

Лекция 4

На предыдущей лекции мы рассмотрели вопросы создания функции пользователя $F(x,a,b)$ средствами *VBA* и использования этой функции для вычисления таблицы значений функции для аргумента x , изменяющегося в диапазоне от -10 до 10 с шагом 1 , и для трёх значений аргумента a .

Продолжим наше знакомство с возможностями организации вычислений в среде программы *Excel* с использованием элементов *VBA*.

Сделаем первое замечание, которое направлено на уточнение наших знаний о программе *Excel* и языке программирования *VBA*.

При создании функции $F(x,a,b)$ команда "Макрос ►", которую мы выполнили из окна "*Microsoft Excel*", была нам нужна только для того, чтобы перейти в среду *Редактора Visual Basic*. (Можно получить этот же результат, нажав клавиши $\langle Alt \rangle + \langle F11 \rangle$)

Перейдя в среду *VBA*, мы получили возможность создать стандартный модуль – объект, в который мы затем записали текст нашей функции $F(x,a,b)$. В дальнейшем, если мы захотим записать какие-то другие пользовательские функции, нет необходимости каждый раз создавать новый модуль. В уже существующий модуль можно записывать и другие, используемые в реализации Вашего проекта функции.

Создание новых модулей целесообразно в тех случаях, когда Вам необходимо выделить некоторую группу функций, а также при создании приложений, связанных с объектами *User Form* (форма пользователя), автономных проектов *VBA* и т.п.

Рассмотрим теперь пример решения задачи табулирования разветвляющейся функции.

1. Табулирование разветвляющихся функций

Разветвляющимися называются функции, вид которых зависит от значения аргумента. К числу таких функций можно отнести функции, используемые в физике для описания различных процессов.

Например, функция, определяющая силу сухого трения.

Известно, что сила сухого трения между поверхностями соприкасающихся твёрдых тел при отсутствии между ними жидкой или газообразной прослойки подразделяется на трение покоя - трение при отсутствии относительного перемещения соприкасающихся тел и трение скольжения – трение при относительном движении соприкасающихся тел.

Сила трения покоя $F_{тр0}$ определяется силой, препятствующей движению одного тела по поверхности другого до момента начала движения. При изменении внешней силы $F_{внешн}$ от нуля до некоторого предельного значения $F_{тр0}^{макс}$ тело остаётся неподвижным. Движение одного тела относительно другого возникает при выполнении условия

$$F_{внешн} > F_{тр0}^{макс} .$$

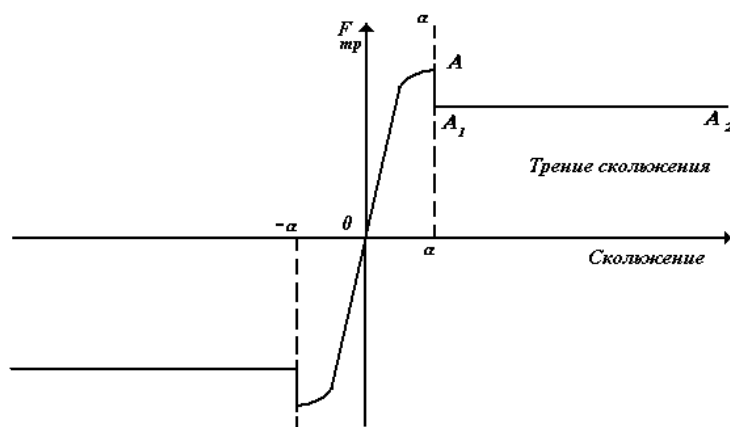


Рис. 4. 1. Значение силы трения в зависимости от относительного смещения трущихся тел при сдвиге, переходящем в скольжение

Если составляющая силы, приложенной к телу, лежащая в плоскости соприкосновения двух тел, недостаточна для того, чтобы вызвать скольжение данного тела относительно другого, то возникающая сила трения называется **неполной силой трения** (участок OA на рис. 4.1). Она вызвана малыми частично обратимыми перемещениями в зоне контакта. Величина таких смещений пропорциональна приложенной силе и изменяется с увеличением силы от 0 до некоторого максимального значения (точка A на рис. 4.1). Эта сила называется **силой трения покоя**, а перемещения – предварительными смещениями. После того как приложенная сила превысит критическое значение, предварительное смещение переходит в скольжение. При этом сила, поддерживающая скольжение, меньше, чем максимальная сила, вызвавшая переход в режим скольжения. Эта сила не зависит от перемещения и называется **силой трения движения** (участок от точки A_1 до A_2).

Приближённое описание такой функции включает три участка и может быть представлено в виде:

$$F(X) = \begin{cases} -F_a & \text{при } X \leq -a, \\ kX & \text{при } -a < X \leq a, \\ F_a & \text{при } X > a. \end{cases}$$

В общем случае разветвляющаяся функция, имеющая n ветвей, может быть представлена как функция вида

$$F(X) = \begin{cases} \varphi_0(X) \text{ при } X < x_0, \\ \varphi_1(X) \text{ при } x_0 \leq X < x_1, \\ \dots \\ \varphi_{n-2}(X) \text{ при } x_{n-2} \leq X < x_{n-1}, \\ \varphi_{n-1}(X) \text{ при } X \geq x_{n-1}. \end{cases} \quad (4.2)$$

Для реализации условных переходов во всех языках программирования используются условные операторы. В число функций программы *Microsoft Excel* включена функция условного перехода **ЕСЛИ**. Обращение к этой функции, как и ко всем остальным функциям программы *Excel* может быть записано в любую ячейку таблицы на листе книги. Рассмотрим формат записи этого обращения к функции.

Начнём рассмотрение с записи обращения к этой функции для реализации разветвляющейся функции с двумя ветвями.

$$F(X, \alpha) = \begin{cases} C * X & \text{при } X \leq \alpha, \\ D * X & \text{при } X > \alpha. \end{cases} \quad (4.3)$$

Алгоритм, реализующий функцию $F(X, \alpha)$, может быть изображён в виде фрагмента блок-схемы, в котором используется графический символ "Решение" (рис. 4.2 блок 1).

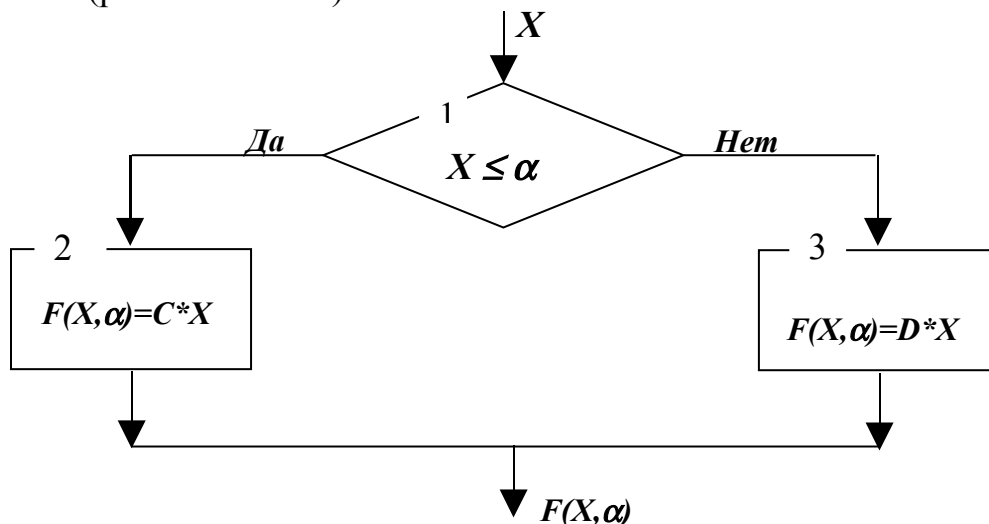


Рис. 4.2. Блок-схема алгоритма, реализующего функцию (4.3)

На листе книги *Excel* заполним ячейки, определяющие постановку задачи.

В ячейку *A1* вводим текст "**Параметры функции:**". Выделяем ячейки *A1*, *B1* и *C1*. Объединяем их и помещаем текст в центре.

В ячейку **D1** вводим текст "*альфа*=", в **E1** вводим значение параметра $\alpha = 0$.

Во второй строке определяем значения параметров функции **C** и **D**.

В ячейку **A3** вводим текст "*Исследуемый интервал*", а в ячейках четвёртой строки определяем значения X_0 , X_k и dX .

В ячейках **A5** и **B5** оформляем шапку таблицы, вводя в них тексты **X** и **F(X,альфа)** соответственно.

В ячейку **A6** вводим формулу $=\$B\4 , определяющую абсолютную ссылку на ячейку **B4**, в которой задается начальное значение аргумента **X**.

В ячейку **A7** вводим формулу $=A6+\$F\4 , по которой вычисляется следующее значение аргумента **X**. Протягиваем эту формулу по столбцу **A** до получения значения $X=X_k$ в ячейке **A26**.

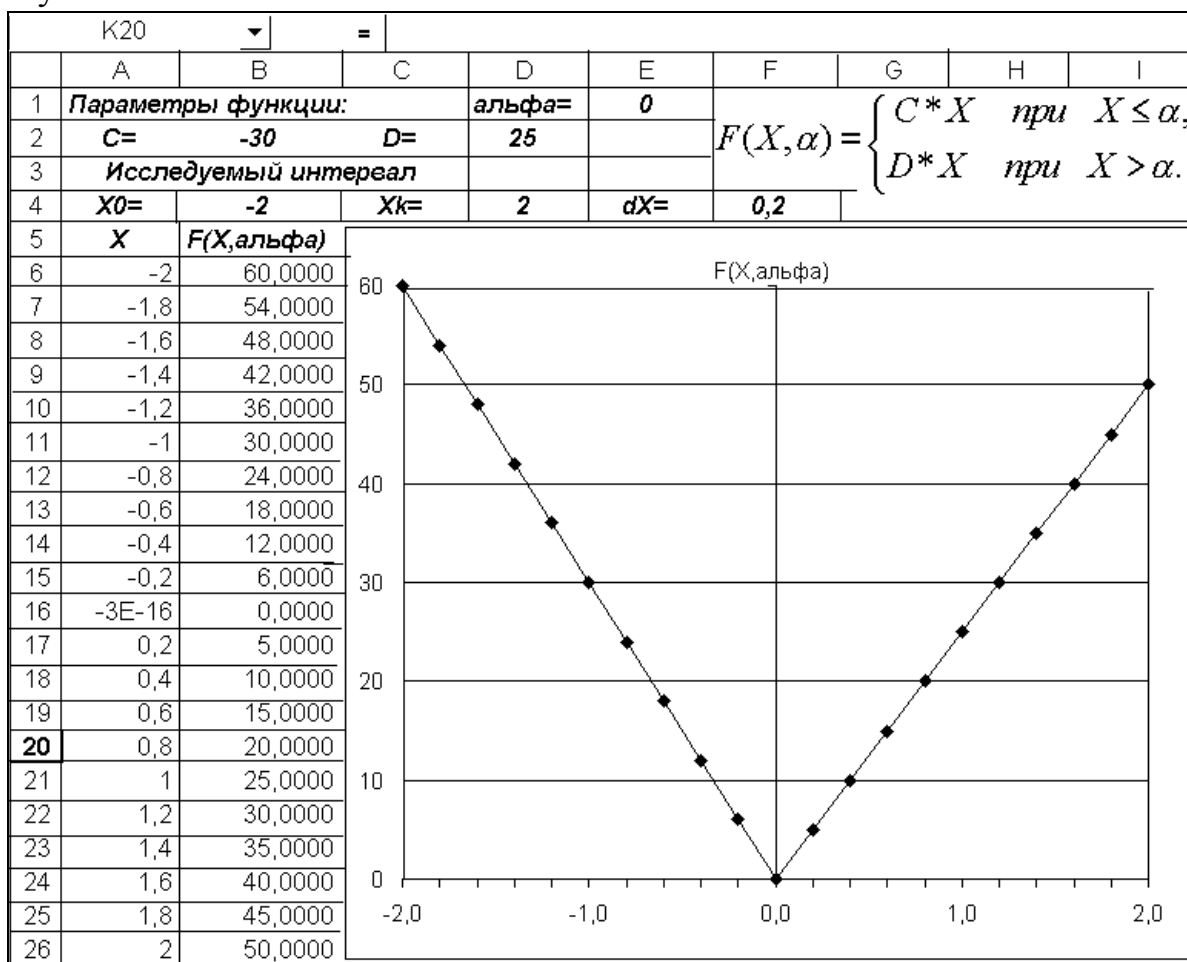


Рис. 4.3. Результирующий документ, полученный при табулировании функции (4.3)

В ячейку **B6** введём формулу для вычисления значения **F(X,альфа)**.

1. Вводим символ "=" и, не найдя в списке функций функцию **ЕСЛИ**, выбираем строку "*Другие функции*". Открывается окно "*Мастер функций – шаг 1 из 2*". В поле "*Категория:*" выбираем категорию

функций "*Логические*". В списке логических функций, находящемся в поле "*Функция:*" выбираем функцию *ЕСЛИ*.

Подсказка в окне первого шага мастера функций, соответствующая функции *ЕСЛИ*, определяет синтаксис обращения к функции:

ЕСЛИ(логическое_выражение;значение_если_истина;значение_если_ложь)

Ниже в поле окна приводится пояснение: "*Возвращает одно значение, если условие истинно, и другое, если оно ложно*".

После нажатия на кнопку **ОК**, открывается окно второго шага мастера функций (рис. 4.4):

ЕСЛИ

Логическое_выражение = логическое

Значение_если_истина = любое

Значение_если_ложь = любое

=

Возвращает одно значение, если указанное условие истинно, и другое, если оно ложно.

Логическое_выражение любое значение или выражение, которое при вычислении дает значение ИСТИНА или ЛОЖЬ.

Значение:

Рис. 4.4. Окно второго шага мастера функция для функции *ЕСЛИ*

В поле "*Логическое выражение*" определим условие, определяющее переход от одной ветви функции к другой:

$$A6 \leq \$E\$1$$

В этом выражении *A6* определяет ссылку на ячейку, в которой находится текущее значение аргумента *X*, а *\$E\$1* – абсолютную ссылку на ячейку таблицы, в которую записано значение параметра α .

Результат вычисления значения этого отношения равен *ИСТИНА*, если $X \leq \alpha$ и *ЛОЖЬ*, если $X > \alpha$.

После того, как ввод этого отношения завершён, справа от поля ввода высвечивается его значение "*ИСТИНА*", определяемое по результату вычисления отношения для $X=X0$.

В поле ввода "Значение_если_истина" вводим формулу, соответствующую первой ветви функции $F(X,\alpha)=C*X$:

$$\$B\$2*A6$$

Здесь $\$B\2 абсолютная ссылка на ячейку со значением константы C , а $A6$ определяет ссылку на ячейку, в которой находится соответствующее значение аргумента X .

В поле ввода "Значение_если_ложь" вводим формулу, соответствующую второй ветви функции $F(X,\alpha)=D*X$:

$$\$D\$2*A6$$

Здесь $\$D\2 абсолютная ссылка на ячейку со значением константы D .

Если значения выражений могут быть вычислены, то они отображаются за знаком "=" справа от поля ввода.

ЕСЛИ

Логическое_выражение	A6<=E\$1	= ИСТИНА
Значение_если_истина	\$B\$2*A6	= -6,4
Значение_если_ложь	\$D\$2*A6	= -13,6

= -6,4

Возвращает одно значение, если указанное условие истинно, и другое, если оно ложно.

Логическое_выражение любое значение или выражение, которое при вычислении дает значение ИСТИНА или ЛОЖЬ.

Значение: [?] [OK] [Отмена]

Рис. 4.5. Заполненное окно второго шага мастера функция для функции (4.3)

Под полем, отведённым для размещения полей ввода всех параметров, необходимых для вычисления значения функции, располагается символ "=", за которым выводится окончательное вычисленное значение функции для текущего значения аргумента X .

Итоговая формула, введённая в ячейку $B6$, имеет следующий вид:

$$=ЕСЛИ(A6<=E$1;$B$2*A6;$D$2*A6)$$

После нажатия кнопки **OK** в ячейку $B6$ будет записано вычисленное значение, а при протягивании формулы по столбцу B в его ячейки будут записаны все значения функции $F(X,\alpha)$ (4.3).

Следует отметить, что в функции, выбранной для этого примера, значения для левой и правой ветвей совпадают в точке переключения.

2. Реализация функций с тремя и более ветвями

Вычисление функции с двумя ветвями реализуется достаточно просто. Более сложным является вариант реализации функции с тремя и более ветвями.

В качестве примера рассмотрим разветвляющуюся функцию

$$F(x) = \begin{cases} \sin(a * x) & \text{при } x \leq \alpha, \\ a * x + b & \text{при } \alpha < x \leq \beta, \\ \text{ctg}(a * x) & \text{при } \beta < x. \end{cases} \quad (4.4)$$

Перед реализацией алгоритма, определяемого формулой (4.4) мы более подробно рассмотрим один из способов выполнения его предварительной разработки. Этот способ – разработка блок-схемы алгоритма. Он позволяет продумать шаги, которые необходимо выполнить при реализации решения поставленной задачи, и часто позволяет сократить время и трудозатраты прилагаемые разработчиком.

Блок-схема алгоритма наглядно представляет предварительный проект реализации алгоритма, который может быть изменён и уточнён в процессе разработки алгоритма и реализации программы. Тщательная проработка алгоритма позволяет существенно сократить время написания и отладки программы.

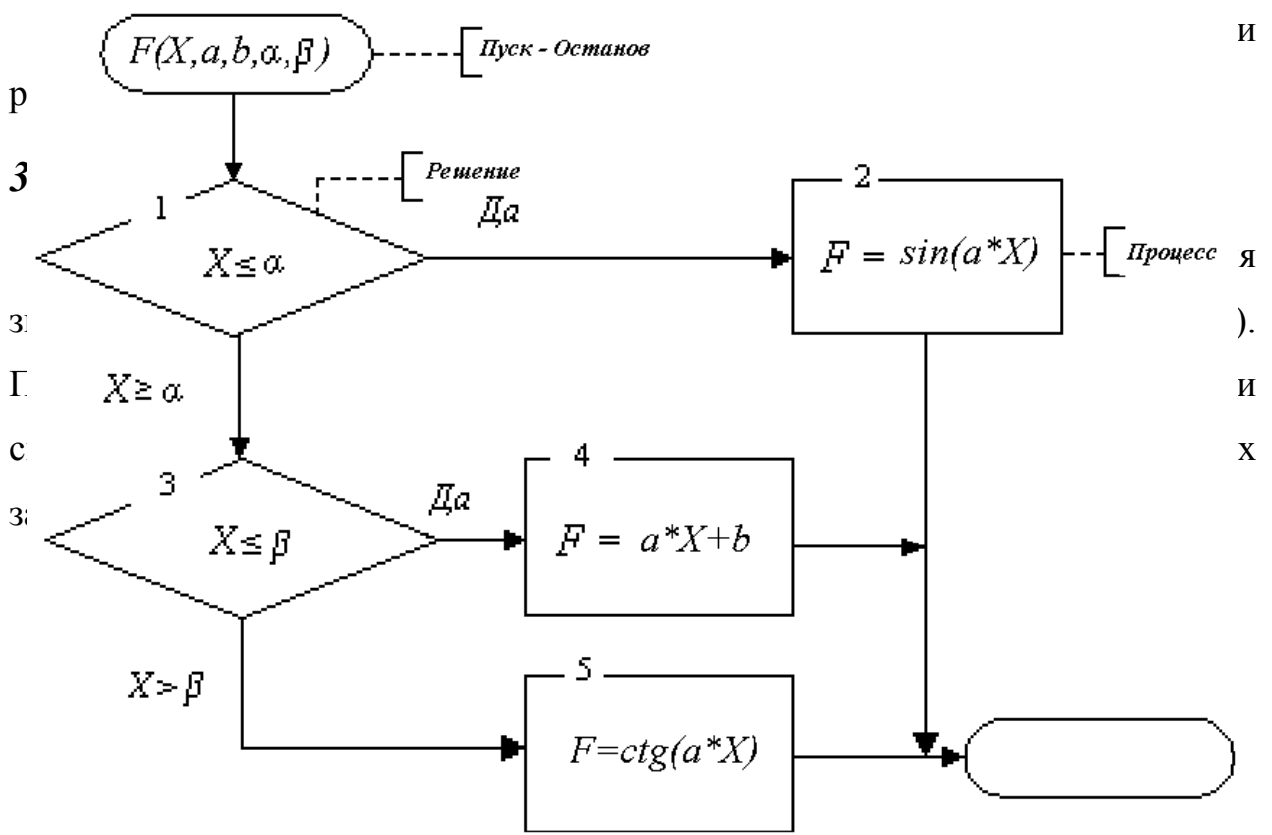


Рис. 4.6. Блок-схема алгоритма вычисления значения разветвляющейся функции $F(x)$

Символы "**Пуск-останов**" используются в блок-схемах для обозначения точек, определяющих начало, прерывания и конец алгоритма.

Символы "**Процесс**" используются для определения одного или нескольких последовательно выполняемых действий.

Символы "**Предопределённый процесс**" определяют некоторые фрагменты алгоритмов, которые могут быть в дальнейшем раскрыты в виде отдельных блок-схем.

Символы "**Решение**" используются для определения в алгоритмах точек разветвления. В этих точках направление передачи управления определяется в зависимости от выполнения некоторого условия. Разделение потока передачи управления в алгоритмах возможно только на блоках "**Решение**". Все остальные блоки имеют один вход и один выход.

Верификация алгоритма и программы – проверка правильности результатов, получаемых при их выполнении, наиболее важные шаги в решении любой задачи.

Анализ ожидаемых и получаемых решений начинается уже на этапе постановки задачи.

Проанализируем функцию $F(x)$, таблицу значений которой мы собираемся получить.

Функции, задающие первую и вторую ветви разветвляющейся функции, определены для любых значений аргумента X в диапазоне от $+\infty$ до $-\infty$.

В определение функции третьей ветви входит тригонометрическая функция котангенс. Как известно, эта функция имеет разрывы в точках со значениями аргумента кратными π . Значения функции в таких точках изменяется от $-\infty$ до $+\infty$.

При табулировании функций на ЭВМ вероятность получения значений аргумента X точно кратных π чрезвычайно мала. Если такое значение всё таки будет получено, то результатом операции будет вывод в ячейку сообщения **#ДЕЛ/0!** – деление на ноль.

Как правило в таких точках вычисляются значения функции с высоким порядком 10^{15} и выше (очень большие по модулю).

Запишем формулу **ЕСЛИ**, реализующую вычисление функции (4.4) в ячейке таблицы *Excel*. Подробное описание алгоритма реализации функций с тремя ветвями дается в методических указаниях по выполнению седьмой лабораторной работы, записанных в файл *ЛР_7*. Поэтому здесь ограничимся записью формулы и краткими пояснениями к ней.

=ЕСЛИ(A5<=\$B\$3;SIN(\$B\$2*A5);ЕСЛИ(A5<=\$D\$3;B2*A5+\$D\$2;1/TAN(\$B\$2*A5)))

Первое логическое выражение соответствует оператору отношения блока 1 блок-схемы рис. 4.6: $X \leq \alpha$ [(A5<=\$B\$3) A5 ссылка на X, \$B\$3 ссылка на α]

Если это логическое выражение истинно, то вычисляется значение функции **SIN(\$B\$2*A5)**; (блок 2 рис.4.6), если же значение логического выражения **ЛОЖЬ**, то записывается следующее обращение к функции **ЕСЛИ**, реализующее блок 3 рис. 4.6.

Логическое выражение этого оператора **A5<=\$D\$3** [$X \leq \beta$]. Если значение этого отношения **ИСТИНА**, то выполняется оператор блока 4: **a*X**

+b, если же это значение ЛОЖЬ, то вычисляется оператор блока 5: $1/\text{TAN}(\$B\$2*\text{A5})$.

В программе на языке *Visual Basic* запишем реализацию функции *FVB*:

Public Function FVB(X, a, b, alfa, betta)

'x – аргумент функции, a,b,alfa,betta - параметры

*If (X <= alfa) Then FVB = Sin(X*a) _*

Else _

*If (X <= betta) Then FVB = a * X + b _*

*Else FVB = 1 / Tan(a * X)*

End Function

4. Представление информации в памяти ЭВМ

Информационные объекты записываются и хранятся в ячейках памяти ЭВМ. Способ представления информации определяется типом информационного объекта – типом данных, записываемых в память ЭВМ.

Тип данных определяет множество допустимых значений, которое может принимать определяемая этим типом информация.

В языке *Visual Basic* определены следующие типы данных:

№	Тип	Описание
1	<i>Byte</i>	Байт
2	<i>Boolean</i>	Логический
3	<i>Integer</i>	Целое
4	<i>Long</i>	Длинное целое
5	<i>Single</i>	С плавающей точкой обычной точности
6	<i>Double</i>	С плавающей точкой двойной точности
7	<i>Currency</i>	Денежный
8	<i>Decimal</i>	Масштабируемое целое
9	<i>Date</i>	Дата и время
10	<i>Object</i>	Объект
11	<i>String</i>	Строка переменной длины
12	<i>String</i>	Строка постоянной длины
13	<i>Variant</i>	Числовые подтипы
14	<i>Variant</i>	Строковые подтипы
15	<i>Type</i>	Типы данных, определяемые пользователем

В памяти ЭВМ объект каждого типа занимает некоторое число байтов, обеспечивающее возможность записи любого из возможных для данного типа значений.

При вычислениях, выполняемых на ЭВМ, точность представления чисел определяется числом двоичных разрядов (битов) в ячейке памяти,

$x_0 = -9,424777961$ $x_k = 9,42477796$ $Dx = 0,942478$
 $a = 1$ $b = 1$
 $\text{alfa} = -3,141592654$ $\text{beta} = 4,71238898$

X	F(X)	FVB(X)
-9,42478	-3,67545E-16	-3,67382E-16
-8,4823	-0,809016994	-0,809016994
-7,53982	-0,951056516	-0,951056516
-6,59734	-0,309016994	-0,309016994
-5,65487	0,587785252	0,587785252
-4,71239	1	1
-3,76991	0,587785252	0,587785252
-2,82743	-0,661923647	-1,827433388
-1,88496	-0,884955592	-0,884955592
-0,94248	0,446025451	0,057522204
0	1	1
0,942478	0,165949004	1,942477796
1,884956	1,840738168	2,884955592
2,827433	3,827433388	3,827433388
3,769911	1,625613006	4,769911184
4,712389	9,674274258	5,71238898
5,654867	-1,37638192	-1,37638192
6,597345	3,077683537	3,077683537
7,539822	0,324919696	0,324919696
8,4823	-0,726542528	-0,726542528
9,424778	-2,72076E+15	-2,72196E+15

