

Глава 5

Операции и выражения

В этой главе...

- ◆ Выражения в языке VBA
- ◆ Совместимость типов данных
- ◆ Оператор присваивания
- ◆ Арифметические операторы
- ◆ Логические операторы
- ◆ Операторы сравнения
- ◆ Строковые операторы
- ◆ Приоритеты выполнения операций
- ◆ Резюме

Выражения в языке VBA

Для манипуляции данными, константами и переменными в языке VBA используются выражения. *Выражение* — это отдельное значение или несколько значений, соединенных знаками операций. Иными словами, выражение — это то, что может быть вычислено, причем полученный результат должен иметь допустимый тип данных. Выражение может содержать любую комбинацию чисел, символов, констант, переменных, свойств объектов, встроенных функций и процедур типа `Function`, связанных между собой допустимыми знаками операции. В сложном выражении, состоящем из нескольких компонентов, каждый из них также может рассматриваться как некоторое промежуточное выражение со своим собственным значением. Отсюда следует, что в языке VBA выражение может представлять собой как единственную константу, так и сколь угодно сложную конструкцию, однако в любом случае оно в конечном итоге дает единственное значение.

При записи выражения отдельные его элементы соединяются знаками операций, обозначающих те арифметические, логические или иные действия, которые должны быть выполнены над соответствующими операндами. При выполнении программы любое выражение вычисляется до получения конечного результата, причем тип данных этого результата принимается как тип данных всего выражения. В зависимости от типа данных результата выражения могут быть числовыми, строковыми, логическими, объектными и с типом даты.



Помимо любых допустимых значений тех типов данных, речь о которых шла в предыдущей главе, результатом вычисления выражения может быть одно из специальных значений, обозначаемых ключевыми словами `Empty` и `Null`. Значение `Empty` в языке VBA представляет неинициализированную переменную типа `Variant`; также это результат вычисления любого выражения, в котором принимает участие подобная неинициализированная переменная. Значение `Null` указывает, что эта переменная не содержит корректных данных.

Выражения в программе строятся для выполнения различных вычислений, построения и сравнения значений, представления изменяемых величин. Они могут использоваться как для присвоения значения переменным, так и для передачи значений аргументов функциям и процедурам.

Совместимость типов данных

В предыдущей главе мы довольно подробно останавливались на описании тех типов данных, которые поддерживает VBA. Из сказанного очевидно, что не все типы данных взаимно совместимы, и поэтому нельзя использовать несовместимые типы данных в одном выражении. При обработке выражения, содержащего различные типы данных, компилятор VBA сначала пробует самостоятельно устранить любое различие типов. Для этого он пытается преобразовать значения в выражении в совместимые типы данных, и только если не удастся автоматически устранить различия в типах данных, компилятор VBA выдает ошибку *несовпадения типов*. Аналогичную ошибку — несовпадение типов — можно получить, если присваивать результат вычисления какого-нибудь выражения переменной, тип которой несовместим с типом этого результата. То же самое можно сказать о функции или о процедуре — ошибка несовпадения типов возникнет в том случае, если использовать результат вычисления не подходящего по типу выражения в качестве их аргумента. В языке VBA есть целый ряд встроенных функций, предназначенных для преобразования одних типов данных в другие (эти функции будут подробно рассмотрены в главе 9). Важно знать, что всегда, когда только это возможно, компилятор VBA автоматически конвертирует данные к подходящему типу. Если переменной целого типа присвоить значение с дробной частью, то компилятор VBA автоматически округлит это значение до ближайшего целого числа. В языке VBA многие типы данных совместимы, что позволяет использовать их в одном выражении — например, смешивать при вычислениях различные числовые типы данных. Компилятор VBA автоматически выполняет необходимые преобразования при использовании в выражении различных числовых типов.



Автоматическое преобразование типов компилятор VBA выполняет гораздо быстрее, когда выражение содержит переменные типа *Variant*, поскольку переменная такого типа может хранить данные любого из допустимых в VBA типов, включая числовые значения, строки, даты и объекты.

Оператор присваивания

Оператор — это наименьшая единица программного кода. Каждый (выполнимый) оператор содержит инструкцию, указывающую, какие именно манипуляции данными или изменения в ходе программы должны быть осуществлены при его выполнении. *Оператор присваивания* представляет собой связанную знаком равенства конструкцию, в которой значение выражения, находящегося справа, присваивается переменной, имя которой указано слева. Например, после выполнения следующей пары операторов переменной *z* будет присвоено значение *a + b*:

```
x = a
z = x + b
```

Оператор присваивания допускается использовать и для манипуляции объектами. Однако, если переменной присваивается значение ссылки на объект, то в операторе присваивания следует указать ключевое слово *Set*, например:

```
Set objA = objB
```

Здесь предполагается, что `objA` и `objB` были объявлены как переменные объектного типа и переменная `objB` в данный момент содержит ссылку на некоторый объект. В результате выполнения приведенного выше оператора переменной `objA` присваивается значение ссылки на тот же объект.

Следует заметить, что если переменной с определенным типом данных присваивается результат вычисления некоторого выражения, тип данных этого результата должен совпадать с типом переменной, получающей новое значение. В противном случае компилятор VBA автоматически предпримет попытку преобразовать тип данных результата вычисления выражения в тип, совместимый с типом указанной переменной.

Арифметические операторы

В языке VBA арифметические операторы позволяют выполнить любые арифметические операции в полном соответствии с правилами арифметики. Ниже, в табл. 5.1, приведены знаки операций, которые в языке VBA могут использоваться при написании арифметических выражений.

Таблица 5.1. Знаки арифметических операций, используемые в языке VBA

Знак операции	Описание
+	Сложение
-	Вычитание
*	Умножение
/	Деление
\	Деление без остатка
^	Возведение в степень
mod	Остаток от деления по модулю

Нет смысла подробно обсуждать здесь использование таких тривиальных знаков операций, как “+”, “-”, “*” или “/”, но несколько полезных замечаний все же следует сделать:

- знак операции сложения можно использовать при построении арифметических выражений с данными типа `Date`;
- если в операции сложения используются данные типа `Integer` и `Long`, результат вычисления выражения будет иметь тип данных `Long`;
- если в операции вычитания один из операндов имеет тип данных `Date`, то результат вычисления выражения будет иметь тип данных `Date`;
- если оба операнда в операции вычитания имеют тип данных `Date`, то результат вычисления выражения будет иметь тип данных `Double`;
- если в операции умножения используются данные различных типов, то результат вычисления выражения будет иметь тип данных, соответствующий типу данных того из сомножителей, который имеет наибольшую точность;
- при умножении значения переменных с типом данных `Variant`, содержащих в данный момент значения типа `Date`, последние будут преобразованы в числовые значения;

- если при выполнении деления делитель окажется равным нулю, будет выдано соответствующее сообщение об ошибке;
- для результата вычисления операции деления чисел с плавающей точкой обычно используется тип данных `Double`, а для чисел типа `Integer` — `Single`.

Результатом операции деления без остатка будет целое число без округления, при этом дробная часть результата просто отбрасывается. Например, после вычисления выражения $17 \setminus 5$ получим 3. Тип данных результата операции деления без остатка — либо `Integer`, либо `Long`, при этом язык VBA использует наименьший тип данных, позволяющий представить значение вычисленного результата.



Если в операции деления любого типа один из операндов имеет значение `Null` (отсутствие конкретного значения), то результатом операции также будет значение `Null`. И вообще, если в выражении присутствует переменная, имеющая значение `Null`, то результатом вычисления всего выражения также будет значение `Null`.

Операция возведения в степень (^) предполагает возведение значения первого операнда выражения в степень, равную значению второго операнда (например, выражение $5 \wedge 4$ означает “пять в четвертой степени” и после вычисления дает значение 625). При этом оба операнда должны быть либо числовыми выражениями, либо строками, которые компилятор VBA может преобразовать в числовое значение. Отрицательное число может быть возведено только в целочисленную степень. Результат операции возведения в степень всегда имеет тип `Double`, за исключением случая, когда один из операндов равен `Null` (здесь результатом возведения в степень также будет `Null`).

При выполнении операции деления по модулю осуществляется деление первого операнда на второй и в качестве результата возвращается целочисленный остаток этого деления. Если при выполнении операции `Mod` используются числа с плавающей точкой, они округляются до ближайших целых до выполнения операции. Тип данных результата операции деления по модулю может быть `Integer` или `Long`, при этом в каждом случае используется наименьший тип данных, который соответствует полученному результату.

Логические операторы

Логические операторы в языке VBA используются для манипулирования логическими значениями — `True` (в числовом представлении это 1) и `False` (в числовом представлении это 0). При выполнении этих операторов логическим переменным присваивается результат вычисления логических выражений, указанных в правой части операторов.

Обычно результатом логических операций является значение типа `Boolean`. В табл. 5.2 кратко описаны логические операторы, поддерживаемые в языке VBA.

Таблица 5.2. Допустимые логические операторы языка VBA

Оператор	Описание
<code>And</code>	Конъюнкция (Логическое И)
<code>Or</code>	Дизъюнкция (Логическое ИЛИ)
<code>Not</code>	Отрицание (Логическое НЕТ)
<code>Xor</code>	Исключение (Логическое исключающее ИЛИ)
<code>Eqv</code>	Логическая эквивалентность
<code>Imp</code>	Логическая импликация

В табл. 5.2 логический оператор And обозначает операцию *конъюнкции*, результатом выполнения которой будет значение True тогда и только тогда, когда оба операнда имеют значение True. В остальных случаях результатом выполнения этой операции будет значение False.

```
(3 > 2) And (4 > 3)
(3 > 6) And (4 > 3)
```

Результатом вычисления первого выражения будет значение True, так как оба операнда имеют значение True. Результатом вычисления второго выражения будет значение False, поскольку первый операнд имеет значение False.

Логический оператор Or обозначает операцию *дизъюнкции*, результатом выполнения которой будет значение True, если хотя бы один из операндов имеет значение True. Результат будет иметь значение False тогда и только тогда, когда оба операнда имеют значение False.

```
(3 > 5) Or (4 > 2)
(2 > 3) Or (2 > 4)
```

Результатом вычисления первого выражения будет значение True, так как один из операндов (второй) имеет значение True. Результат вычисления второго выражения будет иметь значение False, так как оба операнда имеют значение False.

Логический оператор Not обозначает операцию *логического отрицания* и имеет только один операнд. Результатом выполнения этой операции будет значение True, если исходно операнд имеет значение False, и наоборот, результатом будет значение False, если операнд имеет значение True.

```
D1 = ((2 > 5) Or (7 > 3)) And Not ((3 > 1) Or (8 < 10))
```

После вычисления приведенного выше выражения переменной D1 будет присвоено значение False, поскольку, хотя оба помещенные в скобки логические выражения Or имеют значения True, оператор Not превращает второе значение True в False, а значит и результатом всего оператора And будет значение False.

Логический оператор Xor обозначает логическую операцию “исключающее ИЛИ”, результатом выполнения которой будет значение True, если операнды имеют различные значения, и значение False, если оба операнда имеют одинаковые значения (оба — True или оба — False).

```
D2 = ((2 > 5) Xor (7 > 3)) And ((3 > 1) Xor (8 < 10))
```

В приведенном выше примере переменной D2 будет присвоено значение False, поскольку, хотя первый помещенный в скобки оператор Xor дает значение True, второй оператор Xor дает значение False, а значит результатом всего оператора And будет значение False.

Логический оператор Eqv обозначает логическую операцию эквивалентности двух выражений, результатом выполнения которой будет значение True, если оба операнда имеют одинаковые значения (оба — True или оба — False), и значение False, если только один из операндов имеет значение False.

Логический оператор Imp обозначает логическую операцию импликации (следования) двух выражений, результатом выполнения которой будет значение False, если первый операнд имеет значение True, а второй операнд имеет значение False, и значение True во всех остальных случаях.

Следует отметить, что логические операторы чаще всего используются в операторах сравнения.

Операторы сравнения

Операторы сравнения используются в языке VBA для сравнения числовых и строковых значений переменных, констант и результатов вычисления выражений. В результате выполнения операции сравнения всегда получается значение типа `Boolean`: либо `True` (Истина), либо `False` (Ложь). В табл. 5.3 описаны операторы сравнения, поддерживаемые в языке VBA.

Таблица 5.3. Операторы сравнения в языке VBA

Оператор	Описание
=	Равно
>	Больше
<	Меньше
>=	Больше или равно
<=	Меньше или равно
<>	Не равно
Is	Сравнение двух операндов, содержащих ссылки на объекты
Like	Сравнение двух строковых выражений

Если оба операнда в выражении имеют один и тот же тип данных, в языке VBA выполняется простое сравнение для этого типа данных. Если один или оба операнда являются переменными типа `Variant`, то компилятор VBA преобразует тип `Variant` в какой-либо совместимый тип данных, иначе на экран будет выведено сообщение об ошибке времени выполнения.

Знаки операции сравнения можно использовать как для чисел, так и для строковых значений. При сравнении строковых значений компилятор VBA последовательно сравнивает отдельные символы слева направо, определяя старшинство в соответствии с алфавитным порядком для букв и в соответствии с двоичным значением кода ASCII для любых прочих символов. В языке VBA при сравнении строк одна строка будет признана равной другой, если обе строки имеют одну и ту же длину и содержат одинаковые символы, расположенные в одинаковом порядке. Результатом такого сравнения будет значение `True`. Если же хоть одно из перечисленных условий не будет выполнено, результатом сравнения строк будет значение `False`.

Язык VBA предоставляет два способа сравнения символов различных регистров.

- Сравнение строк в двоичном режиме (метод сравнения по умолчанию). При сравнении строковой информации по этому методу для анализа используется действительный двоичный эквивалент кода каждого символа. Буквы верхнего регистра имеют меньший двоичный код, чем буквы нижнего регистра. Например, для буквы "A" код символа верхнего регистра (A) меньше кода символа нижнего регистра (a), т.е. `"a" > "A"` `True`.
- Сравнение строк в текстовом режиме. Согласно этому методу сравнение строк будет выполняться по алфавиту, но без учета регистра букв. При сравнении в текстовом режиме строка "ASD" равна строке "asd". Для того чтобы перейти в текстовый режим сравнения, необходимо в начало модуля перед объявлениями переменных или процедур поместить оператор `Option Compare Text`, устанавливающий данный режим сравнения для этого модуля.

Для сравнения выражений, содержащих ссылки на объекты, в языке VBA используется оператор сравнения `Is`. Результатом выполнения оператора `Is` является значение `True`, если оба выражения типа `Object` ссылаются на один и тот же объект, и значение `False` в противном случае.

```
obj1 Is obj2
```

Оператор `Like` позволяет сравнивать строку с заданным шаблоном (некоторым специальным образцом) и может использоваться только для сравнения двух строковых выражений. С помощью оператора `Like` осуществляется поиск в текстовой информации некоторого слова или фразы, отвечающих условиям, которые описаны заданным шаблоном. При этом оператор `Like` возвращает значение `True`, если в тексте присутствует слово или фраза, соответствующие заданному шаблону, или значение `False` в противном случае. Синтаксис оператора следующий:

```
Strzap1 Like Strzap2
```

Здесь `Strzap1` — любое строковое выражение (сравниваемая строка, в которой выполняется поиск), а `Strzap2` — комбинация специальных символов, определяющая шаблон для сравнения. Перечень допустимых символов шаблона с их кратким описанием приведен в табл. 5.4.

Таблица 5.4. Специальные символы шаблона для оператора `Like`

<i>Символ шаблона</i>	<i>Чему соответствует</i>
?	Одному любому символу
*	Любому количеству символов от 0 и больше
#	Любой цифре от 0 до 9
(List)	Списку (List) определенных символов. Для обозначения диапазона используется знак дефиса (–)
(!List)	Любому символу, не входящему в список (List)

Результат сравнения в операторе `Like` прописных (верхний регистр) и строчных (нижний регистр) букв зависит от установленного режима `Option Compare...` Если задано двоичное сравнение строк, то оператор `Like` различает буквы верхнего и нижнего регистров. Если установлен режим текстового сравнения, то оператор `Like` не реагирует на значение регистров.

Строковые операторы

В языке VBA для строковых значений поддерживается единственный оператор *конкатенации*, выполняющий операцию слияния двух строк в одну. В языке VBA для обозначения операции конкатенации используется знак “&”. Результат операции конкатенации строк всегда имеет тип данных `String`. Например, рассмотрим результат выполнения следующего оператора:

```
MyFIOString = "Иванов" & " " & "Иван" & " " & "Иванович"
```

Здесь в переменную `MyFIOString` будет помещена строка "Иванов Иван Иванович". (Обратите внимание, что пробелы между словами в эту строку добавлены как отдельные операнды-строки.) Все возможные манипуляции со строками подробно рассматриваются в главе 9.



Если в операции конкатенации используются числовые операнды, то перед ее выполнением компилятор VBA автоматически преобразует числовые выражения в операндах в строки символов, соответствующие их текущему значению.

Приоритеты выполнения операций

Если выражение содержит больше одного оператора, то вычисление значения таких многоэлементных выражений регулируется правилами о приоритете выполнения операций, принятыми в языке VBA. Приоритеты основных операторов представлены в табл. 5.5. Операторы, имеющие одинаковый приоритет, выполняются в выражении последовательно, слева направо. Изменить стандартный порядок выполнения операций в выражениях языка VBA можно с помощью скобок.

Таблица 5.5. Приоритеты операторов VBA

Оператор	Название	Приоритет
^	Возведение в степень	1
*, /	Умножение, деление	2
\	Деление без остатка	3
Mod	Остаток	4
+, -	Сложение, вычитание	5
&	Слияние строк	6

Операторы с меньшим значением приоритета согласно табл. 5.5 выполняются первыми. Рассмотрим следующий пример.

$7 + 5 * 3$

В данном примере первой будет выполняться операция умножения, так как она имеет более высокий приоритет, а затем операция сложения, т.е. результат вычисления всего выражения будет равен 22. Следует также отметить, что если в сложных выражениях встречаются части выражений, заключенные в скобках, они всегда вычисляются в первую очередь независимо от приоритета операторов.

$(7 + 5) * 3$

В этом примере первой будет выполнена операция сложения, заключенная в скобки, а лишь затем операция умножения, т.е. результат вычисления всего выражения будет равен 36. В выражении можно использовать сколько угодно уровней скобок, но все они должны быть парными. Если скобки в выражении вложены одна в другую, то вычисления начинаются с самого внутреннего уровня скобок.

Резюме

В этой главе мы продолжили изучение основных концепций языка VBA и ознакомились с выражениями и различными типами операций, поддерживаемых в нем. Выражения — это последовательность значений, соединенных знаками операций. При выполнении программы любое выражение вычисляется и приводится к единственному значению того или иного типа данных. При построении выражений значения соединяются знаками операций, набор которых задается всей совокупностью операторов, поддерживаемых в языке VBA, — арифметических, логических, строковых и операторов сравнения.

Выражения входят в состав любых выполнимых операторов языка VBA, простейшими из которых являются операторы присваивания. С их помощью результат вычисления выражения помещается в указанную переменную. При этом очень важное значение имеет совместимость типов данных переменной и вычисленного выражения.

Тестовые задания

Ниже приведено несколько тестовых заданий, предназначенных для закрепления изложенного в этой главе материала. В каждом задании вам предлагается вопрос и несколько вариантов ответа на него, один (или несколько) из которых является правильным, а остальные — нет. Укажите правильный ответ.

1. В языке VBA выражением является:
 - а) последовательность символов, разделяемая пробелами и знаками операций;
 - б) отдельное значение или несколько значений, соединенных знаками операций;
 - в) последовательность операторов, ограниченная операторами `Sub — End Sub`;
 - г) несколько переменных и/или констант, разделяемых знаками арифметических и логических операций.
2. Язык VBA поддерживает автоматическое преобразование следующих типов данных:
 - а) `Integer` в `Double`;
 - б) `String` в `Object`;
 - в) `Boolean` в `String`;
 - г) `Byte` в `Date`.
3. Язык VBA поддерживает следующие арифметические операторы:
 - а) умножения;
 - б) извлечения квадратного корня;
 - в) взятие модуля числа;
 - г) сложения;
 - д) преобразования двоичного значения в десятичное.
4. Язык VBA поддерживает следующие строковые операторы:
 - а) сложения;
 - б) преобразование в двоичное значение;
 - в) конкатенации;
 - г) суммирования,
5. В сложных выражениях операции выполняются по следующим правилам:
 - а) согласно скобок, а если их нет — слева направо;
 - б) сначала числовые, затем строковые и последними — логические;
 - в) от простых к сложным;
 - г) согласно приоритету операций, слева направо, с учетом скобок.

