

Введение в VBA

Объекты и их семейства

- ▶ Что такое объект? Объект позволяет инкапсулировать данные вместе с кодом, предназначенным для их обработки, т. е. объединить их в нечто целое, именуемое объектом.
- ▶ VBA не является объектно-ориентированным языком в строгом понимании этого слова, однако объектный подход играет в нем большую роль.
- ▶ Все визуальные объекты, такие как *рабочий лист* (*worksheet*), *диапазон* (*Range*), *диаграмма* (*Chart*), *форма* (*userForm*), являются объектами. В VBA имеется более 100 встроенных объектов.

Семейство

- ▶ *Семейство* (объект collection) представляет собой объект, содержащий несколько других объектов, как правило, одного и того же типа.
- ▶ Например, объект Workbooks (рабочие книги) содержит все открытые объекты Workbook (рабочая книга).
- ▶ Каждый элемент семейства нумеруется и может быть идентифицирован либо по номеру, либо по имени.
- ▶ *Worksheets (1), Worksheets ("Лист1")*

Объекты OLE и ActiveX

- ▶ В VBA используется механизм OLE (Object Linking and Embedding — связывание и внедрение объектов), который позволяет взаимодействовать с любыми программами, поддерживающими OLE. Примером элементов, которые можно интегрировать при помощи механизма OLE, являются вставляемые объекты OLEObject, создаваемые, например, при помощи программ WordArt, ClipArt и т. д.
- ▶ Все OLE-объекты рабочего листа образуют семейство oLEObjects. Вручную в рабочий лист OLE-объекты вставляются командой **Вставка, Объект** (Insert, Object) с выбором в появившемся диалоговом окне **Вставка объекта** (Object) из списка на вкладке

Application

Workbook (Workbook)

Charts (Charts)

DocumentProperties

VBProject

CustomView

CommandBar

AddIns (AddIns)

AutoCorrect

Assistant

Debug

Dialog

CommandBar

► Об
VE
10
об
на
ие
оп
об
пу

Полная ссылка на объект

- ▶ `Application.Workbooks("Архив").Worksheets("Лист 1").Range("A1")`
- ▶ `Worksheets("Лист1").Range("A1")`
- ▶ `Range("A1")`

Методы

- ▶ *Метод* как раз и представляет собой действие, выполняемое над объектом.
- ▶ Объект.Метод
- ▶ Application.Quit
- ▶ Worksheets("Лист!").Chartobjects.Delete

Свойства

- ▶ *Свойство* представляет собой атрибут объекта, определяющий его характеристики, такие как размер, цвет, положение на экране и состояние объекта, например, доступность или видимость. Чтобы изменить характеристики объекта, надо просто изменить значения его свойств.
- ▶ Объект.Свойство = ЗначениеСвойства
- ▶ Application.Caption = "Пример"
- ▶ Worksheets.Visible = False

События

- ▶ *Событие* представляет собой действие, распознаваемое объектом (например, щелчок мышью или нажатие клавиши), для которого можно запрограммировать отклик. События возникают в результате действий пользователя или программы, или же они могут быть вызваны системой.
- ▶ Суть программирования на VBA как раз и заключается в этих двух понятиях: *событие* и *отклик* на него. Действия, происходящие в системе, являются событиями, а отклики на них — процедурами. Этот специальный вид процедур, генерирующих отклик на события, называется *процедурами обработки событий*. В целом программирование на VBA состоит в создании кода программ, которые генерируют прямо или косвенно отклики на события.

VBA-редактор

- ▶ **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 (то есть — программный код) или форму.

VBA-редактор

- ▶ **File • Export File** — экспортирует данные из редактора во внешний файл. Например, этой командой можно сохранить редактируемую форму и передать ее другому разработчику.
- ▶ **File • Close and Return To Microsoft Word** — закрывает VBA-редактор и возвращается в Microsoft Word (аналогичная команда есть и для MS Excel). Вы можете просто переключаться между редактором и основным приложением в Панели задач Windows, не закрывая редактор.

VBA-редактор

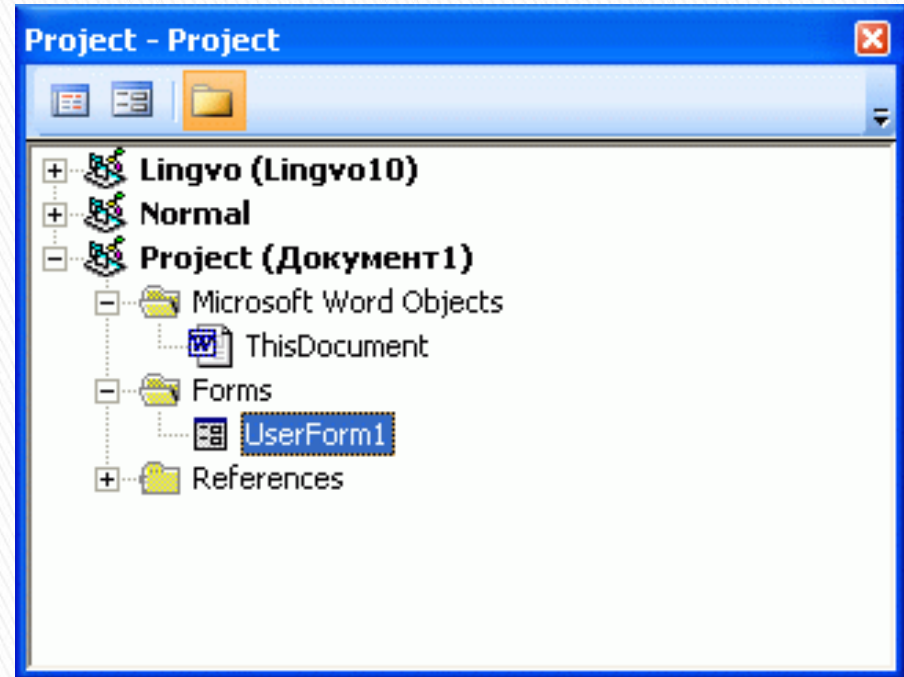
- ▶ **Edit** (Правка) — содержит команды для правки. Помимо стандартных команд отмены и возврата последнего действия (**Undo, Redo**), вырезания, копирования, вставки (**Cut, Copy, Insert**), поиска (**Find**), это меню содержит несколько особенных команд. В частности, это **List Properties/Methods** (Список Свойств/Методов) и другие.
- ▶ **View** (Вид) — содержит команды для отображения различных окон редактора. Названия команд соответствуют названиям окон.

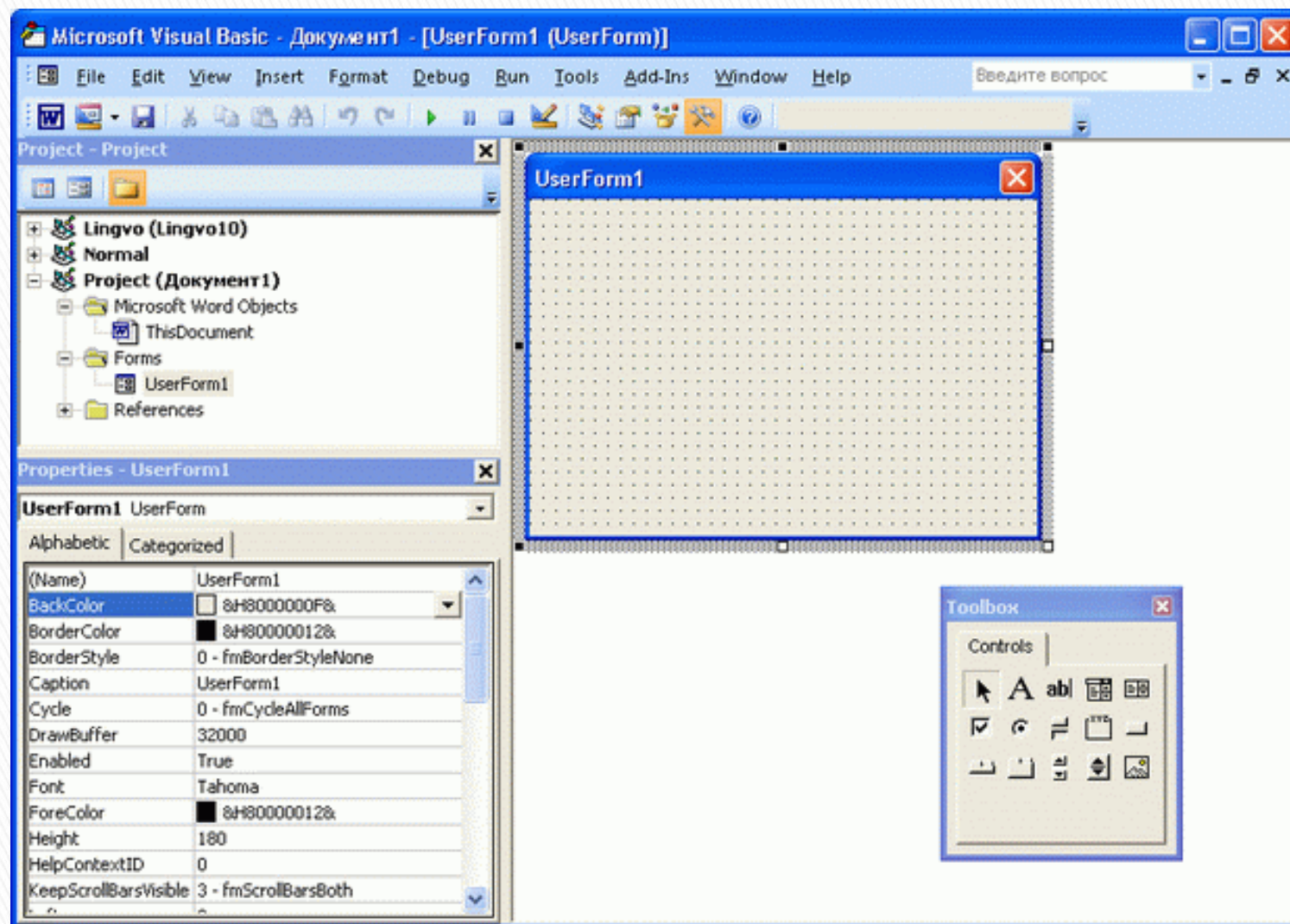
VBA-редактор

- ▶ **Insert** (Вставка) — служит для вставки в проект форм (**Insert • User Form**), модулей (**Insert • Module**), процедур (**Insert • Procedure**), файлов (**Insert • File**) и модулей класса (**Insert • Class Module**). Чаще всего вам придется вставлять в проект формы. 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. В частности, команда **Run • Sub/User Form** (Запуск • Процедура/Форма) запускает на выполнение активную процедуру или форму (рядом с этой командой стоит характерный зеленый треугольник). Команда **Run • Break** (Запуск • Приостановить) — приостанавливает выполнение программы, команда **Run • Reset** (Запуск • Перезапуск) — останавливает выполнение программы. Эти команды продублированы на панели инструментов редактора в виде кнопок с соответствующими пиктограммами.
- ▶ **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.

Окно Project

- ▶ служит для навигации по проекту
- ▶ Каждый открытый документ представляет собой отдельный проект.
- ▶ Проект имеет иерархическую структуру





Связь объекта Range и свойства Cells

- ▶ Так как ячейка является частным случаем диапазона, состоящим только из единственной ячейки, объект Range также позволяет работать с ней. Объект cells (ячейки) — это альтернативный способ работы с ячейкой.
- ▶ Range ("A2") --- cells (1,2).
- ▶ Range ("A2:C3") --- Range(Cells(1,2), Cells(3,3))

Свойства и методы объекта Range

Value	Возвращает значение из ячейки или в ячейки диапазона. В данном примере переменной x присваивается значение из ячейки C1 : <pre>x = Range ("C1") .Value</pre> В следующем примере в диапазон A1 : B2 введена 1 : <pre>Range ("A1:B2") .Value = 1</pre>
Name	Возвращает имя диапазона. В данном примере диапазону A1:B2 присваивается имя итоги: <pre>Range ("A1 :B2") .Name = "Итоги"</pre>

CurrentRegion

Возвращает число строк текущего диапазона.

Текущим является диапазон, ограниченный пустыми строками и столбцами и содержащий данный элемент. В следующем примере переменной *y* присваивается значение, равное числу строк в текущем диапазоне, содержащем ячейку A1 :

```
y = Range ( "A1" ). CurrentRegion. Rows .  
Count
```

WrapText

Позволяет переносить текст при вводе в диапазон. Допустимые значения True и False. В следующем примере в ячейку B2 вводится текст длинный текст и в этой ; ячейке устанавливается режим ввода текста с переносом:

```
With Range ("B2") .Value =  
"Длинный текст" .WrapText =  
True End With
```

Comment

Возвращает объект comment (примечание), который связан с левым верхним углом диапазона при отображении на экране. Объект comment является элементом семейства comments. Метод AddComment, примененный к диапазону, создает новое примечание. Среди методов объекта comment отметим только метод Text, который задает текст, выводимый в примечании. Синтаксис:
Text (Text, Start, Overwrite)

- Text — строка, выводимая в качестве примечания
- start — с какого символа вводится текст в уже существующее примечание. Если аргумент опущен, то из примечания удаляется весь ранее введенный текст
- overwrite — допустимые значения: True (вводимый текст записывается поверх уже существующего) и False (вводимый текст вставляется в уже существующий)
- Среди свойств объекта Comment отметим только свойство visible, устанавливающее отображение примечания при активизации диапазона, имеющего определенное примечание.

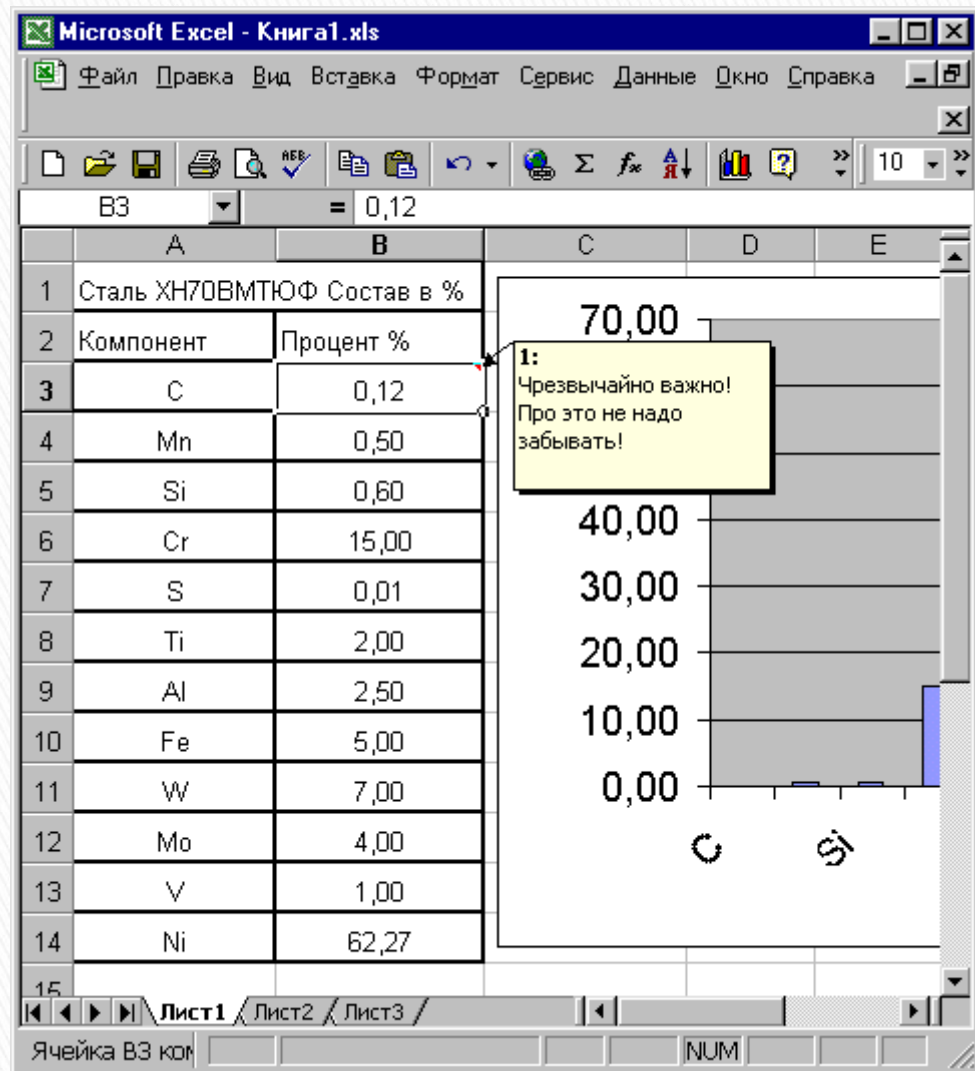
With Range ("B3")

.AddComment

.Text Text:= "Чрезвычайно важно!" &
Chr(10) & "Про это никак нельзя забыть ! "

.Visible = True

End With



Font	Возвращает объект Font (шрифт). Объект Font имеет следующие свойства: Name — строка, указывающая имя шрифта, например "Arial Cyr" FontStyle — СТИЛЬ, возможен Regular (обычный), Bold (ПОЛУЖИРНЫЙ), Italic(курсив), Bold italic (полужирный курсив) size - размер Superscript — допустимы два значения: True (текст используется как верхний индекс) и False (не используется как верхний индекс) Subscript — допустимы два значения: True (текст используется как нижний индекс) и False (не используется как нижний индекс) Underline-допустимыми являются значения: - xlNone (нет подчеркивания) - xlSingle (одинарное, по значению)
------	--

В следующем примере устанавливается для диапазона A1: B2 полужирный шрифт, красного цвета и с высотой символов 14:

```
With Range ("A1:B2").Font  
    .Size = 14  
    .FontStyle = Bold  
    .ColorIndex = 3  
End With
```

Microsoft Excel - для занятия 2								
Файл Правка Вид Вставка Формат Сервис Данные Окно Справка								
Arial Cyr 10 Ж К Ч [бул] [бул] [бул] [бул] % 000 [бул] [бул] 100% [бул]								
J21 fx								
	A	B	C	D	E	F	G	H
1	Табель успеваемости							
	Литера Русский Алгебра Геомет Инфор Средний							
2	№п/т	Фамилия, Имя	тура	язык	рия	матика	балл	
3	1	Иванов М.	4	5	4	3	3	3,8
4	2	Петров С.	5	4	5	5	5	4,8
5	3	Сидоров К.	3	3		3	3	3
6	4	...						
7	5							
8	6							
9	7							
10	8							
11	9							
12	10	...						
13								
14	Средний балл по предмету:		4	4	4,5	3,666667	3,666667	
15								
16								