



**Преподавание
программирования в среде
Visual Basic в средней
школе**

Основы объектно-ориентированного визуального программирования

- Приложения на языках объектно-ориентированного программирования Visual Basic строятся из объектов, подобно тому, как из блоков и различных деталей строятся дома.
- Системы объектно-ориентированного программирования дают возможность визуализировать процесс создания графического интерфейса разрабатываемого приложения, то есть позволяют создавать объекты и задавать значения их свойств с помощью диалоговых окон системы программирования.
- Взаимодействие программных объектов между собой и их изменения описываются с помощью программного кода. Создание программного кода в объектно-ориентированном программировании базируется на использовании алгоритмических структур различных типов (линейной, ветвления, цикла), исполнителями которых выступают программные объекты.

Объекты: свойства, методы, события

Программные объекты обладают

свойствами,

могут использовать

методы

и реагируют на

события

Графический интерфейс и событийные процедуры

- Основой для создания графического интерфейса разрабатываемого приложения являются *форма* , представляющая собой окно, в котором размещаются управляющие элементы.
- **Форма – это объект, представляющий собой окно на экране, в котором размещаются управляющие элементы.**
- Визуальное конструирование графического интерфейса приложения состоит в том, что на форму с помощью мыши помещаются и «рисуются» те или иные *управляющие элементы*.

Некоторые классы объектов, их свойства, методы и события

Класс объектов	Свойства	Методы	События
Form (форма)	Name (Имя) Caption (Надпись) Font (Шрифт) Height (Высота) Width (Ширина)	Show(Показать) Move (Переместить)	Load (Загрузка)
CommandButton (командная кнопка)	Name (Имя) Caption (Надпись) Font (Шрифт) Height (Высота) Width (Ширина)	Move (Переместить)	Click (Щелчок)
TextBox (текстовое поле)	Name (Имя) Text (Текст) Font (Шрифт) Height (Высота) Width (Ширина)	Move (Переместить)	DoubleClick (Двойной щелчок)

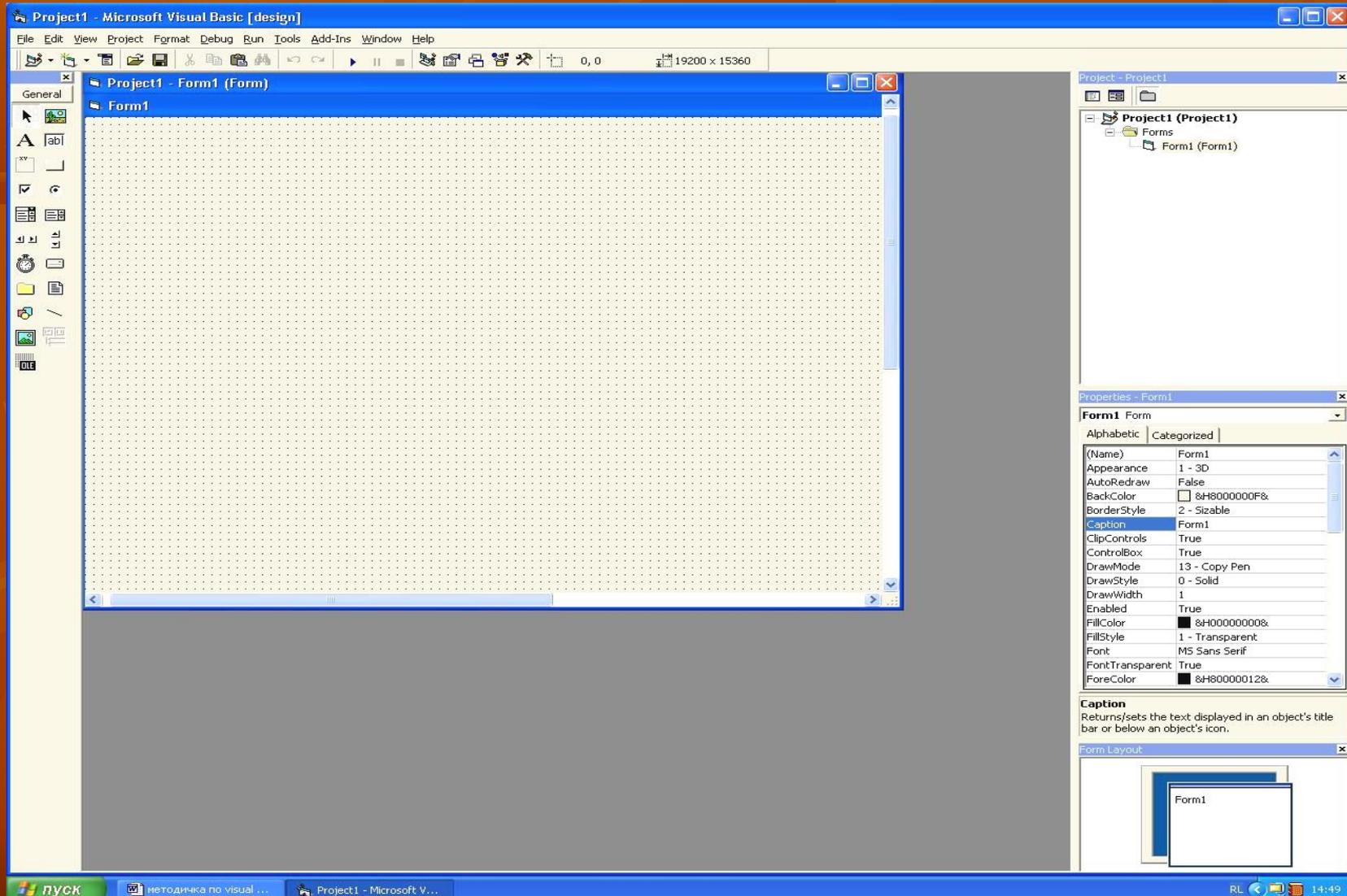
Событийные процедуры

Если пользователь производит какое-либо воздействие на элемент графического интерфейса (например, щелчок), в качестве отклика выполняется некоторая последовательность действий (событийная процедура).

Каждая процедура представляет собой отдельный программный модуль, который реализует определенный алгоритм. В терминологии процедурного программирования такие процедуры соответствуют подпрограммам.

Каждая из событийных процедур начинается с ключевого слова Sub (subroutine - подпрограмма) и заканчивается ключевыми словами End Sub:

Интегрированная среда разработки языка Visual Basic



Типы переменных

Тип переменной	Возможные значения	Объем занимаемой памяти
Byte	Целые числа от 0 до 255	1 байт
Integer	Целые числа от -32768 до 32767	2 байта
Long	Целые числа двойной длины	4 байта
Single	Десятичные числа одинарной точности от $1.401298E-35$ до $3,4022823E38$	4 байта
Double	Десятичные числа двойной точности от $1,94065645841 247E-324$ до $1, 797693 13486232E+308$	8 байтов
Boolean	Логическое значение True или False	2 байта
String	Строка символов	1 байт на каждый символ
Currency	Число в денежном формате	8 байтов
Date	Дата от 1 января 100 г. до 31 декабря 9999г	8 байтов

Для объявления типа переменной используется:

- Dim ИмяПеременной [As ТипПеременной]

Арифметические выражения

Арифметические выражения должны быть записаны в так называемой линейной записи согласно следующим правилам:

- - выражение должно быть записано в виде линейной цепочки символов (надо писать $x1, v0$);
- — для обозначения операции умножения используется звездочка (*), для операции деления — косая черта (/), для операции возведения в степень — (^);
- нельзя опускать знаки операций, например писать $4a$. Для записи произведения чисел 4 и a надо писать $4*a$;
- аргументы функций (\sin, \cos и др.), как и — аргументы вспомогательных алгоритмов, записываются в круглых скобках, например $\sin(x), \cos(4*a)$.

Переменные величины в Visual Basic

Линейный алгоритм

Задача: Скорость первого автомобиля $v_1=5$ км/ч, второго $v_2=3$ км/ч, расстояние между ними $s=12$ км. Какое расстояние будет между ними через $t=5$ ч, если автомобили движутся в разные стороны?

$s_1 = s + (v_1 + v_2)t$ (если автомобили изначально двигались в противоположные стороны) или $s_2 = |(v_1 + v_2)t - s|$ (если автомобили первоначально двигались навстречу друг другу).

В окне *Программный код* ввести событийную процедуру целиком:

```
Dim v1, v2, t, s, s1, s2 As Single
v1 = 5
v2 = 3
t = 5
s = 12
s1 = s + (v1 + v2) * t
s2 = (Abs(v1 + v2) * t - s)
Print s1, s2
```

Запустить проект на выполнение. Посмотреть результаты, выведенные на форму.

Алгоритм «ветвление»

Задача: Решить квадратное уравнение ax^2+bx+c

- Разместить на форме кнопку command1. Создать событийную процедуру command1_Click () .
- В окне *Программный код* ввести событийную процедуру целиком:

```
Private Sub Command1_Click()
```

```
Dim a, b, c, d, x1, x2 As Single
```

```
a = InputBox("введите a:", «ВВОД »)
```

```
b = InputBox("введите b:", «ВВОД »)
```

```
c = InputBox("введите c:", «ВВОД »)
```

```
d = Val(b) ^ 2 - 4 * Val(a) * Val(a)
```

```
If d < 0 Then Print «нет решения » Else x1 = (-b + Sqr(d)) / (2 *  
a): x2 = (-b - Sqr(d)) / (2 * a)
```

```
Form1.Print x1, x2
```

```
End Sub
```

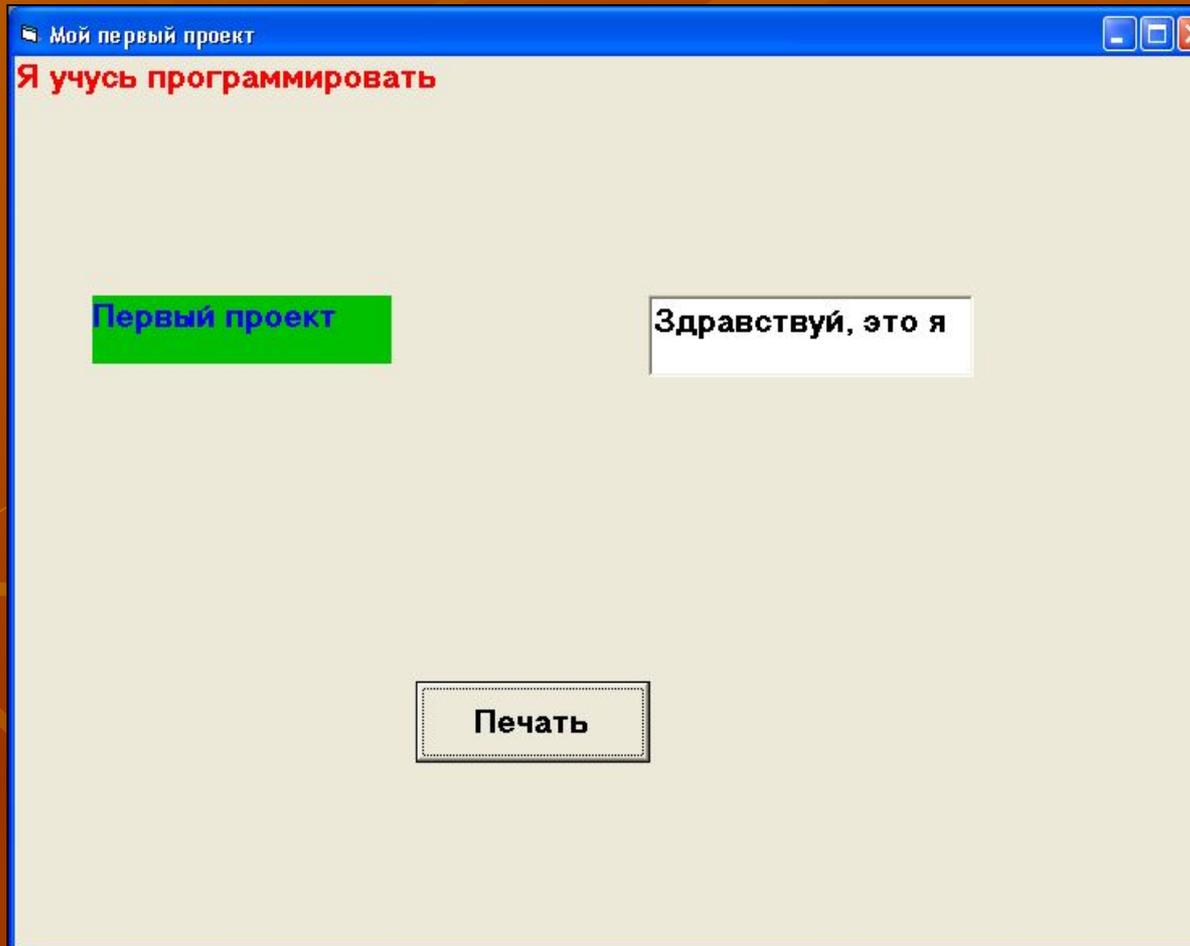
- **Задание на самостоятельное выполнение:**

Найти периметр и площадь треугольника по его сторонам.

Форма и размещение на ней управляющих элементов

1. Запустить Visual Basic. Создать новый проект командой [**File-New Project –Standard.exe**]
2. С помощью Панели инструментов на форму Form1 поместить метку (**Label1**), текстовое поле (**Text1**) и командную кнопку (**Command1**).
3. В окне **Свойства объекта** выбрать свойства:
Для формы:
BackColor = голубой
Caption= «Мой первый проект»
Font =размер 14, жирный
ForeColor=красный
Для метки:
Caption= «Первый проект»
BackColor = зеленый
Font =размер 10, курсив
Для командной кнопки:
Caption= «Печать»
4. Ввод команд в процедуры вводится двойным щелчком по кнопке. Ввести команды в процедуры:*form1.print «Я учусь программировать»*
Text1.text=«Здравствуй, это я»
5. Запустить проект [**Run - Start**]
6. Сохранить файл [**Save Project As ...**]

Результат работы программы:



Графические возможности

- `Scale (X1,Y1) - (X2,Y2)`- позволяет задать систему координат и масштаб для формы или графического окна: `X1, Y1` — координаты левого верхнего угла объекта и `X2, Y2` — координаты правого нижнего угла объекта.
- `Pset (X,Y) [,color]` - установка точки с заданными координатами и цветом: `X, Y` — координаты точки и `color` -- цвет линии. `Line` — рисование линии, прямоугольника или закрашенного прямоугольника заданного цвета:
 - 
- `Line (X1,Y1) - (X2,Y2)[,color][,B][F]`, `X1, Y1` и `X2, Y2` — координаты концов линии (левого верхнего и правого нижнего угла прямоугольника), `color` — цвет линии. Флажок `B` задает рисование прямоугольника, а флажок `F` — его закрашивание.
 - 
 - 
- `Circle (X, Y) ,radius [,color, start, end, aspect]`- рисование окружности, овала или дуги с заданными координатами центра, радиусом, цветом, начальным и конечным углами дуги и коэффициентом сжатия: `X, Y` — координаты центра окружности, `radius` -- радиус окружности, `color` цвет окружности, `start` и `end` — начальный и конечный угол дуги, `aspect` -- коэффициент сжатия. Для получения закрашенного круга требуется установить требуемое значение свойств `FillColor=[color]` и `Style=0`
 - 
 - 

Задание цвета

- Значение аргумента color можно задать различными способами:
- с помощью одной из восьми констант, определяющих цвет (vbBlack — черный, vbBlue — синий, vbGreen — зеленый, vbCyan — голубой, vbRed — красный, vbMagenta — сиреневый, vbYellow — желтый, vbWhite — белый);
- с помощью функции QBColor (number), аргументом которой являются числа от 0 до 15, а результат соответствует одному из основных 16 цветов;
- с помощью функции RGB (bytRed, bytGreen, bytBlue), аргументами которой являются три числа в диапазоне от 0 до 255 (интенсивности базовых цветов), а результатом — число типа Long в диапазоне от 0 до 2563 (16 777 215). Таким образом, определяется цветовая палитра с более чем 16 миллионами цветов, а каждый цвет задается числом. В случае отсутствия аргумента color рисование будет производиться цветом, принятым по умолчанию (черным).

Создание анимации

алгоритм цикл

Для создания анимации (иллюзии движения на экране какого-либо объекта) применяется принцип смены кадров (изображений), как это делается в мультипликации. Программа, имитирующая движение, должна реализовывать следующие этапы:

- создание изображения на экране;
- реализация временной паузы для того, чтобы глаз зафиксировал изображение;
- проведение коррекции изображения.

Анимация часто используется для изображения движения объектов. Для регулирования скорости движения объекта используют пустой цикл: чем большее количество раз он будет выполняться, тем медленнее будет двигаться объект.

Для выполнения задания применить цикл со счетчиком:

```
For Счетчик=НачЗнач To КонЗнач Step  
Тело цикла  
Next
```



АНИМАЦИЯ

Разместить на форме графическое поле `picture1`, в котором будет производиться движение круга.

Поместить на форму кнопку `cmdStart` и создать событийную процедуру, реализующую анимацию:

```
Dim intX As Integer, lngI As Long
Form1.Scale (-10, 10)-(10, -10)
For intX = -10 To 10
Form1.FillStyle = 0
Form1.FillColor = vbBlue
Form1.Circle (intX, 0), 1, vbBlue
For lngI = 1 To 10000000
Next lngI
Form1.FillStyle = 0
Form1.FillColor = vbWhite
Form1.Circle (intX, 0), 1, vbWhite
Next intX
```

Запустить проект и щелкнуть по кнопке *Старт*. В графическом поле начнется движение синего круга по оси *OX*.

- Создать аналогичное движение круга по оси *Y*.
- Создать аналогичное движение круга по диагонали графического поля

Простейший графический редактор

Задание. Разместить на форме графические примитивы (точку, линию, прямоугольник, закрашенный прямоугольник, окружность).

Поместить на форму 7 командных кнопок.

Создать событийную процедуру для задания масштаба рисования:

```
Private SUB cmd1_Click()  
Form1.Scale (0, 100) – (100,0)  
End Sub
```

Создать событийную процедуру для очистки графического поля.

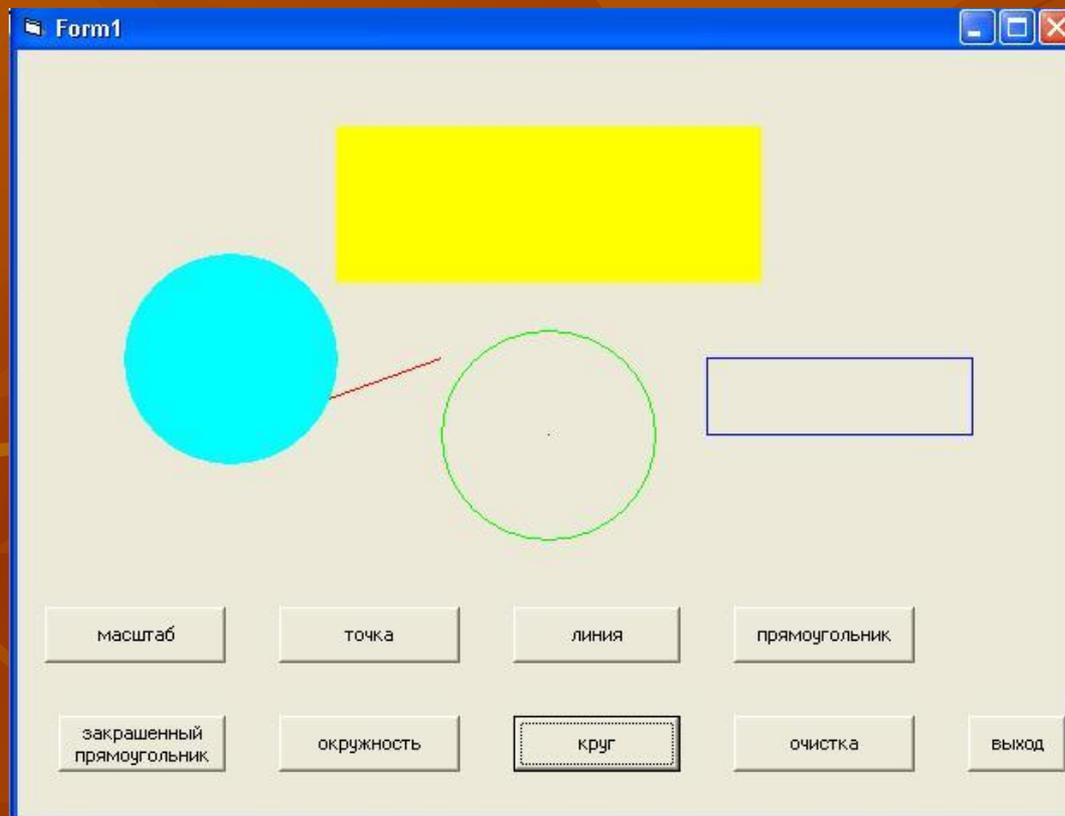
```
Private SUB cmd2_Click()  
Form1.cls  
End Sub
```

Создать 5 событийных процедур рисования графического примитива.

Например, событийная процедура рисования закрашенного прямоугольника выглядит следующим образом:

```
Private SUB cmd1_Click()  
Form1.Line (20, 50)-(40,60),vbRed,BF  
End Sub
```

Результат работы программы:



Массивы

Заполнение массива случайными числами

```
Dim bytA(1 To 100), bytI As Byte
Private Sub Command1_Click()
For bytI = 1 To 10
bytA(bytI) = Int(Rnd * 10)
Next bytI
End Sub
```

Поиск в массивах минимального значения и его индекса

```
Private Sub Command2_Click()
```

```
    bytmin = bytA(1)
```

```
    bytI = 1
```

```
    For bytI = 2 To 10
```

```
        If bytA(bytI) < bytmin Then bytmin = bytA(bytI): bytn = bytI
```

```
    Next bytI
```

```
    Print "минимальный элемент"; bytmin; "его индекс"; bytn
```

```
End Sub
```

Результат работы программы:

The screenshot displays the Microsoft Visual Basic IDE with the following components:

- Code Editor:** Contains two event procedures:

```
Dim bytA(1 To 100), bytI As Byte, bytmin, bytj, bytk, bytr, bytn As Byte

Private Sub Command1_Click()
For bytI = 1 To 10
bytA(bytI) = Int(Rnd * 10)
Print bytA(bytI)
Next bytI
End Sub

Private Sub Command2_Click()
bytmin = bytA(1)
bytI = 1
For bytI = 2 To 10
If bytA(bytI) < bytmin Then bytmin = bytA(bytI): bytn = bytI
Next bytI
Print "минимальный элемент"; bytmin; "его индекс"; bytn
End Sub
```
- Form1 (Form1):** A window with a list of 10 numbers: 7, 5, 5, 2, 3, 7, 0, 7, 8, 7. Below the list, the text "минимальный элемент 0 его индекс 7" is displayed. Two buttons are present: "Заполнить массив" (Fill array) and "Найти мин элемент" (Find min element).
- Project Explorer:** Shows the project structure with "Project1 (Project1)" and "Form1 (Form1)".

Visual Basic for Applications

Visual Basic воспринимается легче, чем строгий Pascal или виртуозный C++. Именно по этой причине я считаю Visual Basic наилучшим языком программирования для детей.

Учиться программированию можно прямо в среде Microsoft Office, в которую встроен специальный редактор Visual Basic for Applications:

Например, создавать макросы в Word, Excel, Access