# Информатика

Введение в программирование на Visual Basic.

Лекция 2.

Стандартные элементы интерфейса

- **Элементы интерфейса** – это базовые объекты, из которых строятся приложения в VB. С их помощью решается большинство задач взаимодействия пользователя с программой путем установления требуемых значений свойств объектов и обращения к их методам
- VB позволяет обращаться к элементам интерфейса не только во время проектирования приложения, но и во время выполнения программы как к обычным переменным. При обращении программе к свойству или методу объекта сначала записывается его имя, а затем через точку имя этого свойства или метода.
- Например:

cmdCommand1.Caption = “Ok”

присвоение свойству Caption элемента интерфейса cmdCommand1 значения Ok.

frmForm1.Show

обращение к методу Show (отображение формы на экране) объекта (формы) frmForm1.

# Группы стандартных элементов

- Поля ввода-вывода информации
- Элементы управления процессом выполнения программы
- Элементы оформления
- Списки
- Полосы прокрутки
- Элементы интерфейса, связанные с файловой системой компьютера
- Прочие элементы

# Свойства, общие для большинства объектов

| Категория | Свойство | Название |
|---|---|---|
| Appearance (внешний вид) | **Appearance**<br>**BackColor**<br>**BorderStyle**<br>**Caption**<br>**ForeColor** | Определяет стиль объекта на экране (трехмерный или плоский)<br>Определяет цвет фона объекта<br>Определяет особенности границы<br>Задает надпись на объекте<br>Определяет цвет выводимого на объекте изображения |
| Behavior (поведение) | **Enabled**<br>**Visible** | Определяет возможность реагирования объекта на действия пользователя (доступен или недоступен)<br>Определяет видимость объекта во время исполнения программы (отображается или не отображается) |
| Font (шрифт) | **Font** | Устанавливает параметры шрифта (вид, размер и др.) |
| Pisition (позиция) | **Left**<br>**Top**<br>**Width**<br>**Height** | Устанавливает положение левой границы объекта<br>Устанавливает положение правой границы объекта<br>Устанавливает ширину объекта<br>Устанавливает высоту объекта |

# Методы, общие для большинства объектов

| Название | Назначение метода |
|---|---|
| **Drag** | Поддержка операции перетаскивания объектов в пределах контейнеров<br>(контейнер – объект, содержащий в себе другие объекты, например, форма ) |
| **Move** | Изменение положения объектов |
| **SetFocus** | Активизация объекта (получение объектом фокуса)<br>Фокус-понятие операционной системы, с помощью которого указывается, какой элемент активен. Если элемент получает фокус, то это соответствующим образом отображается на экране (например, в текстовом поле мигает курсор) |

# События, общие для большинства объектов

| Событие | Источник возникновения события |
|---|---|
| **Change** | Изменение значения текстового поля или выбор элемента в поле со списком |
| **Click** | Щелчок мышью по объекту |
| **DbClick** | Двойной щелчок мышью по объекту |
| **DragDrop** | Перемещение объекта на другое место |
| **DragOver** | Перемещение объекта поверх на другого объекта |
| **GotFocus** | Получение объектом фокуса |
| **KeyDown** | Нажатие клавиши, когда объект находится в фокусе |
| **KeyPress** | Нажатие и отпускание клавиши, когда объект находится в фокусе |
| **KeyUp** | Отпускание клавиши, когда объект находится в фокусе |
| **LostFocus** | Потеря объектом фокуса |
| **MouseDown** | Нажатие основной кнопки мыши, когда указатель помещен на объект |
| **MouseMove** | Перемещение указателя за пределы объекта |
| **MouseUp** | Отпускание основной кнопки мыши, когда указатель помещен на объект |

# Формы. Основные свойства

- **BorderStyle** определяет особенности границы окна и возможности управления им. Может иметь шесть значений, в зависимости от которых окно может иметь способность изменять или не изменять размеры, быть перемещенным или не перемещенным по экрану и т.д.
- **Icon** задает значок пиктограммы.
- **WindowState** задает состояние окна при его запуске. Может иметь три значения:
  - Normal окно открывается в нормальном состоянии;
  - Minimized окно открывается в свернутом состоянии;
  - Maximized окно открывается развернутым на весь экран.

# Формы

**Основные методы:**

- **Show** отображает форму на экране (делает ее видимой)
- **Hide** удаляет форму с экрана (делает ее невидимой)
- **Unload** выгружает форму из памяти (удаляет ее как объект)

**Основные события:**

- **Load** происходит при загрузке формы в память. Используется для инициализации объектов и переменных, принадлежащих форме
- **Unload** происходит при выгрузке формы из памяти
- **GotFocus** происходит при получении формой фокуса (когда форма загружается или пользователь обращается к ней)

# Обработка событий, связанных с формами

- Синтаксис процедуры обработки события формы отличается от синтаксиса процедур обработки событий других элементов интерфейса
- Имя процедуры обработки события формы всегда содержит слово “Form”. При этом не важно, как фактически называется форма
- Например, процедура обработки события, которое заключается в загрузке формы в оперативную память, имеет вид

```
Private Sub Form_Load()
……..
End Sub
```

# Текстовой поле (TextBox). Свойства

- **Locked** - устанавливает возможность редактирования текста в поле.
- **MaxLength** - ограничивает количество символов, вводимых в поле.
- **Multiline** - определяет возможность ввода в поле многострочного текста.
- **PasswordChar** - задает символ, который отображается в поле вместо введенного символа.
- **ScrollBars** - определяет наличие полос прокрутки в текстовом поле.
- **Text** - задает содержимое поля.
- **SelStart** - задает начальный символ выделенного текста
- **SelLength**- задает конечный символ выделенного текста
- **SelTex** - возвращает содержимое выделенного фрагмента текста

# Текстовое поле (TextBox)

**Методы:**

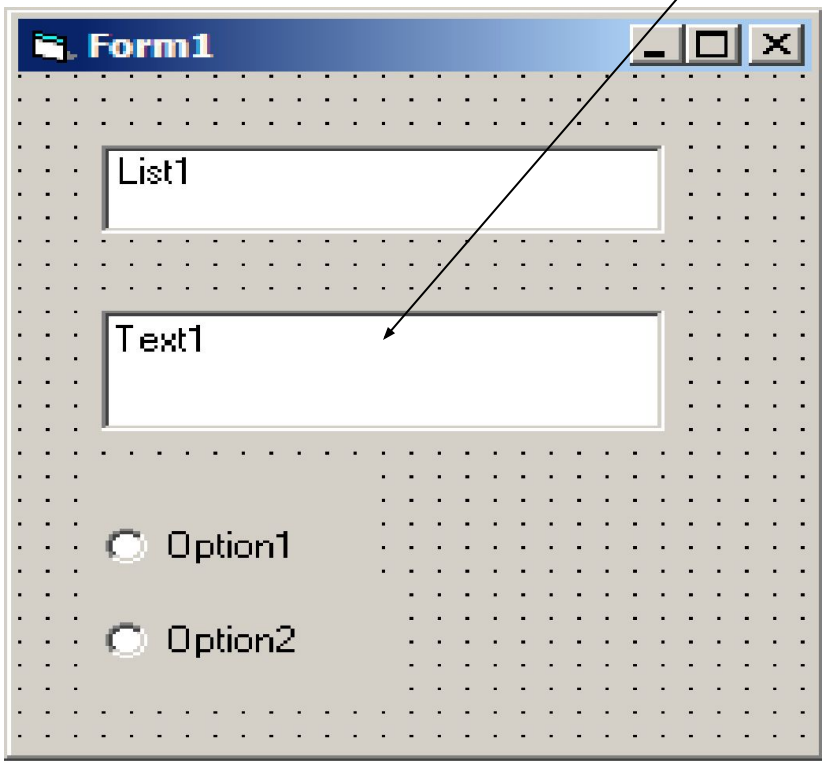
- **SetFocus** - активизирует (устанавливает фокус) поле (или устанавливает на нем курсор)

**События:**

- **Change** - изменение значения текстового поля. Происходит каждый раз, когда вводится, удаляется или изменяется символ в поле.
- **GotFocus** - получение полем фокуса

# Пример

**Текстовой поле - TextBox**



# Надпись (Label)

- Предназначена для отображения (вывода) информации. Обычно используется в качестве надписей для пояснения других элементов интерфейса

**Свойства:**

- **Caption** - важнейшее свойство, определяет текст надписи.
- **AutoSize** - устанавливает возможность автоматической регулировки размера надписи по горизонтали.
- **Font** - задает шрифт надписи.
- **TabIndex** - определяет порядок перебора элементов интерфейса при нажатии клавиши **Tab**.
- **WordWrap** - устанавливает возможность автоматической регулировки размера надписи по горизонтали. Работает только тогда, когда свойство **AutoSize** имеет значение **True**

**События и методы надписей обычно не используются**

# Кнопки (CommandButton). Свойства

- **Default** - задает реакцию кнопки на нажатие клавиши **Enter**. Если это свойство кнопки имеет значение **True**, то нажатием клавиши **Enter** генерируется событие **Click** данной кнопки, независимо от того какой элемент интерфейса имеет фокус (обычно это кнопки “**Ok**”). Если на форме находится несколько кнопок, то только у одной из них свойство **Default** может иметь значение **True**.
- **Cansel** - задает реакцию кнопки на нажатие клавиши **Esc**. Если это свойство кнопки имеет значение **True**, то нажатием клавиши **Esc** генерируется событие **Click** данной кнопки, независимо от того какой элемент интерфейса имеет фокус (обычно это кнопки “**Cansel**”). Если на форме находится несколько кнопок, то только у одной из них свойство **Cansel** может иметь значение True.
- **Enabled** - определяет возможность обращения пользователя к кнопке. Если это свойство имеет значение **False**, то кнопка недоступна.
- **TabIndex** - определяет порядок перебора элементов интерфейса при нажатии клавиши Tab.
- **Visible** - определяет видимость кнопки. Если это свойство имеет значение False, то кнопка на форме не отображается (невидима)

# Кнопки (CommandButton)

Методы

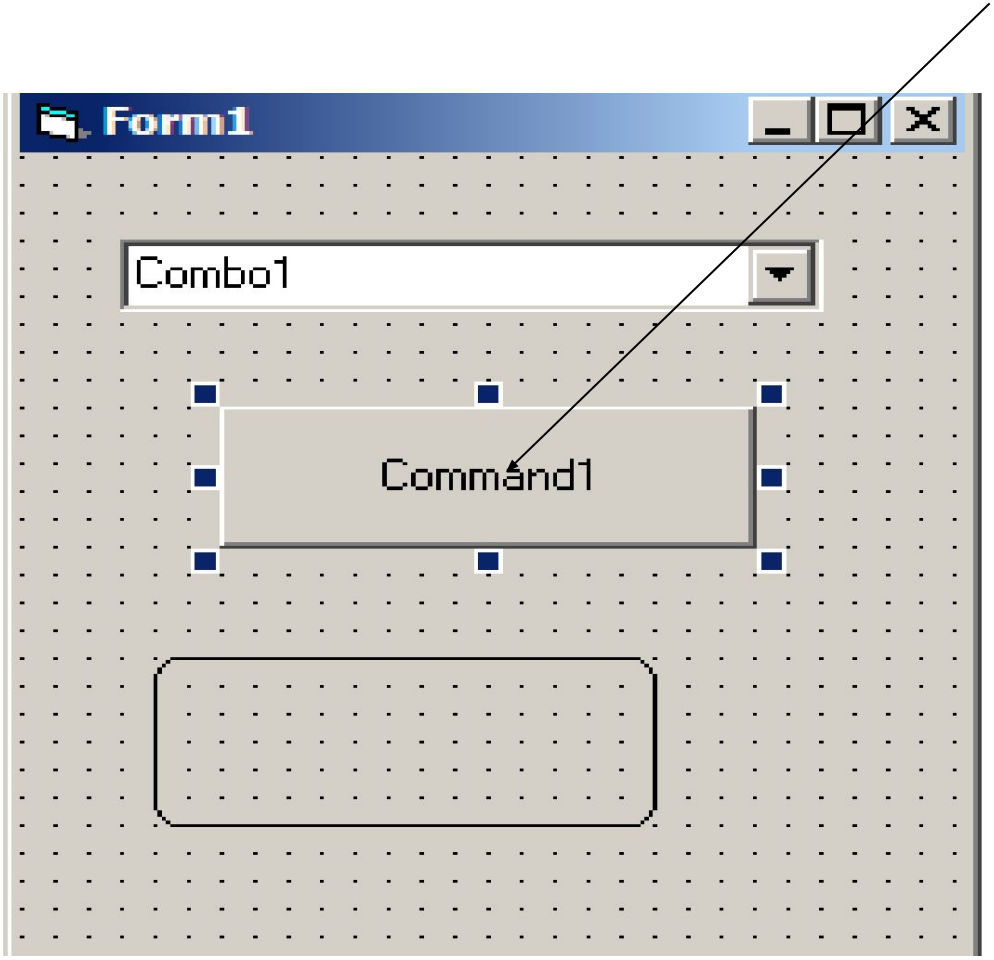
- **SetFocus** - активизирует кнопку.

События

- **Click** - щелчок мышью по кнопке.
- **GotFocus** - получение кнопкой фокуса.
- **MouseDown** - нажатие основной кнопки мыши, когда указатель помещен на объект (кнопку).
- **MouseUp** - отпускание основной кнопки мыши, когда указатель помещен на объект (кнопку).
- **MouseMove** - перемещение указателя мыши за пределы кнопки

# Пример

**Кнопка - CommandButton**



# Переключатель (OptionBox)

- **Позволяет выбрать один (и только один) вариант из нескольких взаимоисключающих вариантов.**

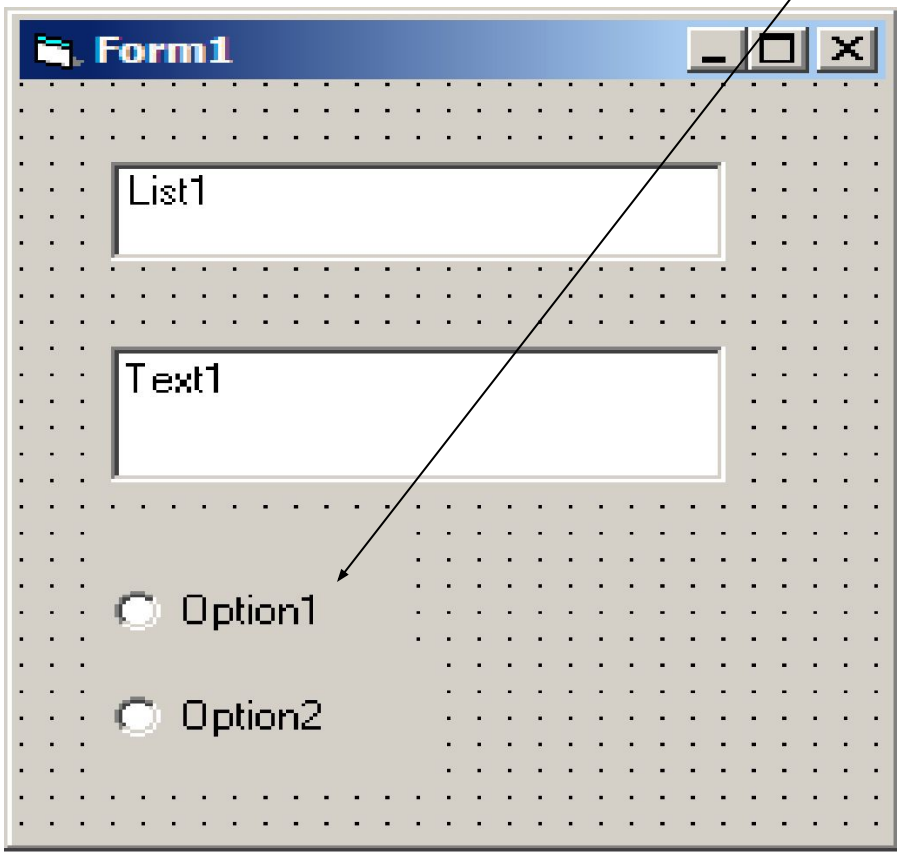
Свойства переключателей:

- **Value** - задает состояние переключателя.

Методы переключателей практически не используются. Из событий в основном используется только событие **Click**

# Пример

**Переключатель - OptionBox**



# Флажки (CheckBox)

**Флажки позволяют производить выбор нескольких вариантов из множества предложенных.**

Свойство **Value** флажка может иметь три значения:

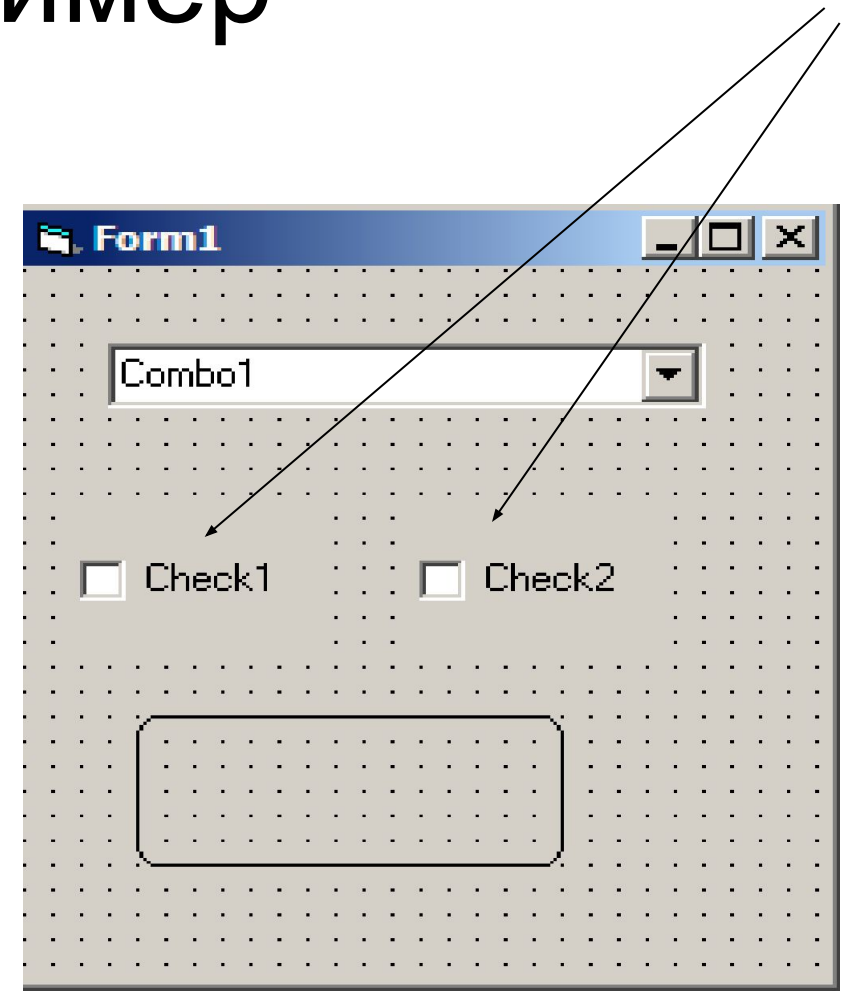
- 1 - флажок установлен;
- 0 - флажок снят;
- 2 - флажок находится в неопределенном состоянии. Это состояние используется для того, чтобы сообщить пользователю о том, что в программе есть еще одна группа флажков, причем некоторые из них (но не все) уже установлены.

Методы флажков практически не используются

Из событий в основном используется только событие **Click**

# Пример

**Флажок - CheckBox**

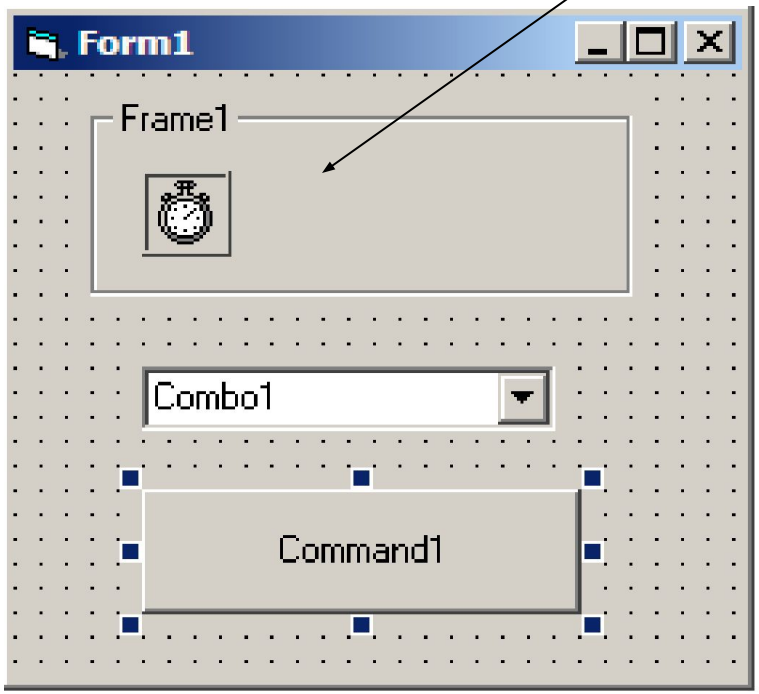


# Рамка (Frame)

- **Рамка служат для объединения в группу нескольких других элементов. Объекты, объединенные с помощью рамки, можно как единое целое перемещать, активизировать, делать видимыми или невидимыми**
- Для того, чтобы какой-либо элемент располагался внутри рамки, нужно сначала на форме разместить саму рамку, и только потом поместить в ней этот элемент. Чаще всего рамки используются для группировки переключателей. Когда в группе устанавливается один переключатель, то все остальные переключатели группы автоматически снимаются. Если же переключатели разместить на форме без рамок, то они будут действовать как одна большая группа
- Рамка – элемент интерфейса, который особых, присущих ему свойств не имеет. События и методы рамок обычно не используются

# Пример

**Рамка - Frame**



# Графические средства VB

Они применяются для следующих целей:

- Для выделения отдельных элементов интерфейса
- Для обеспечения отображения графической информации (рисунков, схем, графиков, диаграмм)
- Графические изображения в VB создаются двумя способами:
  - с помощью элементов интерфейса или
  - с использованием графических методов

  Графические элементы интерфейса работают подобно другим элементам. Их можно подключать во время разработки программы, задавая требуемые значения

  Графические методы – это функции, которые встроены в язык VB и вызываются во время выполнения программы

# Линия (Line) и фигура (Shape)

- **Элементы Line и Shape представляют собой средства добавления в форму простейших графических примитивов: линий и фигур**
- Во время разработки программы они помещаются в нужном месте формы. Во время выполнения программы эти элементы могут скрываться или перемещаться

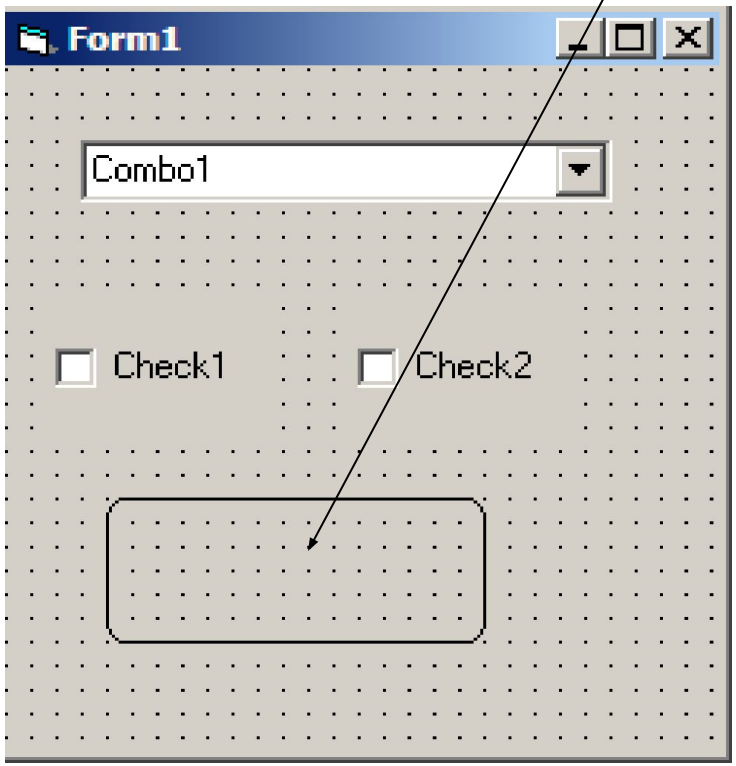
# Свойства

- Свойство **BackStyle** задает способ отображения примитива. Может иметь два значения:
  - 0 (Transparent) фон отображаемого примитива прозрачный (отображается только контур примитива);
  - 1 (Opaque) фон отображаемого примитива непрозрачный. В этом случае с помощью свойства BackColor можно устанавливать цвет фона.
- Свойство **Shape** элемента Shape может иметь шесть значений, определяющих вид фигуры:
  - 0 (Rectangle)- прямоугольник;
  - 1 (Square) - квадрат;
  - 2 (Oval) - овал;
  - 3 (Circle) - круг;
  - 4 (Rounded Rectangle) - прямоугольник со скругленными углами;
  - 5 (Square Rectangle) - квадрат со скругленными углами

  События линий и фигур, как правило, не обрабатываются. Методы не используются

# Пример

**Графический объект – Shape**



# Графическое поле (PictureBox) и рисунок (Image)

- **Элементы используются для отображения на формах изображений, подготовленных в графических редакторах.**
- С их помощью могут быть отображены изображения, хранящиеся в файлах следующих типов:
  - Bmp –файлы растровой графики;
  - Ico – пиктограммы;
  - Wmf – метафайлы (файлы векторной графики);
  - Jpg, jpeg – графические файлы.

- Элементы PictureBox и Image во многом подобны, но работа с ними производится по-разному. Рисунки (Image) занимают больше памяти и быстрее отображаются. Графические поля (PictureBox) обеспечивают более высокое качество изображения. Кроме того, во время выполнения программы в них можно выводить текст и рисовать различные примитивы (линии, окружности и т.п.). с помощью методов графического поля.
- В графическом поле можно можно располагать другие элементы, т.е. он также как рамка является контейнером. Фактически элемент Image – это очень упрощенный вариант элемента PictureBox. Поэтому, несмотря на то, что у них много общих параметров их количество и состав для этих элементов различны

**Свойства:**

- Picture - указывает на файл, содержимое которого будет выводиться.
- Stretch - определяет возможность масштабирования графического изображения (только для элемента Image).
- AutoSize - определяет возможность автоматического изменения размеров графического поля под размеры выводимого графического изображения (только для элемента PictureBox).

**Методы графических полей предназначены для вывода примитивов и текста в режиме выполнения программы, к ним относят:**

- Line - рисует линию или прямоугольник.
- Circle - рисует окружность, эллипс или их дуги.
- Print - выводит текст. При вызове метода указывается текстовая строка, подлежащая выводу. Например: Print “Текстовая строка”

# Список (ListBox)

- **Элемент список используется для работы с перечнем записей.**
- Пользователь может просмотреть перечень, выбрать одну или несколько строк для последующей обработки.
- Непосредственное редактирование записей невозможно

# Свойства

- Строки списков хранятся в виде массива. Свойство List задает содержимое строк, входящих в этот массив, т.е. в список.
- ListCount - определяет количество строк в списке (свойство доступно только во время выполнения программы).
- ListIndex - определяет номер выделенной строки (доступно только во время выполнения программы).
- Text - определяет содержимое выделенной строки (доступно только во время выполнения программы).
- Sorted - задает сортировку списка по алфавиту.
- MultiSelect - задает возможность выбора нескольких строк

# Методы

- При размещении элемента список на форму он пуст, т.е. не имеет элементов
- Элементы списка добавляются с помощью метода **AddItem**. Оператор обращения к этому методу в программе имеет вид:

  ListBox.AddItem <Строка> [, <Индекс>]
- С помощью параметра Индекс указывается место в список новой строки. Индексация строк начинается с нуля. Данный метод должен вызываться при создании каждой строки. Как правило, заполнение списка производится при загрузке формы, т.е. операторы обращения к этому методу записываются в процедуре обработки события **Load** формы (Sub Form_Load).
- Метод RemoveItem - удаляет строку из списка. Оператор обращения к методу имеет вид:

  ListBox. RemovItem <Индекс>
- Метод Clear - очищает список (удаляет все строки из списка).

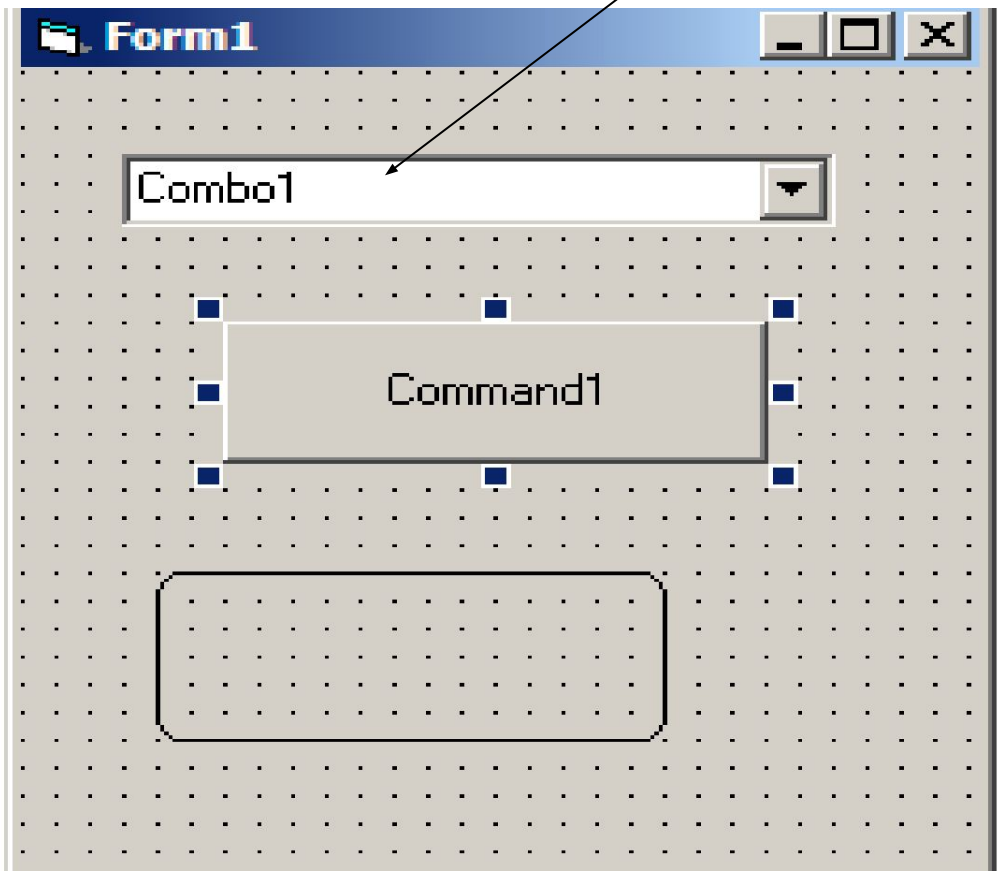
# События

- Click - щелчок мышью по кнопке.
- DblClick - двойной щелчок мышью по кнопке

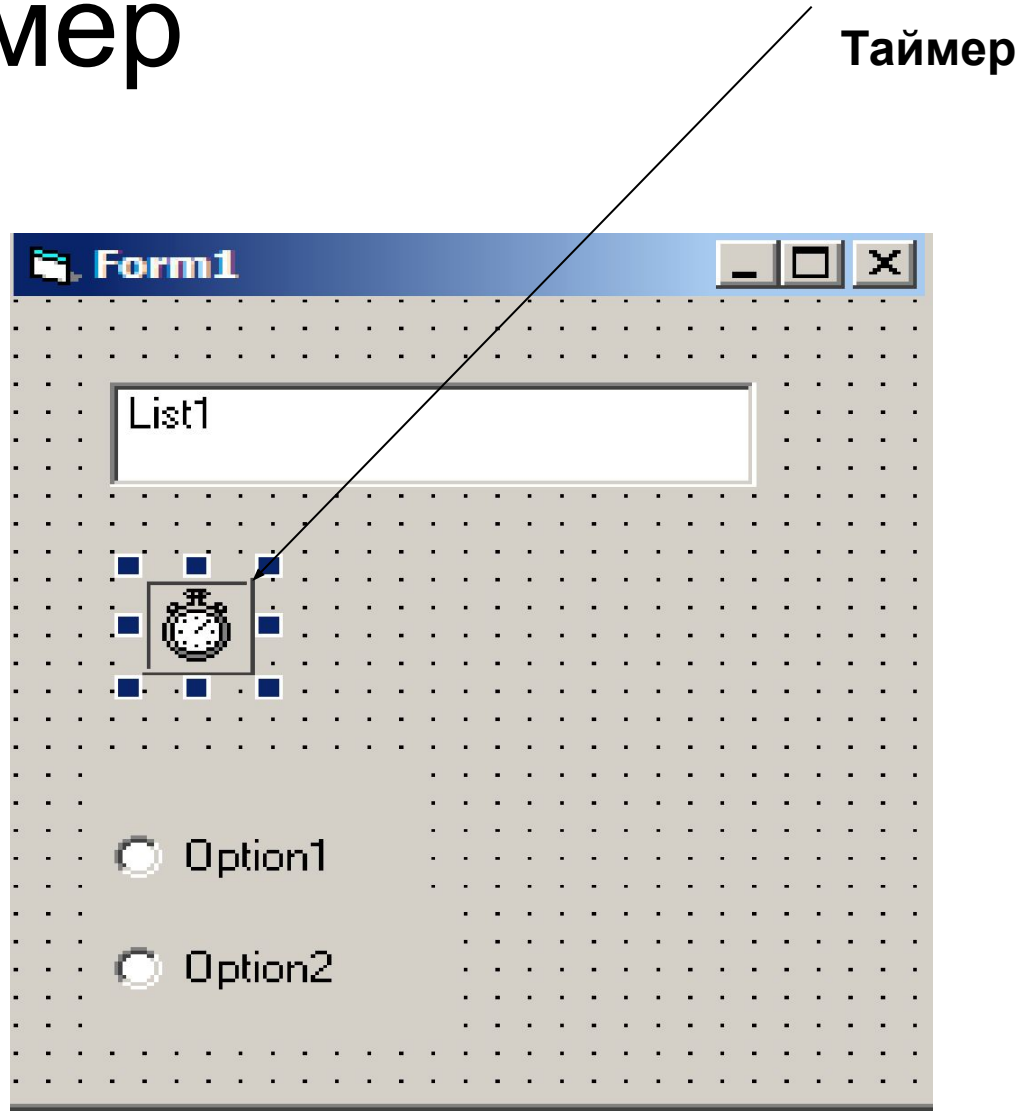
# Комбинированное поле (ComboBox)

- **Используется для работы с перечнем записей. В отличие от элемента ListBox элемент ComboBox позволяет не только выбирать строку списка, но и вводить ее непосредственно в поле ввода, после чего она автоматически помещается в список.**
- Имеется три разновидности этого элемента, которые определяются значением свойства Style.
  - 0 (DropDown Combo) - текстовое поле для редактирования и раскрывающийся список. Это значение принимается по умолчанию.
  - 1 (Simple Combo) - текстовое поле для редактирования и постоянно открытый список.
  - 2 (DropDown List) - поле с раскрывающимся списком. Отличается от первой разновидности тем, что пользователь не имеет возможности вводить текст в поле ввода.
- По своим свойствам, событиям и методам комбинированные поля подобны спискам.

# Пример



# Таймер (Timer)

- **Основное назначение элемента Timer – выполнение заданной последовательности действий по истечении установленного промежутка времени.**
- Требуемый промежуток времени (измеряемый в миллисекундах) задается с помощью свойства **Interval**, значения которого могут находиться в диапазоне от 0 (отключение таймера) до 65535. Как только заданный свойством Interval промежуток времени пройдет, наступает событие Timer, и выполняется процедура обработки этого события.
- Элемент Timer никогда не отображается на экране в режиме выполнения программы. Поэтому он может располагаться на форме где угодно

# Пример