



# **ОСНОВЫ ОФИСНОГО ПРОГРАММИРОВАНИЯ VBA**

Доцент кафедры «Бизнес-информатика и информационные технологии», к.пед.  
н. Валентина Игоревна Кузнецова



# ПЛАН:

1. Знакомство с редактором VBA
2. Элементы языка VBA
3. Процедуры и модули VBA
4. Комментарии в VBA
5. Тестирование и отладка программ



# 1. Знакомство с редактором VBA

Процесс программирования состоит из следующих этапов:

- 1. Постановка задачи** – определение требований к программному продукту.
- 2. Анализ условия** – определение исходных данных и результата.
- 3. Поиск путей решения** – определение методов решения: как из исходных данных можно получить результат, построение математической модели.
- 4. Построение алгоритма** – разработка структуры программы, выбор структур для хранения данных и т.д.
- 5. Реализация** – написание программы на выбранном языке программирования.
- 6. Отладка** - поиск и устранение ошибок, анализ эффективности.
- 7. Эксплуатация (сопровождение)** – выпуск новых версий, исправление выявленных ошибок.

# 1. Знакомство с редактором VBA

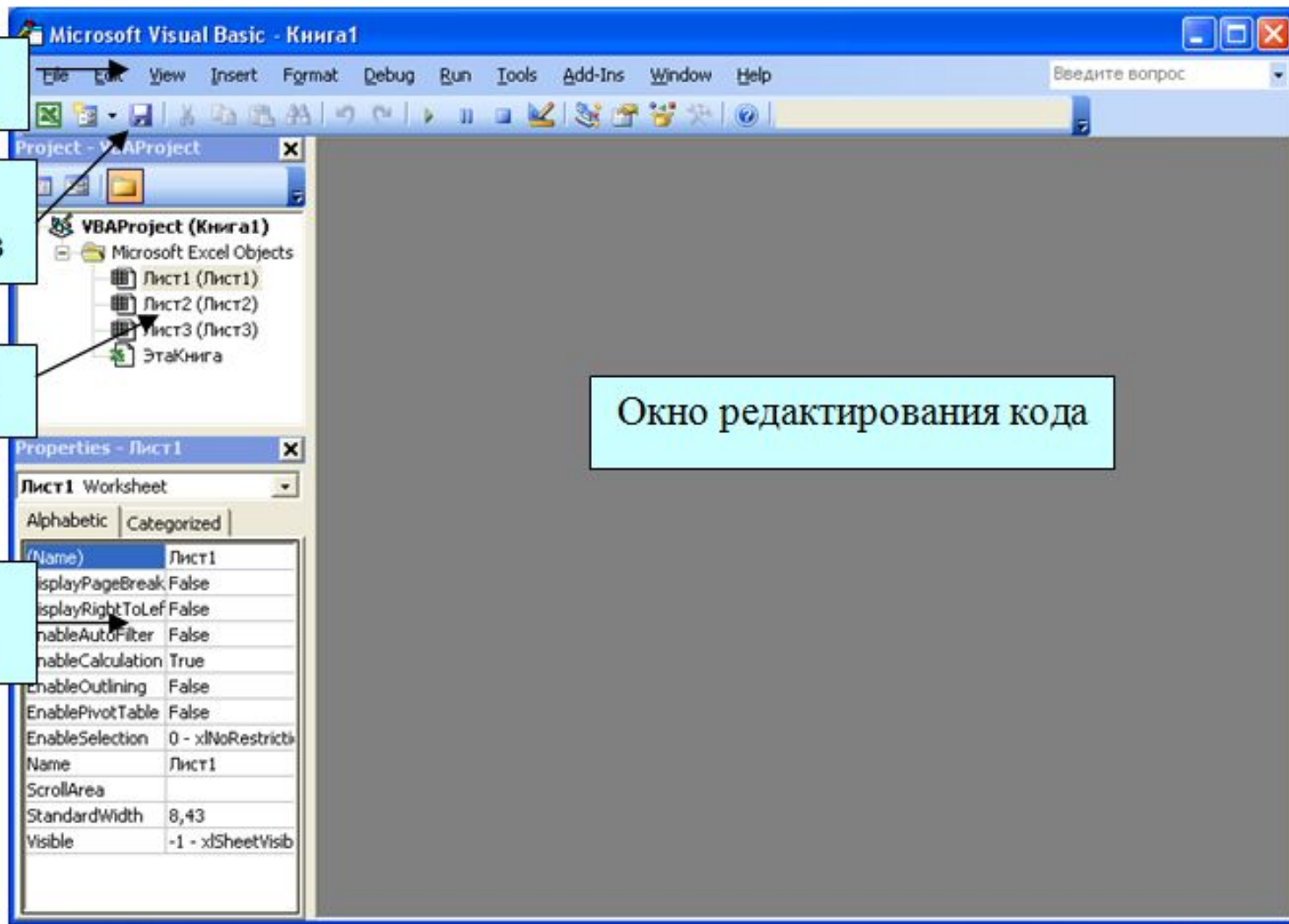
Меню

Панель инструментов

Окно проекта

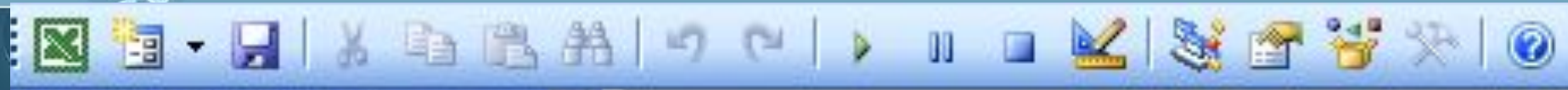
Окно свойств

Окно редактирования кода





# Стандартная панель инструментов редактора Visual Basic

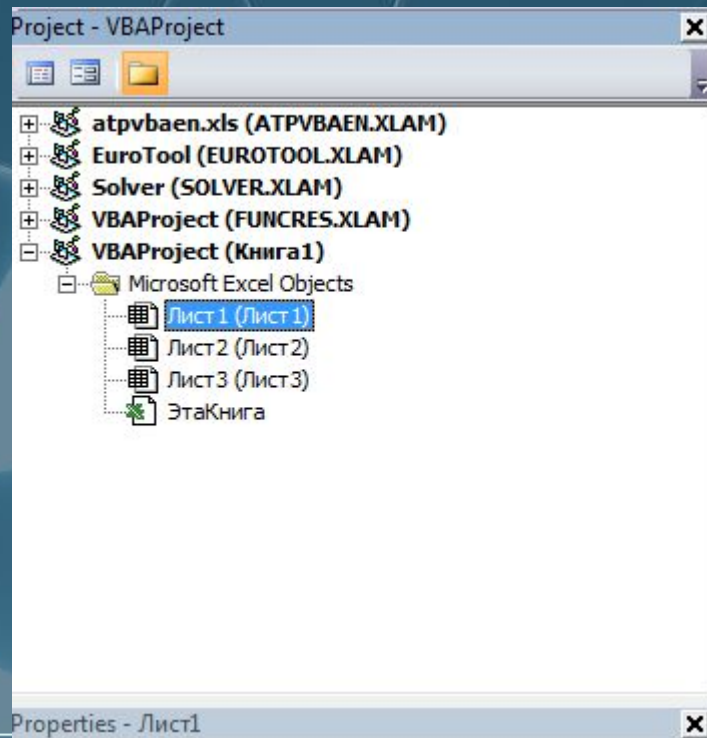


1. *Кнопки первого сегмента:* кнопки для возврата в Excel (Word), вставки элементов (модулей, процедур, экранных форм) в проект и сохранение рабочей книги.
2. *Кнопки второго сегмента:* Кнопки для вырезания, копирования, вставки и поиска фрагментов кода.
3. *Кнопки третьего сегмента:* Кнопки отмены действий и повторения отмененных действий.
4. *Кнопки четвертого сегмента:* Кнопки для выполнения, остановки и прекращения выполнения процедуры, а также смены режима отображения экранной формы.
5. *Кнопки пятого сегмента:* Кнопки, управляющие отображением окон проектов, свойств и просмотра объектов, а также панели инструментов.
6. *Последняя кнопка* – это кнопка вызова справочной системы.


# Окно проекта

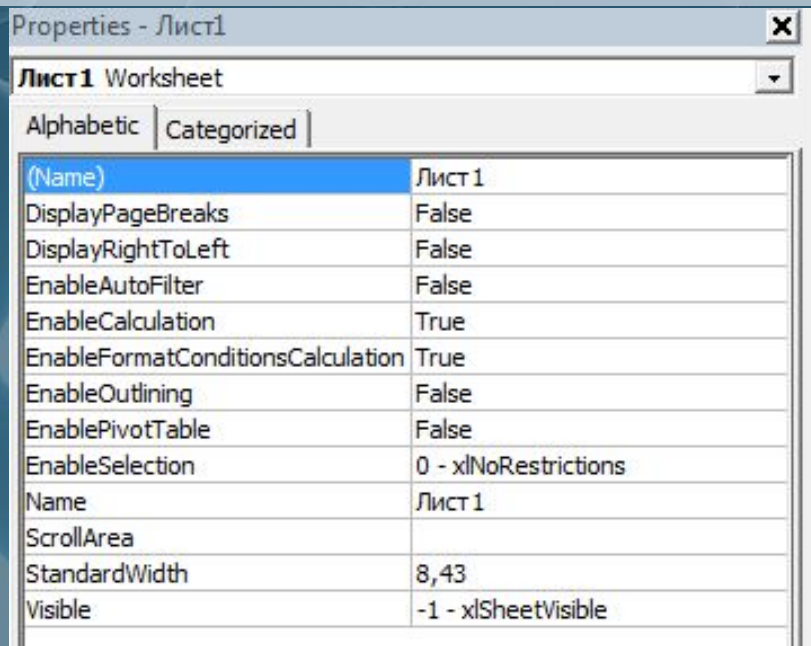
Активизируется выбором команды **Вид** → **Окно проекта** (View, Project window) или нажатием кнопки **Окно проекта**

В окне проекта представлена иерархическая структура файлов, форм и модулей текущего проекта.




# ОКНО СВОЙСТВ

В окне свойств перечисляются основные значения свойств выбранного объекта. Используя это окно, можно просматривать свойства и изменить их значения. Для просмотра выбранного объекта надо выполнить команду **Вид** → **Окно свойств** (Properties Windows) или нажатием кнопки 



(Name)	Лист1
DisplayPageBreaks	False
DisplayRightToLeft	False
EnableAutoFilter	False
EnableCalculation	True
EnableFormatConditionsCalculation	True
EnableOutlining	False
EnablePivotTable	False
EnableSelection	0 - xlNoRestrictions
Name	Лист1
ScrollArea	
StandardWidth	8,43
Visible	-1 - xlSheetVisible

# Окно для просмотра объектов (Object Browser)

Вызывается командой Вид → Просмотр объектов (View, Object Browser) или нажатием кнопки 

В этом окне можно просматривать все объекты проекта. Здесь вы найдете все свойства, методы и события, связанные с любым объектом.







## 2. Элементы языка VBA

1. **Объекты** – основные элементы языка VBA. Объектами VBA Excel являются рабочая книга, рабочий лист и его составляющие.
2. **Константы** – данные, не изменяющиеся в процессе решения задачи. Константы бывают двух видов: числовые и символьные.
  - **Числовые константы** – это целые либо вещественные числа.
  - **Символьные константы** – текст, заключенный в кавычки.

**Синтаксис:**

**Const** Имя\_константы **As** тип\_данных = значение

**Const** Годы\_учебы **As** Byte = 11



## 2. Элементы языка VBA

**3. Переменные** – данные, значения которых меняются в ходе выполнения программы. Для переменной задается имя и тип.

*Имя переменной* должно начинаться с буквы, и может содержать любую комбинацию букв, цифр и символов за исключением точек, пробелов и следующих символов: «!», «%», «&», «\$», «#», «@».

Длина имени не должна превышать 255 символов. Не следует использовать имена, совпадающие с ключевыми словами VBA и именами встроенных функций и процедур.


Переменные типа **Variant** могут хранить все, что в них поместят. Их тип изменяется в зависимости от последнего присвоения.

**Dim <имя переменной> As <тип>**



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

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



# Основные типы переменных, их размеры и диапазоны принимаемых значений

Тип	Размер (байт)	Диапазон значений
Byte (байт)	1	От 0 до 255
Boolean (логический)	2	True или False
Integer (целое число)	2	От -32768 до 32767
Long (длинное целое число)	4	От -2147483648 до 2147483647
Single (число с плавающей запятой обычной точности)	4	Для отрицательных значений от -3,4E38 до -1,4E-45. Для положительных – от 1,Е-45 до 3,4E38.
Double (число с плавающей запятой двойной точности)	8	Для отрицательных значений от -1,79E308 до -4,94E-342. Для положительных – от 4,94E-324 до 1,79 E308.
Currency (денежный)	8	От 922337203685477,5808 до 922337203685477,5807
Date (дата и время)	8	От 1 января 100 г. до 31 декабря 9999 г.
String (строка)	10+длина строки	От 0 до $2 \times 10^9$ .
Variant (вариант)		Зависит от содержимого переменной



# Примеры объявления переменных:

```
Dim intNumber As Integer
```

```
Dim strName As String
```

```
Dim dtTime As Date
```



# Видимость переменных:

В VBA переменные, описанные в модуле отдельно от всех его программ, делают их доступными в теле любой программы модуля.

Например, описание:

```
Dim a As Integer, b As Integer
```

```
Sub P1 (x As Single, y As String) As Single
```

```
..... a = 7.....
```

```
End Sub
```

```
Sub P2 ( x As Integer) As Boolean
```

```
..... b = a*3 + x.....
```

```
End Sub
```



# Видимость переменных:

**Для VBA действуют следующие правила видимости:**

1. Переменные, описанные в процедуре или функции, видны только в ней. Они объявляются с помощью ключевого слова `Dim` или `Static` и называются локальными.
2. 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.
3. Переменные, описанные в модуле вне описаний программ модуля с ключевым словом `Public`, видны во всех модулях (процедурах, функциях) проекта.



## 2. Элементы языка VBA

**Массив** – упорядоченная совокупность однотипных переменных. Массивы имеют имя и размерность. Имя массива подбирается с учетом тех же правил, что и имена переменных.

**Размерность** – это количество элементов (переменных), составляющих массив.

4. В VBA имеется большой набор **встроенных функций**, которые разделяются на категории.
5. Из констант, переменных и встроенных функций с помощью скобок и знаков арифметических операций («+», «-», «\*», «/», «^») можно составлять **выражения**. Частным случаем выражения может быть просто одиночный элемент, т.е. константа, переменная или обращение к встроенной функции.

Примеры выражений: Z; 45; (a+b)^2; Sin(y).





# ФУНКЦИИ В VBA

Функция	Выражение
	<i>Математические функции</i>
<b>Abs (x)</b>	Модуль (абсолютная величина числа)
<b>Cos (x)</b>	Косинус
<b>Exp (x)</b>	Экспонента, т.е. результат возведения основания натурального логарифма в указанную степень
<b>Log (x)</b>	Натуральный логарифм
<b>Sin (x)</b>	Синус
<b>Sqr (x)</b>	Квадратный корень
	<i>Логические функции</i>
<b>x1 AND x2</b>	True, если x1 и x2 равны True; False, если либо x1, либо x2 равно False
<b>x1 OR x2</b>	True, если хотя бы один из x1, x2 равен True; False, если и x1, и x2 равны False
	Функции проверки типов
<b>IsNumeric</b>	True, если переменная – число; False, если переменная – не число
<b>IsNull</b>	True, если переменная является пустым значением (Null); False, если переменная – не пустое значение
	<i>Функции преобразования форматов</i>
<b>CInt (x)</b>	x преобразуется в целое
<b>CSng (x)</b>	x преобразуется в число с плавающей точкой обычной точности
<b>CStr (x)</b>	x преобразуется в строку
	Функции времени и даты
<b>Date</b>	Текущая дата
<b>Now</b>	Текущие дата и время



## 2. Элементы языка VBA

*Application. Функция Рабочего Листа (Аргументы).*

### Примеры

**Application.Sum(Sheets(«Проверка»).Range(«A1:B20»))** – суммируются значения из ячеек диапазона A1:B20, расположенного на листе «Проверка»;

**Application.CountA(Sheets(«Ученики»).Range(«A:A»))** – подсчитывается количество непустых ячеек в столбце A на листе «Ученики».



## 2. Элементы языка VBA

К другим элементам VBA относятся: **встроенные диалоговые окна, операторы.**

**= оператор присвоения**

*Пример:*

```
Dim a, b As Single  
L1.Caption = CStr(a + b)
```



## 3. Процедуры и модули VBA

*Модуль* – это совокупность объявлений (описательная часть) и процедур, хранящихся как единое целое.

*Процедура* – любая совокупность кода VBA, рассматриваемая как единое целое.

Как правило, процедура состоит из операторов, выполняющих какую-либо задачу.

Каждая процедура идентифицируется своим *уникальным именем*.

Часто выполнение процедуры является реакцией на какое-либо событие. В этом случае говорят, что процедура обрабатывает событие.



# В VBA предусмотрены следующие типы процедур:

**1. Процедура типа Sub (подпрограмма)** — универсальная процедура для выполнения каких-либо действий:

```
Sub Farewell()
```

```
MsgBox "Goodbye"
```

```
End Sub
```

**2. Макрос в VBA** — это просто процедура типа Sub, не имеющая параметров. Только макросы можно вызывать по имени из редактора VBA или приложения Office. Все другие процедуры нужно вызывать либо из других процедур, либо специальными способами.

**3. Процедура типа Function (функция)** — тоже набор команд, которые должны быть выполнены. Принципиальное отличие только одно: функция возвращает вызвавшей ее программе какое-то значение, которое там будет использовано.



## 3. Процедуры и модули VBA

Пример процедуры:

```
Function Tomorrow()  
Tomorrow = DateAdd("d", 1, Date())  
End Function
```

и пример ее вызова:

```
Private Sub Test1()  
Dim dDate  
dDate = Tomorrow  
MsgBox dDate  
End Sub
```

**В тексте функции необходимо предусмотреть оператор, который присваивает ей какое-либо значение.**

```
Private Sub Test 1()  
MsgBox Tomorrow()  
End Sub
```



# Создание процедур

**Создание первой процедуры требует выполнения двух последовательных шагов:**

1. Вставить модуль в рабочую книгу.
2. В этом модуле создать процедуру.


Для любого создаваемого приложения надо создать свой модуль. Приложение может содержать несколько модулей, но это не обязательно.



# Выполнение процедур

После создания процедуры мы можем ее сразу выполнить.

Для этого существует несколько путей:

1. Выбрать команду меню **Run** → **Run Sub** → **UserForm** (**Выполнить** → **Выполнить подпрограмму** → **пользовательскую форму**) или щелкнуть на кнопке  — **Run Sub/UserForm** стандартной панели инструментов.
2. Нажать клавишу F5.
3. На листе Excel запустить макрос на выполнение.
4. Создать на листе Excel автофигуру и привязать к ней подготовленную процедуру.





## 4. КОММЕНТАРИИ В VBA

В языке VBA существуют два способа ввода комментариев:

- 1) применение апострофа ( ' ), который можно поставить в любом месте строки;
- 2) зарезервированное слово **Rem** вместо апострофа. При этом комментируется текст до конца строки.
- 3) Оператор **With / End with** избавляет программиста от большого количества повторений имени одного и того же объекта.

### Синтаксис:

**With** объект  
Оператор 1  
Оператор 2  
... ..  
Оператор N  
**End with**



# 5. ТЕСТИРОВАНИЕ И ОТЛАДКА ПРОГРАММ

Процесс исправления ошибок называется **отладкой**.

*Отладка программ* – это проверка и внесение исправлений в программу при ее разработке. Отладка позволяет идентифицировать ошибки, допущенные при программировании. Например, синтаксические ошибки в тексте программы, именах функций и переменных или логические ошибки.

*Обработка ошибок* – это задание реакции на ошибки, которые возникают во время выполнения программы. Причиной ошибок могут быть различные обстоятельства, находящиеся вне сферы влияния программиста. Например, отсутствие файлов, к которым происходит программное обращение, отказ аппаратных средств или неправильные действия пользователя.



# 5. ТЕСТИРОВАНИЕ И ОТЛАДКА ПРОГРАММ

**Тестирование** – это процесс выполнения процедуры и исследование всех аспектов ее работы.

**Цель тестирования** – проверить правильность результатов выполнения процедуры и ее реакцию на разнообразные действия пользователя.



**Все возможные ошибки можно разделить на три вида:**

1. **Ошибки компиляции.** Эти ошибки возникают, если VBA не может интерпретировать введенный текст.
2. **Ошибки выполнения.** Эти ошибки возникают при выполнении программы, т.е. после успешной компиляции. Причиной таких ошибок может быть отсутствие данных или неправильная информация (например, данные, введенные пользователем).
3. **Логические ошибки.** Эти ошибки труднее всего заметить и устранить. Логические ошибки не приводят к прекращению компиляции или выполнения. Однако они являются причиной того, что программа не выдает желаемых результатов.



# 5. ТЕСТИРОВАНИЕ И ОТЛАДКА ПРОГРАММ

**Компиляция** – это процесс преобразования программы, написанной на алгоритмическом языке, в язык машинных кодов.

Если в программе есть синтаксические ошибки, то процесс компиляции прекращается, строки с ошибкой закрашиваются желтым цветом и выдается соответствующее сообщение. Для продолжения выполнения программы необходимо исправить ошибку и нажать кнопку «Continue»  на стандартной панели редактора VBA, или прервать выполнение программы, нажав на кнопку «Reset» , исправить ошибку в программе, а затем заново ее запустить.

При обнаружении ошибки компилятор выдает сообщение с указанием номера ошибки. В этом случае полезно, воспользовавшись справочной системой редактора VBA, определить характер ошибки и исправить ее.