

ОСНОВЫ
Visual Basic
for
Applications

VBA

Visual Basic for Applications (VBA, Visual Basic для приложений) — немного упрощённая реализация языка программирования Visual Basic, встроенная в линейку продуктов Microsoft Office (включая версии для Mac OS), а также во многие другие программные пакеты, такие как AutoCAD, SolidWorks, CorelDRAW, WordPerfect и ESRI ArcGIS.



VBA

VBA является интерпретируемым языком. В будущем Microsoft планирует заменить VBA на **Visual Studio Tools for Applications (VSTA)** — инструментарий расширения функциональности приложений, основанный на Microsoft .NET.

К достоинствам языка можно отнести сравнительную **лёгкость освоения**, благодаря которой приложения могут создавать даже пользователи, не программирующие профессионально. К особенностям VBA можно отнести выполнение скрипта именно в среде офисных приложений.

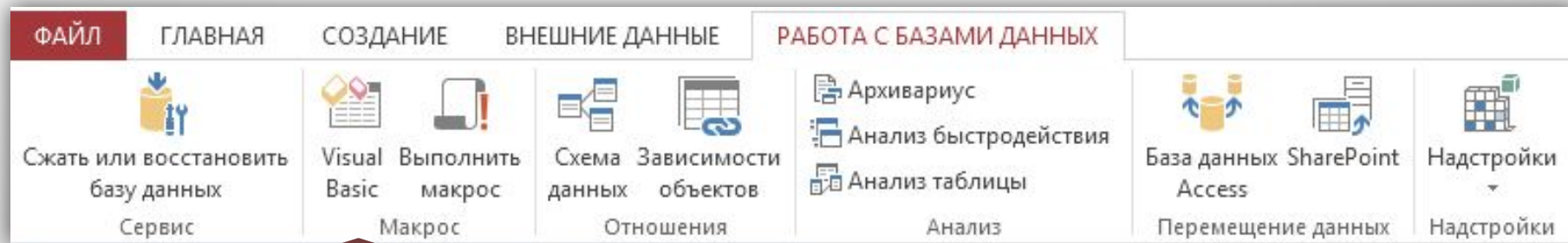
Недостатком являются **проблемы с обратной совместимостью** разных версий. Также к недостаткам часто относят и **слишком высокую открытость кода** для случайного изменения, тем не менее, многие программные продукты (например, Microsoft Office и IBM Lotus Symphony) позволяют пользователю использовать шифрование исходного кода и установку пароля на его просмотр.

Среда VBA

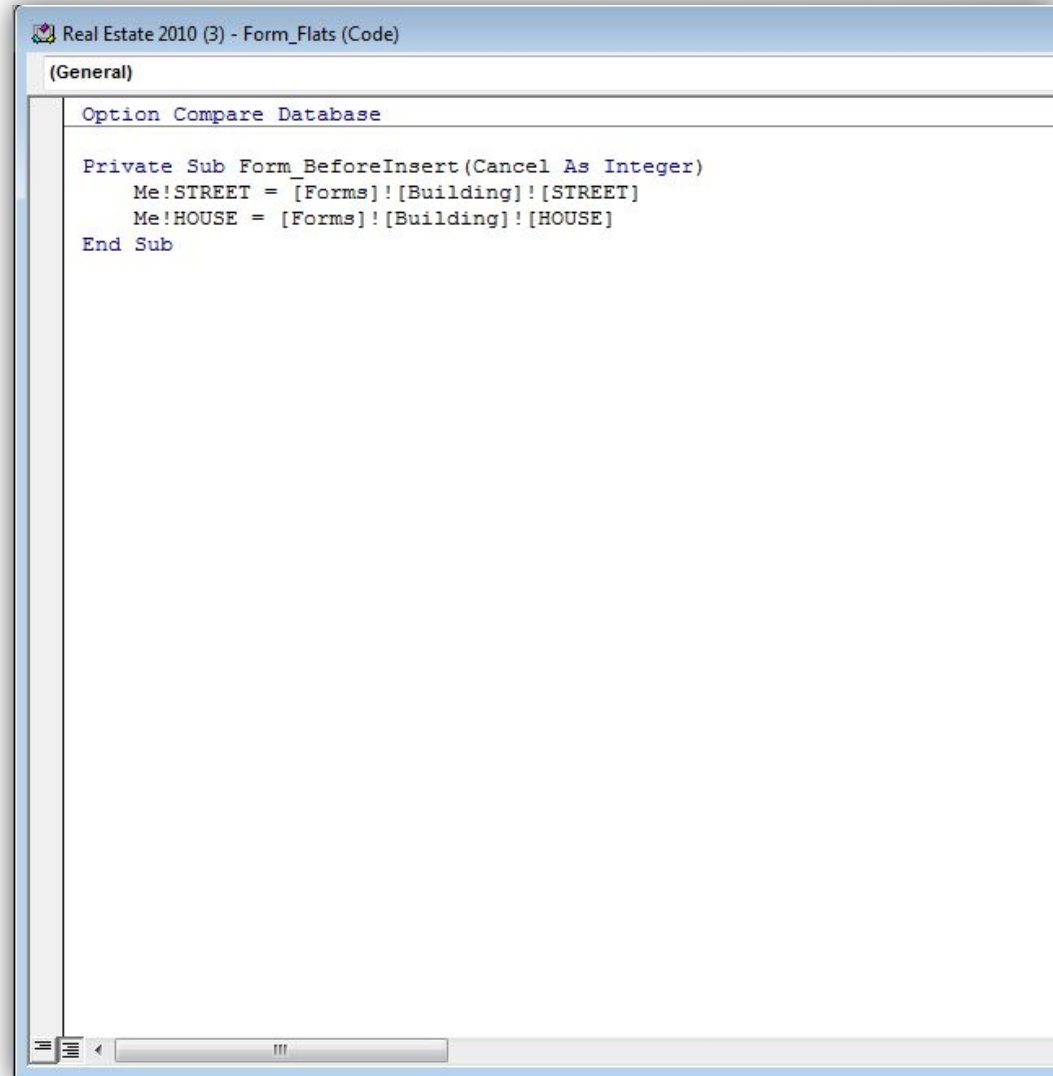
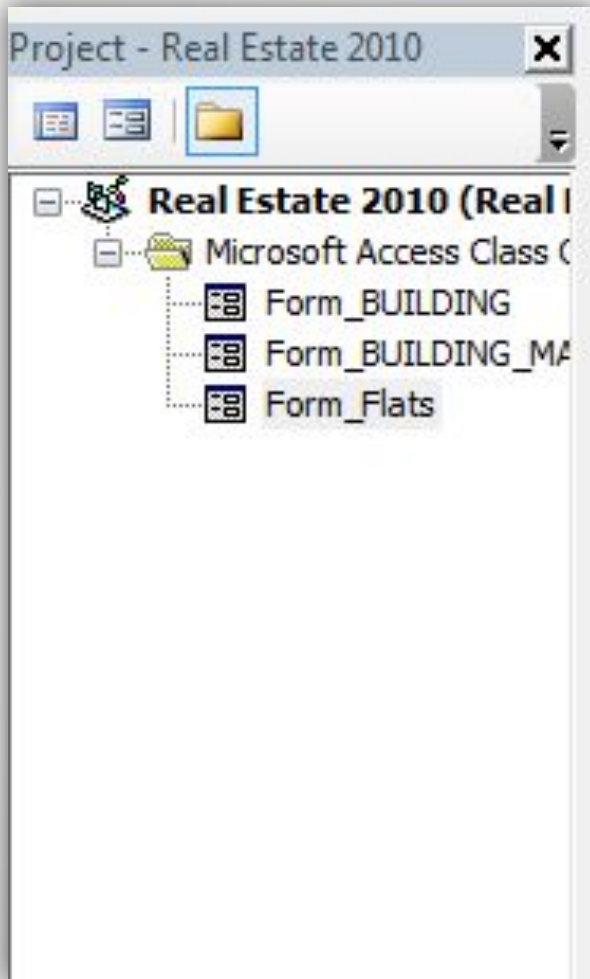
Создав базу данных MS Access автоматически создается проект VBA с названием базы данных.

Чтобы открыть **среду разработки VBA необходимо:**

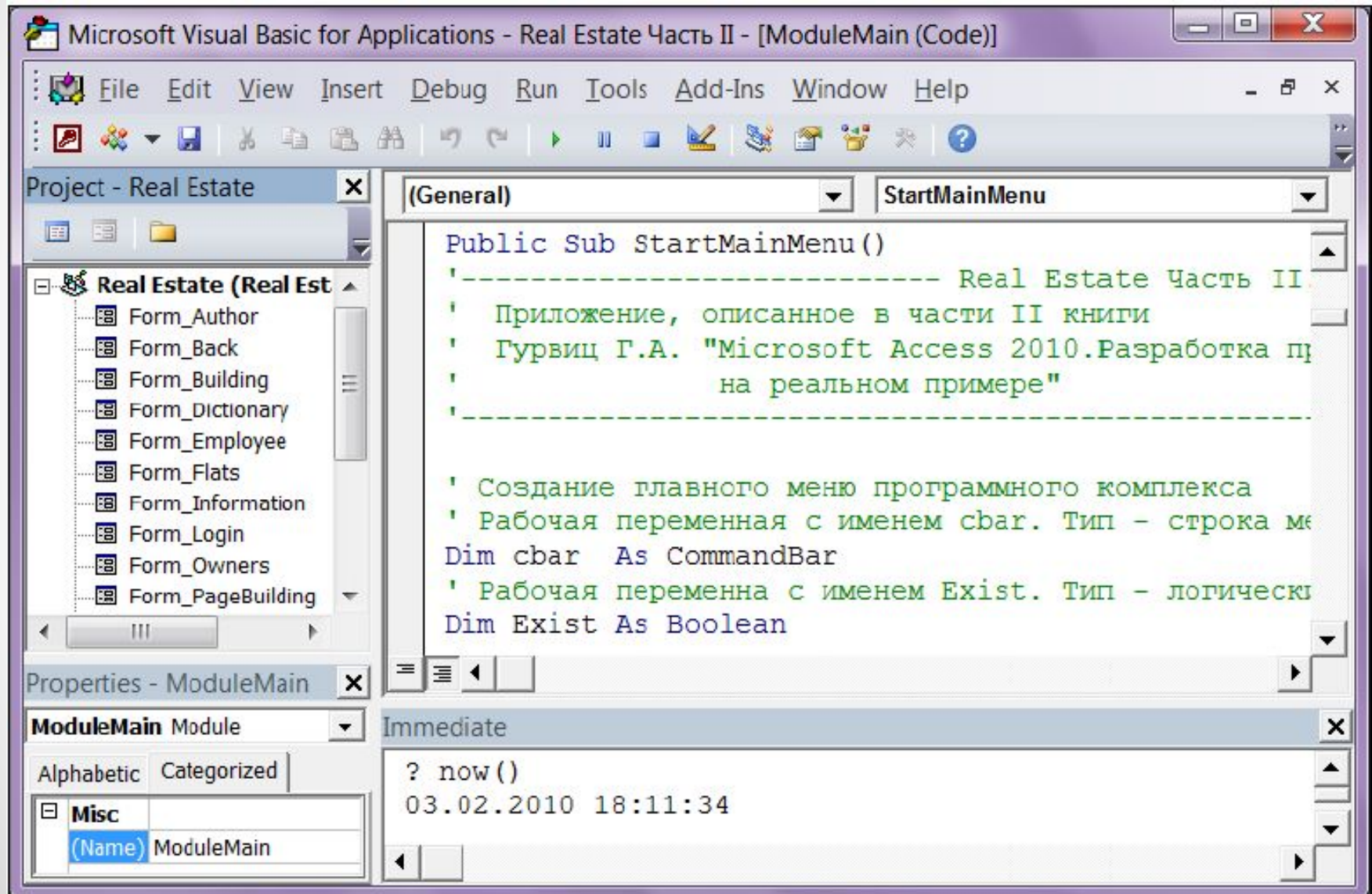
1. перейти на вкладку Работа с базами данных;
2. выбрать самую левую пиктограмму — **Visual Basic** — откроется окно с собственной средой разработки, в которой имеется меню, окна и другие элементы, использующиеся при работе с проектом.



Среда VBA



Среда VBA



Среда VBA

Проект VBA носит **имя базы данных** и содержит формы, модули класса и модули кода.

На представлены четыре основных окна среды VBA:

1. окно проекта (заголовок окна: Project — sample);
2. окно текста (на рисунке заголовка окна нет, но хорошо видны две надписи: General и StartMainMenu);
3. окно свойств (заголовок: Properties — ModuleMain);
4. окно отладки (Immediate).

В окне проекта можно выбрать требуемый объект. Для его редактирования сделайте по объекту двойной щелчок мышью. В окне кода появится его текст, а в окне **Properties** — свойства объекта.

Окно свойств предназначено для задания свойств объектов. Например, можно указать в окне свойств формы фон, заголовок, номер темы в файле справки и т. д.

На первой вкладке свойства расположены в алфавитном порядке **(Alphabetic)**. Выбор второй вкладки этого окна **(Categorized)** отсортирует свойства по категориям.

Среда VBA

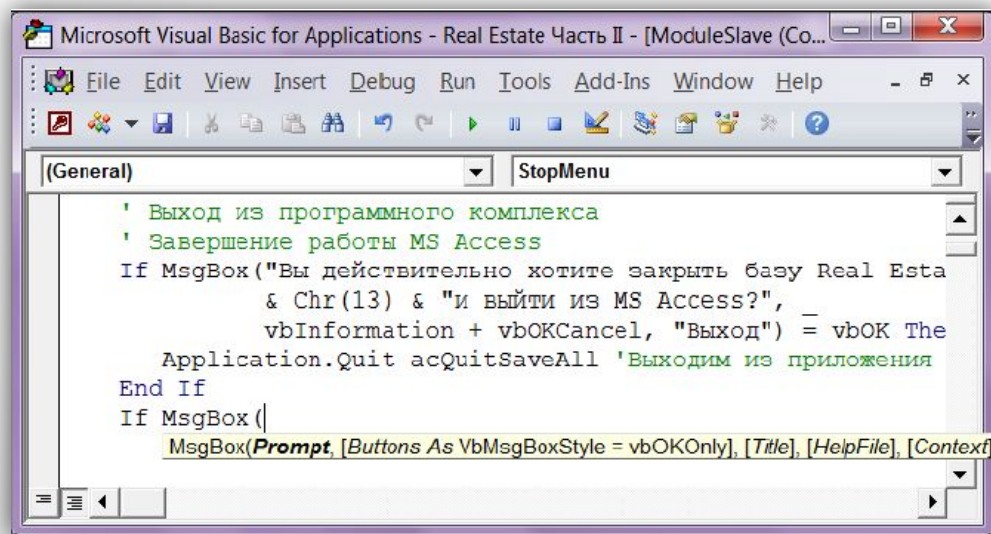
Окно редактирования кода предназначено для ввода текста **процедур VBA**. Код внутри модуля сгруппирован в отдельные разделы — **процедуры**. Редактор VBA может отображать текст в **двух режимах**: просмотр всего модуля или просмотр отдельной процедуры.

В режиме просмотра всего модуля процедуры отделяются одна от другой горизонтальной чертой.

Два раскрывающихся списка (в них отображены надписи General и StartMainMenu) предназначены для выбора объекта и его составляющих соответственно. Если в VBA идет работа с кодом обработки событий формы, то левый список содержит объект формы, а правый — перечень событий, допустимых для выбранного объекта.

Возможности редактора кода

Редактор кода VBA наделен способностью автоматически завершать написание составных частей строки текста (операторов, параметров, свойств). Он сам предлагает разработчику список компонентов, которые могут завершить введенную инструкцию или отобразить на экране сведения о процедуре или функции после введения имени



Редактор кода выполняет автоматическую проверку синтаксиса введенной строки после нажатия клавиши <Enter>. Если строка выделяется красным цветом — ищите ошибку.

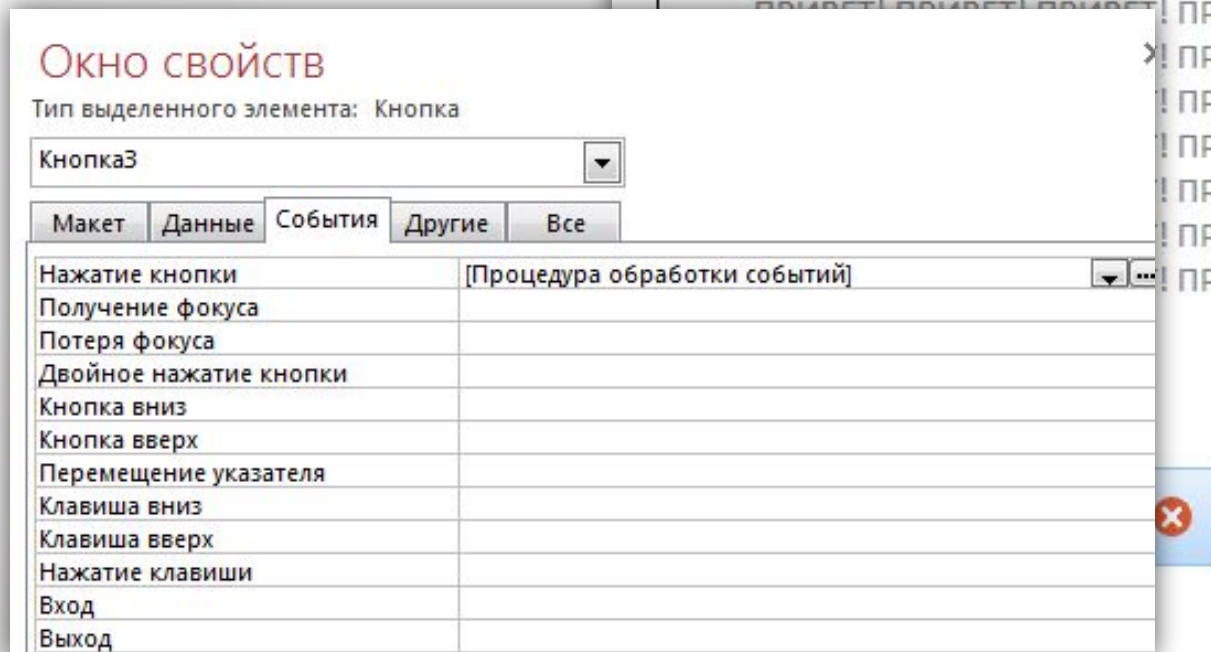
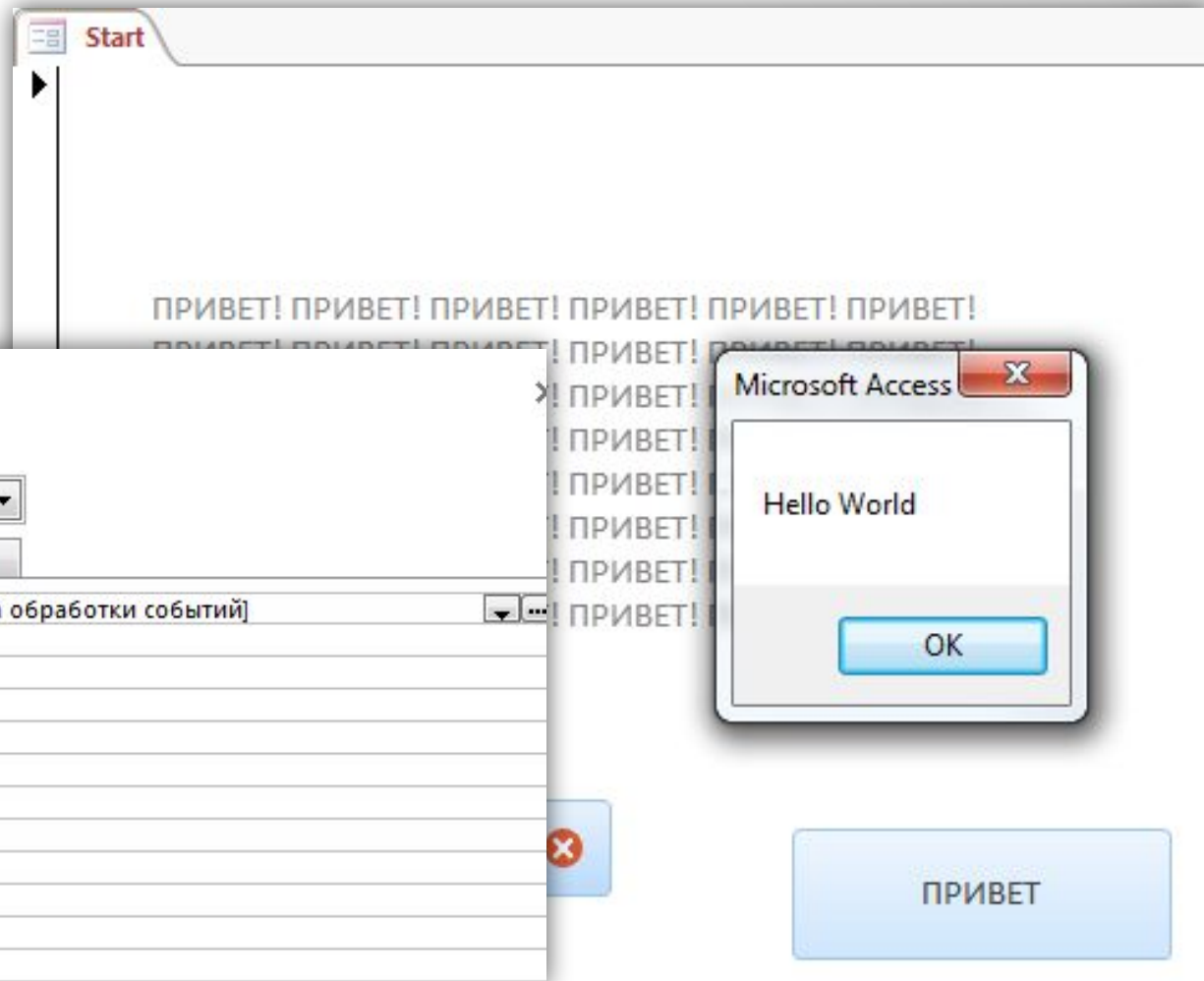
Первая программа VBA

VBA является **языком сценариев** – это значит, что каждый раз когда пользователь совершает определенное действие срабатывает заранее написанный код (происходит обработка события).

Для написания первой программы:

1. Создайте на стартовой форме кнопку «Привет» не выбирая ни какого события (нажать Отмена);
2. В свойствах кнопки в разделе «События» выбрать **«Процедура обработки событий»** напротив пункта **«Нажатие кнопки»** и нажать кнопку «...»;
3. В открывшемся окне редактора кода введите **«MsgBox “Hello”** - этот код автоматически сгенерирует сообщение, когда кнопка будет нажата;
4. Закройте редактор, сохраните форму и откройте ее для использования, опробуйте работу кнопки.

Первая программа VBA



Принципы синтаксиса VBA

- **Регистр** не играет роли – вы можете прописать оператор как в верхнем, так и в нижнем регистре, или вообще в вперемешку, например, dim, DIM, Dim, разницы нет.
- **Комментарии** – для комментирования строки, перед ней прописывается **одинарная кавычка** или **ключевое слово REM**.
- **Максимальная длина любого имени** (будь то **имя переменной**, **константы** или **процедуры**) составляет **255 знаков**.
- В отличие от JavaScript, **Jscript** или C языков, в VBA в конце строки с операторами **не нужно ставить запятой**.
- **Двоеточие** – чтобы не прописывать каждый оператор в отдельной строке, их можно с помощью двоеточия прописать в одну строку, например: **MsgBox "Строка1" : MsgBox "Строка2"**
- **Перенос операторов** – если строка с операторами слишком длинная, то ее можно разделить на несколько при помощи символа подчеркивания и знака **&**, например:
MsgBox **"Это** **простая"** **_**
& "строка"

Создание макроса (модуля)

- Для того чтобы открыть редактор VBA, выберите команду Сервис, **Макрос, Редактор Visual Basic** или нажмите комбинацию клавиш **<Alt>+<F11>**.
- Далее можно воспользоваться меню редактора: выбрать команды **Insert, Module**.
- На листе созданного модуля появятся стандартные строки начала и конца процедуры:

```
Public Sub ИмяПроцедуры()  
End Sub
```

- Между ними можно набрать код (инструкции) программы.
- Для запуска программы выбрать в меню **Run** команду **Run Sub/UserForm** или нажать **кнопку Плэй**.
- Если в программе обнаружены ошибки, то компилятор выдает сообщение и отладчик отмечает строку с ошибкой желтым цветом. Чтобы снять это выделение, можно нажать **кнопку Стоп**.

Переменные VBA

Переменная — это область памяти компьютера, имеющая имя. В VBA переменные используются для хранения данных в оперативной памяти. После создания переменная указывает на одну и ту же область памяти до тех пор, пока не будет уничтожена. Разработчику не нужно знать, где находится эта область. Достаточно лишь сослаться на **имя переменной**.

При выборе имени переменной следует руководствоваться правилом:

- ❖ имя должно начинаться с буквы;
- ❖ имя не может содержать пробел и точку, а также @, &, !, \$, #;
- ❖ имя не должно быть длиннее 255 символов;;
- ❖ не рекомендуется назначать имена, совпадающие с названием функций и методов VBA.

Переменные VBA

Перед началом работы с переменной ее нужно **объявить**. Сделать это можно при помощи **операторов Dim, Public, Private и Static**. В зависимости от используемых операторов и месте их появления в коде программы, переменной будет назначена область видимости и время жизни.

В VBA существуют **три области видимости** переменной:

Переменная уровня проекта.

Описывается с помощью инструкции **Public**. Также называется *открытой*. Является доступной для всех процедур проекта VBA. Пример:

```
Public Account As Integer
```

Переменная уровня модуля.

Описывается с помощью инструкции Dim или Private. Является доступной только в том модуле, в котором описана. Описание такой переменной необходимо разместить перед описанием всех процедур модуля. Пример:

```
Private LastName As String  
Dim LastName As String
```

Переменная уровня процедуры.

Описывается с помощью инструкции Dim или Static. Является доступной только в той процедуре, в которой описана. Если используется конструкция Static, то переменная сохраняет свое значение, пока выполняются другие процедуры. Если Dim, то после завершения работы процедуры значение такой переменной всегда теряется.

Переменные VBA

The screenshot displays the Microsoft Visual Basic for Applications (VBA) environment. The main window title is "Microsoft Visual Basic for Applications - Real Estate Часть II [break] - [ModuleSlave (Co...". The menu bar includes File, Edit, View, Insert, Debug, Run, Tools, Add-Ins, Window, and Help. The "Project - Real Estate" window shows a tree view of the project structure, including various forms and modules. The "Debug" menu is open, showing options like Debug, Edit, Standard, UserForm, and Настройка... The "Run" menu is also open, showing a toolbar with icons for Run, Break, Step Into, Step Over, Step Out, and other debugging actions. The code editor shows the following VBA code:

```
Public Sub Proba()  
    ' Процедура для демонстрации работы  
    ' с окном отладки Immediate  
    Static X As Integer  
    X = 8  
    Debug.Print "Значение переменной X=" & X  
End Sub
```

The line `Debug.Print "Значение переменной X=" & X` is highlighted in yellow. Below the code editor, the "Immediate" window is open, displaying the output of the debug print statement: "Значение переменной X=8".

Переменные VBA

Типы данных в языках программирования можно условно разбить на следующие группы:

1. **примитивные типы** (целые, вещественные числа, логические и символьные значения);
2. **конечные последовательности** однотипных элементов (линейные массивы, матрицы, многомерные массивы);
3. **конечные последовательности** разнотипных элементов (записи, структуры);
4. **динамические последовательности** со специальными операциями доступа (очереди, стеки, списки);
5. **сложные типы**, ориентированные на специальные предметные области (электронные таблицы, листы Excel, диапазоны ячеек листов, базы данных и т.п.);
6. **произвольные типы**, определяемые пользователем (например, самолет, автомобиль, животное, собаки, студенты).

Переменные VBA (с. 160)

VBA дает возможность описывать и использовать **переменные различных типов**.

Выбор типа переменной основан на требованиях разрабатываемого приложения. В VBA имеются **строки, числа, даты, объекты, логические значения**, а также общий тип данных — **Variant**. Он применяется по умолчанию и может содержать данные любого типа, кроме строк фиксированной длины.

Категория	Тип данных	Размер	Диапазон
Числовые типы	Byte	1 байт	От 0 до 255
	Integer	2 байта	От -32 768 до 32 767
	Long	4 байта	От 2 147 483 648 до 2 147 483 647
	Single	4 байта	Для отрицательных чисел: от -3.402823×10^{38} до $-1.401298 \times 10^{-45}$ Для положительных чисел: от 1.401298×10^{-45} до 3.402823×10^{38}
	Double	8 байтов	Для отрицательных чисел: от $-1.79769313486232 \times 10^{308}$ до $-4.94065645841247 \times 10^{-324}$ Для положительных чисел: от $4.94065645841247 \times 10^{-324}$ до $1.79769313486232 \times 10^{308}$
	Currency	8 байтов	От -922 337 203 685 477.5808 до 922 337 203 685 477.5807

Переменные VBA (с. 160)

Категория	Тип данных	Размер	Диапазон
Логический тип	Boolean	2 байта	False или True. Если используется 0, то он интерпретируется как False. Любое другое значение — True. При обратном преобразовании True рассматривается как -1
Тип даты	Date	8 байтов	От 1 января 100 года до 31 декабря 9999 года
Тип объект	Object	4 байта	Ссылка на объект
Строковые типы	String	Длина строки плюс 10 байтов	Строки переменной и фиксированной длины

Переменные VBA

Для обязательного объявления всех переменных в начало модуля надо поместить директиву:

Option Explicit

Использование этой директивы не допускает возможности работы с необъявленными переменными.

Константа — это область памяти компьютера, имеющая имя и не меняющая свое значение на протяжении всей работы программы. В VBA константы, как и переменные, используются для хранения данных в оперативной памяти.

В VBA можно как описать собственные константы, так и применять **стандартные**, которых в языке великое множество. При описании константы необходимо сразу присвоить ей значение, например:

Public Const Pi As Double=3.141592653589793

Описав константу один раз, можно использовать ее в разных выражениях столько раз, сколько это требуется. Область видимости констант VBA регламентируется теми же правилами, что и у переменных.

Функция MsgBox

Функция **MsgBox()** — стандартная функция вывода сообщений на экран. Для работы с ней применяется 21 константа.

Добавим вывод сообщения на экран при закрытии приложения:

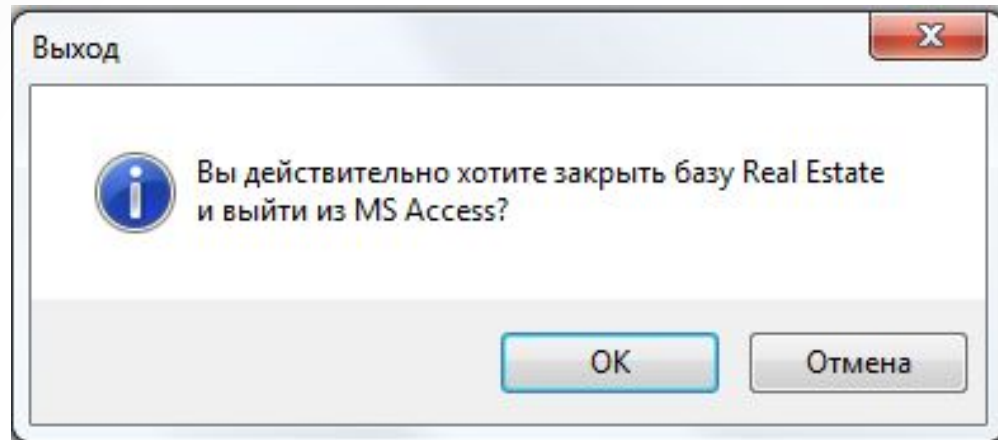
1. Откроем стартовую форму на которой есть кнопка закрытия приложения;
2. Выберем **обработку событий по клику** для кнопки закрытия приложения;
3. Добавим следующий код:

```
Option Compare Database

Private Sub Start_Stop_Click()
    ' Выход из программного комплекса
    ' Завершение работы MS Access
    If MsgBox("Вы действительно хотите закрыть базу Real Estate " & Chr(13) & "и выйти из MS Access?", _
vbInformation + vbOKCancel + VbDefaultButton1, "Выход") = vbOK Then
        Application.Quit acQuitSaveAll ' Выходим из приложения
    End If
End Sub
```

4. Откроем форму для работы и нажмем на кнопку выхода.

Функция MsgBox



Текст, следующий за **СИМВОЛОМ** ' до конца строки, представляет собой **комментарий** и игнорируется транслятором, нужен для пояснения.

Обратите внимание на **константы VBA**. Их в этом примере используется **четыре**, причем значения трех складываются:

vbInformation=64 - используется значок информации

vbOKCancel=1 - отображаются кнопки ОК и Отмена

VbDefaulButton1=0 - номер выделенной кнопки

И если нажата кнопка ОК, то значение **VbOK=1**.

Функция MsgBox (с 163-164)

Далее приведены константы, которые можно указать в качестве аргументов функции MsgBox() для определения состава кнопок в диалоговом окне, константы, определяющие отображаемые в диалоговом окне значки, константы, определяющие основную кнопку в диалоговом окне, константы, соответствующие выбранной кнопке.

Константа	Значение	Отображение кнопок
VbOkOnly	0	
VbOkCancel	1	 
VbAbortRetryIgnore	2	  
VbYesNoCancel	3	  
VbYesNo	4	 
VbRetryCancel	5	 

Функция MsgBox (с 163-164)

Константа	Значение	Значок
VbCritical	16	 Текст сообщения
VbQuestion	32	 Текст сообщения
VbExclamation	48	 Текст сообщения
VbInformation	64	 Текст сообщения

Константа	Значение	Номер основной кнопки
VbDefaultButton1	0	1
VbDefaultButton2	256	2
VbDefaultButton3	512	3
VbDefaultButton4	768	4

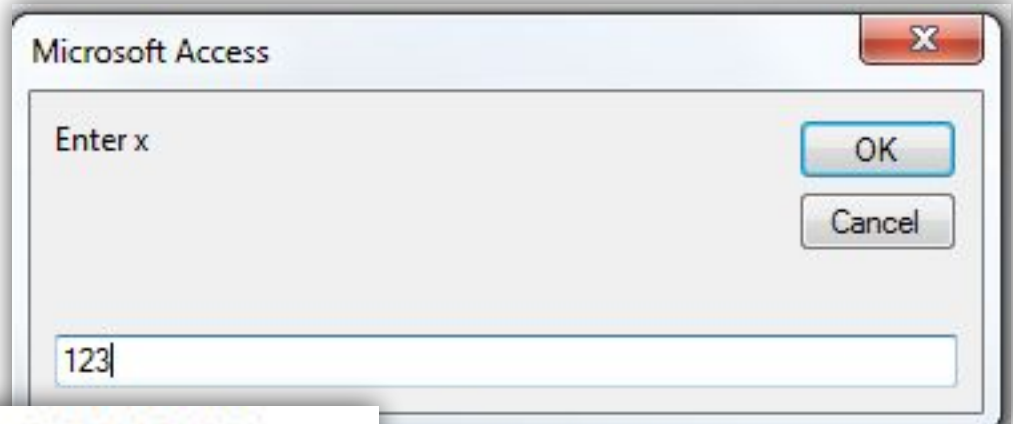
Функция MsgBox (с 163-164)

Константа	Значение	Отображение кнопок
VbOk	1	
VbCancel	2	
VbAbort	3	
VbRetry	4	
VbIgnore	5	
VbYes	6	
VbNo	7	

Ввод значений в VBA

В языке VBA ввод значения переменной *x* можно осуществить с помощью функции **InputBox**, например:

```
x = InputBox("Введи значение x")
```



```
Private Sub Кнопка4_Click()  
X = InputBox("Enter x")  
End Sub  
|
```

Описание программы на VBA

Описание программы $P(x,y,z)$ – это текст программы P , в котором x, y, z являются обозначениями входных (и, возможно, выходных) данных. Эти обозначения называют формальными параметрами программы P . Описание обычно делят на заголовок и тело.

В заголовке указываются типы входных данных и результата (выходных данных). Если программа не вычисляет конкретное значение, например, просто печатает текст или производит различные действия со своими собственными переменными, не сообщая результаты в качестве выходных данных, то считается, что результат имеет **неопределенный тип**. Такие программы называют часто **процедурами**. В отличие от них программы, вычисляющие определенное значение (некоторое число, текст, массив данных и т.п.), называют **функциями**. Говорят тогда, что *функция возвращает значение*.

Общий синтаксис на VBA:

```
Function <имя функции> (<список формальных параметров>)  
    <тип результата>  
    или  
Procedure <имя процедуры> (<список формальных параметров>)
```

Описание программы на VBA

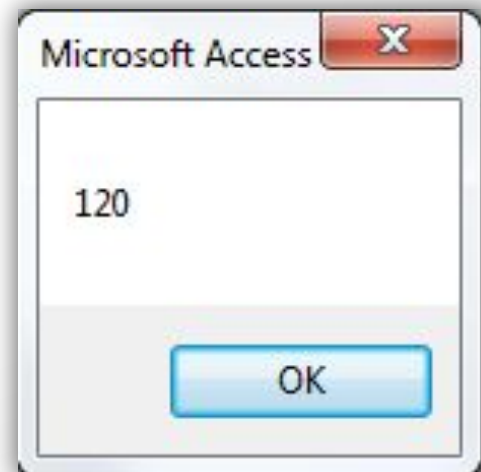
Тело программы – это сама программа, описание ее собственных переменных и операторы, реализующие алгоритм.

Функция вычисления факториала:

```
Function fact(n As Integer) As Integer
fact = 1
If n > 1 Then
    For i = 1 To n
        fact = fact * i
    Next
End If
End Function
```

Процедура вызова функции факториала:

```
Sub vysov()
Dim k As Integer
k = fact(5)
End Sub
```



Описание программы на VBA

Программа, используемая другой программой, может иметь статус **подпрограммы**, и тогда ее описание может быть помещено внутри использующей программы, среди других ее описаний (описание одной программы содержит описание другой, оно, в свою очередь, содержит описание третьей и т. д.). В языках С++, VBA программы имеют **независимый статус**, т.е. описываются отдельно от использующей программы, вложения описаний недопустимы. По описанию программы компилятор выделяет память, содержащую команды программы, память для переменных программы и память для ее формальных параметров (пока пустую). Для того, чтобы программа, содержащая параметры, заработала, нужно ее вызвать, т.е. использующая программа должна содержать **оператор вызова**.

Вызов программы должен включать имя программы и набор конкретных значений, подставляемых вместо ее формальных параметров. Синтаксис вызова процедуры в VBA такой:

```
<имя процедуры> <список фактических параметров>  
или  
Call <имя процедуры>(<список фактических параметров>).
```

Вызов функции имеет вид:

```
<имя функции> <список фактических параметров>.
```

Описание программы на VBA

Список фактических параметров – это последовательность фактических параметров, разделенных запятой. В качестве фактического параметра может выступать значение, переменная, выражение, вызов функции.

Вызов процедуры – это независимый оператор, он не может стоять в правой части присваиваний, поскольку не возвращает значения.

Вызов функции должен стоять в правых частях присваиваний, он может быть операндом выражения, т.к. возвращает значение, которое и используется при вычислении выражения.

Механизм реализации вызова:

1. в точке вызова управление передается указанной программе;
2. переменные, указанные в качестве имен формальных параметров получают конкретные значения, заданные через фактические параметры;
3. программа исполняется с этими фактическими значениями параметров;
4. если программа является функцией, то ее значение (результат работы) возвращается в точку вызова и может быть использовано, например, в выражении;
5. если программа – процедура, то после ее исполнения просто происходит возврат управления в точку вызова, следующему за вызовом оператору.

Описание программы на VBA

Список фактических параметров – это последовательность фактических параметров, разделенных запятой. В качестве фактического параметра может выступать значение, переменная, выражение, вызов функции.

Вызов процедуры – это независимый оператор, он не может стоять в правой части присваиваний, поскольку не возвращает значения.

Вызов функции должен стоять в правых частях присваиваний, он может быть операндом выражения, т.к. возвращает значение, которое и используется при вычислении выражения.

Механизм реализации вызова:

1. в точке вызова управление передается указанной программе;
2. переменные, указанные в качестве имен формальных параметров получают конкретные значения, заданные через фактические параметры;
3. программа исполняется с этими фактическими значениями параметров;
4. если программа является функцией, то ее значение (результат работы) возвращается в точку вызова и может быть использовано, например, в выражении;
5. если программа – процедура, то после ее исполнения просто происходит возврат управления в точку вызова, следующему за вызовом оператору.

Описание программы на VBA

Например, программа, содержащая вызовы процедуры Sum и функции fact, может быть такой (после знака ' идет однострочный комментарий):

```
Sub vysov()  
Dim k As Integer  
k = fact(5)      'функция fact вызвана с фактическим параметром  
                  '5, результат 120  
k = fact(4)+10+k 'функция fact вызвана с фактическим  
                  'параметром 4, ее результат 64  
MsgBox k  
Sum 5, 10      'процедура Sum вызвана с фактическими параметрами  
               '5 и 10  
Call Sum(10,20) 'другой способ вызова Sum с фактическими  
                  'параметрами 10 и 20  
Sum k, 2      'процедура Sum вызвана с фактическими параметрами  
              'k и 2  
End Sub
```


Описание программы на VBA

Программы – процедуры и функции могут быть собраны в автономно существующие, например, в рамках проекта, более «крупные» единицы – **модули**.

Видимость переменных: При всякой сборке возникает вопрос: могут ли программы использовать имена и значения переменных из других программ? Этот вопрос решается для каждого языка программирования в разделе, называемом **правилами видимости переменных**.

Для VBA действуют следующие **правила видимости**:

- ❖ переменные, описанные в процедуре или функции, видны только в ней. Они объявляются с помощью **ключевого слова Dim** или **Static** и называются локальными.
- ❖ переменные, описанные в модуле вне описаний программ модуля с **ключевыми словами Dim** или **Private**, видны во всех программах этого модуля. В других модулях они не видны.
- ❖ Переменные, описанные в модуле вне описаний программ модуля с **ключевым словом Public**, видны во всех модулях (процедурах, функциях) проекта.

Практика использования VBA

Ознакомьтесь с документом **«Учебное пособие по VBA. Глава 1. Алгоритмы и программы»** и составьте следующие программы:

- 1) Найти значения переменной y , заданные формулой $y = x^2 + \sqrt{x} - \sin(x)$ для заранее заданного и введенного с клавиатуры положительного числа x . Оформить вычисление в процедуру или функцию.
- 2) Найти минимальное из двух чисел вводимых с клавиатуры.
- 3) Найти среднее арифметическое четных чисел из N чисел, вводимых с клавиатуры.

Лабораторная работа №6

- 1) Ознакомиться с пунктом учебника 5.4 Стандартные константы;
- 2) Внедрить в разрабатываемую БД несколько кнопок с различным функционалом;
- 3) Составить на языке VBA следующие программы:
 - a) посчитать: $y = x^3 + z^2 - z/x$;
 - b) найти максимальное из трех чисел, вводимых с клавиатуры;
 - c) Посчитать произведение чисел, вводимых с клавиатуры до тех пор, пока не встретится 0.