

Основные инструкции языка Visual Basic

Основные инструкции языка Visual Basic

Сгруппируем основные инструкции языка Visual Basic по типу операций:

- присваивания, в том числе с выполнением математических операций в правой части инструкции;
- проверки условий, управления последовательностью исполнения инструкций, организации циклов;
- инструкции для работы с файлами.

Основные операторы VBA

Операторы присваивания

Dim-объявляет переменные и выделяет память под них

Data- устанавливает значение системной даты

Операторы условия

If <Выражение> Then <Инструкция, исполняемая, если выражение-условие истинно>.(простая структура)

If..... Then.....Else
(блочная структура)

Примеры алгоритмов

If $x > 0$ Then

$y = 2 * x$

$z = 1$

Else

$y = x - 5$

$z = 0$

End If

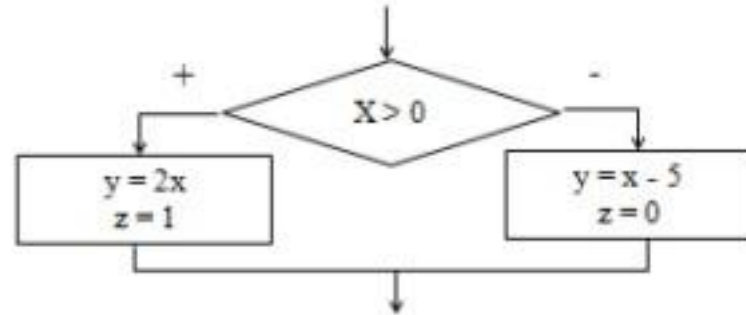


Рис. 3.3

Ветвление

If $x > 0$ Then

$y = 2 * x$

Else

If $x = 0$ Then

$y = x - 5$

Else

$y = x^2$

End If

End If

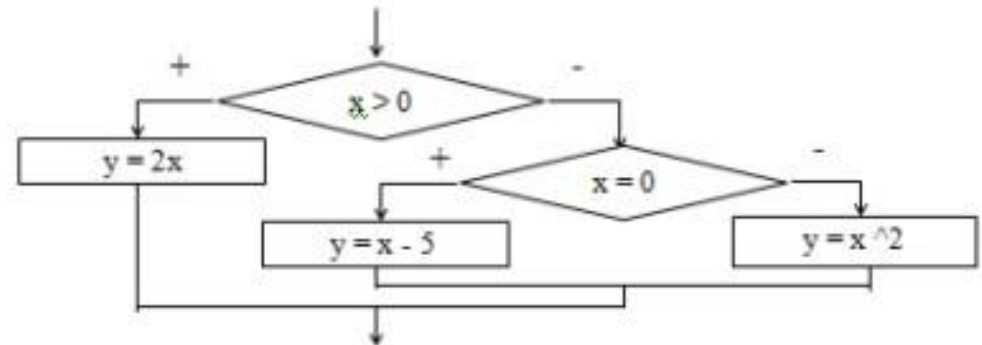


Рис. 3.4

Основные операторы VBA

Операторы цикла

For НАЧ_ЗНАЧЕНИЕ **To** КОН_ЗНАЧЕНИЕ **Step** ШАГ
БЛОК_ОПЕРАТОРОВ
[Exit For]
БЛОК_ОПЕРАТОРОВ
Next СЧЕТЧИК

для организации циклов с неизвестным заранее числом повторений используются

циклы с предусловием - **Do While ... Loop**

циклы с постусловием – **Do ... Loop While**

Примеры алгоритмов

$S = 0$

$P = 1$

For $i = 1$ to 5

$S = S + 2$

$P = P * S$

Next i

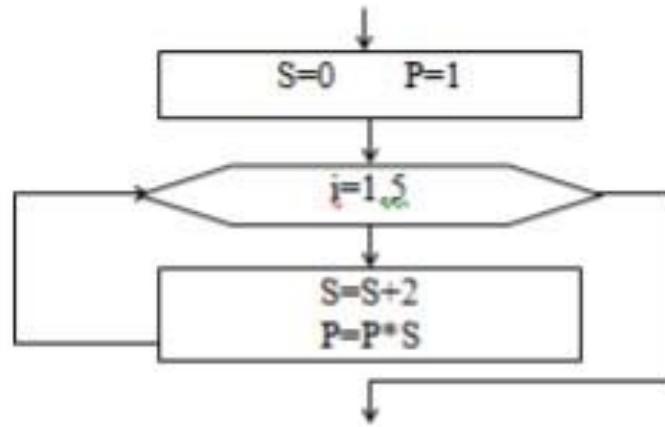


Рис. 3.6

$k = 0$

$f = 1$

Do While $f \leq 15$

$k = k + 1$

$f = f + 2$

Loop

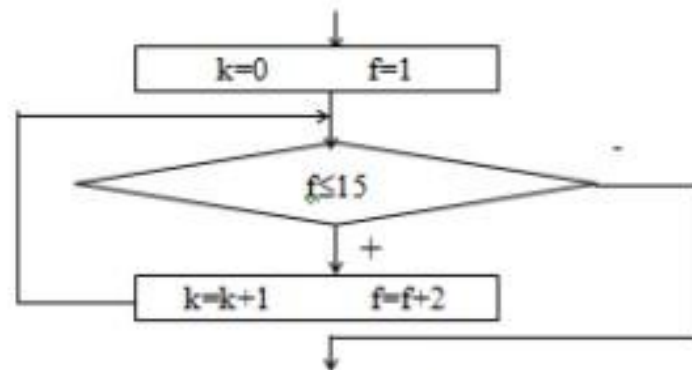


Рис. 3.7

Арифметические операции

Операция	Содержание операции	Приоритет
\wedge	Возводит первое арифметическое выражение в степень, задаваемую вторым арифметическим выражением	1
-	Знак минус ставится перед именем переменной, константы или перед числом	2
*	Перемножает два арифметических выражения	3
/	Делит первое арифметическое выражение на второе	3
\	Округляет два арифметических выражения до целых значений и делит первое число на второе. Результат округляется до целого	4
Mod	Округляет оба арифметических выражения до целых чисел, делит первое число на второе и возвращает в качестве результата остаток	5
+	Складывает два арифметических выражения	6
-	Вычитает из первого арифметического выражения второе	6

Операции сравнения

Операция	Наименование	Пример (R имеет тип Boolean)	Результат (R)
=	Равно	$R = (10=5)$	False
>	Больше	$R = 10>5$	True
<	Меньше	$R = 10<5$	False
>=	Больше или равно	$R = \text{"a"} \geq \text{"b"}$	False
<=	Меньше или равно	$R = \text{"a"} \leq \text{"b"}$	True
<>	Не равно	$R = \text{"a"} \neq \text{"b"}$	True

Логические операции

Операция	Наименование
And	Логическое умножение
Or	Логическое сложение
Xor	Исключающее Or (или)
Not	Логическое отрицание

Тип данных	Размер в байтах	Разрядность (цифр)	Диапазон, пояснение
Boolean	2	1	True, False – логические значения
Byte	1	≤ 3	0, 255 – целые положительные числа
Integer	2	≤ 5	-32768, 32767 – целые числа
Long	4	≤ 10	-2147483648, 2147483647 – длинные целые
Single	4	≤ 7	$-3.402823 \cdot 10^{38}$, $-1.401298 \cdot 10^{-45}$ – отрицательные числа
Double	8	≤ 15	$1.401298 \cdot 10^{-45}$, $3.402823 \cdot 10^{38}$ – положительные числа
Currency	8	≤ 19	Действительные числа с двойной точностью
Date	8		Действительные числа с 4 знаками после десятичной
String	1+L		точки. Используются для точных денежных расчетов
Object	4		01.01.0100, 31.12.9999 – даты
Variant			Строка из L символов; $L \leq 65535$ Объект – фактически, ссылка на него, т.е. адрес его размещения в оперативной памяти Может принимать любой тип данных, т.е. настройка на тип операнда происходит во время исполнения инструкций или процедур

Встроенные функции Visual Basic

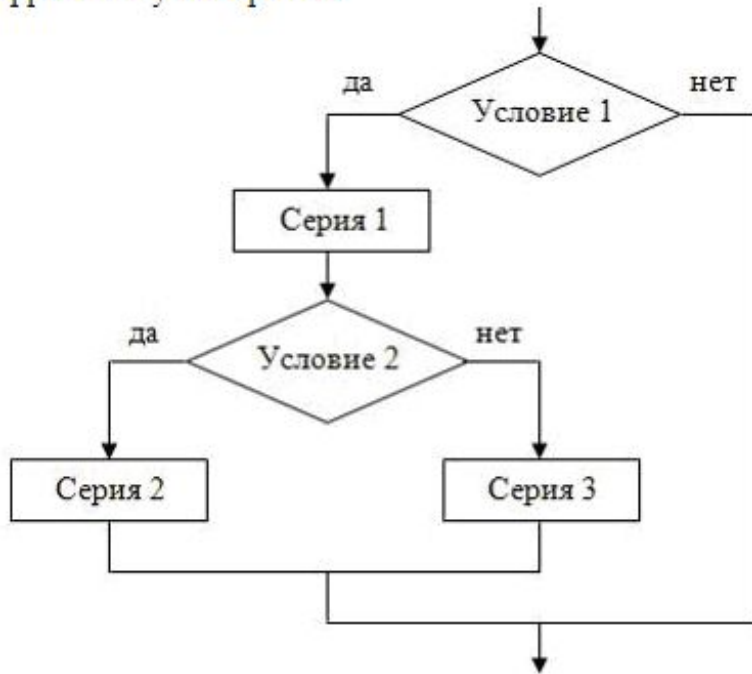
Функция	Содержание арифметической функции
1	2
Abs(x)	Возвращает значение, тип которого совпадает с типом переданного аргумента X, равное абсолютному значению указанного числа
Atn(x)	Возвращает значение типа Double, содержащее арктангенс числа X
Cos(x)	Возвращает значение типа Double, содержащее косинус угла X
Exp(x)	Возвращает значение типа Double, содержащее результат возведения числа e (основание натуральных логарифмов) в указанную степень
Fix(x)	Возвращает значение типа, совпадающего с типом аргумента, которое содержит целую часть числа, ближайшее отрицательное целое число, большее либо равное указанному X

Int(x)	Возвращает значение типа, совпадающего с типом аргумента, которое содержит целую часть числа, ближайшее отрицательное целое число, меньшее либо равное указанному X
Log(x)	Возвращает значение типа Double, содержащее натуральный логарифм числа X
Rnd(x)	Возвращает значение типа Single, содержащее случайное число
Sgn(x)	Возвращает значение типа Variant (Integer), соответствующее знаку указанного числа. Обязательный аргумент: число X может представлять любое допустимое числовое выражение. Возвращаемое значение: $X > 0$, то Sgn(x) возвращает 1; если X равняется нулю, то 0; если $X < 0$, то -1
Sin(x)	Возвращает значение типа Double, содержащее синус угла X

Sqr(x)	Возвращает значение типа Double, содержащее квадратный корень указанного числа
Tan(x)	Возвращает значение типа Double, содержащее тангенс угла X
Строковые	LCASE() – преобразование строки в строчные буквы
	UCASE() – преобразование строки в заглавные буквы
	LEN() – определение длины строки и др.
	INSTR() – позиция первой встречи одной строки внутри другой

Даты и времени	DATE() – текущая дата
	TIME() – текущее время
	NOW() – текущая дата и время
	DAY() – номер дня года и др.
Преобразования типов данных	CINT() – действительного числа в целое (с округлением)
	CSTR() – числа в строку символов
	CVAR() – преобразование в тип данных VARIANT
	FORMAT() – форматирование данных, формирование строк
	VAL() – преобразование строки символов в число и др.

Укажите фрагмент программы, соответствующий приведенному фрагменту алгоритма.



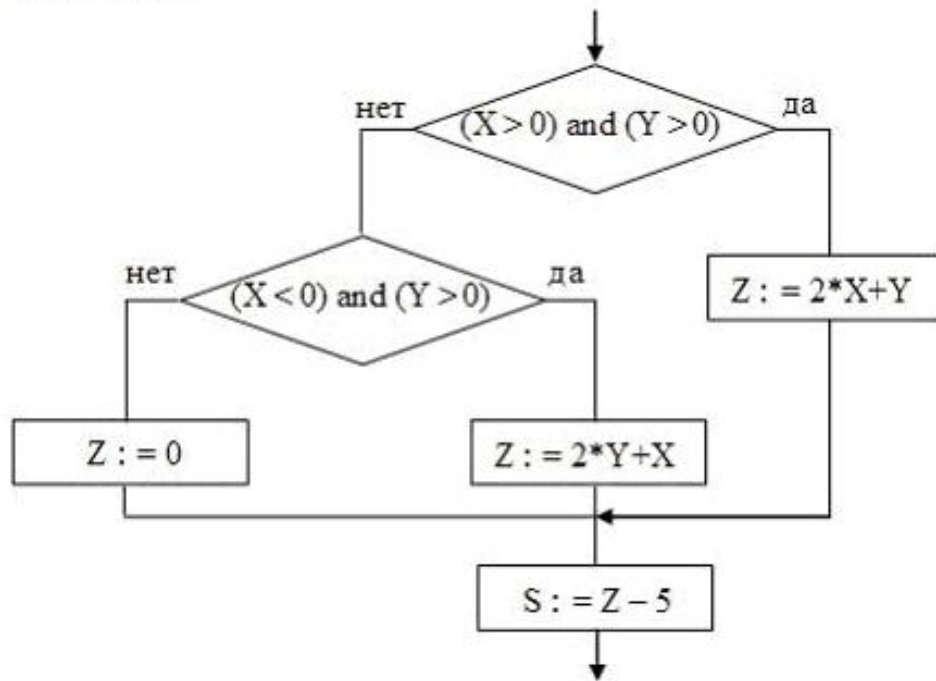
если Условие 1
то Серия 1
иначе
если Условие 2
то Серия 2
иначе Серия 3
все
все

если Условие 1
то Серия 1
все
если Условие 2
то Серия 3
иначе Серия 2
все

если Условие 1
то Серия 1
все
если Условие 2
то Серия 2
иначе Серия 3
все

если Условие 1
то
иначе
если Условие 2
то серия 2
иначе серия 3
иначе серия 1
все
все

Дан фрагмент алгоритма. Логическая операция *and* используется для обозначения одновременности выполнения условий. При начальных значениях переменных $X=2$, $Y=10$ переменные Z и S примут значения ...



- $Z=22, S=17$
- $Z=14, S=9$
- $Z=14, S$ не определено
- $Z=36, S=31$

Разветвляющаяся конструкция относится к базовым алгоритмическим конструкциям и обеспечивает выбор между двумя альтернативами в зависимости от входных данных. Вначале проверяется *условие* (вычисляется, логическое выражение). Если условие истинно, то выполняются *действия 1* – последовательность команд, на которую указывает стрелка с надписью «да» (положительная ветвь). В противном случае выполняются *действия 2* (отрицательная ветвь). Различают полное и неполное ветвления.

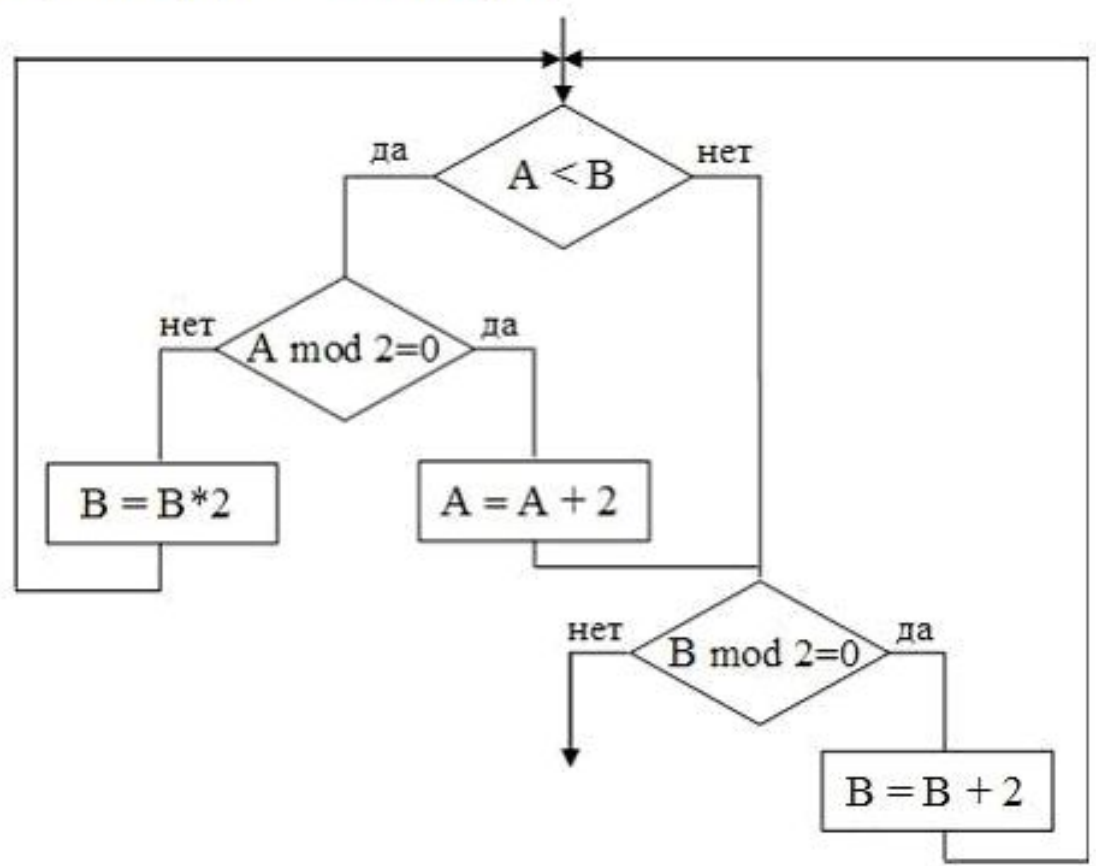
Полное ветвление позволяет организовать две ветви в алгоритме, каждая из которых ведет к общей точке их слияния, так что выполнение алгоритма продолжается независимо от того, какая ветвь была выбрана.



В предложенной задаче в блоках условий в логическом выражении используется логическая связка **and** (**И**) – конъюнкция. Если она объединяет истинные условия – все логическое выражение будет истинным, а во всех остальных случаях – ложным.

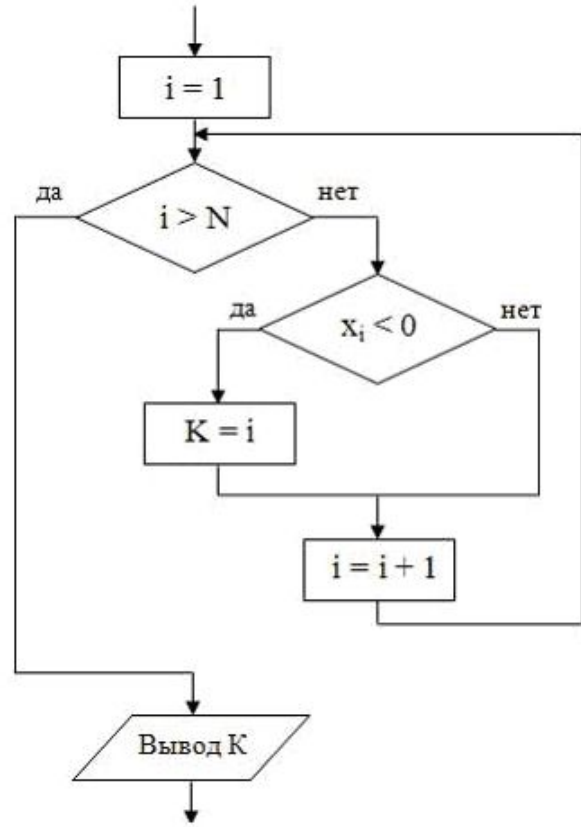
Проверяем первое логическое выражение. Оно истинно, так как истинны оба условия, в него входящие ($X > 0$ и $Y > 0$). Поэтому дальнейшее вычисление пойдет по ветке «да». Вычисляем значение переменной $Z = 2 * 2 + 10 = 14$. Значение переменной S вычисляется независимо от того, какая ветвь была выбрана, поэтому $S = 14 - 5 = 9$.

Определите, при каких начальных значениях переменных A и B алгоритм, представленный следующей блок-схемой, закончит работу (mod – функция, вычисляющая остаток от деления нацело первого аргумента на второй).

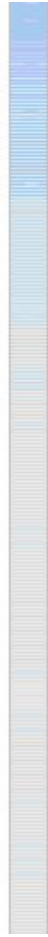


- A=1, B=6
- A=5, B=3
- A=3, B=5
- A=4, B=2

Задан одномерный массив X_1, X_2, \dots, X_N . Фрагмент алгоритма



определяет ...



- индекс первого отрицательного элемента
- количество отрицательных элементов
- минимальный элемент массива
- индекс последнего отрицательного элемента

Функция *mod* вычисляет остаток от деления нацело первого аргумента на второй. Значение переменной *K* после выполнения следующей программы:

k := 0;

нц

для *i* **от** 1 **до** 100

если (*mod*(*i*, 3) = 2) **и** (*mod*(*i*, 5) = 1)

то *k* := *k* + 1

все

кц

будет равно ...

6

7

288

3

Значение переменной *m* после выполнения фрагмента алгоритма (операция *mod* (*x*, *y*) – получение остатка целочисленного деления *x* на *y*)

k := 70

выбор

при *mod*(*k*, 12) = 5: *m* := *k*;

при *mod*(*k*, 12) < 5: *m* := 2;

при *mod*(*k*, 12) = 13: *m* := 3;

иначе *m* := 1;

все

будет равно ...

2

1

3

70