

ИНФОРМАТИКА

Туркин О. В.



VBA

Практическое программирование



**Поурочное
планирование**

О программировании - доступно
Реализация алгоритмических конструкций
Мощности Excel и возможности программирования
Офисное программирование - это просто
Неожиданные возможности VBA
Программы на VBA для психологов
математиков, физиков



ЭЛЕКТИВНЫЙ КУРС * ПРОФИЛЬНОЕ ОБУЧЕНИЕ

УДК 004
ББК 32.973
Т 88

О. В. Туркин

VBА. Практическое программирование. —

М.: СОЛОН-ПРЕСС, 2010. — 128 с.: ил. —

(Серия «Элективный курс * Профильное обучение»)

ISBN 5-98003-304-1

Книга посвящена применению известного для многих пользователей компьютера языка программирования Visual Basic for Application в различных областях. Популярность этого языка объясняется тем, что он основан на широко известном языке Basic и одновременно связан с офисными приложениями. Эти особенности позволяют легко решить многие задачи связанные с работой в электронной таблице и другими продуктами компании Microsoft Office. В первой главе изложены основные принципы программирования, там же читатель может познакомиться с основными объектами, их свойствами и методами. Вторая глава посвящена особенностям применения этого языка при создании самостоятельно всевозможных программ для специалистов самого разного профиля. В третьей главе можно познакомиться с примерами программ, с помощью которых решаются практические задачи. Особенностью этой книги является то, что она может служить учебным пособием при изучении языка программирования. Текст сопровождается рядом упражнений, поэтому ее можно рекомендовать также учителям, студентам и школьникам.

КНИГА — ПОЧТОЙ

Книги издательства «СОЛОН-ПРЕСС» можно заказать наложенным платежом (оплата при получении) по фиксированной цене. Заказ оформляется одним из двух способов:

1. Послать открытку или письмо по адресу: 123242, Москва, а/я 20.
2. Оформить заказ можно на сайте www.solon-press.ru в разделе «Книга — почтой».

Бесплатно высылается каталог издательства по почте.

При оформлении заказа следует правильно и полностью указать адрес, по которому должны быть высланы книги, а также фамилию, имя и отчество получателя. Желательно указать дополнительно свой телефон и адрес электронной почты.

Через Интернет Вы можете в любое время получить свежий каталог издательства «СОЛОН-ПРЕСС», считав его с адреса www.solon-press.ru/kat.doc.

Интернет-магазин размещен на сайте www.solon-press.ru

По вопросам приобретения обращаться: **ООО «АЛЬЯНС-КНИГА КТК»**

Тел: (495) 258-91-94, 258-91-95, www.abook.ru

Сайт издательства «СОЛОН-ПРЕСС»: www.solon-press.ru

E-mail: solon-avtor@coba.ru

ISBN 5-98003-304-1

© Туркин О. В., 2010

© Макет и обложка «СОЛОН-ПРЕСС», 2010

1. Основы VBA

Первые шаги — создание макросов

Изучение VBA мы начнем с создания макросов. Макрос — это и есть та самая запись последовательности нажимаемых клавиш. Записав макрос, мы потом посмотрим, какую же программу мы с Вами практически создали и попытаемся разобраться в ней. Работать мы будем в электронной таблице (ЭТ) Excel. Это офисное приложение самое эффективное в плане применения VBA.

Упражнение 1

Откройте электронную таблицу (ЭТ) Excel с новым файлом и убедитесь, что Вы находитесь на 1 листе.

1. В ячейку A4 введите число **2**, а в ячейку B4 — число **3**
2. В ячейку C4 введите формулу: $=A4*B4$. В результате этих действий в C4 получен результат **-6**. То есть мы с Вами выполнили одно из стандартных действий в ЭТ.

Теперь попробуем записать эти действия в виде макроса. Для этого необходимо сделать следующее

3. Очистите содержимое ячеек A4:C4 и установите курсор в любую ячейку, кроме очищенных.

4. Выберите в меню пункты **Сервис/Макрос/Начать запись**. Появится окно в котором для нас важным является имя макроса «*Макрос1*», который Вам предлагается сохранить в «*Этой книге*». Нажмите «ОК». На листе появится окошко **Остановка записи** (В заголовке будет видна лишь часть записи — «Останов...»). С этого момента все Ваши действия будут записаны, и по этим действиям будет составлена программа.

5. Опять проделайте пункты 1 и 2 из этого задания.
6. Нажмите кнопку с прямоугольником в окне **Остановка записи** Таким образом Вы остановили процесс записи.

Итак, Вы записали свой первый макрос.

Важное замечание

Многие из Вас, наверно, заметили, что после выбора в меню команд «Сервис», «Макрос» можно увидеть строчку «Безопасность». Это строка позволяет подготовить офисные приложения для работы с VBA. По умолчанию, в начальном состоянии все приложения

имеют высокий уровень безопасности и поэтому программы, составленные на языке VBA запускаться не будут (исключения составляют программы, входящие в шаблоны и надстройки, а также записанные Вами макросы). Для работы с программами на VBA необходимо установить средний или низкий уровень безопасности и перезагрузить приложение. В случае если Вы установите среднюю безопасность, то при запуске приложения оно будет проверять наличие программ на VBA и формировать запрос, в котором пользователь информируется о наличии макросов. Пользователь должен принять решение — отключать макросы или оставить их включенными. Форма запроса зависит от версии пакета MS Office. Понятно, что программы на VBA будут работать, если пользователь выберет режим, при котором макросы **ВКЛЮЧЕНЫ**. В случае низкого уровня безопасности без предупреждения будут работать и программы, о которых знает пользователь, и программы — вирусы.

Меню макроса. Запуск макросов. Чтение макросов

Макрос записан. Посмотрим, как он будет работать. Для этого необходимо очистить заполненные Вами ячейки рабочего листа, выбрать в меню **Сервис/Макрос/Макросы** и выбрав «Макрос1» нажать кнопку «Выполнить». В результате, созданный Вами макрос выполнит все записанные Вами действия. Отметим, что если Вы во время записи нажимали еще какие-то клавиши, то эти нажатия также будут зафиксированы.

Теперь давайте познакомимся с текстом записанного макроса. Выберите **Сервис/Макрос/Макросы**, выберите «Макрос1» и нажмите кнопку «Войти».

В результате откроется окно **Microsoft Visual Basic**, внутри которого будет располагаться несколько других окошек, но в центре вы увидите окно, в котором будет текст Ваших макросов.

Вот как они примерно будут выглядеть

```
Sub Макрос1()
```

```
,
```

```
'Макрос1 Макрос
```

```
'Макрос записан 28.08.05 (Туркин О.В.)
```

```
Range("A4").Select
```

```
ActiveCell.FormulaR1C1 = "2"
```

```
Range("B4").Select
```

```
ActiveCell.FormulaR1C1 = "3"
```

```
Range("C4").Select
```

```
ActiveCell.FormulaR1C1 = "="+RC[-2]*RC[-1]"
```

```
End Sub
```

Проанализируем содержимое макроса. Макрос начинается словами Sub... и заканчивается End Sub — это обязательные компоненты. После апострофов следуют комментарии о макросе.

```
'Макрос1 Макрос
```

```
'Макрос записан 28.08.05 (Туркин О.В.)
```

```
— необязательные компоненты.
```

```
Затем в строках
```

```
Range("A4").Select
```

```
ActiveCell.FormulaR1C1 = "2"
```

указывается на ячейку A4, дается команда на ее выделение (**Select**), и в выделенную, активную ячейку (**ActiveCell.**) вводится формула (**FormulaR1C1**), а в данном случае просто число **2**. Затем также вводится в другую выделенную ячейку число **3**, и в очередную ячейку вводится формула, в которой перемножаются два числа находящиеся в двух ячейках той же строки, но соответственно на 2 и 1 столбец левее ("**="+RC[-2]*RC[-1]"**)

Редактирование макросов

Теперь, чтобы отредактировать, изменить записанный макрос, нам нужно поработать в среде «Microsoft Visual Basic». В эту среду можно перейти несколькими способами. Один способ реализуется через команды *Сервис/Макрос/Редактор Visual Basic*, а второй — через панель *Visual Basic*. Эта панель может быть выведена через команды *Вид/Панели инструментов/Visual Basic*. В этой панели также есть кнопка *Редактор Visual Basic*.

Итак, мы оказались в редакторе VBA и имеем возможность изменить содержимое макросов. Пока мы может сделать совсем немного — а именно изменить содержимое формул. Например, можно вместо 2 или 3 в код макроса внести другую цифру, но более интересным представляется такое редактирование, когда результат вычисления будет представлен не в ячейке «C4», а в другой ячейке, но при этом необходимо изменить относительные ссылки в самой формуле. Ниже предлагается вариант отредактированного

макроса1, когда происходит перемножение 4 на 7, но результат помещается не в «C4», а в «D6».

```
Sub Макрос1()  
'Макрос1 Макрос  
'Макрос записан 28.08.05 (Туркин О.В.)  
  Range("A4").Select  
  ActiveCell.FormulaR1C1 = "4"  
  Range("B4").Select  
  ActiveCell.FormulaR1C1 = "7"  
  Range("d6").Select  
  ActiveCell.FormulaR1C1 = "="+R[-2]C[-3]*R[-2]C[-2]"  
End Sub
```

Особенности применения макросов в различных приложениях

Подведем небольшие итоги. Фактически мы познакомились с самым простым способом создать программу (грамотнее употребить слово «процедуру») — это записать ее. Этот метод пригоден для Word, Excel и PowerPoint. Макрос можно рассматривать в данном случае как простую процедуру Sub. Но для каждого приложения существуют свои особенности.

При записи в Excel имеется возможность записывать как относительные, так и абсолютные ссылки на ячейки. В первом случае записывается перемещение или выбор ячеек относительно активной ячейки. Например, при нажатии клавиши <стрелка вверх> Excel регистрирует перемещение на одну ячейку вверх, поэтому при последующем выполнении макроса выделяется ячейка, находящаяся над активной. Если то же самое действие записывать с помощью абсолютных ссылок, то всегда выбирается одна и та же ячейка, а именно та, которую выделили при записи.

При записи макроса Excel выводится панель инструментов **Остановка записи (Stop Recording)**, на которой имеется кнопка **Относительная ссылка (Relative Reference)**, служащая для того, чтобы переключиться между записью с относительными и абсолютными ссылками

Word не позволяет при записи использовать мышшь для работы с текстом, поэтому необходимо пользоваться клавиатурой, чтобы переместить курсор или выделить текст,

В Access макросы создаются с помощью собственного языка макросов, а не записываются автоматически. Более того, макросы Access не являются процедурами VBA.

Прежде чем записать макрос, требуется продумать свои действия, а именно

1. Какие условия должны выполняться при запуске макроса:

- Должен ли быть открыт некоторый файл?
- Должен ли курсор находиться в определенном положении или должна быть активной определенная ячейка?
- Требуется ли включить определенный режим работы приложения?

2. Какие действия должен выполнять макрос? Рекомендуется перед записью потренироваться и выполнить пробный заход. Во многих случаях не играет роли, что используется при записи: мышь, клавиатура или команды меню, однако в записи будет меньше ошибок, если заранее продумать последовательность своих действий.

3. Тщательно продумайте, что необходимо проделать при завершении работы макроса:

- Требуется ли установить курсор в некоторую позицию или сделать активной некоторую ячейку?
- Требуется ли закрыть открытый файл или вернуть к исходным любым измененным установкам?

Хороший макрос характеризуется тем, что по завершении его работы приложение имеет то же состояние, что и до его запуска, за исключением тех изменений, которые должен был произвести по Вашему плану макрос.

Чтобы начать запись макроса, необходимо проделать следующие действия

1. Активизируйте основное приложение (Word или Excel или PowerPoint).

2. Откройте документы (в Word — это текст, в Excel — книга), используемые при записи макроса.

3. Выберите команду **Сервис/Макрос**.

4. Выберите команду «Начать запись». Выводится диалоговое окно **Запись макроса**.

5. Введите имя записываемого макроса в поле «Имя макроса».

6. Выберите в списке «Макрос доступен для» документ, в который требуется поместить макрос. Вы имеет возможность разместить макрос или только в данном документе или в Ваших шаблонах

7. Введите или измените текст в поле «Описание».

8. Нажмите кнопку «ОК», чтобы начать запись.

9. Выполните те действия, которые Вы хотите записать в макросе.

10. По окончании записи нажмите на панели инструментов **Остановка записи** кнопку «Остановить запись» или выберите команду **Сервис/Макрос/Остановить запись**.

Вновь созданный макрос теперь содержится в документе, который был активным при запуске записи или в Ваших шаблонах. При записи документа на диск макрос также сохраняется.

В ряде случаев записанные макросы выполняют не совсем те действия, которые требовались от них. Основная причина заключается в том, что информация в командах макроса в точности соответствует введенной при записи, а условия работы с документом могли измениться. По этой причине часто необходимо отредактировать записанные макросы.

Тем не менее, запись позволяет существенно сократить время на создание приложений, а также является эффективным способом освоить VBA.

Переходим к VBA

Запись макроса — не самый лучший способ создать процедуру. По мере возрастания сложности макросов требуется использовать переменные, формы, а также инструкции, управляющие порядком выполнения команд программы, и поэтому многие из средств языка автоматически записать невозможно. Постепенно осваивая язык и разрабатывая свой стиль программирования, можно полностью отказаться от записи макросов и создавать процедуры, работая непосредственно в редакторе VBA.

Прежде чем перейти в рабочую среду программирования VBA рекомендуется открыть в электронной таблице панель элементов управления. Используем команды **Вид/Панели инструментов/элементы управления**.

Познакомимся с этой панелью. Она по смыслу разделена на три области. Нас сначала интересует первая область, в которой на-

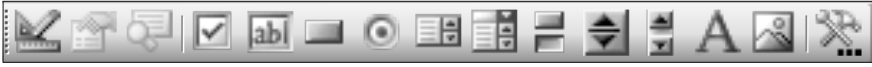


Рис. 1.1. Панель элементов управления

ходятся три кнопки — «Режим конструктора», «Свойства», «Исходный текст». Нажав на кнопку **Исходный текст** мы попадаем в среду программирования VBA.

Но прежде чем туда отправиться, давайте запомним очень важное правило — все наши программы и макросы будут работать, если кнопка «Режим конструктора» будет отжата, и в противном случае мы можем заниматься конструированием наших элементов управления, но программу запустить не сможем.

Что же касается кнопки **Свойства** то она открывает для нас окно свойств различных объектов и элементов управления. Обращаем внимание на то, что эти кнопки могут встречаться не только в этой панели, но и в других панелях. Назначение этих кнопок, естественно такое же.

Вторая часть этой панели содержит кнопки — символы, относящиеся к различным элементам управления, с которыми мы будем знакомиться позже. И, наконец, в третьей области располагается одна кнопка «Дополнительные элементы», которая позволяет расширить список используемых элементов управления.

Итак, нажав кнопку **Исходный текст** переходим в среду программирования VBA. В этой среде нам и предстоит писать коды программ. Сначала, для определенности выберем **View/Project Explorer**. С левой стороны Вы должны увидеть появившееся (или оно уже было) окно проектов, в котором в виде дерева отображаются названия открытых Вами книг, а также стандартных шаблонов. Работая в этом окне, так же как и с деревом каталогов в Проводнике Windows, Вы можете раскрыть содержание Вашей книги, увидеть ссылки на рабочую книгу в целом, и на каждый рабочий лист этой книги. Кроме того, Вы можете увидеть, что дерево также содержит и модули.

Выбрав одну из строчек в этом дереве, Вы справа получите раскрытое окно кода для данного элемента рабочей книге. Также это окно можно раскрыть через **View/Code**. Другими словами, если вы собираетесь писать программу для конкретного рабочего листа, то Вы в окне проекта должны выбрать строку с названием этого листа, и тогда раскроется окно кода именно для этого рабочего листа. Если Вы захотите написать код в модуле, то тогда выбирайте

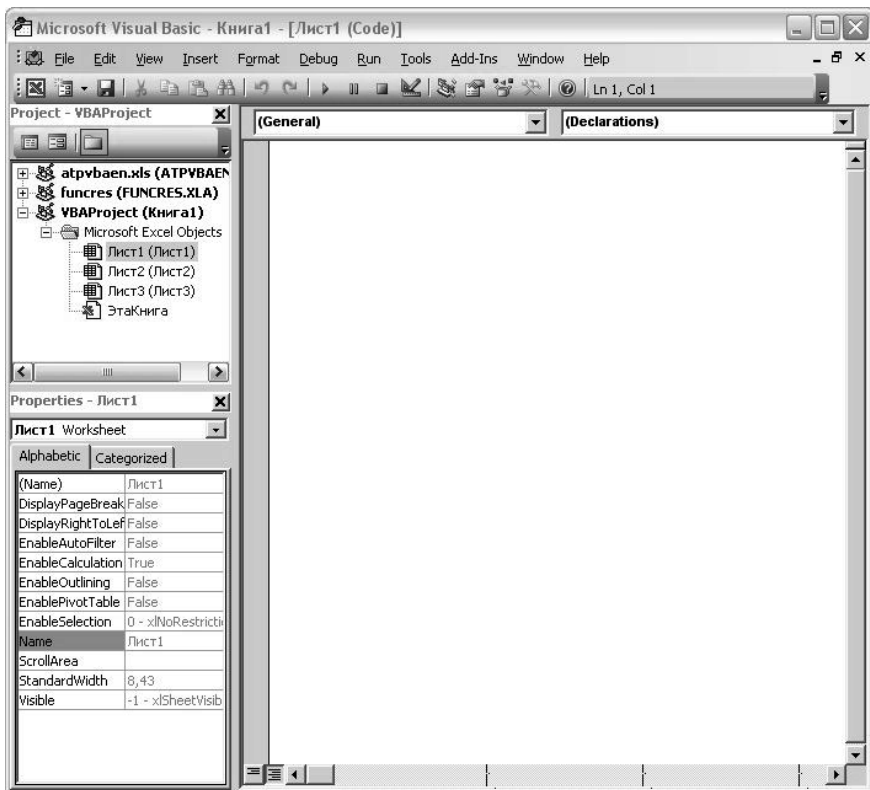


Рис. 1.2. Среда программирования VBA

модуль, или Вы можете создать его. Вставить новый модуль можно либо через команду **Insert/Module** либо через контекстное меню, находясь в окне проекта. Итак, мы можем работать с окном проекта или с окнами кодов для различных объектов или модулей. Кроме того, можно вызвать окно свойств либо через соответствующую кнопку на верхней панели кнопок, либо через **View/Properties Window**.

Обратим внимание на команды в главном меню, которые могут понадобиться в первую очередь. В разделе «Edit» («Правка») располагаются команды, которые будут Вам полезны при написании кодов программы и выполняют функции форматирования, но также имеет смысл обратить внимание на строчки «List Properties/Methods» («Список свойств/методов») и «List Constants» («Спи-

сок констант»). С помощью этих команд Вы получаете справочную информацию о возможностях языка программирования.

Команда меню «View» («Вид») позволяет Вам открывать различные объекты Вашего проекта, описывать их свойства, просмотреть используемые объекты и скрывать или раскрывать различные панели, а также управлять режимом отладки.

Команда меню «Insert» («Вставка») позволяет вставить в проект новые элементы (см. содержание этой компоненты меню).

Команда «Format» («Формат») позволяет управлять расположением выделенных Вами в режиме конструктора элементов управления, то есть реализовывает хорошо известные команды форматирования.

Команда «Debug» («Отладка») используется в режиме отладки проекта.

Особый интерес представляет интерес команда «Run» («Запуск»). Здесь необходимо отметить следующее. Иногда возникают ситуации, когда выполнение программы по каким либо причинам прерывается, и программа находится в режиме отладки или паузы. Но пользователь или программист об этом забывает и никак не может запустить программу вновь. Давайте запомним, что если программа почему-то не запускается (хотя Вы не находитесь в режиме конструктора), то, скорее всего, необходимо нажать кнопку «Reset» («Сброс»), чтобы иметь возможность запустить программу сначала.

Не будем подробно останавливаться на командах «Tools» («Сервис») и «Windows» («Окно»), так как их назначение достаточно прозрачно. Отметим только, что команда «Tools/Option» («Сервис/Параметры») относится к установке параметров самой среды программирования, а команда «Tools/VBAProject Properties» («Сервис/Свойства проекта») позволяет установить ряд свойств Вашего проекта, и в частности установить на него защиту от вмешательства «нехороших людей».

Объекты, их свойства, события и методы

Теперь наступила пора познакомиться с объектами, событиями, свойствами и методами VBA. К объектам в первую очередь можно отнести те элементы различных приложений, которыми Вы пользуетесь при работе на компьютере. Это хорошо известные Вам кнопки, полосы прокрутки, различные иконки, строки в

Содержание

Введение	3
1. Основы VBA	5
Первые шаги — создание макросов	5
Упражнение 1	5
Меню макроса. Запуск макросов. Чтение макросов	6
Редактирование макросов	7
Особенности применения макросов в различных приложениях	8
Переходим к VBA.....	10
Объекты, их свойства, события и методы	13
Стандартные элементы управления	16
Упражнение 2	16
Структура проекта на языке VBA.....	20
Создание процедур	22
Упражнение 3	22
Типы данных, переменные, константы	26
Упражнение 4	32
Основные конструкции языка	34
Процедуры и функции	42
Описание переменных.....	48
Массивы.....	49
Отладка программ.....	51
2. Особенности применения VBA в офисных приложениях или «Быстрое» программирование	54
Основные объекты электронной таблицы Excel	54
Использование справочной системы VBA.....	59
Упражнение 5	60
О «быстром» программировании.....	61
Создание пользовательской функции	62
Упражнение 6	62
Ввод и вывод данных.....	64
Упражнение 7	64
Быстрое построение графика.....	67
Выделяем ячейку...и получаем массу возможностей	69
Однорукий бандит.....	71
Секундомер.....	72

Живая диаграмма	75
Упражнение 8	75
Создание собственного меню	76
Применение VBA в редакторе Word	84
3. Проекты на VBA.....	89
Анализ текста	89
Предварительные замечания	89
Программа для лингвистов	90
Подготовка данных к экспорту.....	91
Об объектных переменных и подсчет числа букв.....	92
Код программы	93
Проект — игра «Теннис».....	95
Создание изображений в диаграмме	95
Программирование движения.....	96
О человечке.....	97
О работе процедуры <code>prov</code>	97
У снарядов разные скорости	98
Обработка социометрических исследований.....	98
Цель проекта.....	98
Структура таблицы.....	98
Новые данные	99
Использование массивов и циклов	100
Сведения для печати.....	101
Личные сведения.....	102
В целом о коде	102
Моделирование физического процесса.....	109
Немного физики и математики	109
Оформляем рабочий лист.....	110
Построение диаграммы.....	111
Шарики должны двигаться	111
Проверка программы	115
Создание оригинальных тестов.....	115
Цель проекта.....	115
План проекта	116
Работаем с ячейками.....	116
Особенности программы	117
Развитие проекта	120
Заключение	122