


## Практическая работа №1

### РЕДАКТОР VBA. СТРУКТУРА ПРОГРАММЫ. ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА

**ЦЕЛЬ РАБОТЫ.** Изучить структуру редактора VBA в MS Excel, научиться формировать диалоговые окна.

#### ТЕОРЕТИЧЕСКАЯ ЧАСТЬ. ВЫПОЛНЕНИЕ РАБОТЫ

##### 1. Структура редактора VBA

1. Запустите программу MS Excel.
2. Выполните команду **Файл – Сохранить как...**
3. В окне **Сохранение документа** укажите место хранения файла и его имя – **Лаб1**.
4. Щелкните значок **Кнопка Microsoft Office** , а затем щелкните **Параметры Excel**.
5. В категории **Основные** в группе **Основные параметры работы с Excel** установите флажок **Показывать вкладку "Разработчик"** на ленте, а затем нажмите кнопку **ОК**.
6. На вкладке **Разработчик** в группе **Код** (Рисунок 1.1) нажмите кнопку **Visual Basic**.

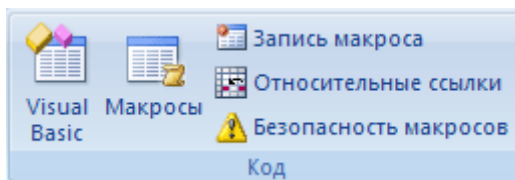


Рисунок 1.1 Группа Код

На экране появится окно редактора Visual Basic (Рисунок 1.2). Основные компоненты интерфейса VBA:

- меню;
- окно проекта;

- окно редактирования кода;
- окно свойств;
- окно форм;
- панели инструментов.

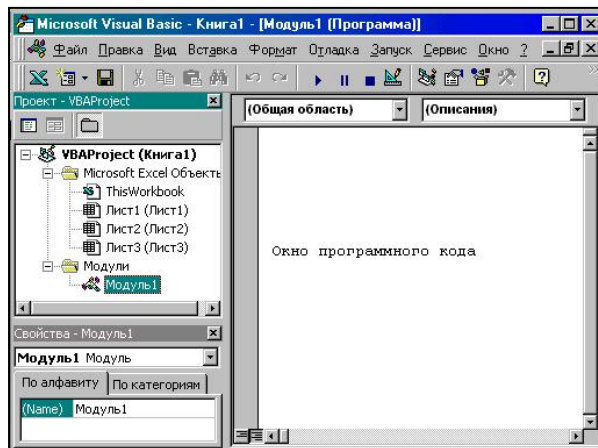


Рисунок 1.2. Окно редактора Visual Basic

### 1.1. Окно проекта

Окно проекта можно активизировать командой **Вид/Окно проекта** или нажатием кнопки **Окно проекта**. Окно проекта содержит иерархическую структуру файлов форм и модулей текущего проекта (Рисунок 1.3).

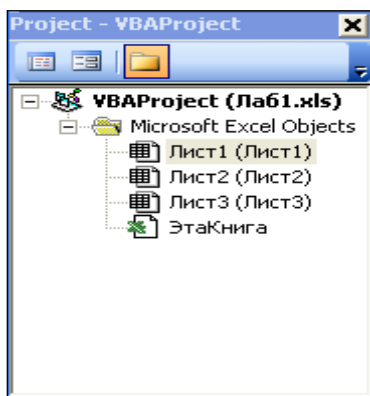


Рисунок 1.3 Иерархическая структура файла форм и модулей текущего проекта

Модуль для каждого рабочего листа и для всей рабочей книги в проекте создается автоматически. Модули по назначению делятся на стандартные и модули объектов. Стандартные модули это те, которые содержат макросы. Такие модули добавляются в проект командой **Вставка/Модуль**. Модули, связанные с рабочими листами, рабочей книгой, формами, и модули класса

относятся к модулям объектов. Модули класса создаются командой **Вставка/Модуль**. Формы создаются командой **Вставка/UserForm**.

Проекты всех открытых рабочих книг выводятся в окне проекта. Это ускоряет процесс создания новых приложений, так как позволяет легко копировать формы и коды из одного проекта в другой. Двойной щелчок мышью на значке файла в окне проекта открывает окна редактора для соответствующего модуля.

- Щелкните дважды на объекте **Лист1**. Справа появится окно для редактирования кода (Рисунок 1.4).

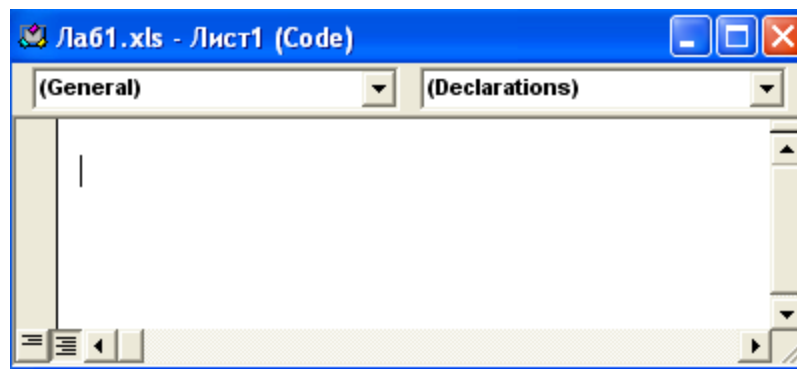


Рисунок 1.4. Окно для редактирования кода

## 1.2. Интеллектуальные возможности редактора кода

Значительно облегчается написание программ за счет способности редактора кода автоматически завершать написание операторов, свойств и параметров. В процессе набора команды, редактор сам предлагает пользователю список компонентов, логически завершающих вводимую пользователем команду. Сделав двойной щелчок на выбранном элементе из этого списка, или нажав клавишу Tab, выбранное имя вставляется в код программы.

Если курсор расположить на ключевом слове языка VBA, имени процедуры, свойства или метода и нажать клавишу F1, то на экране появится окно со справочной информацией об этой функции.

С помощью команды **Сервис/Параметры** и используя вкладку **Редактор** диалогового окна **Параметры** можно настроить редактор VBA:

- **Проверка синтаксиса.** Автоматическая проверка синтаксиса введенной строки программы.
- **Явное описание переменных.** Устанавливает обязательное явное описание переменных в модулях.
- **Список компонентов.** Выводит список данных, логически завершающих инструкцию.
- **Краткие сведения.** Сразу после ввода имени процедуры обеспечивает вывод на экран сведений о процедурах: функциях, подпрограммах, свойствах, методах – и их параметрах.
- **Подсказки значений данных.** В режиме прерывания отображает значение переменной, на которой установлен курсор.
- **Автоотступ.** Задание положения табуляции для первой и последующих строк.
- **Интервал табуляции.** От 1 до 32 символов можно задать величину табуляции.
- **«Перетаскивание» текста.** Дает возможность перетаскивать текст с помощью мыши.
- **Просмотр всего модуля.** Для вновь открываемых модулей устанавливает режим просмотра всех процедур.
- **Разделитель процедур.** Отображает или скрывает полосу, разделяющую каждую процедуру в окне модуля.

### 1.3. Окно свойств

Основные параметры активного элемента отображаются в окне свойств (рисунок 1.5), которое состоит из двух частей: раскрывающегося списка, из которого можно выбрать любой элемент управления текущей формы или саму форму и рабочей части, состоящей из двух вкладок: **По алфавиту** и **По категориям**. Имя элемента управления (свойство **Name**) идет первым. Значения свойств можно поменять путем ввода с клавиатуры или выбором из раскрывающегося списка.

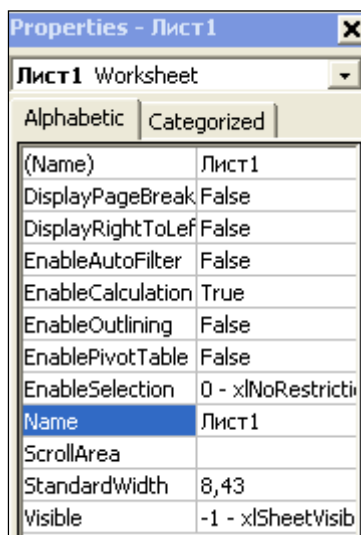



Рисунок 1.5 Окно свойств

#### 1.4. Окно просмотра объектов Object Browser

Командой **View(Вид)/Object** или нажатием кнопки Просмотр объектов  открывается окно **Object Browser** (Рисунок 6), которое состоит из трех частей:

1. **<All libraries> (Все библиотеки)** в левом верхнем углу окна. В этом раскрывающемся списке можно выбрать различные библиотеки объектов Excel, VBA, Office и VBAProject.
2. **Классы (Classes)**. Содержимое данного списка зависит от выбранной библиотеки. Например, после выбора из раскрывающегося списка VBA, все классы этой библиотеки будут выведены в списке Классы.
3. **Компоненты (Members)**. Содержимое данного списка зависит от выбора Класа просматриваемой библиотеки, все компоненты выбранного класса выводятся в списке Компоненты.

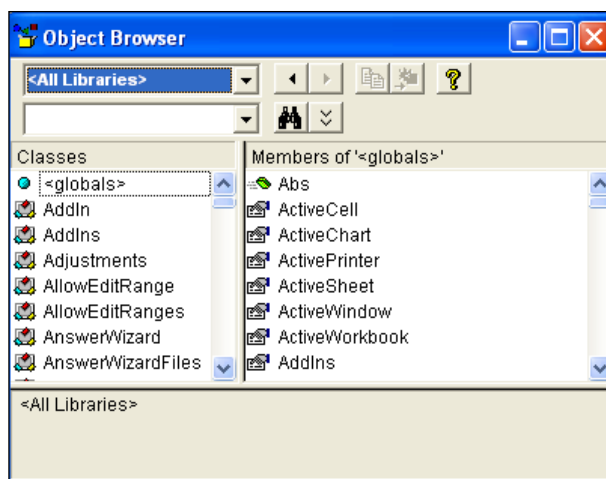


Рисунок 1.6 Окно просмотра объектов

- Закройте окно **Object Browser**.

## 1.5. Структура программного кода

**Программа** представляет собой последовательность инструкций, которые компьютер выполняет одну за другой. Программный код VBA состоит из следующих блоков:

**Проект** – включает в себя все модули, формы и связанные с приложением объекты, относящиеся к конкретному документу. Проект содержит модули.

**Модуль** – это функционально завершенный фрагмент программного кода, имеющий имя и который может быть использован как самостоятельный компонент в других проектах.

**Процедура** – это наименьшая единица программного кода, имеющая имя, которая может выполняться независимо. VBA распознает два главных типа процедур: **Sub** и **Function**. Главное отличие между процедурами **Function** и **Sub** состоит в том, что процедура **Function** возвращает результат, процедура **Sub** – нет:

- ♦ процедура **Sub** – набор команд, решающих определенную задачу. При ее запуске выполняются команды процедуры, а затем управление передается в приложение пакета MS Office или процедуру, которая вызвала данную процедуру. В упрощенном виде структура любой процедуры выглядит:

```
Sub <имя> [(<аргументы>)]
```

```
.....
```

```
End Sub
```

♦ процедура Function (функция) также представляет собой набор команд, который решает определенную задачу. Такая процедура заключается между вот такими открывающим и закрывающим операторами:

```
Function
```

```
...
```

```
End Function
```

Приведем пример процедуры Function на VBA именем SumMinus, которая получает три аргумента a, b, c типа Double.

```
Function SumMinus(a As Double,b As Double, c As Double) As Double
```

```
SumMinus = a + b - c
```

```
End Function
```

Вызвать процедуру Function можно из другой процедуры VBA при помощи простого присваивания этой процедуры переменной. В следующем примере показано обращение к процедуре SumMinus, которая была создана выше.

```
Sub main()
```

```
Dim total as Double
```

```
total = SumMinus(5, 4, 3)
```

```
End Sub
```

Область действия процедуры, также как и область действия переменных, можно задать при помощи ключевых слов **Public** и **Private**.

**Оператор** – это элементарная единица программного VBA-кода, способная выполняться. Например, оператор присваивания, оператор цикла и др.

На данном этапе мы будем создавать программы в рамках следующей синтаксической конструкции:

```
[Option Explicit]
```

[Private | Public] Sub <Имя> ([<Список аргументов>])

[Dim <Имя> [As <Тип>]]

[Const <Имя> [As <Тип>] = <Выражение>]

[Инструкции]

[Exit Sub]

[Инструкции]

End Sub

В этой записи:

- **Option Explicit** – инструкция, которая обязывает явное описывать все переменные, встречающиеся в программе;
- **Public** – ключевое слово, которое указывает на область видимости процедуры - доступна для всех других процедур во всех модулях;
- **Private** – процедура доступна для других процедур модуля, в котором она описана;
- <Имя> – имя процедуры;
- <Список аргументов> – список переменных, представляющий аргументы, которые передаются в процедуру при ее вызове. Имена переменных разделяются запятой.
- **Dim <Имя> [As <Тип>]** – блок описания переменных;
- **Const <Имя> [As <Тип>] = <Выражение>** – блок описания констант;
- **Инструкции** – набор любых операторов VBA;
- **Exit Sub** –немедленный выход из процедуры;
- **Sub, End Sub** –служебные слова VBA.

*Внимание! Здесь и далее по тексту в приведенных синтаксических конструкциях операторов VBA угловые скобки < > свидетельствуют об*



обязательном аргументе, а квадратные [ ] – необязательный параметр, т.е. может быть, а может не быть.

## 2. Ввод и запуск программы на выполнение

Будем осуществлять написание собственного кода на языке VBA в среде редактора Visual Basic приложения MS Excel.

- Для написания кода выполните команду **Insert** (Вставить)/**Procedure**(Процедуру).
- В появившемся окне **Add Procedure** (Рисунок 1.7) в поле **Name** необходимо ввести имя процедуры или функции и задать область ее действия.

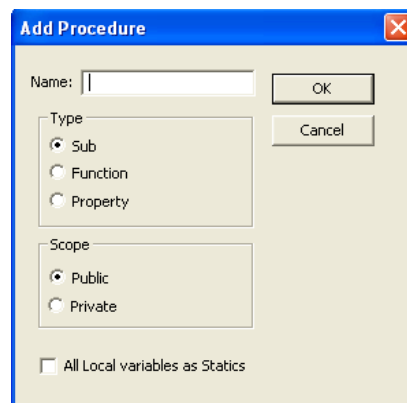



Рисунок 1.7. Окно Добавление процедуры

- Введите пример, затем нажмите **OK**.

После этого будут автоматически сформированы операторы начала и конца процедуры и можно переходить непосредственно к набору операторов процедуры.

Для набора следующей процедуры в том же модуле необходимо повторить команду **Insert**(Вставить)/**Procedure**(Процедуру). Если нужно создать новый модуль повторяется команда **Insert**(Вставить)/**Module**(Модуль).

Проверка правописания осуществляется на этапе компиляции командой **Debug**(Отладка)/**Compile VBAproject**(компилировать).

Для запуска программы требуется выполнить команду **Run**(Запуск)/**Run Sub/UserForm** (Запуск подпрограммы/UserForm) или нажать клавишу <**F5**> или соответствующую кнопку панели инструментов .

### 3. Встроенные диалоговые окна

#### 3.1. Операторы ввода и вывода

В программном коде на VBA ввод и вывод организуется с помощью встроенных диалоговых окон: окон ввода и окон сообщений. Окно сообщений (процедура MsgBox) выводит простейшие сообщения для пользователя, а окно ввода (Функция InputBox) обеспечивает ввод информации.

Функция InputBox выводит на экран диалоговое окно, содержащее сообщение и поле ввода, устанавливает режим ожидания ввода текста пользователем или нажатия кнопки. Затем возвращает введенному значению тип String (текстовый).

Синтаксис:

InputBox (Prompt, [, title] [, default])

Аргументы:

Prompt – текст в диалоговом окне;

title – заголовок диалогового окна. Если этот аргумент не указан, диалоговое окно называется по имени приложения;

default – значение по умолчанию в поле ввода. Если этот аргумент опущен, поле ввода изображается пустым.

#### Пример

- Введите команду  $A = \text{InputBox} ("Введите A", , 2)$

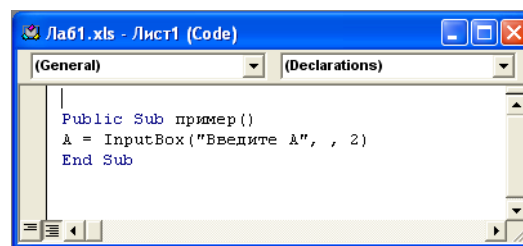


Рисунок 1.8. Процедура Пример

- Запустите на выполнение. На экране появится диалоговое окно, где в поле ввода по умолчанию будет число 2 (Рисунок 1.9).

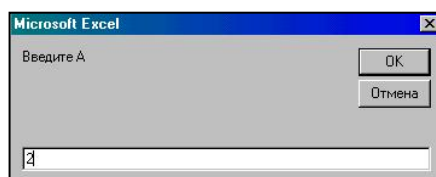


Рисунок 1.9. Стандартное окно ввода

В программе можно использовать данные, расположенные в ячейках рабочего листа MS Excel.

#### Пример

$A = Cells(1, 1)$

После выполнения этого оператора переменной A присвоится значение, которое хранится в ячейке, находящейся в первой строке (первая цифра) и в первом столбце A (вторая цифра), т.е. в ячейке A1 электронной таблицы.

Вывод информации в VBA осуществляется с помощью оператора вывода. Процедура MsgBox выводит на экран диалоговое окно, содержащее сообщение, устанавливает режим ожидания нажатия кнопки пользователем.

Синтаксис:

MsgBox (Prompt, [, button][, title])

Аргументы:

Prompt – строковое выражение, отображаемое как сообщение в диалоговом окне.

button – числовое выражение, представляющее сумму значений, которые указывают число и тип отображаемых кнопок, тип используемого значка, основную кнопку. Значение этого аргумента по умолчанию равняется 0.

Таблица 1

Значения аргумента, определяющие отображаемые кнопки

Константа	Значение	Отображаемые кнопки
VbOKOnly	0	ОК
VbOKCancel	1	ОК, Отмена
VbAbortRetryIgnore	2	Стоп, Повтор, Пропустить
VbYesNoCancel	3	Да, Нет, Отмена
VbYesNo	4	Да, Нет
VbRetryCancel	5	Повтор, Отмена

Таблица 2

Значения аргумента, определяющие отображаемые значки:

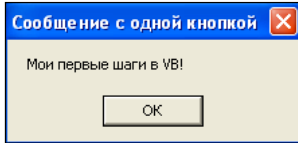
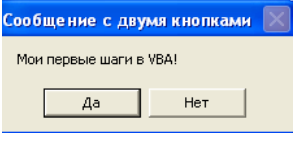
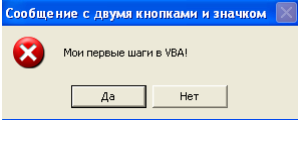
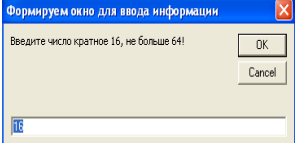
Константа	Значение	Значок сообщения
VbCritical	16	
VbQuestion	32	
VbExclamation	48	
VbInformation	64	

`title` – строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот аргумент опущен, в строку помещается имя приложения.

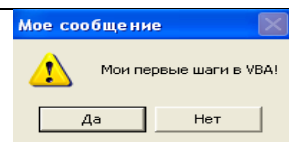
Этот оператор выводит на экран диалоговое окно, содержащее сообщение, устанавливает режим ожидания нажатия пользователем кнопки, а затем возвращает в программу.

### Пример

В результате выполнения приведенных ниже строк программы на экране будут появляться окна. В таблице отражены окна, соответствующие определенным участкам кода.

MsgBox “Мои первые шаги в VB!”, , “Сообщение с одной кнопкой”	
MsgBox “Мои первые шаги в VBA!”, vbYesNo, “Сообщение с двумя кнопками”	
MsgBox “Мои первые шаги в VBA!”, vbYesNo + vbCritical, “Сообщение с двумя кнопками и значком”	
<code>x = InputBox (“Введите число, кратное 16, не больше 64!”)</code> , “Формируем окно для ввода информации”, 16)	

MsgBox “Мои первые шаги в VBA!”, vbYesNo + 48,  
“ Мое сообщение”



- Добавьте в процедуру пример приведенные команды (Рисунок 10) и запустите программу на выполнение.

```
Ла61.xls - Лист1 (Code)
(General)
пример

Public Sub пример()

A = InputBox("Введите A", , 2)
MsgBox "Мои первые шаги в VBA!", , "Сообщение с одной кнопкой"
MsgBox "Мои первые шаги в VBA!", vbYesNo, "Сообщение с двумя кнопками"
MsgBox "Мои первые шаги в VBA!", vbYesNo + vbCritical, "Сообщение с двумя кнопками"
MsgBox "Мои первые шаги в VBA!", vbYesNo + vbCritical, "Сообщение с двумя кнопками и значком"
x = InputBox("Введите число, кратное 16, не больше 64!", "Формируем окно для ввода информации", 16)
MsgBox "Мои первые шаги в VBA!", vbYesNo + 48, "Мое сообщение"

End Sub
```

Рисунок 10 Программный код формирования диалоговых окон

В операторе можно выводить также значения нескольких переменных.

### Пример

*MsgBox (“Значение A=” & A & “, значение B=” & B)*

Символ “&” в операторе означает слияние в одну строку всех символьных строк, записанных в скобках.

## ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Каждый студент выполняет два варианта задания. В отчет записать исходное задание. Составить программу его выполнения. Ввести программу в редакторе VBA, запустить на выполнение.

### Вариант 1

Напишите команды для организации ввода в программу двух чисел. Окна ввода должны иметь разные параметры (строку заголовка, сообщение в окне, значения по умолчанию).

### Вариант 2

Напишите программу диалога с пользователем. Вначале появляется окно ввода, в котором ваше имя (значение по умолчанию). После обработки

полученных данных, на экране должно появиться окно сообщения с текстом приветствия, содержащего обращение по имени, введенному ранее.

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите основные компоненты окна редактора VBA.
2. Структура программного кода.
3. Правило написания операторов ввода и вывода.
4. Как в одном диалоговом окне вывести значения нескольких переменных?