

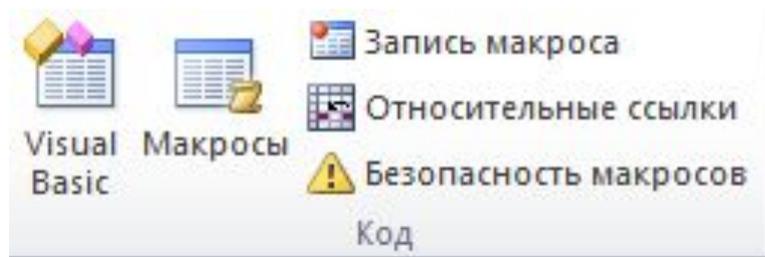
Знакомство с редактором Visual Basic. Принципы создания собственных макросов. Средства отладки.

- Общие сведения.
- Окно проводника проекта (Project Explorer) и структура проекта VBA.
- Работа с редактором кода (Code Editor).
- Работа со справкой.

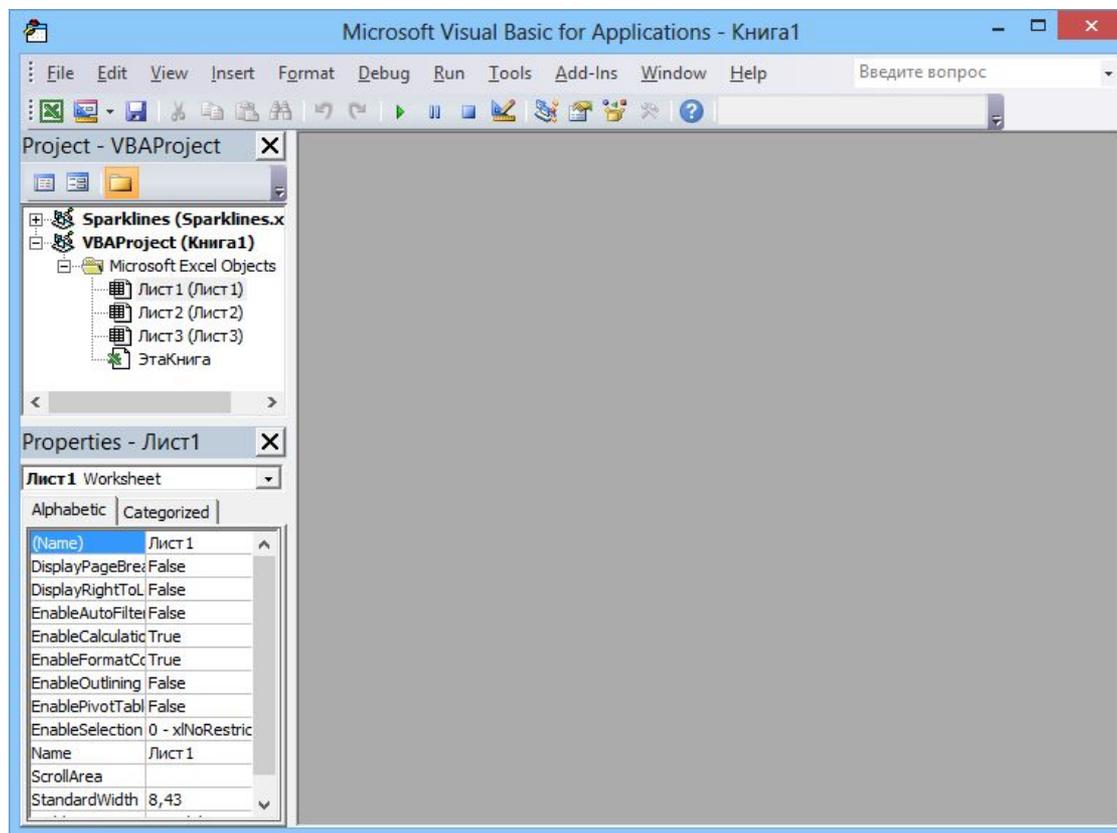
Создание макросов в редакторе Visual Basic

Для ввода команд и формирования программы, т.е. создания макроса необходимо открыть специальное окно - редактор программ на VBA, встроенный в Microsoft Excel.

- В старых версиях (Excel 2003 и старше) для этого идем в меню Сервис - Макрос - Редактор Visual Basic (Tools - Macro - Visual Basic Editor).
- В новых версиях (Excel 2007 и новее) для этого на вкладке Разработчик (Developer) служит кнопка Редактор Visual Basic (Visual Basic Editor)



Интерфейс редактора VBA и большинство файлов справки не переводятся компанией Microsoft на русский язык



Но много информации по VBA можно найти на официальном сайте <https://msdn.microsoft.com/ru-ru>

Всего в редакторе Visual Basic предусмотрено 9 дополнительных окон.

- **Project Explorer** – окно проводника проекта. По умолчанию оно открыто и находится в левой части окна редактора Visual Basic. В нем можно просмотреть компоненты проекта и выполнить множество операций.
- **UserForm** – окно формы. Появляется тогда, когда вы редактируете пользовательскую форму при помощи дизайнера форм.
- **Toolbox** – панель инструментов управления. Из нее можно добавить элементы управления в форму или в сам документ. Появляется вместе с окном формы.
- **Properties** – одно из самых важных окон. Через него можно просмотреть свойства элемента управления или компонента проекта и изменить их.

- **Code** – окно программного кода. В этом окне выполняется основная работа по написанию кода макроса. При открытии программного модуля открывается автоматически.
- **Object Browser** – обозреватель объектов. Необходим для получения информации о классах, доступных программе.
- **Watch** – окно контролируемых выражений. Используется во время отладки для отслеживания значений выбранных переменных программы и выражений.
- **Locals** – окно локальных переменных. Нужно для отслеживания во время отладки значений переменных текущей процедуры;
- **Immediate** – окно для немедленного выполнения команд в ходе отладки. Оно позволяет выполнить отдельные строки программного кода и немедленно получить результат.

Чтобы найти какое-либо окно нужно выбрать в меню **View** соответствующую команду, и если окно было скрыто, оно появится в редакторе.

Особенности главного меню VBA-редактора

File (Файл) — служит для работы с файлами.

File • Save — сохраняет файл.

File • Import File — позволяет импортировать внешний файл в редактор. Например, таким образом можно добавить в свой проект модуль.

File • Export File — экспортирует данные из редактора во внешний файл. Например, этой командой можно сохранить редактируемую форму и передать ее другому разработчику.

Edit (Правка) — содержит команды для правки. Помимо стандартных команд отмены и возврата последнего действия (Undo, Redo), вырезания, копирования, вставки (Cut, Copy, Insert), поиска (Find), это меню содержит несколько особенных команд. В частности, это List Properties/Methods (Список Свойств/Методов) и другие.

View (Вид) — содержит команды для отображения различных окон редактора. Названия команд соответствуют названиям окон.

Insert (Вставка) — служит для вставки в проект форм, модулей, процедур, файлов и модулей класса. Эти команды продублированы на панели инструментов редактора.

Format (Форматирование) — служит для управления расположением элементов управления на формах.

Debug (Отладка) — содержит команды, позволяющие отлаживать программы.

Run (Запуск) — содержит команды для управления выполнением программ.

Run • Sub/User Form (Запуск • Процедура/Форма) запускает на выполнение активную процедуру или форму (рядом с этой командой стоит характерный зеленый треугольник).

Run • Break (Запуск • Приостановить) — приостанавливает выполнение.

Run • Reset (Запуск • Перезапуск) – останавливает выполнение.

Tools (Инструменты) — содержит средства для настройки свойств редактора, подключения дополнительных библиотек объектов.

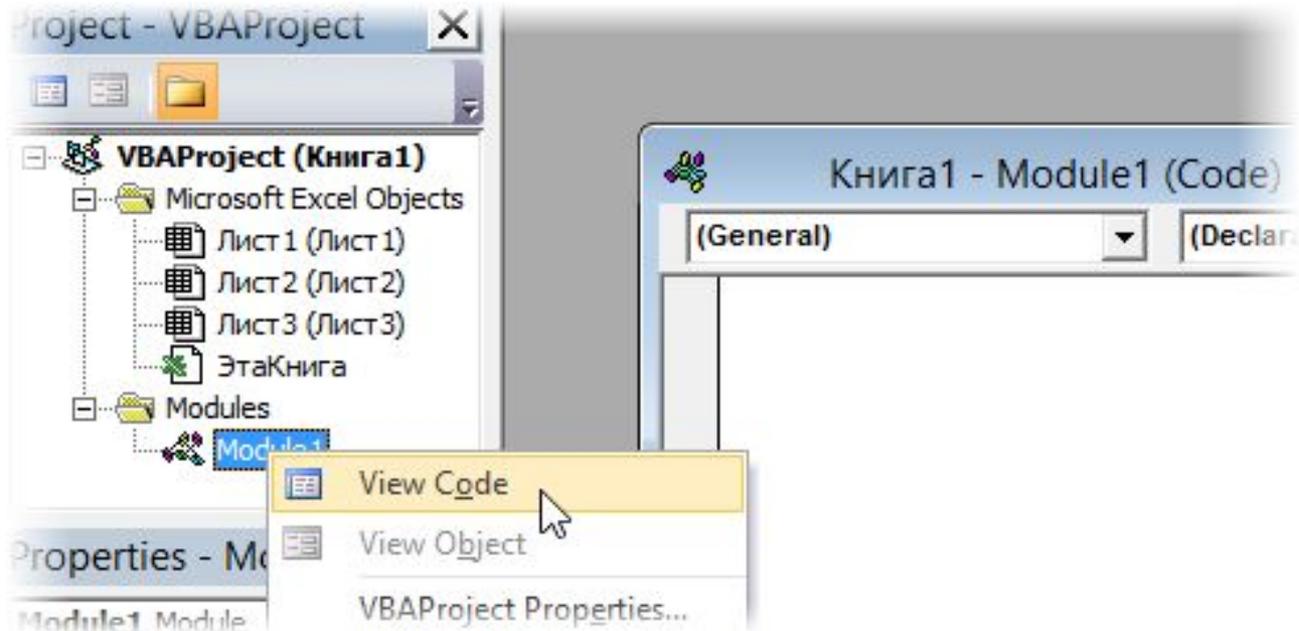
Add-Ins (Дополнения) — позволяет управлять дополнениями. По умолчанию это меню содержит лишь одну команду, запускающую менеджер дополнений.

Window (Окно) — стандартные команды для работы с окнами.

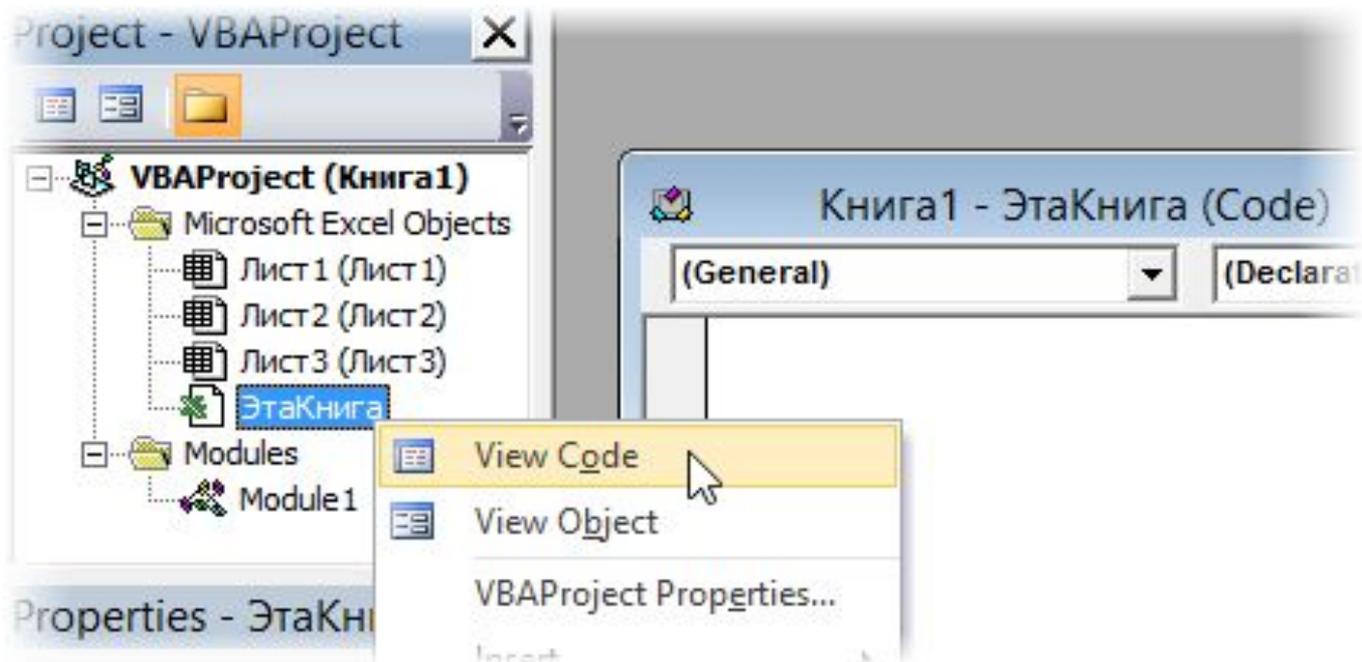
Help (Помощь) — помощь по VBA.

- Макросы (т.е. наборы команд на языке VBA) хранятся в **программных модулях**.
- В любой книге Excel мы можем создать любое количество программных модулей и разместить там наши макросы.
-
- Один модуль может содержать любое количество макросов.
- Доступ ко всем модулям осуществляется с помощью окна **Project Explorer** в левом верхнем углу редактора (если его не видно, нажмите CTRL+R).
- Программные модули бывают нескольких типов для разных ситуаций.

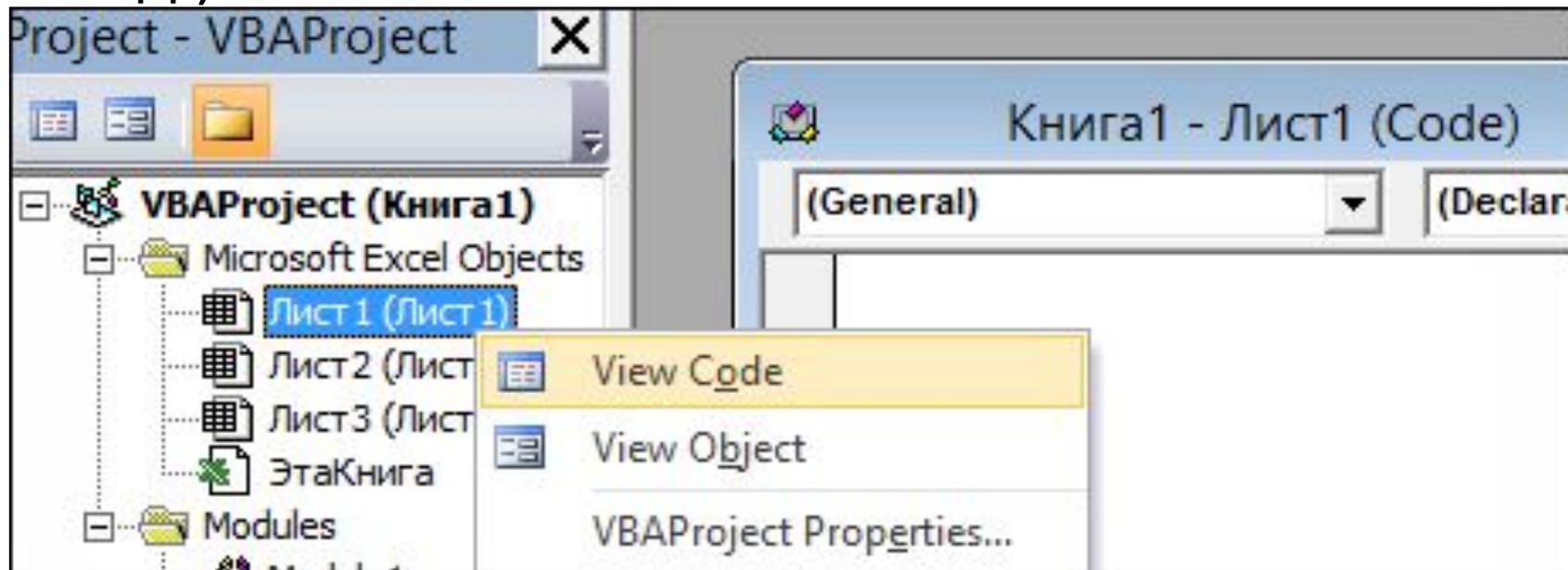
- **Обычные (стандартные) модули** - используются в большинстве случаев, когда речь идет о макросах.
- Для создания такого модуля выберите в меню Insert - Module. В появившееся окно нового пустого модуля можно вводить команды на VBA, набирая их с клавиатуры или копируя их из другого модуля, и т.д.



- **Модуль Эта книга** - также виден в левом верхнем углу редактора Visual Basic в окне, которое называется Project Explorer.
- В этот модуль обычно записываются макросы, которые должны выполняться при наступлении каких-либо событий в книге (открытие или сохранение книги, печать файла и т.п.)



- **Модуль листа** - доступен через Project Explorer и через контекстное меню листа, т.е. правой кнопкой мыши по ярлычку листа - команда Исходный текст (View Source).
- Сюда записывают макросы, которые должны выполняться при наступлении определенных событий на листе (изменение данных в ячейках, пересчет листа, копирование или удаление листа и т.д.)



Окно проводника проекта (Project Explorer)

- В окне **Project Explorer** представлено **дерево компонентов** приложения VBA.
- Самый верхний уровень — это **проект (Project)**, которому соответствует документ Word, рабочая книга Excel, презентация PowerPoint и прочие файлы, с которыми работает данное приложение.
- Например, если вы открыли редактор Visual Basic из Word, то в Project Explorer будут представлены все открытые в настоящее время файлы Word и еще шаблон Normal.dot. Если редактор Visual Basic открыт из Excel, то в Project Explorer будут открытые книги Excel и книга PERSONAL.XLS.

- Помимо того, что обычно содержится в документах Office, каждый проект (который и есть документ) — это одновременно и контейнер для хранения **стандартных модулей, модулей классов и пользовательских форм.**
- Добавить в проект каждый из этих компонентов можно при помощи меню **Insert** или через контекстное меню в **Project Explorer.**
- При работе макрорекордера в Excel автоматически создается стандартный модуль **Module1**, куда и записываются все создаваемые макрорекордером макросы.

Стандартные (обычные) модули — это просто блоки с текстовым представлением команд VBA. В них может быть только два раздела:

- ✓ раздел объявлений уровня модуля (объявление переменных и констант уровня модуля);
- ✓ раздел методов модуля (расположение процедур и функций).

В большинстве проектов VBA используется только один стандартный модуль, куда и записывается весь код.

Создавать новые стандартные модули (**Insert –Module**) есть смысл для:

- удобства экспорта и импорта;
- повышения производительности. При вызове любой процедуры модуля происходит компиляция всего модуля, поэтому иногда выгоднее разместить процедуры в разных модулях, чтобы компилировать только нужный в данный момент код;
- улучшения читаемости. Если ваше приложение выполняет разные группы задач, то код, относящийся к каждой группе, лучше поместить в свой модуль

- **Модули классов** позволяют создавать свои собственные классы, по которым можно создавать свои собственные объекты.
- Создать модуль класса можно с помощью контекстного меню **Insert - Class Module**.
- **Пользовательские формы** являются одновременно хранилищем элементов управления (кнопок, переключателей, полей ввода и прочего) и программного кода, который относится к ним, самой форме и происходящими с ними событиями.
- Создать пользовательские формы можно с помощью контекстного меню **Insert – UserForm**.

Еще одна полезная возможность **Project Explorer** — **настройка свойства проекта**.

Для этого нужно щелкнуть правой кнопкой мыши по узлу **VBAProject** и в контекстном меню выбрать **Project Properties** (окно свойств проекта можно открыть и через меню **Tools | Project Properties**).

В этом окне можно:

- изменить имя проекта;
- ввести описание проекта, информацию о файле справки и параметры, которые будут использоваться компилятором;
- защитить проект, введя пароль.

Обязательно подумайте, где вам будет нужен создаваемый код — только в одном документе или во всех документах данного приложения.

Если он будет нужен только в одном документе, используйте стандартный программный модуль этого документа.

Если во всех, то используйте программные модули проекта PERSONAL.XLS.

Работа с редактором кода (Code Editor)

В редакторе кода выполняется основная часть работы по программированию.

Открыть окно редактора кода можно множеством способов:

- выбрать нужный элемент (в Project Explorer, в дизайнера форм и т.п.) и в контекстном меню выбрать View -> Code;
- нажать на кнопку <F7>;
- выбрать View -> Code из меню;
- дважды щелкнуть по объекту модуля в Project Explorer (или выделить его и нажать на кнопку <Enter>).

Редактор программного кода — это по сути обычный текстовый редактор

- В верхней части окна редактора кода находятся два списка.
- Список слева — это **список объектов**. В нем вы можете выбрать объект, к которому будет относиться ваш код. Если вы открыли программный код модуля, то здесь будет только пункт (**General**). Если открыта форма — в этом списке вы сможете выбрать саму форму или любой ее элемент управления и записать для него код.
- Список справа — это **список процедур/событий**. В нем есть раздел (**Declarations**) — объявления уровня всего модуля и список всех процедур (макросов) для стандартного модуля или событий, если создается код для формы. При выборе нужного события будет автоматически создана нужная процедура, обрабатывающая это событие.

- Иногда в процессе написания программного кода в одном месте вам в голову приходит идея, относящаяся к другой части кода. В этом случае используют **закладки**.
- **Закладка** (как и в случае с обычной книгой) — это метка, при помощи которой можно быстро найти нужное место. Работа с закладками производится либо из панели инструментов **Edit** , либо через меню **Edit -> Bookmark**.
- Для того, чтобы включить или отключить закладку, нужно установить указатель ввода на нужную строку и воспользоваться командой **Toggle Bookmark**.
- Часто бывает очень удобно разделить окно редактирования на две части — для просмотра разных частей модуля, для копирования и т.п. Делается это при помощи линии разбивки — маленького бегунка сразу над полосой прокрутки.

Основы объектно-ориентированного программирования

Программа (макрос, функция, процедура) создается в редакторе кода.

Во всех языках высокого уровня программный код состоит из:

- Переменных
- Выражений
- Операторов
- Управляющих структур
- Функций
- Классов и объектов

- VBA является современным языком программирования, полностью оснащенный всеми необходимыми структурными операторами и типами данных.
- Кроме того, в VBA применяются объекты, что делает его использование при разработке прикладных программ более понятным и легким.
- В объектно-ориентированном программировании данные и код, который манипулирует этими данными, объединены в структуре, называемой **объектом**. Объект обладает определенными свойствами и методами.
- **Свойства** - это характеристики объекта, такие как размер, цвет, положение на экране, или состояние объекта, например доступность или видимость.
- **Методы** – это действия, выполняемые над объектом.

Пример.

- `Worksheets("Лист1").Visible = False` - С помощью установки свойству *Видимость* значения *Ложь* скрывается рабочий лист "Лист 1".
- `Worksheets ("Лист 2").Delete` - При помощи метода *Delete* удаляется рабочий лист "Лист 2" .

Программный объект может являться частью другого, большего программного объекта.

Для доступа к свойствам и методам объекта, являющегося составной частью более крупного объекта, нужно определить каждый из сборных объектов, начиная с самого левого (большого объекта), а затем поставить точку и указать следующий, составной объект, затем опять ставится точка и определяется следующий внутренний объект, пока не будет определен объект, к свойствам и методам которого необходимо получить доступ. (аналог адреса)

Пример.

`Workbooks("Книга1").Worksheets("Лист1").Range("A14").Font` – *Получаем доступ к рабочей книге "Книга1", рабочему листу "Лист1", шрифту ячейки "A14".*

Объектами Excel являются таблицы, рабочие книги, диаграммы, области ячеек и др.

Семейство представляет собой объект, содержащий несколько других объектов, как правило, одного и того же типа.

Пример.

Семейство Workbooks объединяет все открытые рабочие книги.

Обратиться к элементу семейства можно по имени или номеру.

Пример.

Worksheets (“Лист1”) или Worksheets (1)

Редактор кода VBA: получение списка свойств и методов, список параметров, автоматическое дополнение слов

В редактор кода встроено множество средств, которые облегчают жизнь разработчику.

- Самое полезное средство — это **получение списка свойств и методов**. В большинстве VBA-программ используются свойства и методы различных объектов, при этом многие методы принимают параметры. Помнить точное название каждого свойства и метода, очередность передачи параметров невозможно, а разыскивать каждый раз справку по этому объекту в документации — непроизводительная трата времени.

1. Чтобы получить **список свойств и методов**, доступных для объекта достаточно напечатать **имя объекта** и за ним — **точку**.

Этот способ работает если включен **автоматический показ** (он включен по умолчанию), если автопоказ отключен, то можно воспользоваться командой **List Properties/Methods** в меню **Edit** или нажать **<Ctrl>+<J>**.

Выбрав нужное свойство/метод, нужно нажать на клавишу **<Tab>**.

Это средство работает и для пользовательских классов/переменных. (Если не работает, необходимо проверить настройки параметра **Auto List Members** в диалоговом окне **Options** меню **Tools->Options**).

2. Получить список аргументов для метода и информацию о них можно автоматически после того, как вы напечатали имя метода, принимающего параметры.

Вручную вызвать список аргументов можно при помощи `<Ctrl>+<I>`.

Включить/отключить автоматический вывод аргументов можно при помощи меню `Tools->Options->Auto Quick Info`.

Комбинация клавиш `<Ctrl>+<Shift>+<I>` — показывает информацию о параметрах, список аргументов для самой внешней функции (в случае вложенности).

3. Список констант (то есть допустимые значения для данного свойства) также появляется автоматически после того, как напечатан знак равенства (=). Можно воспользоваться также комбинацией **<Ctrl>+<Shift>+<J>**.

4. Ключевые слова VBA и имена доступных в данный момент классов очень удобно вводить при помощи **автоматического дополнения слов (Complete Word)**.

Для этого достаточно нажать на **<Ctrl>+<Пробел>**. Можно предварительно ничего не печатать, а можно впечатать одну-две буквы.

Еще несколько моментов, связанных с редактором кода:

- если вы напечатаете одну строку кода с отступом, то тот же отступ будет установлен для следующих строк. Изменить поведение можно при помощи параметра **Auto Indent** в диалоговом окне **Options**;
- если редактор кода распознает ключевое слово, он автоматически делает его первую букву заглавным и выделяет синим цветом;
- часто бывает необходимо закомментировать или раскомментировать несколько строк сразу. Для этой цели можно включить отображение панели инструментов **Edit** и воспользоваться кнопками **Comment Block** и **Uncomment Block**.
- если при создании процедуры вы пишете ключевое слово **Sub** или **Function**, то редактор автоматически дописывает оператор **End Sub** или **End Function**. Между процедурами вставляется строка-разделитель;

- если при переходе на новую строку редактор кода обнаружит синтаксическую ошибку, то вам будет выдано предупреждение. Отменить предупреждения редактора можно, сняв флажок **Auto Syntax Check** в диалоговом окне **Options**. Синтаксически неверные строки в любом случае будут автоматически выделяться красным цветом;
- в редакторе Visual Basic вполне допускается работа сразу с несколькими окнами редактирования кода. Переход между ними осуществляется по **<Ctrl>+<Tab>** или **<Ctrl>+<F6>** или мышкой;
- по умолчанию редактор кода работает в режиме **Full Module View** — показ всего содержимого модуля. Если вы хотите просматривать процедуры по отдельности, переключитесь в режим **Procedure View**. Кнопки для переключения — в левом нижнем углу окна редактора кода.

Работа со справкой

Вызов окна **справки** производится из редактора Visual Basic:

- по нажатию на кнопку <F1>.
- кнопкой Справка на панели инструментов Standard.
- установкой указателя мыши в нужное место в окне редактора кода (например, на имя вызываемого метода или используемого свойства) и нажатием на кнопку <F1>