label1 (Этикетка первая) на форму приложения. В окне Properties (Свойства) для этого объекта измените значение свойства Caption (Заголовок) на произвольную надпись, например, «Мир мелодий так же неисчерпаем, как и сама вселенная». Самостоятельно поэкспериментируйте со свойствами этикетки. Обязательно посмотрите, как работает свойство этикетки BackStyle (Стиль фона). Затем создайте вторую этикетку – Label2. Ее свойству Caption придайте значение сегодняшней даты. Для этого в процедуру Form_Load допишите строку:

```
Label2.Caption = Date
                    ' показать дату
```

Запустив приложение, Вы обнаружите в его окне сегодняшнюю дату (рисунок 11).

Если Вы не любите «мышиную возню», то попробуйте управлять данным приложением только с клавиатуры. Visual Basic автоматически назначает наиболее употребительные клавиши управления приложениями для Windows. Это клавиши – «Enter», «Курсоры», «Пробел», «Tab», «Esc» и другие. Отодвиньте мышь на дальний край Вашего рабочего стола, с клавиатуры запустите приложение и поработайте в нем. Затем, по-прежнему не прибегая к помощи мыши, закройте приложение. В уроке № 4 Вы узнаете, как можно назначить любую клавишу для реакции приложения на действия пользователя.

А теперь вкратце остановимся на возможности построения многооконных приложений. В только что созданное приложение добавьте второе окно, где расположите сведения об авторстве на это приложение. Из папки Lesson31 запустите проект приложения Project1.mak. В области панели инструментов щелкните по значку «Открытие

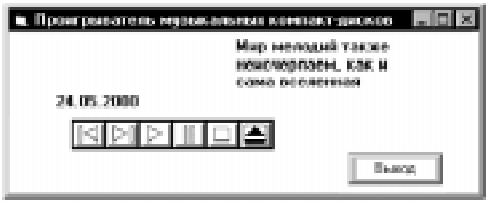


Рисунок 11.

новой формы» или в меню File (Файл) выберите New Form (Новая форма). Ваше приложение дополнится еще одной формой Form2. Уменьшите размеры новой формы в расчете на то, чтобы на ней поместились минимальные сведения об авторе. Переместите ее в нужную для Вас область экрана. Окрасьте окно этой формы в произвольный цвет, например, серый. Для вызова второго окна установите на форме Form1 первого окна кнопку Command3. Ее свойству Caption (Надпись) придайте значение «О программе» и в процедуру прерывания Command3_Click впишите программное утверждение:

```
Form2.Show
                ' показать вторую форму
```

Эта строка при нажатии на кнопку «О программе» в поле первого окна покажет второе окно в определенном месте экрана. Для формы Form2 свойству Caption (Надпись) придайте значение «О программе», свойству BorderStyle (Граничный стиль) придайте значение 3-Fixed Double. Во втором окне расположите небольших размеров рисунок или пиктограмму (расширение ICO). Если у Вас окажется под рукой сканер, поместите во второе окно свою фотографию. Поместите в это окно сведения о названии программы, о дате ее создания, Ваше имя, телефон и другое. Для этой цели несколько раз используйте объект Label (Этикетка).

Во избежание путаницы при создании многооконного приложения необходимо указывать перед именем объекта имя формы, которой принадлежит тот или иной объект. Например, строка из программного кода приложения «Проигрыватель музыкальных компакт-дисков» теперь будет выглядеть так:

```
Form1.Label2.Caption = Date
                    ' показать дату
```

А для второй этикетки второй формы можно записать, например:

```
Form2.Label2.Caption = _
                "Омск 2000 Россия"
```

Если Вы хотите скрыть вторую форму Form2 на каком-либо этапе работы Вашего приложения, то в нужную процедуру прерывания включите строку:

```
Form2.Hide    ' скрыть вторую форму
```

Можно данное программное утверждение поместить, например, в процедуру прерывания второй формы Form_Deactivate. Эта процедура сработает при свершении такого события, как наступление пассивного состояния второго окна.

Для загрузки формы в память существует оператор Load (Загрузить). Например, для второй формы он применяется так:

```
Load Form2
' загрузка второй формы в память
```

Чтобы выгрузить форму, используйте оператор Unload. Для удаления второй формы из памяти необходимо программное утверждение:

```
Unload Form2
' выгрузка второй формы из памяти
```

Если применена команда Show (Показать) до выполнения команды Load (Загрузить), то команда Show автоматически загрузит форму, а затем покажет ее. И все-таки часто возникает необходимость в использовании оператора Load, например, для предварительной загрузки формы, содержащей большое число объектов. Тогда неприятное «притормаживание» будет скрыто от взора пользователя. Иногда во время работы приложения полезно выгружать из памяти формы, которые не отображены на экране в данное время. Тем самым Вы справитесь с перегрузкой памяти.

После урока

1. В одно из приложений первого или второго уроков добавьте два новых окна. Первое из этих окон должно содержать сведения о текущих дате и времени, а второе превратите в проигрыватель музыкальных компакт-дисков.

2. Самостоятельно изучите объект Line (Линия) и его свойства. Подумайте, как применить этот объект в ранее созданных Вами приложениях.

3. Самостоятельно исследуйте объект типа Frame (Фрейм) и его свойства. Используйте этот объект для оформления Ваших программ.

4. Непременно исследуйте все имеющиеся у Вас оптические диски (даже с игровыми программами) путем запуска их из созданного Вами проигрывателя. Иногда совершенно неожиданно начинают звучать мелодии из программ, в которых они должны воспроизводиться только во время работы этих программ.

Рисунок 12. Некоторые наиболее употребительные команды языка Visual Basic

And	и	**Load**	загрузить в память
Beep	звуковой сигнал	**Loop**	цикл
Circle	рисовать окружность	**Mid**	середина
Cls	очистить экран	**Next**	конец оператора For
Dim	размерность	**Not**	не
Do	выполнить	**Or**	или
Else	иначе	**Print**	вывод на экран
End If	конец оператора If	**PSet**	рисовать точку
Exit Do	выход из цикла Do	**Rnd**	случайное число
For	для	**Step**	шаг
GoTo	идти к	**Then**	то
If	если	**To**	до
Input	ввод с клавиатуры	**UnLoad**	удалить из памяти
Len	длина	**Wend**	выход из цикла While
Line	рисовать линию	**While**	пока

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