

Основы программирования на Visual Basic

A series of horizontal lines in teal and light blue colors, with varying lengths and slight offsets, creating a modern, layered effect across the middle of the slide.

Visual Basic - это объектно-ориентированный язык. Основой языка являются объекты. Например: окно, кнопка, поле со списком, с которыми работает программа.

Достоинства

- ✓ Высокая скорость создания приложений с графическим интерфейсом для MS Windows.
- ✓ Простой синтаксис, позволяющий очень быстро освоить язык.
- ✓ Возможность компиляции как в машинный код, так и в Р-код. В режиме отладки программа всегда компилируется в Р-код, что позволяет приостанавливать выполнение программы, вносить значительные изменения в исходный код, а затем продолжать выполнение: полная перекомпиляция и перезапуск программы при этом не требуется.
- ✓ Защита от ошибок, связанных с применением указателей и доступом к памяти. Этот аспект делает Visual Basic приложения более стабильными, но также является объектом критики.
- ✓ Возможность использования большинства WinAPI функций для расширения функциональных возможностей приложения. Данный вопрос наиболее полно исследован Дэном Эпплманом, написавшим книгу «Visual Basic Programmer's Guide to the Win32 API».

Недостатки

- ✓ Поддержка операционных систем только семейства Windows и Mac OS X.
- ✓ Отсутствие механизма наследования реализации объектов. Существующее в языке наследование позволяет наследовать только интерфейсы, но не их реализацию.
- ✓ Требуется установленную `msvbvmXX.dll` для работы программы.
- ✓ Низкая скорость работы, обусловленная тем, что практически все встроенные функции языка реализованы через библиотеку времени исполнения, которая, в свою очередь, производит много «лишней» работы по проверке и/или преобразованию типов.

Каждый объект имеет:
свойства
методы
события

Свойства - это показатели, характеризующие объект.

Методы - это действия, которые можно произвести с объектом.

События - это действия, которые происходят с объектом.

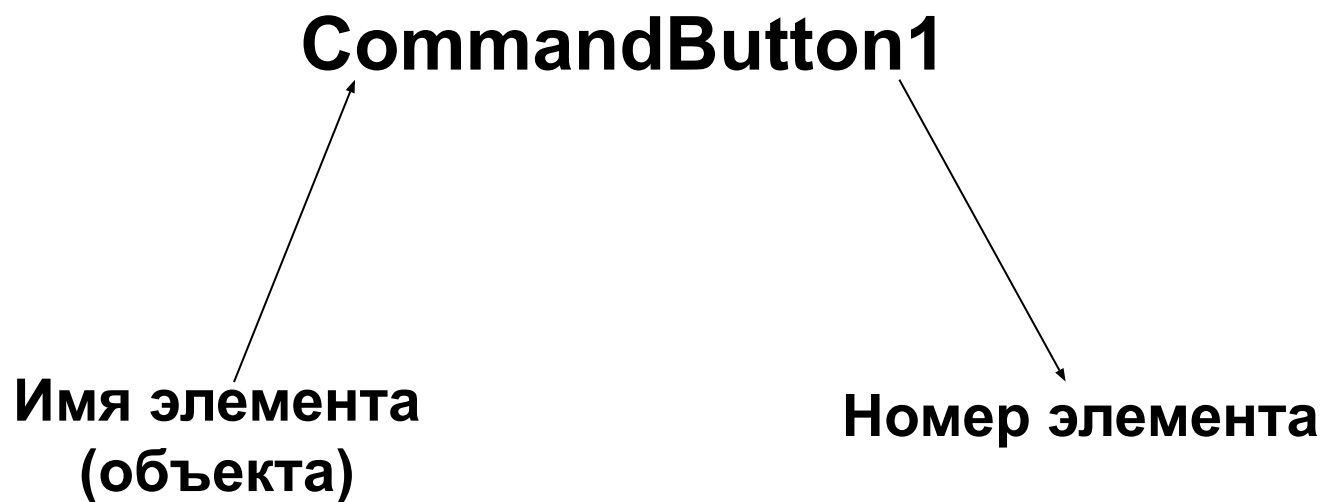
Управляющие элементы

- Текстовые поля (TextBox) - **txt**
- Метки (надписи) (Label) - **lbl**
- Графические окна (PictureBox) - **pic**
- Командные кнопки
(CommandButton) - **cmd**

Форма - это

Основной объект (элемент),
на котором могут
располагаться другие объекты

Командная кнопка



Переменные

- Предназначены для хранения и обработки данных.
- Переменные задаются **именами**.
- Имя определяет область памяти, в которой хранится **значение** переменной.
- Значениями переменных могут быть данные различных типов (числа, символы, логические значения и т.д.).

Тип переменной

- Зависит от типа данных, которые могут быть значениями переменных.

Тип переменной	Возможные значения	Приставка к имени
Byte	Целые числа от 0 до 255	byt
Integer	Целые числа от -32768 до 32767	int
Long	Большие целые числа	lng
Single	Десятичные числа одинарной точности	sng
Double	Десятичные числа двойной точности	dbl
String	Строка символов	str
Date	Дата	dtm
Boolean	Логическое значение True или False	bln

Тип переменной

- Над различными типами данных можно выполнять различные операции.
- **Числовые переменные** – арифметические операции.
- **Строковые переменные** – операции преобразования символьных строк.
- **Логические переменные** – логические операции.

Имя переменной

- Имя переменной уникально и не может меняться в процессе выполнения программы.
- Может состоять из различных символов (латинские и русские буквы, цифры и др.).
- Должно обязательно начинаться с буквы и не должно включать знак «.» (точка).

Имя переменной

- **Примеры имен:**
 - числовые: A, D, B1, Число и др.;
 - строковые: A, D, S, Слово и др.
- В Visual Basic в имя переменной включают особую **приставку**, которая обозначает тип переменной.
- **Например:**
 - числовые: intA, bytD, intB1, sngЧисло и др.;
 - строковые: strA, strD, strS, strСлово и др.

Объявление типа переменной

- Необходимо чтобы компьютер при исполнении программы мог знать тип переменной и сколько памяти выделить для хранения переменной.
- **Dim** **ИмяПеременной** **As** **ТипПеременной**
- Пример объявления двух переменных:
- **Dim** intA **As** **Integer**, strСтрока **As** **String**

Арифметические выражения

- В состав арифметических выражений могут входить переменные числового типа и числа.
- Над переменными и числами могут производиться различные арифметические и математические операции (функции).

Строковые выражения

- Строка – любая последовательность символов, заключенная в кавычки.
- Например: «информатика», «2008», «3+3».
- Операция **конкатенации** – объединение строки или значения строковых переменных в единую строку.
- Обозначается знаком «+».
- **Пример:** «теле»+«визор»=«телевизор»,
- «2»+«8»=«28»

Ввод данных в программу

- Можно вводить данные к текстовое поле.

Функция **Val** - преобразование текста в число.

- Функция вызова окна ввода

InputBox (Приглашение, Заголовок)

Вывод данных

- Вывод данных на форму:
- **Frm.Print “Комментарий”
<Переменная>**
- Вывод данных в текстовое поле
- **Txt1.Text = <выражение>**

Объект "телефон"



Свойства:	Методы:	События:
цвет размер расположение вес объем	снять трубку набрать номер передвинуть телефон	звонок длинный гудок короткий гудок

В проектируемом приложении события будут возникать в основном в результате действий пользователя. Например, пользователь нажал на кнопку мыши или клавиатуры, запустил или завершил программу. Часто используется метод **Move** - переместить объект (кнопку, форму), который меняет свойства объекта, такие как расстояние от левого и верхнего края экрана до объекта.

Практическое задание №1

Выберите в реальной жизни любой объект (*холодильник, магнитофон, машина*) и опишите его **свойства, методы, события**.

Выберите любой объект на экране компьютера (*кнопка, окно, текстовое поле*) и опишите его **свойства, методы, события**.

Практическое задание №2

Откройте среду программирования **Visual Basic**, создайте новый проект **Стандартный EXE**.

В окне среды программирования найдите все элементы, уберите их с экрана и выведите вновь, переместите элементы, подберите их оптимальные размеры.

Добавьте новую форму, обратите внимание на ее имя. Закройте и откройте окна форм, измените надписи в заголовках форм:

у первой **Обычный калькулятор**

у второй **Инженерный калькулятор**

Откройте и закройте окна кода первой формы, затем - второй.

Запустите программу. Какая форма появится? Сделайте так, чтобы при запуске появлялась другая форма.

Удалите вторую форму из проекта, не сохраняя ее.

Сохраните проект и форму в отдельной папке. Папку создайте внутри папки Мои документы. В качестве имени папки используйте свою фамилию.