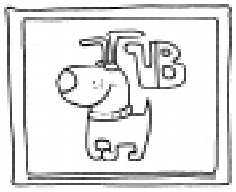
Паньгина Нина Николаевна

ЗАНЯТИЕ 1. ПЕРВОЕ ЗНАКОМСТВО С VISUAL BASIC

ВВЕДЕНИЕ



Важнейшие качества языка BASIC (Beginner's All-purpose Symbolic Instruction Code – символический командный код общего назначения для начинающих) – простота и компактность – оказались решающими в период перехода на микрокомпьютеры. Билл Гейтс и Пол Аллен, основатели корпорации MICROSOFT (MS), создали версию BASIC, способную работать в 4 Килобайтах оперативной памяти (!), со временем превратив эту версию в один из самых популярных программных продуктов для персональных компьютеров (ПК).

Впоследствии потребность в более быстром, компактном, простом и структурированном процедурном языке привела к появлению MS QUICK BASIC, поднятого на уровень технологии программирования 80-х годов.

Но большие перемены в компьютерном мире, связанные с принятием стандарта на графический интерфейс пользователя (Graphical User Interface, GUI), требовали средств разработки приложений под Windows. И такое средство появилось в 1991 году – MS Visual Basic, позволяющее, «отстранившись» от сложнейшей внутренней структуры Windows, творить в свое удовольствие.

Нельзя также забывать, что первая версия Visual Basic была создана еще под DOS и позволяла не в графической среде оперировать такими объектами, как окна, кнопки и т.д., приводя в неописуемый восторг начинающих программистов.

ЧТО ТАКОЕ VISUAL BASIC?

Visual Basic (VB) – это *среда разработки программ*, которая позволяет быстро и легко создавать приложения (прикладные программы) для Windows. В нее включено все, что необходимо для создания, модификации, тестирования, корректирования и компиляции Ваших программ.

Visual Basic – это *полноценный язык программирования* высокого уровня.

Слово *Visual* – «визуальный», «наглядный» – означает способ разработки пользовательского интерфейса программы. Еще на этапе создания программы Вам видно, как будет выглядеть программа в действии. Вы «рисуете» окна, кнопки, текстовые панели, линейки прокрутки и другие компоненты пользовательского интерфейса подобно тому, как рисуют объекты в графическом редакторе для Windows.

Слово *Basic* – «основной» – описывает тип программного кода, который Вы создаете. Программа Visual Basic представляет собой вариант хорошо известного языка программирования.

Получили распространение такие подмножества языка Visual Basic, как VB For Applications (встроенный язык написания макросов для MS Office приложений) и VB Script (язык для работы в среде WEB-браузера с объектами и элементами HTML).

ЧТО МОЖЕТ VISUAL BASIC?

Visual Basic позволяет создавать прикладные программы, которые могут:

- Использовать стандартные элементы пользовательского интерфейса Windows, такие, как окна, меню, кнопки, линейки прокрутки и т.п.
- Создавать, читать и записывать текстовые файлы и файлы баз данных.
- Выводить данные на печать при помощи стандартных драйверов принтеров Windows.
- Читать графические файлы в форматах .bmp, .jpg, .tif, .ico, .wmf и др.
- Обращаться к базам данных таких форматов, как SQL, dBase, Microsoft Access и другие.

· Взаимодействовать с другими прикладными программами через буфер обмена, DDE (Динамический Обмен Данными) и OLE (Связывание и Встраивание Объектов).
· Использовать элементы управления OCX ActiveX и API-функции.
· Взаимодействовать с Интернет.

ЗАПУСК VISUAL BASIC ОСНОВНЫЕ ОКНА

После запуска Visual Basic (VB) на экране появляются пять окон (рисунок 1):
· основное окно;
· окно формы;
· окно инструментария;
· окно проекта;
· окно свойств.

Основное окно (или Main window) содержит стандартные пункты меню File и Edit, как и во всех приложениях Windows, собственные пункты Visual Basic: View, Project, Format, Run, Tools и др., а также панель инструментов (Toolbar).

Окно формы (Form) – это окно будущего приложения, на которое, как на холст, наносятся различные части программы – объекты (objects) или элементы управления (controls).

Окно инструментария (Toolbox) содержит набор элементов управления, представленных каждый своим значком (icon) или инструментом (tool), позволяет вставлять в прикладные программы такие элементы, как кнопки, текстовые окна и диалоговые окна...

Окно проекта (Project) содержит список всех файлов, необходимых для выполнения программы на VB.

Окно свойств (Properties) содержит атрибуты конкретного объекта управления, позволяет менять параметры компонентов проекта.

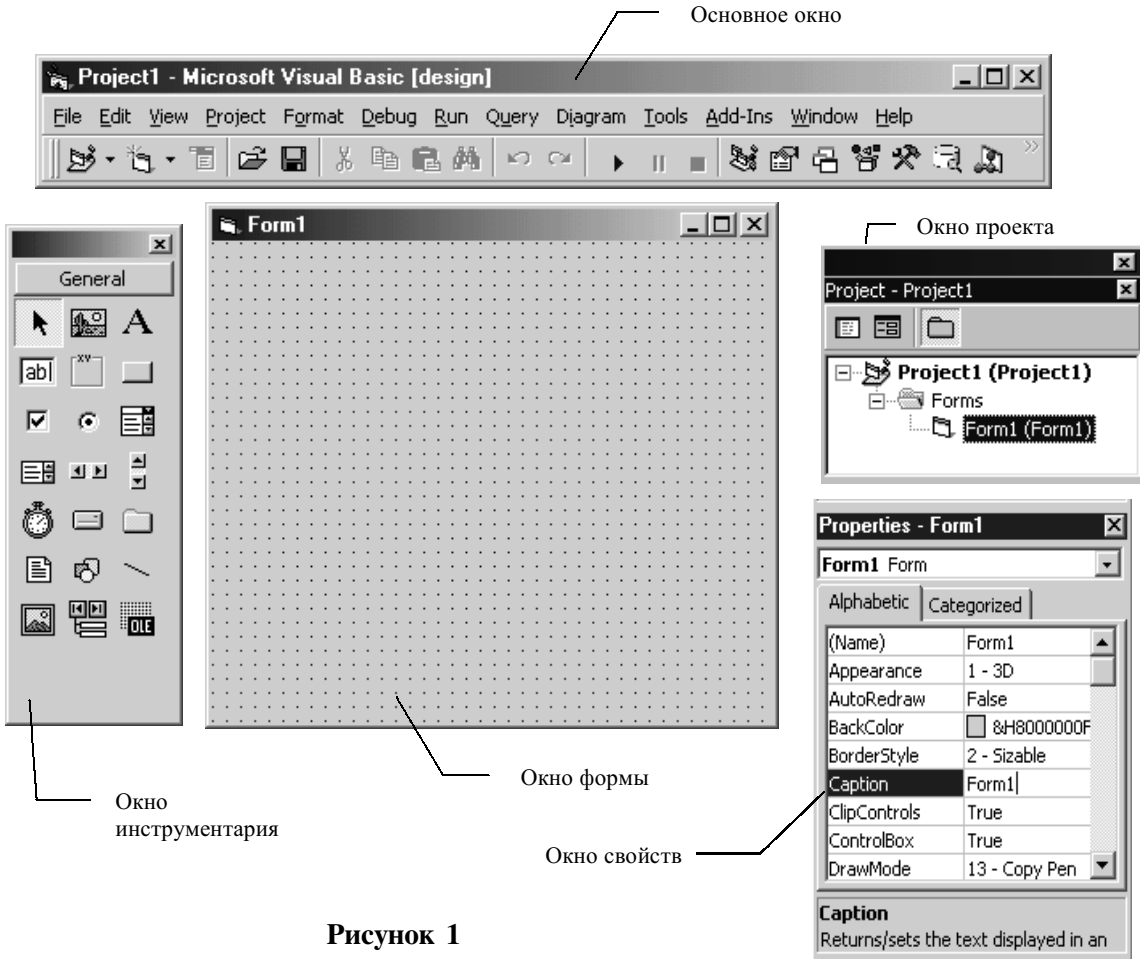
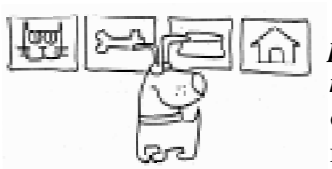


Рисунок 1

ОСНОВНЫЕ ТЕРМИНЫ, СВЯЗАННЫЕ С СИСТЕМОЙ VISUAL BASIC



Объект управления и контроля (Control) – общий термин, который определяет как экранные формы, так и графические элементы внутри этих форм, в том числе текстовые окна, окна-списки, командные кнопки, графические окна, линейки прокрутки, пиктограммы и т.д. Над объектами управления можно осуществлять только операции, определенные в рамках метода доступа. В системе VB термины «объект управления и контроля » (Control) и «объект» (Object) равнозначны.

Событие (Event) – действие, которое распознается объектом управления VB.

Экранная Форма (Form) – окно, созданное пользователем в соответствии с требованиями прикладной программы.

Метод доступа (Method) – ключевое слово VB, аналогичное по смыслу понятиям «функция» или «оператор», но которое воздействует всегда на конкретный объект управления. Для каждого объекта система VB определяет несколько стандартных методов, которыми может оперировать пользователь.

Процедура (Procedure) – этим термином обозначаются как подпрограммы, так и функции. Это просто последовательность операторов VB, к которым система обращается, как к связанной группе во время выполнения. Существует два типа процедур: событийная процедура и общая процедура.

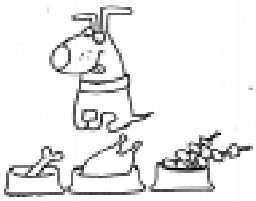
Событийная процедура используется только с экранными формами и объектами управления, общие же процедуры используются в любом месте программы и могут вызываться событийными процедурами.

Проект (Project) – совокупность всех файлов, составляющих прикладную программу.

Параметр или свойство (Property) – характеристика или атрибут объекта управления. Для каждого типа объекта управления VB определяет набор параметров, относящихся только к данному объекту.

Значение (Setting) – значение параметра. Можно изменить значение большинства параметров при разработке программы. Программа может изменить значения параметров и во время работы.

ВОЗМОЖНОСТИ МЕНЮ VISUAL BASIC



File – содержит команды, позволяющие создавать, открывать, сохранять, печатать, компилировать и закрывать проекты VB.

Edit – используется для изменения содержимого экранной формы и программного кода, установки связи при динамическом обмене данными; содержит команды отмены, записи в буфер памяти, чтения из памяти, команды контекстного поиска и замены.

View – содержит команды для просмотра компонентов VB.

Insert (Project) – позволяет добавлять в проект новые формы и модули.

Run – содержит команды для выполнения и компиляции проекта.

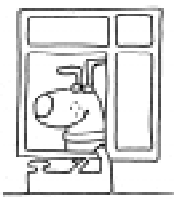
Tools – содержит команды для конфигурирования среды программирования VB и текущего проекта.

Add-Ins – содержат дополнительные средства, расширяющие возможности VB.

Window – содержит команды для работы с окнами.

Help – обеспечивает доступ к справке VB и справочному руководству.

ОКНО ИНСТРУМЕНТАРИЯ (TOOLBOX)



Инструментарий – это набор сервисных программ или инструментов (рисунок 2), которые используются для изменения содержимого экранной фор-

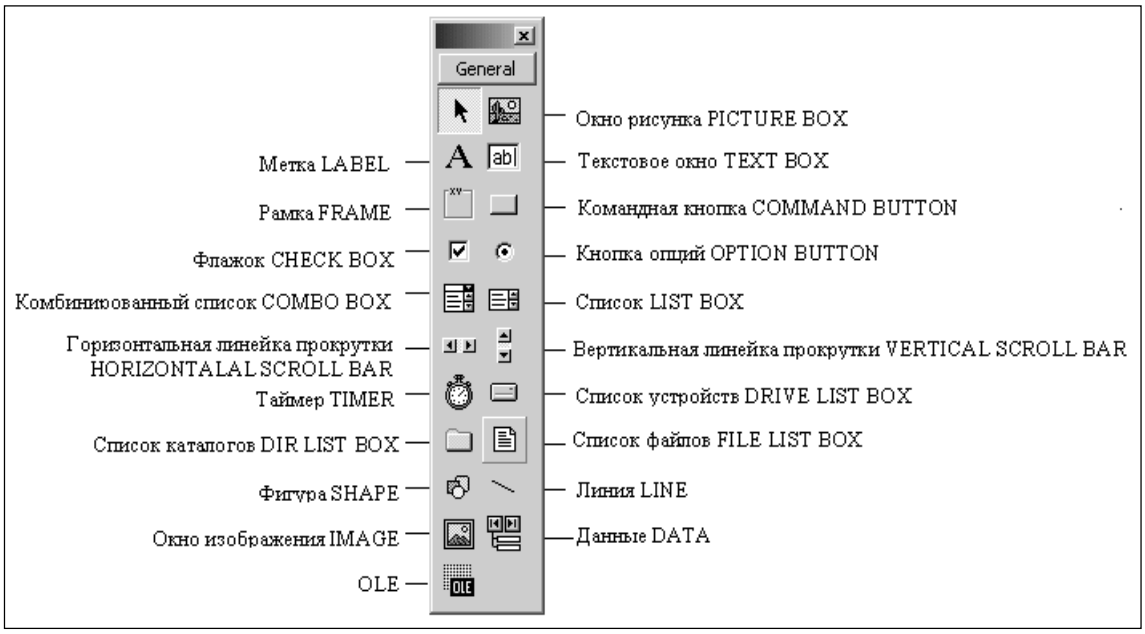


Рисунок 2

мы и программного кода, установки связи при динамическом обмене данными.

РАЗРАБОТКА ПРОГРАММ НА VISUAL BASIC

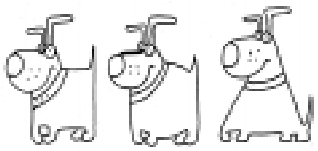
Существует два этапа создания программы на Visual Basic:

1. Этап *визуального программирования*, когда Вы создаете графический интерфейс пользователя с будущей программой, то есть создаете окно Windows-приложения, на котором размещаете элементы управления и контроля и присваиваете значения их свойствам.

2. Этап *программирования в исходном коде*. Здесь Вы вводите программный код для обработки определенных событий для заданных объектов. Чтобы ввести программный код, соответствующий объекту управления и контроля, необходимо

- двойным щелчком на объекте открыть окно кода;
- развернуть список событий (справа вверху) и выбрать нужное;
- ввести программный код между Private Sub и End Sub.

ОБЪЕКТ УПРАВЛЕНИЯ И КОНТРОЛЯ FORM



Объект ***Form (Форма)*** представляет собой стандартное Windows-окно (рисунок 3), которое служит основой для создания интерфейса прикладной программы. Формы имеют некоторое множество параметров, которые определяют их внешний вид и доступные режимы работы с ними.

Основные свойства:

BackColor – цвет фона (задается 16-ричной константой)

BorderStyle – тип границы

0 – нет
1 – одинарная фиксированная
2 – изменяемая
3 – двойная фиксированная

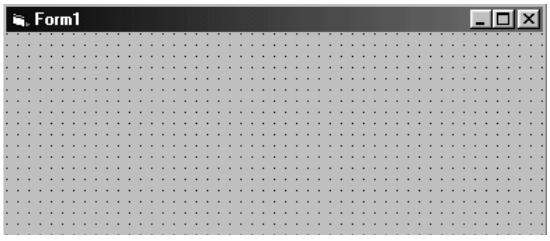


Рисунок 3

Caption – название формы (в заголовке)
Drawstyle – стиль рисования
Solid – сплошная
Dash – пунктир
Dot – точки
Transparent – прозрачная
Enabled – доступность формы
FillColor – цвет заполнения
FillStyle – стиль заполнения
FontBold – полужирный шрифт
FontItalic – курсив
FontName – имя шрифта
FontSize – размер шрифта
ForeColor – цвет переднего плана
Height – высота формы
Icon – рисунок в поле заголовка
Left – расстояние до формы от левой границы экрана
ScaleMode – единица измерения в координатной системе объекта
Top – расстояние до формы от верхней границы экрана
Visible – видимость
Name – имя формы в программе
MinButton – кнопка минимизации формы
MaxButton – кнопка максимизации формы
MousePointer – указатель мыши
WindowState – состояние окна

События для формы:
Load – загрузка формы
Unload – выгрузка формы
Click – щелчок мыши
DoubleClick – двойной щелчок

Процедуры и методы:
Cls – очистка формы
Print – вывод текста на форму
Line – линия
Circle – окружность
LoadPicture (имя файла) – функция загрузки картинки, записанной в виде файла

При написании программного кода каждый объект определяется идентификатором, указанным в его свойстве Name. Обращение к свойствам или методам объектов производится по уточненному имени с точкой-разделителем. Например:

Form1.Caption = «Первая программа» меняет название заголовка формы.

Form1.Circle (1000, 1000), 500 рисует окружность с центром в точке с координатами (1000; 1000) относительно левого верхнего угла формы и радиусом 500. По умолчанию используется единица измерения 1 twip = 1/1440 дюйма. Единицы измерения можно изменить в свойстве формы ScaleMode.

Упражнения
(для самостоятельного выполнения):

1. Измените название формы «Form1» на «Моя форма», используя окно свойств.

2. Измените цвет фона формы на голубой, а стиль границы на фиксированный через окно свойств.

3. Проверьте действие свойств Icon, Visible, Enabled, MousePointer, WindowState.

4. При щелчке на форму измените ее название на «Первая программа на Visual Basic», выведите на форме слово «Здравствуйте» и нарисуйте в центре формы окружность. Используйте событийную процедуру Click.

5. При двойном щелчке на форму очистите ее и завершите работу программы, используя событийную процедуру DoubleClick.

ОБЪЕКТ УПРАВЛЕНИЯ И КОНТРОЛЯ COMMAND BUTTON

Щелчок ***Command Button (Командной кнопки)*** активизирует выполнение какой-то команды (рисунок 4).

Рисунок 4

Основные свойства:
Caption, Enabled, FontName, FontItalic, FontSize, FontBold, Height, Width, Left, Top, Mousepointer, Visible, Name – свойства, аналогичные свойствам формы.

Cancel – отмена (если true, то щелчок на командную кнопку аналогичен клавише Esc)

Default – по умолчанию (если true, то нажатие Enter аналогично щелчку этой кнопки)

Основные события для командной кнопки:

Click – щелчок мыши.

ОБЪЕКТ УПРАВЛЕНИЯ И КОНТРОЛЯ LABEL

Label (Метка) – поле, заполняемое программистом текстовой информацией и недоступное пользователю для редактирования (рисунок 5). Содержание Метки определяется ее значением Caption. Прямой вывод текста или рисование на метке не допускается.

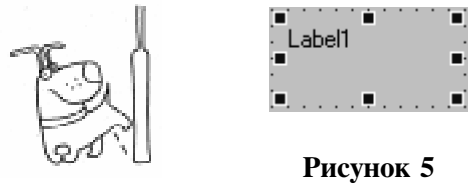


Рисунок 5

Основные свойства:

Backcolor, Enabled, FontName, FontItalic, FontSize, FontBold, Forecolor, Height, Width, Left, Top, Mousepointer, Visible, Name – свойства, аналогичные свойствам формы.

Alignment (выравнивание) – определяет, каким образом размещается название метки. По умолчанию оно равно 0 – Left Justify, что выравнивает текст на метке по ее левой границе. Прочие значения: 1 – Right Justify (выравнивание по правой границе) и Center (по центру).

AutoSize (автоподстройка размера). Если это свойство приравнено True, размер поля метки автоматически подгоняется под размер текста, заданный свойством Caption. Если же это свойство соответствует False, метка сохраняет размер, установленный при проектировании; лишние символы длинного текста просто отсекаются.

BorderStyle (тип границ). Это свойство способно принимать всего два значения: 0 – контур поля метки отсутствует, устанавливается по умолчанию, и 1 – метка очерчивается одинарными линиями.

Enabled (доступ). Обычно равно True, а если присвоить False, текст метки станет серым (поблекнет), и обработка событий, связанных с действиями мышки, будет запрещена.

Основные события для Метки:

Объект Метка воспринимает события Click и DblClick так же, как и окно формы.

Задание 1. Первая программа.

1. Создать форму, как на рисунке 6, переименовать ее.

2. Нанести на форму Метку и три Командные Кнопки. Изменить их свойства в соответствии с рисунком.

3. Написать программу, выполняющую следующие функции:

- при нажатии на Кнопку «Привет» в окне Метки появляется сообщение «Здравствуйте, друзья!»;
- при нажатии на Кнопку «Очистка» окно Метки очищается;
- при нажатии на Кнопку «Выход» программа завершает свою работу.

Выполнение задания 1.

Первая часть – *визуальное программирование*, то есть разработка внешнего вида приложения.

1. Необходимо создать форму и переименовать ее в соответствии с рисун-

Рисунок 6

ком 6, свойству Caption (название) присвоив значение «Первая программа на Visual Basic».

2. Нанести на форму Метку, изменив ее свойство Caption на пустую строку, свойству Alignment (выравнивание) придать значение Center (по центру), изменить значения свойства Font (шрифт), выбрав название шрифта Arial, размер 16, стиль полужирный курсив.

3. Нанести на форму Командные кнопки 1, 2 и 3, изменив их размеры, и свойство Font, а также свойство Caption на «Привет», «Очистка» и «Выход», соответственно.

Вторая часть – *написание кода программы*. Для этого необходимо следующее:

- Определить, какие будут использоваться глобальные переменные, и описать их. Описание глобальных переменных (используемых в нескольких процедурах) выполняется в процедуре Declarations объекта General.
- Определить, какие события и в какой последовательности должны происходить с объектами управления и контроля. Первое событие – загрузка формы. При загрузке формы выполняется инициализация переменных и объектов.
- Наметить алгоритм выполнения необходимых действий по каждому из событий.
- Описать эти действия с помощью операторов и методов в соответствующих событийных процедурах.

4. В данном приложении не будут использоваться никакие переменные, поэтому описание не требуется.

5. В данном задании не нужно выполнять инициализации.

6. Второе событие после загрузки формы – щелчок на кнопку «Привет». В процедуре обработки данного события свойству Caption Метки необходимо присвоить значение символьной строки «Здравствуйте, друзья!»

```
Private Sub Command1_Click()
Label1.Caption = «Здравствуйте, друзья»
End Sub
```

7. Третье событие – щелчок на кнопку «Очистка». В процедуре обработки этого события свойству Caption Метки необходимо присвоить значение пустой символьной строки.

```
Private Sub Command2_Click()
Label1.Caption = «»
End Sub
```

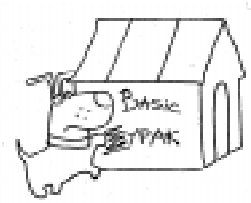
8. Последнее событие – щелчок на кнопку «Выход». В процедуре обработки этого события необходимо завершить работу программы.

```
Private Sub Command3_Click()
END
End Sub
```

Третья часть – *выполнение программы*. Для этого необходимо выбрать пункт горизонтального меню Run и подпункт Start, либо нажать кнопку Start на стандартной панели инструментов, либо клавишу F5. Важно уметь определять, в каком состоянии находится приложение: в состоянии дизайна или в состоянии выполнения. Это определяется, в первую очередь, по строке заголовка основного окна VB – design или run, соответственно. В программе всегда должен быть предусмотрен выход. В крайнем случае, можно воспользоваться подпунктом меню End в пункте Run или аварийным завершением Ctrl+Break.

9. Запустить программу на выполнение, протестировать ее и завершить.

ОБЪЕКТ УПРАВЛЕНИЯ И КОНТРОЛЯ TEXT BOX



Text Box (Текстовое окно) – позволяет пользователю вводить и редактировать текст.

Основные свойства:

Текстовые окна, наряду со стандартными свойствами FontBold, FontItalic, FontName, FontSize, FontUnderline, BorderStyle, Enabled, Left, Name, Top, Visible, Width и Height, обладают еще и такими:

Text (текст) – содержимое Текстового окна (символьные данные). Программа распознает с помощью этого свойства, какой именно текст введен пользователем. Кроме того, оно позволяет изменять отображаемый на экране текст.

MaxLength (максимальная длина). Оно обычно равно 0 (по умолчанию), то есть в текстовое окно можно вводить любое количество символов. Если установить значение, не равное 0, Visual Basic ограничит возможность ввода до заданного количества символов.

MultiLine (многострочность). Если это свойство приравнено к False, то разрешен ввод не более чем одной строки текста. Установив True, можно вводить по несколько строк, нажимая клавишу Enter (то есть вставлять в текст символ возврата каретки) и продолжая набор текста с новой строки.

Alignment – выравнивание текста (работает только при Multiline = true)

PasswordChar (символ пароля). Это свойство определяет: защищено данное текстовое окно паролем или нет. Если текст ввести в защищенное паролем окно, символы на экране будут отличаться от набираемых на клавиатуре. По умолчанию свойство PasswordChar соответствует пустой строке, а это означает, что какие символы пользователь вводит, такие символы он видит. Если же это свойство приравнять какому-нибудь символу (допустим, звездочке), то на экране, вместо набираемых знаков, появятся звездочки. Но, на самом деле, содержимое текстового окна соответствует введенному тексту – звездочки высвечиваются только на дисплее.

ScrollBar (линейки прокрутки). Этому свойству можно присвоить 0 – линеек прокрутки в текстовом окне нет, 1 – появляется только горизонтальная линейка прокрутки, 2 – появляется только вертикальная линейка прокрутки и 3 – видны обе линейки.

SelLength (количество выделенных символов). Это свойство определяет количество символов, выделенных в данный момент. Его значение меняется при выделении знаков в текстовом окне. То же самое можно сделать программным путем, присвоив свойству нужную величину (целого типа), и, кстати, тем самым изменить размер выделенного фрагмента.

SelStart (начало выделенного блока). Параметр, указанный в этом свойстве, определяет, откуда начинается выделенный фрагмент текста: с первого символа (0), со второго (1) и т. д. Свойство SelStart тоже допускается модифицировать программным путем.

SelText (выделенный текст). Это свойство содержит текстовую строку, соответствующую выделенному в данный момент фрагменту. Если текст не выделен, то это свойство содержит пустую строку. Устанавливая данное свойство в коде программы, Вы заменяете выделенный в текстовом окне фрагмент новым значением SelText. Например, если в текстовом окне с именем Text1 набрана строка: «Здраствуйте, друзья!» и слово «Здраствуйте» выделено, его можно исправить, выполнив оператор Text1.SelText = «Здравствуйте».

Свойства SelLength, SelStart, SelText доступны только при выполнении программы.

Основные события для Текстового окна:

Change (изменение). Сигнал об этом событии поступает в программу при изменении свойства Text пользователем (при вводе нового текста) или программой, устанавливающей новое значение этого свойства. Обратите внимание: если Вы набираете слово «Привет», сигнал о событии Change поступает шесть раз – по одному на каждую букву.

LostFocus (уход из фокуса). Это событие возникает при перемещении пользователем курсора ввода за пределы данного текстового окна или начале работы с мышью над какими-нибудь другими объектами на форме. Проверять значение свойства Text эффективнее в процедуре обработки события LostFocus, а не Change.

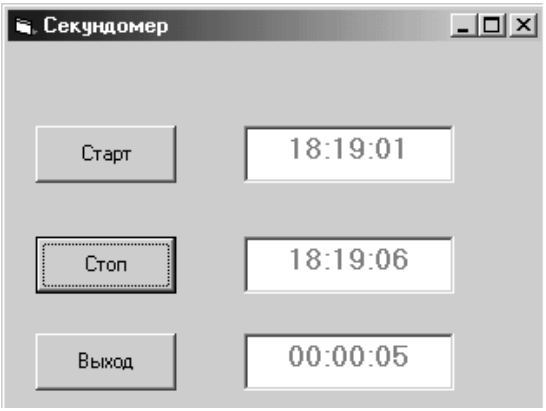


Рисунок 7

Задание 2. Секундомер.

1. Создать форму, как на рисунке 7, переименовать ее.

2. Нанести на форму три Текстовых окна и три Командные Кнопки. Изменить их свойства в соответствии с рисунком.

3. Написать программу, выполняющую следующие функции:

- при нажатии на Кнопку «Старт» в первом Текстовом окне появляется текущее время;
- при нажатии на Кнопку «Стоп» во втором Текстовом окне появляется текущее время, а в третьем Текстовом окне – разность времен;
- при нажатии на Кнопку «Выход» программа завершает свою работу.

Выполнение задания 2.

Первая часть – *визуальное программирование*.

1. Необходимо создать форму и переименовать ее в соответствии с рисунком 7, свойству Caption (название) присвоив значение «Секундомер».

2. Нанести на форму три Текстовых окна, изменив их свойство Text на пустую строку, свойству Alignment (выравнивание) придать значение Center (по центру), свойству Multiline присвоить значение True, изменить значения свойства Font (шрифт), выбрав название шрифта Arial, размер 14, стиль полужирный.

3. Нанести на форму Командные кнопки 1, 2 и 3, изменив их размеры и свойство Font, а также свойство Caption на «Старт», «Стоп» и «Выход», соответственно.

Вторая часть – *написание кода программы.*

4. В данном приложении будут использоваться две глобальные переменные: начальное время tn и конечное время tk, поэтому требуется описание в процедуре Declarations объекта General.

```
Dim tk, tn
```

5. Второе событие после загрузки формы – щелчок на кнопку «Старт». В процедуре обработки данного события свойству text первого Текстового окна необходимо присвоить значение текущего системного времени (встроенная функция Time) и запомнить его в переменной tn.

```
Private Sub Command1_Click()
tn = Time: Text1.Text = tn
End Sub
```

6. Следующее событие – щелчок на кнопку «Стоп». В процедуре обработки данного события свойству text второго Текстового окна необходимо присвоить значение текущего системного времени (встроенная функция Time) и запомнить его в переменной tk. Вычислить разность двух значений времени и вывести ее по формату (чч:мм:сс) в Текстовое окно 3 (встроенная функция Format).

```
Private Sub Command2_Click()
tk = Time
Text2.Text = tk
Text3.Text = Format(tk - tn, «hh/
mm/ss»)
End Sub
```

7. Последнее событие – щелчок на кнопку «Выход». В процедуре обработки этого события необходимо завершить работу программы.

```
Private Sub Command3_Click()
END
End Sub
```

8. Запустить программу на выполнение, протестировать ее и завершить.

Рисунок 8

Задание 3.
Перевод из радианов в градусы
(для самостоятельного выполнения)

1. Создать форму, как на рисунке 8, переименовать ее.

2. Нанести на форму две Метки, два Текстовых окна и две Командные Кнопки. Изменить их свойства в соответствии с рисунком.

3. Написать программу, выполняющую следующие функции:

- при щелчке на Текстовые окна они очищаются;
- при нажатии на Кнопку «Перевод», если в первом Текстовом окне записано число, то во втором появляется соответствующее число в радианах и, наоборот, если во втором Текстовом окне записано число, то в первом появляется соответствующее число в градусах;
- при нажатии на Кнопку «Выход» программа завершает свою работу.

Задание 4. Перевод чисел
из одной системы счисления в другую
(для самостоятельного
творческого выполнения)

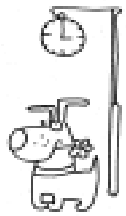
1. Создать форму для программы, выполняющей перевод из 10-й системы счисления в 2-ую систему счисления и наоборот.

2. Нанести на форму соответствующие задаче Объекты управления и контроля. Изменить их свойства.

3. Написать программу, выполняющую следующие функции:

- очистку Текстовых окон или Меток;
- при нажатии на соответствующие Кнопки должен осуществляться перевод из одной системы счисления в другую или программа должна завершать свою работу.

ОБЪЕКТ УПРАВЛЕНИЯ И КОНТРОЛЯ TIMER



Timer (Таймер) – объект управления и контроля, который способен инициировать события через регулярные промежутки времени.

Основные свойства:

Enabled, Left, Top, Name – свойства, аналогичные свойствам других объектов.

Interval – определяется интервал между двумя событиями (в миллисекундах). Интервал, равный нулю, отключает таймер.

Задание 5. Электронные часы.

1. Создать форму, как на рисунке 9, переименовать ее.

2. Нанести на форму Текстовое окно и три Командные Кнопки. Изменить их свойства в соответствии с рисунком.

3. Поместить на форму Таймер, задать свойство Interval.

4. Написать программу, выполняющую следующие функции:

- при нажатии на Кнопку «Старт» в Текстовом окне появляется текущее время, окно становится недоступным, часы «идут», то есть каждую секунду меняется время;

Рисунок 9

· при нажатии на Кнопку «Стоп» в Текстовом окне останавливается текущее время, а при повторном нажатии на кнопку «Старт» часы продолжают идти;
· при нажатии на Кнопку «Выход» программа завершает свою работу.

Выполнение задания 5.

Первая часть – *визуальное программирование*.

1. Необходимо создать форму и переименовать ее в соответствии с рисунком 9, свойству Caption (название) присвоив значение «Электронные часы».

2. Нанести на форму Текстовое окно, изменив его свойство Text на пустую строку, свойству Alignment (выравнивание) придать значение Center (по центру), свойству Multiline присвоить значение True, изменить значения свойства Font (шрифт), выбрав название шрифта Arial, размер 16, стиль полужирный.

3. Нанести на форму Командные кнопки 1, 2 и 3, изменив их размеры и свойство Font, а также свойство Caption на «Старт», «Стоп» и «Выход», соответственно.

4. Нанести на форму объект Timer, присвоить свойству Interval значение 1000 (соответствует 1 секунде).

Вторая часть – *написание кода программы*.

5. В данном приложении не будут использоваться переменные, поэтому не требуется делать описания в процедуре Declarations объекта General.

6. Первое событие – загрузка формы. Необходима инициализация объекта Timer, свойству Enabled присвоить значение False, чтобы таймер не начал работу сразу после запуска программы.

```
Private Sub Form_Load()
Timer1.Enabled = False
End Sub
```

7. Второе событие после загрузки формы – щелчок на кнопку «Старт». В процедуре обработки данного события свойству Enabled объекта Timer присвоить значение True, чтобы «включить» таймер.

```
Private Sub Command1_Click()
Timer1.Enabled = True
End Sub
```

8. Следующее событие Timer применимо к объекту Timer. Событие наступает, когда исчерпывается интервал времени в 1сек. В процедуре обработки данного события необходимо отразить в Текстовом окне текущее время и дату. Для этого можно использовать функцию Now (встроенная функция, возвращающая текущую дату и время).

```
Private Sub Timer1_Timer()
Text1.Text = Now
End Sub
```

9. Следующее событие – щелчок на кнопку «Стоп». В процедуре обработки данного события свойству Enabled объекта Timer присвоить значение False, чтобы «выключить» таймер, а значит, и электронные часы.

```
Private Sub Command2_Click()
Timer1.Enabled = False
End Sub
```

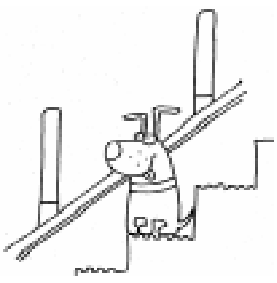
10. Последнее событие – щелчок на кнопку «Выход». В процедуре обработки этого события необходимо завершить работу программы.

```
Private Sub Command3_Click()
END
End Sub
```

11. Запустить программу на выполнение, протестировать ее и завершить.

ОБЪЕКТЫ УПРАВЛЕНИЯ И КОНТРОЛЯ SCROLL BARS

Horizontal Scroll Bar и ***Vertical Scroll Bar (Горизонтальная и вертикальная линейки прокрутки)*** – наиболее распространенные управляющие элементы в Windows-приложениях (рисунок 10). Дают возможность пользователю выбирать некоторое числовое значение, передвинув движок (ползунок) на поло-



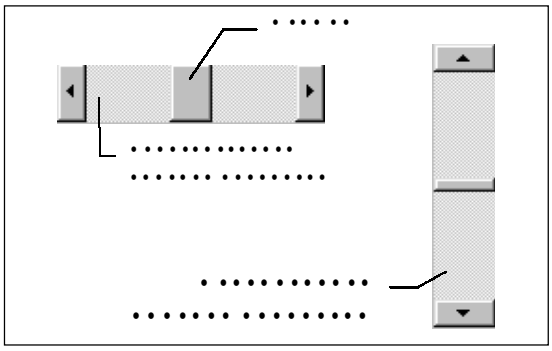


Рисунок 10

се. Эта шкала может использоваться самостоятельно и служить устройством ввода или индикатором скорости, количества и т.п.

Основные свойства:

Свойства, аналогичные свойствам других объектов – Enabled, Height, Left, Name, Top, Visible, Width.

Value (Текущая позиция). Это свойство содержит число, которое отражает текущую позицию движка на линейке прокрутки.

Min (Минимум). Значение этого свойства в пределах целого типа. Определяет крайнюю левую или самую верхнюю позицию движка.

Max (Максимум). Значение этого свойства в пределах целого типа. Определяет крайнюю правую или самую нижнюю позицию движка.

LargeChange (Постраничное изменение). Это свойство определяет величину, которая добавляется или вычитается из значения свойства Value при щелчке внутри линейки прокрутки.

SmallChange (Построчное изменение). Это свойство определяет величину, которая добавляется или вычитается из значения свойства Value при щелчке одной из стрелок, расположенных на концах линейки.

Основные события для линеек прокрутки:

Scroll (Прокрутка) непрерывно генерируется при перемещении (мышью) движка по линейке.

Change (Изменение). Это событие возникает после изменения позиции движка.

Задание 6. Метроном.

1. Создать форму, как на рисунке 11, переименовать ее.

2. Нанести на форму две Командные Кнопки. Изменить их свойства в соответствии с рисунком.

3. Нанести на форму Горизонтальную линейку прокрутки, изменив ее свойства.

5. Поместить на форму Таймер, задать свойство Interval.

4. Написать программу, выполняющую следующие функции:

- при нажатии на Кнопку «Старт» движок Линейки прокрутки начинает менять свое положение и перемещаться каждую секунду из крайнего левого до крайнего правого положения;
- при нажатии на Кнопку «Стоп» движок линейки прекращает свое движение;
- при нажатии на Кнопку «Выход» программа завершает свою работу.

Выполнение задания 6.

Первая часть – *визуальное программирование.*

1. Необходимо создать форму и переименовать ее в соответствии с рисунком 11, свойству Caption (название) присвоив значение «Метроном».

2. Нанести на форму Горизонтальную Линейку прокрутки, изменив свойства движка Min и Max на –1 и 1 соответственно, свойство Value на 0.

3. Нанести на форму Командные кнопки 1 и 2, изменив их размеры и свойство Font, а также свойство Caption на «Старт» и «Стоп», соответственно.

4. Нанести на форму объект Timer.

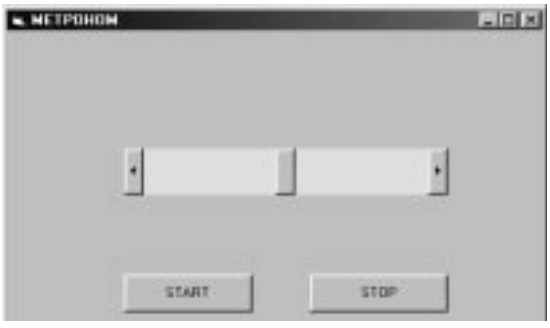


Рисунок 11

Вторая часть – *написание кода программы*.

5. В данном приложении будет использоваться переменная-счетчик k целого типа, поэтому требуется сделать ее описание в процедуре Declarations объекта General.

```
Dim k
```

6. Второе событие после загрузки формы – щелчок на кнопку «Старт». В процедуре обработки данного события свойству Interval объекта Timer присвоить значение 1000 (1 сек), чтобы «включить» таймер.

```
Private Sub Command1_Click()
Timer1.Interval = 1000
End Sub
```

7. Следующее событие Timer применимо к объекту Timer. Событие наступает, когда исчерпывается интервал времени в 1сек. В процедуре обработки данного события необходимо увеличить счетчик на 1 и проверить его на четность. При четном значении переместить движок Линейки прокрутки влево, при нечетном – вправо, сопровождая перемещение движка звуковым сигналом (BEEP).

```
Private Sub Timer1_Timer()
k = k + 1
ost = k Mod 2
If ost = 1 Then
  HScroll1.Value = HScroll1.Max
Else
  HScroll1.Value = HScroll1.Min
End If
Beep
End Sub
```

8. Следующее событие – щелчок на кнопку «Стоп». В процедуре обработки данного события необходимо завершить программу оператором END.

Рисунок 12

```
Private Sub Command2_Click()
End
End Sub
```

9. Запустить программу на выполнение, протестировать ее и завершить.

Задание 7. Скорость.
(для самостоятельного выполнения)

1. Создать форму, как на рисунке 12, переименовать ее.

2. Нанести на форму Командную Кнопку. Изменить ее свойства в соответствии с рисунком.

3. Нанести на форму Горизонтальную линейку прокрутки, изменив ее свойства в соответствии с рисунком.

6. Поместить на форму Текстовое окно, задав такие свойства, чтобы начальное значение скорости было 50 км/час.

4. Написать программу, выполняющую следующие функции:

- при изменении движка Линейки прокрутки начинает меняться значение скорости в Текстовом окне, в крайнем левом положении достигая нулевого значения, а в крайнем правом – 100 км/час;
- при нажатии на Кнопку «Выход» программа завершает свою работу.

НАШИ АВТОРЫ

Паньгина Нина Николаевна,
преподаватель ОИиВТ
школы-лицея № 8, г. Сосновый Бор