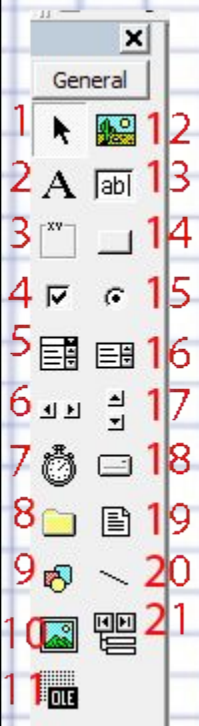


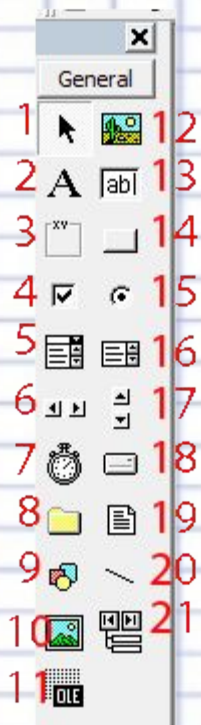


ОСНОВЫ *Visual Basic*



№ рис.	Название	Назначение
1	Pointer	Указатель для выбора активного объекта
2	Label	Метка для вывода текста Command Button – Командная кнопка – элемент управления для выполнения программных откликов на события
3	Frame	Рамка – объект группировки
4	CheckBox	Флажок – элемент выбора одной из нескольких альтернатив
5	ComboBox	Комбинированный список – выбор элементов из списка или ввод новых
6	HScrollBar	Полосы прокрутки – горизонтальная полоса
7	Timer	Таймер – использование функции времени
8	DirListBox	Список каталогов – окно списка каталогов
9	Shape	Фигура – геометрические фигуры 5 типов
10	Image	Картинка – графическое изображение
11	OLE	Контейнер – организация связи с другими приложениями





№ рис.	Название	Назначение
12	PictureBox	Графическое окно – для вывода графических изображений
13	TextBox	Текстовое окно – для ввода, вывода и редактирование текста
14	CommandButton	Командная кнопка – выполнение программных функций
15	OptionButton	Кнопка переключатель – элемент выбора одной из нескольких альтернатив
16	ListBox	Простой список – выбор элементов из списка или заполнение списка в процессе работы приложения
17	VScrollBar	Полосы прокрутки – вертикальная полоса
18	DriveComboBox	Список логических дисков – окно выбора диска
19	FileListBox	Список файлов – окно списка файлов – выводит список файлов в выбранном каталоге
20	Line	Линия – линии и прямоугольники
21	Data	Данные – осуществляет связь с файлами баз



Чтобы объект выполнил какую-либо операцию, необходимо задать метод (описать оператор, команду). Для применения некоторого набора команд, возможных для данного объекта, применим следующий синтаксис:

<объект> . <метод> [= <значение>]

Например:

CommonDialog. ShowOpen – активизировать метод **ShowOpen** для открытия файлов с помощью **CommonDialog**



Для внесения изменений в состояние объекта следует определить новые значения его свойств. В программном коде для изменения свойств объекта принято использовать следующий синтаксис:

<Объект> . <свойство> = значение

Например:

`Shape1.BackColor = vbYellow` – назначить свойство `vbYellow` – желтый цвет для графического объекта `Shape1`.



Язык программирования Visual Basic



Операторы



Операторы

Оператор – последовательность команд, из которых состоит код программы

Простые

Структурированные

Не содержит в себе других операторов

Строятся из других операторов



Комментарии

Используются для пояснения отдельных конструкций.

Строка комментария начинается со слова

Rem

Или со знака « ' » (апостроф), далее следует пробел.

Комментарии можно вставлять в любом месте процедуры.



Идентификаторы

При присвоении идентификатора следует придерживаться следующих правил:

- Имя должно начинаться с буквы
- Имя не может содержать пробел, точку, восклицательный знак, а так же символы:
@, &, \$, #
- Имена не должны содержать более **255** символов
- Не следует использовать имена, совпадающие с названием функций, инструкций и методов языка Visual Basic, так как при этом прекращается выделение в тексте одноименных ключевых слов языка
- Нельзя использовать одни и те же имена переменных, отличающихся только типом
- Не допускается использование повторяющихся имен на одном уровне области определения
- Регистр значения не имеет



Константа

Параметр, значение которого в процессе выполнения программы остается постоянным



Переменная

Параметр, значение которого в процессе выполнения программы может принимать различные значения



Данные

**Информация, представленная в
формальном виде, который
обеспечивает возможность ее
хранения, обработки, передачи**

Объявление данных

В Visual Basic используется 14 стандартных типов данных

Название	Тип	Интервал
Boolean	Логические данные	True/false
Byte	Целые положительные числа	От 0 до 255
Date	Дата	
Double	Тип данных с двойной точностью	
Integer	Целые числовые значения	
Long	«длинные целые числа»	
Single	Числовые значения с плавающей точкой	
String	Символьные (строковые) данные	
Variant	Для хранения любых	



Синтаксис конструкции Dim

Для объявления типа индекса лучше всего использовать оператор Dim, который служит для объявления типов переменной.

Dim <имя переменной1> [(<индексы>)] [As <тип>] [,<Имя переменной2> [(<индексы>)] [As <тип>]]

Например

Dim Y As Integer

Dim Amt As Single

Объявления можно комментировать в одной строке

Dim Y As Integer, Amt As Single

Dim Result – без типа присваивает переменной тип Variant

Данные, объявленные в **Dim**, доступны только на уровне модуля формы или той процедуры, где были объявлены. Для определения области видимости данных используются так же **Public** и **Private**. Эти синтаксисы аналогичны синтаксису **Dim**

Public – глобальная (открытая) переменная. Данные доступны всему приложению

Private - данные доступны на уровне модуля формы.