

**Министерство образования Московской области
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
"СЕРПУХОВСКИЙ КОЛЛЕДЖ"**



"СОГЛАСОВАНО"
Зам.директора по УМР

Г.В. Вяльых
" 06 " 12 2016 г.

**Методическая разработка пособия
по выполнению самостоятельных работ
по дисциплине "Информатика и ИКТ"
для студентов I курса**



Рассмотрена на заседании предметно-
цикловой комиссии общеобразовательных,
общих гуманитарных, социально-
экономических и естественнонаучных

дисциплин

Протокол № 4 от 06.12 2016г.

Закатина И.В.

РЕЦЕНЗИЯ

"Методическое пособие по изучению студентами алгоритмизации и программирования , а также выполнению самостоятельных работ при изучении дисциплины "Информатика и ИКТ"на I курсе "Серпуховского колледжа".

Разработчик: Андреева Нина Анатольевна преподаватель ГБПОУ МО "Серпуховский колледж".

Пособие составлено в соответствии с Федеральным государственным образовательным стандартом (далее - ФГОС) среднего профессионального образования по дисциплине "Информатика и ИКТ"

Пособие содержит систематизированное изложение основных понятий и методов алгоритмизации и программирования на Quick Basic.

Разделы пособия представляют собой своеобразные опорные конспекты, которые удобно применять как при объяснении нового материала, так и при повторении, обобщении уже изученного.

Для более четкой организации работы обучающихся по программированию в пособии даны рекомендации по составлению программ.

Использование пособия на уроках при изучении алгоритмизации и программирования позволяет учителю варьировать время на изучение теоретического материала и выполнение практических работ, построить для каждого учащегося индивидуальную образовательную траекторию.

Данное пособие также может быть использовано учащимися для самостоятельного изучения алгоритмизации и программирования, углубления своих знаний по составлению программ на Quick Basic.

Пособие составлено методически грамотно, выбранный автором стиль изложения при описании материала адекватен поставленной задаче и учитывает круг потенциальных читателей. Методическое пособие предусматривает:

- усвоение обучающимися необходимых теоретических знаний по алгоритмизации и программированию на языке Quick Basic;
- освоение всевозможных методов решения задач, реализуемых на Quick Basic;
- углубление знаний по программированию на Quick Basic;
- формирование интереса к профессиям, связанным с элементами программирования.

"Методическое пособие по программированию на Quick Basic" может быть рекомендовано к использованию на уроках информатики и ИКТ колледжа, а также для изучения алгоритмизации и программирования на языке Quick Basic учащимися общеобразовательных школ и студентами средних профессиональных образовательных учреждений.

Рецензенты:

Бковская Е.В., методист ГБПОУ МО "Серпуховский колледж"
Андреев Борис Михайлович , преподаватель информационных технологий ГБПОУ МО "Серпуховский колледж"

Введение

Язык программирования - это система команд, "понятных" ЭВМ. Разделяют языки низкого и высокого уровня. Первые (в качестве примера можно назвать Ассемблер) позволяют управлять вычислительным процессором напрямую, при помощи машинных команд. Написание программ на языках низкого уровня - процесс сложный и трудоемкий: это удел узкого круга специалистов. Языки высокого уровня требуют для общения с машиной "переводчика" - транслятора, но процесс программирования при этом существенно упрощается. В настоящее время используют большое количество высокоуровневых языков в различных версиях, например, Basic, Pascal, C, C++, Perl и т.д. Каждый из них имеет свои особенности, но при этом общие принципы программирования на разных языках одни и те же. Освоение одного из языков облегчает изучение других.

В настоящем пособии изложены начала программирования на языке QBasic. Подобный выбор продиктован тремя обстоятельствами.

Во-первых этот язык достаточно прост в освоении, как и следует из его названия - Beginner's Allpurpose Symbolic Instruction Code - универсальный язык символьических инструкций для начинающих.

Во-вторых, этот язык рассматривается в курсе основ информатики и вычислительной техники во многих средних профессиональных учебных заведениях.

В-третьих, он является родоначальником таких мощных современных языков программирования для среды Windows, как VB (Visual Basic) и VBA (Visual Basic For Application), что позволит студенту использовать полученные знания при изучении электронных таблиц MS Excel.

Существует множество версий языка Basic разного уровня сложности и совершенства, созданных для ЭВМ различных типов (QBasic, GWBasic, MSX-Basic, HBasic, Visual Basic, VBA и др.). Подчас данные версии несовместимы, но, несмотря на это, все они имеют общий фундамент - совокупность основных операторов и систему команд, которые и рассмотрены в настоящем пособии.

Практическая разработка программ на алгоритмическом языке Quick-Basic может вызывать ряд трудностей у студентов I курса, изучающих этот язык программирования. В предлагаемом пособии на доступном языке, изложен необходимый теоретический материал по алгоритмизации и программированию на языке Quick-Basic и представлены некоторые примеры использования конструкций языка. Данное пособие можно использовать как раздаточный дидактический материал для каждого студента.

1. Языки парограммирования.

Программа – это набор машинных команд, которые следует выполнить компьютеру для реализации того или иного алгоритма.

Программы для компьютера (ЭВМ) составляются с помощью языка программирования. Всё множество языков программирования можно разделить на классы:

- **машинные языки** – языки программирования, воспринимаемые аппаратной частью компьютера (машинные коды - последовательность из 0 и 1);
- **машинно-ориентированные языки**, предназначенные для использования на тех компьютерах, для которых эти языки разработаны (ассемблеры - системы условных обозначений машинных команд);
- **алгоритмические языки** – языки программирования, не зависящие от архитектуры компьютера (Паскаль, Фортран, Бейсик и др.).

Машинные и машинно-ориентированные языки относятся к языкам низкого уровня т.к. ориентированы на аппаратную часть компьютера. Работа по составлению программ на данных языках является трудоемким процессом и требует высокой квалификации от программистов.

Алгоритмические языки относятся к языкам высокого уровня т.е. в достаточно сильной степени приближены к человеческому языку (английскому), но для выполнения на компьютере необходим предварительный перевод на машинный язык специальной программой –транслятором.

Трансляторы бывают двух типов: компиляторы и интерпретаторы.

Компилятор – читает всю программу целиком, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется.

Интерпретатор – переводит и выполняет исходную программу построчно.

К языкам высокого уровня принадлежит и язык программирования QBasic к изучению которого мы приступаем.

2. Общие сведения об языке программирования Qbasic

Язык BASIC (Beginner's All-purpose Instruction Code) разработан Джоном Кимини и Томасом Куртцем в Дартмутском колледже, США, в середине 1960 г.

BASIC занимает особое место среди всех языков высокого уровня. С самого начала он задумывался как универсальный язык для начинающих, и средства программирования на BASIC фирмой Microsoft включались в комплект поставки операционной системы MS-DOS. За выходом первой версии BASIC последовал выпуск других, улучшенных и обладающих большими возможностями, версии – GW-BASIC, QuickBASIC, QBASIC, Basic Professional.

Принципиально новым словом фирмы явилось появление системы Visual BASIC for Windows предназначенной для профессиональных разработок мощных программных комплексов.

Рабочая среда QBASIC ничем не отличается от сред других современных языков программирования. К ней относится и удобная система меню, и возможность работы с манипулятором "мышь", и многооконность, а также удобная справочная система. Кроме того знания полученные при изучении конструкций среды программирования QBASIC успешно будут использоваться для работы в офисном приложении MS Excel.

Вот некоторый достоинства QBASIC (с точки зрения массового пользователя):

- простота синтаксиса;
- простота организации данных и управляющих структур;
- большое число *встроенных* команд и функций, позволяющих без труда выполнять такие операции, как управление текстовым и графическим экраном, обработка символьных строк и т.п.)

Особым достоинством QBASIC следует считать возможность работы в режиме *интерпретации*, который резко упрощает процесс отладки программ: исполнение почти каждой команды можно проверить сразу после написания.

2.1 Алфавит языка QBASIC.

Основными элементами любого языка программирования являются: алфавит, оператор, синтаксис, семантика.

Алфавит языка – набор разрешенных символов языка.

Оператор - основная конструкция языка, определяющая конкретное действие.

Синтаксис - правила образования отдельных конструкций языка (команд, операторов) из символов языка.

Семантика - правила истолкования конструкций языка.

В языке QBASIC используются следующие символы:

- Строчные и заглавные буквы латинского алфавита от A до Z.
- Цифры 1,2,3,4,5,6,7,8,9,0.
- Знаки арифметических и логических операций.
- Специальные символы:

. – точка;	# - признак вещественной величины двойной точности;
, – запятая;	% - признак целой величины;
; – точка с запятой;	\$ - признак текстовой величины;
: – двоеточие;	() - круглые скобки;
! – признак вещественной величины;	" – кавычки.

Буквы русского алфавита можно использовать только в комментариях и в текстовых константах. Кроме того, в качестве символов языка используются некоторые слова английского языка (REM, GOTO, END и т.д.), которые называются **ключевыми** (служебными) в среде QBASIC отображаются большими буквами независимо от регистра клавиатуры. Необходимые ключевые слова будут вводится по мере необходимости.

2.2 Данные в QBASIC.

В языке могут использоваться следующие виды данных:

- Константы;**
- Переменные.**

Константы. Константами называются величины, которые имеют постоянное значение, не меняющееся в процессе выполнения программы. В языке QBASIC используются константы двух типов: числовые и строковые (текстовые).

Числа. В языке используются целые и вещественные числа. Знак "+" перед числом не ставится, а для отделения целой части при записи десятичных дробей используется точка. Нулевую целую часть дробного числа можно опускать. Для отличия буквы О от цифры О последняя перечеркивается Ø.

Вещественные (действительные) числа с большими или малыми порядками могут представляться в экспоненциальной форме. Где латинская буква Е имеет смысл "возвести 10 в степень" за которой следует показатель степени (порядок).

Примеры записи чисел:

Целые		Вещественные	
Запись в математике	QBASIC	Запись в математике	QBASIC
+0,12	.12	-24,345	-24.345
- 456	-456	0,005342	.5342E-2

Строковые константы - это набор буквенно-цифровых и/или специальных символов алфавита языка. Строковые константы ограничиваются с двух сторон кавычками.

Целая константа должна лежать в диапазоне – 32768 до + 32768.

Длина строковой константы не более 255 символов.

Пример записи строковой величины:

Запись в QBASIC	Вывод на экран
PRINT "Информатика"	Информатика

2.3 Переменные.

Переменная – это величина, значение которой может изменяться в процессе выполнения программы. Переменные в QBASIC бывают числовые и строковые (текстовые). Переменная имеет имя (идентификатор, обозначение). Имя может состоять из совокупности латинских букв и цифр (до 40 символов) и может заканчиваться знаком типа переменной. Первым символом в имени переменной обязательно должна быть буква.

Используемые признаки типа переменной:

% - целый тип;

! - вещественный тип одинарной точности (с точностью до 7 знаков после запятой);

- вещественный тип двойной точности (с точностью до 17 знаков после запятой);

\$ - текстовой тип.

Имя целой переменной заканчивается знаком %.

Пример: X12%, Y\$%

Имя текстовой переменной заканчивается знаком \$.

Пример: A\$, DK\$.

Наряду с рассмотренными выше простыми переменными в QBASIC могут использоваться переменные с индексами, служащие для обозначения элементов массивов.

Имя массива – переменная, за которой следует один или два индекса, заключенные в скобки (в QBASIC допускаются одномерные и двухмерные массивы).

Пример: A(2), B(2,2), CR3(8).

Одномерный массив A(4) состоит из четырех элементов и его распределение в памяти ПК имеет вид:

A(0) A(1) A(2) A(3) A(4)

Двумерный массив B(1,1) состоит из четырех элементов и его распределение в памяти ПК имеет вид:

B(0,0) B(0,1) B(1,0) B(1,1)

Нельзя использовать одно и то же имя для одномерного и двухмерного массива в одной программе.

2.4 Выражения и операции в QBASIC.

Выражение – это константы, переменные, индексные переменные, стандартные функции соединенные между собой знаками операций.

Операции в языке QBASIC делятся на:

1. Арифметические;
2. Отношения;
3. Логические;
4. Стандартные функции.

Арифметические операции

Символ операции	Назначение	Алгебраическая запись	Пример на QBASIC	Приоритет
^	Возведение в степень	x^2	X^2	3
*	Умножение	$3 \cdot d$	3*D	4
/	Деление	$\frac{x}{y}$	X/Y	4
MOD	Вычисление целой части при делении двух целых чисел	A : B	A MOD B	5
+	Сложение	a +b	A+B	6
-	Вычитание	a-b	A-B	6

()	Изменение последовательности действий	$\frac{2x}{1+y}$	$2^*X/(1+Y)$	1
SIN(X)	Вычисление стандартной функции sin(x)	sin(x)	SIN(X)	2

Для указания порядка выполнения арифметических действий используются круглые скобки. В первую очередь вычисляются части выражений, заключенные в круглые скобки, даже если операция в скобках имеет более низкий приоритет, чем операция вне скобок. Вычисление групп с вложенными скобками начинается с внутренних скобок и идет в направлении к внешним скобкам. Операции, размещенные в одной строке и имеющие равный приоритет, выполняются последовательно слева направо.

Операции отношения.

Операции отношения используются в качестве условий для сравнения двух величин в операторах условного перехода.

Символ оператора	Назначение	Пример
=	равно	a = b
<>	не равно	a <> b
<	больше	a > b
>	меньше	a < b
<=	меньше или равно	a <= b
>=	больше или равно	a >= b

Логические операции.

Логические операции применяются для объединения нескольких операций отношения в составные (сложные) условия в операторах условного перехода :

- NOT – логическое отрицание "НЕ";
- AND – логическое умножение "И";
- OR - логическое сложение "ИЛИ".

Стандартные функции.

Стандартные функции используются для вычисления часто употребляемых функций. Аргумент стандартной функции заключается в круглые скобки. В QBASIC могут быть использованы следующие стандартные функции:

Математическая запись	Запись в QBASIC	Назначение
sin (x)	SIN (X)	Синус x (аргументом служит радианная мера угла)
cos (x)	COS (X)	Косинус x (аргументом служит радианная мера угла)
tg(x)	TAN (X)	Тангенс x
arctg (x)	ATN (X)	Арктангенс x
int (x)	INT (X)	Целая часть x
\sqrt{x}	SQR (X)	Корень квадратный из x
ln(x)	LOG (X)	Натуральный логарифм x
e^x	EXP (X)	Экспонента x
X	ABS(X)	Модуль числа x
-	RND(X)	Генератор случайных чисел от 0 до 1
Функция знака числа x (сингум)	SGN (X)	Возвращает 1, если X>0; 0, если X=0; -1, если X<0
-	CHR\$(X)	Возвращает символ, десятичный код которого равен X

Пример: Записать на QBASIC следующее математическое выражение:

Обычная запись

$$2x^2 + \frac{y + \sin(x)}{3 - z\sqrt{a + 4b}}$$

Запись на QBASIC

$$2*X^2 + (Y + SIN(X)) / (3 - Z * SQR(A + 4 * B))$$

При записи формул на QBasic необходимо учитывать следующие правила:

1. Порядок действий при вычислении арифметических выражений определяется приоритетом операции:
 - вычисляются выражения внутри скобок;
 - вычисляются значения стандартных функций;
 - производится операция возведения в степень;
 - производятся операции умножения/деления;
 - производятся операции сложения/вычитания.
2. Операции одного ранга (приоритета) проводятся слева направо.
3. Нельзя опускать знаки операций, например символ умножения "*" или ставить два знака подряд.
4. Количество открывающих скобок должно быть равно количеству закрывающих.

Упражнения для самостоятельной работы:

1. Какие из следующих последовательностей символов являются идентификаторами, а какие нет.

- а) X б) X1 в) X^T г) X1X2 д) AB е) ABCD
ж) SIN з) SIN (X) и) A-1 к) 2A л) MAX15

2. Записать по правилам Бейсика следующие выражения:

а) $\frac{X^3 + \sqrt{(A + B)^4}}{1 - B^2 + C^3}$ б) $1 + X + X^2$ в) $1 + |X| + |1 + X|$ г) $\frac{A(SINX^2 + Y)}{(C - D)^3}$
д) $\frac{A+B-1,7}{E+F+0,5}$ е) $\frac{1,2-9,8X}{1-Y(54,264-X)}$ ж) $\frac{X^2 + Y^2 + 100}{(2X+50,2)}$

3. Переписать следующие выражения, записанные по правилам Бейсика, в традиционной математической форме:

- а) $A + B / (C + D) - (A + B) / C + D$ б) $A * B / (C + D) - (C - D) / B * (A + B)$
в) $1 + SQR(COS(X + Y) / 2)$ г) $2.56 + ABS(X^2 - Y^6)$
д) $INT(X * 5.234 - A * (X + Y))$

3. Операторы языка QBasic.

3.1. Общие понятия об операторах языка.

Любая программа на языке QBasic состоит из операторов. Каждый оператор записывается строго определенным образом.

Операторы условно делятся на две группы: исполняемые и неисполнимые.

Исполнимые операторы- это инструкция программы, которая сообщает транслятору языка, что делать при выполнении программы (например: передать управление, вывести результат на экран и т.д.).

Неисполнимый оператор- оператор, который не вызывает никаких программных действий, а является указанием транслятору(например: указание о комментариях в программе, о формировании блока данных и т.д.).

По своему функциональному назначению операторы делятся на: общесистемные операторы, операторы присваивания и ввода данных, операторы вывода данных, операторы структуры программы и подпрограмм и операторы графического режима.

Формат оператора в общем случае имеет вид:

[n] <ключевое слово> <тело оператора>

где:

n – номер строки(метка), целое число, обычно с шагом 10 т.е. 10,20,30,..(для QBasic указывать не обязательно);

ключевое слово – служебное слово, используемое для команд (**REM,GOTO, PRINT** и т.д.);

тело оператора- параметры, списки, аргументы, имена, метки, выражения, команды;

[] - запись в квадратных скобках означает не обязательные параметры в данной конструкции оператора;

< > - запись в угловых кавычках означает, место, где должен находиться элемент указанный внутри этих кавычек.

Максимально допустимое число символов в строке программы, занимаемой одним оператором равно 255. Допускается запись нескольких операторов в одной строке, разделяемых символом ":" (двоеточие).

Пример записи программы на QBasic:

10 CLS

20 REM Программа

30 COLOR 7, 1

40 Y1 = 2+SIN(2)

50 PRINT "Ввод данных": PRINT Y1

60 END

3.2. Общесистемные операторы языка.

Оператор очистки экрана CLS. QBasic не очищает экран автоматически при запуске новой программы, поэтому рекомендуется начинать программу с оператора **CLS**.

Оператор REM. Позволяет включать в текст программы комментарии, поясняя тот или иной её фрагмент.

Формат оператора:

[n] REM [текст]

где:

n – номер строки;

REM – ремарки;

текст – текст комментарий, любая последовательность символов.

Знак " ' " (апострофа) также начинает строку комментарий.

Пример:

10 CLS

20 REM Моя первая программа

30 ' Начало программы

Оператор END. Это последний оператор в программе, при его выполнении интерпретатор QBasic закрывает все открытые файлы и прекращает выполнение программы.

Оператор STOP. Прерывает выполнение программы и при следующем запуске начинает выполнение с оператора следующего за **STOP** (используется при отладке программ).

Оператор GOTO. Предназначен для безусловного перехода на программную строку с указанным номером (меткой).

Формат оператора:

[n] GOTO <n>

где:

n – номер строки;

GOTO – перейти;

n – номер строки перехода(метка).

Оператор COLOR. Устанавливает цвет символов текста (линий в графическом режиме) и цвет фона экрана.

Формат оператора:

[n] COLOR <C>, <F>

где:

n – номер строки;

COLOR – цвет;

C – номер цвета символов(целое число от 1 до 7);

F – номер цвета фона экрана (целое число от 1 до 7).

Номер цвета символов и фона выбирается из списка:

0 – черный

4 – красный

1 – синий

5 – малиновый

2 – зеленый

6 – коричневый

3 – голубой

7 - белый

Использование оператора **CLS** после оператора **COLOR** позволяет очистить экран и "зализть" весь экран цветом фона.

Пример:

10 CLS

20 COLOR 7 , 2

30 CLS

40 PRINT "Моя программа"

50 END

В результате на экране будет напечатан текст "Моя программа" белым цветом на зеленом фоне экрана.

3.3. Оператор присваивания.

Оператор присваивания позволяет присвоить значения переменной или вычисленного значения, стоящего справа арифметического выражения.

Формат оператора:

[n] [LET] <X> = <A>

где:

n – номер строки;

LET – пусть(не обязательный элемент);

X – имя переменной(длина до 40 символов);

A – константа , арифметическое выражение или текстовая величина.

Пример:

```
10 CLS  
20 A = 3  
30 B = 7  
40 C = A+B  
50 D$ = "ЭВМ"  
60 PRINT C, D$  
70 END
```

Работа оператора присваивания: выполняя оператор ЭВМ вычисляет значение выражения стоящего справа от знака "="(равно), а полученный результат заносит в ячейку памяти имя которой указано слева от знака "="(равно) т.е. символ "="(равно) означает в данном случае не факт равенства, а требование выполнить указанные действия(операции).

Если **X** – целая, а значение **A**- вещественная величина, то результат операции округляется до ближайшего целого.

Допустим оператор вида **X=X+A** в этом случае символ **X** справа от знака "=" рассматривается как предшествующее значение, а тот же символ слева от знака "=" как последующее значение переменной **X**.

3.4. Оператор ввода данных.

Оператор ввода INPUT. Оператор служит для ввода исходных данных с клавиатуры при выполнении программы.

Формат оператора:

[n] INPUT ["подсказка"] ; < список имен переменных>

где:

n – номер строки;

INPUT – ввод;

подсказка - текстовая константа (для пояснения);

список – имя или имена переменных(массива);

Пример:

```
20 INPUT "Введите X и Y"; X, Y
```

Данный оператор выведет на экран сообщение:

Введите X и Y ?

и будет ожидать ввода с клавиатуры значений для **X** и **Y**. В ответ необходимо, через запятую ввести числовые значения для переменных и нажать **Enter**- работа программы будет продолжена.

Оператор ввода блока данных. Операторы **READ** и **DATA** используются для организации блока данных, который считывается во время выполнения программы.

Формат оператора:

[n] READ X₁,X₂,X₃,...,X_n

[n] DATA C₁,C₂,C₃,...,C_n

где:

n – номер строки;

READ – читать;

DATA – данные;

X₁,X₂,X₃,...,X_n – имена переменных;

C₁,C₂,C₃,...,C_n – значения переменных.

Замечание! Оператор **DATA** может стоять в любом месте программы.

Пример:

```
10 CLS  
20 READ A, B , C$  
30 DATA 2 , 4.5, "ЭВМ"
```

В результате выполнения оператора в строке 20 значения переменных A,B,C\$ будут прочитаны из оператора строки 30 и получат следующие значения:

A = 2
B = 4.5
C\$ = "ЭВМ"

Оператор RESTORE (восстановить) - позволяет повторно считать данные очередному оператору **READ** , следующему после оператора **RESTORE** т.е. повторно выбрать значения из списка .

Пример:

```
10 CLS  
20 READ A, B , C$  
30 DATA 2 , 4.5, "ЭВМ"  
40 RESTORE  
50 READ F, M , K$
```

В результате переменные получат значения:

A = 2 F = 2
B = 4.5 M = 4.5
C\$ = "ЭВМ" K\$ = "ЭВМ"

3.5. Оператор вывода данных.

Оператор **PRINT** служит для вывода данных на экран монитора (**LPRINT** – на принтер) в процессе выполнения программы.

Формат оператора:

[n] PRINT ["подсказка"] ; X₁,X₂,X₃,.....,X_n

где:

n – номер строки;

PRINT – печатать;

подсказка - текстовая константа (для пояснения);

X₁,X₂,X₃,.....,X_n – список имен переменных, значения которых выводятся на экран(печатать) или текст в кавычках.

Особенности записи и работы оператора PRINT:

1. Запись оператора **PRINT** без списка выводит на экран пустую строку(строку пробелов).
2. Если в качестве разделителя списка имен переменных используется "," (запятая), то в этом случае экран разбивается на пять колонок по 14 позиций в каждой. Значения каждой величины выводится с начала очередной колонки строки.
3. Если в качестве разделителя списка имен переменных используется ";" (точка с запятой), то в этом случае числовые значения выводятся через 2 пробела, а текстовые величины через 1 пробел вплотную друг к другу.
4. Если в конце оператора **PRINT** отсутствует знак ","(запятая) или ";"(точка с запятой), то следующий за ним оператор **PRINT** выводит значения величин с начала новой строки.
5. Номера строк экрана от 1 до 49, а номера позиций символов в строке от 1 до 80.

Пример1:

```
10 CLS  
20 B = -10  
30 H = 7  
40 D = 4.5
```

50 PRINT B, H, D, "Список"

60 END

На экране получим такую строку:

1	16	30	43	
-10	7	4.5	Список	

Пример2:

10 CLS

20 B = -10

30 H = 7

40 D = 4.5

50 PRINT B; H; D; "Список"

60 END

На экране получим такую строку:

1	6	9	12	
-10	7	4.5	Список	

Для управления выводом в операторе **PRINT** используется функция **TAB(m)** (табуляция). Которая служит для вывода значения величин в определенных функцией табуляции позиций строки экрана.

Формат функции:

TAB(m)

где:

TAB- табуляция;

m – номер позиции строки число от 1 до 80.

Работа оператора **PRINT** с функцией **TAB(m)**:

Пример3:

10 CLS

20 PRINT TAB(12); "Наименование"

Данный оператор выведет текст "**Наименование**" с 12-й позиции строки.

Для управления положением курсора (определяющего место вывода информации на экране) служит оператор **LOCATE** (размещать).

Формат оператора:

[n] LOCATE < X, Y >

где:

n – номер строки;

LOCATE – размещать;

X – номер строки от 1 до 40;

Y – номер позиции в строке от 1 до 80.

Внимание ! Действие оператора **LOCATE** управляет выводом для следующего за ним оператора **PRINT**.

Пример4:

10 LOCATE 5 ,20

20 PRINT "Компьютер"

На экране слово "**Компьютер**" будет выведено в 5-ой строке с 20-ой позиции строки.

3.6 Операторы условного перехода.

Линейные алгоритмы и соответствующие им программы редко встречаются в практике решения реальных задач на компьютере. Чаще всего порядок выполнения действий

(ход алгоритмического процесса) зависит от определенных условий и для обеспечения подобных алгоритмов в языке QBasic имеется оператор условного перехода.

Оператор условного перехода обеспечивает в программе проверку заданных условий и организацию разветвляющихся алгоритмов в зависимости от результатов проверки этих условий.

Существует два варианта оператора условного перехода - полная форма записи оператора и сокращенная форма.

Формат оператора:

Вариант 1 (полная форма).

[n] IF <условие> THEN <P1> ELSE <P2>

где:

n – номер оператора;

IF – ключевое слово (ЕСЛИ);

THEN - ключевое слово (ТО);

P1 – оператор или группа операторов;

ELSE - ключевое слово (ИНАЧЕ);

P2 – оператор или группа операторов;

Вариант 2 (сокращенная форма).

[n] IF <условие> THEN <P1>

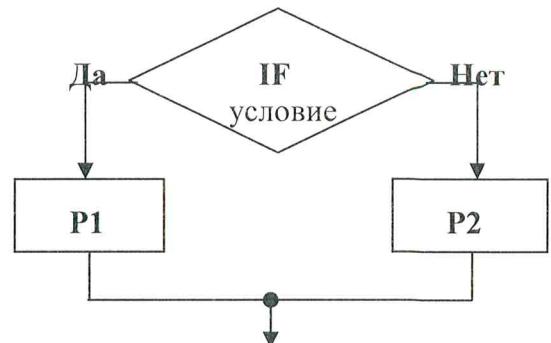


Рис. 1. Вариант 1

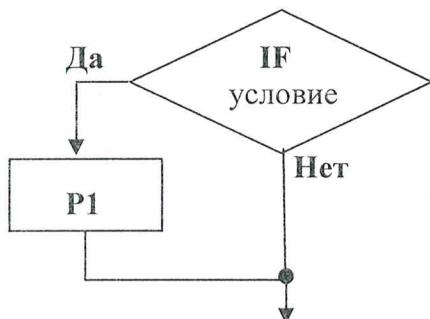


Рис. 2. Вариант 2

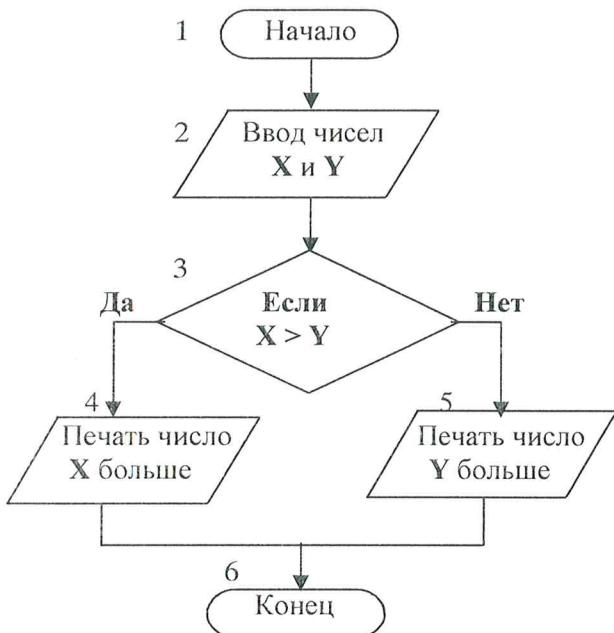
Работа каждого варианта условного оператора определяется соответствующей схемой представленной на рисунках 1 и 2.

В первую очередь проверяется условие, записанное после ключевого слова IF (ЕСЛИ). Если условие является верным (выполняется), то выполняется оператор или группа операторов (P1), расположенных после ключевого слова THEN (ТО). Если условие не является верным (не выполняется), то выполняется оператор или группа операторов (P2), расположенных после ключевого слова ELSE (ЕСЛИ) или в сокращенном варианте, управление передается следующему по порядку оператору программы.

Пример 1: Составить программу которая определяет наибольшее из двух заданных с клавиатуры чисел X и Y.

Решение задачи:

1. Составим блок-схему программы.
2. Написать программу на языке QBasic.



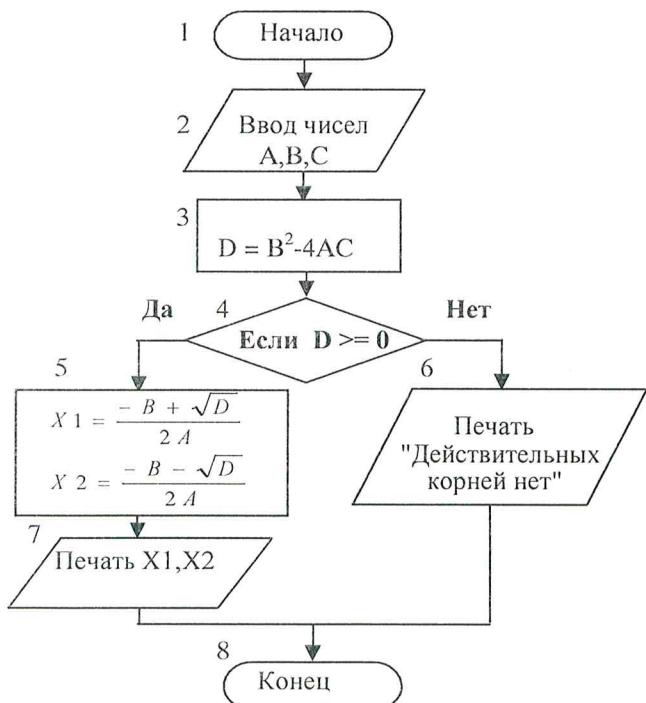
```

10 CLS
20 INPUT "Введите числа X и Y"; X,Y
30 IF X>Y THEN PRINT "Число X
больше";X ELSE PRINT "Число Y
больше";Y
40 END
    
```

Пример2: Даны действительные числа A,B,C. Составить программу которая определяет имеет ли уравнение $Ax^2 + Bx + C = 0$ действительные корни. Если действительные корни имеются, то найти их. В противном случае ответом должно служить сообщение, что действительных корней нет.

Решение задачи:

1. Составим блок-схему программы.
2. Написать программу на языке QBasic.



```

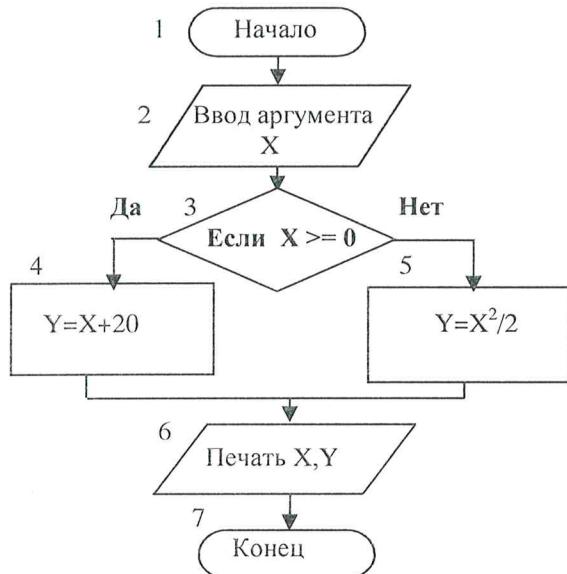
10 CLS
20 INPUT "Введите числа A,B,C"; A,B,C
30 D=B^2-4*A*C
40 IF D>=0 THEN GOTO 50 ELSE GOTO 100
100
50 X1=(-B+SQR(D))/(2*A)
60 X2=(-B-SQR(D))/(2*A)
70 PRINT "X1=";X1
80 PRINT "X2=";X2
90 GOTO 110
100 PRINT "Действительных корней нет"
110 END
    
```

Пример 3: Использование сокращенной формы записи оператора перехода. Для заданных с клавиатуры значений аргумента X составить программу вычислений значений функции Y, определяемых по следующему правилу:

$$Y = \begin{cases} X + 20, & \text{если } X \geq 0 \\ \frac{X^2}{2}, & \text{если } X < 0 \end{cases}$$

Решение задачи:

1. Составим блок-схему программы.
2. Написать программу на языке QBasic.



```

10 CLS
20 INPUT "Введите значение X ";X
30 IF X>=0 THEN Y=X+20: GOTO 50
40 X=X^2/2
50 PRINT "X=";X
60 PRINT "Y=";Y
70 END
  
```

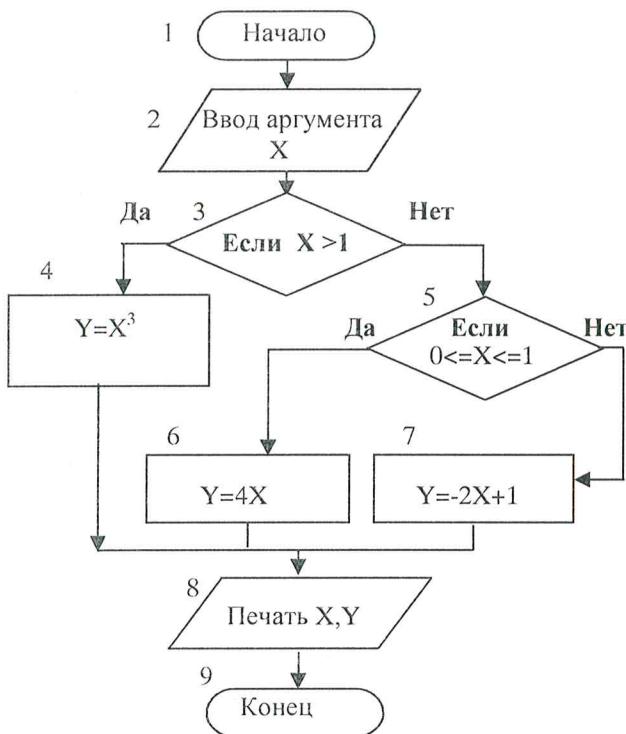
Пример 4: Сложные условия (вложенные операторы перехода). Для заданных с клавиатуры значений аргумента X составить программу вычислений значений функции Y, определяемых по следующему правилу:

$$Y = \begin{cases} X^3, & \text{если } X > 1 \\ 4X, & \text{если } 0 \leq X \leq 1 \\ -2X + 1, & \text{если } X < 0 \end{cases}$$

Решение задачи:

В отличии от других версий в QBasic допускается запись сложного условия в виде неравенства $0 \leq X \leq 1$ именно в такой форме без использования ключевой связки AND.

1. Составим блок-схему программы.
2. Написать программу на языке QBasic.



```

10 CLS
20 INPUT "Введите значение X ";X
30 IF X>1 THEN Y=X^3
40 IF 0<=X<=1 THEN Y=4*X ELSE
Y=-2*X+1
50 PRINT "X=";X
60 PRINT "Y=";Y
70 END
  
```

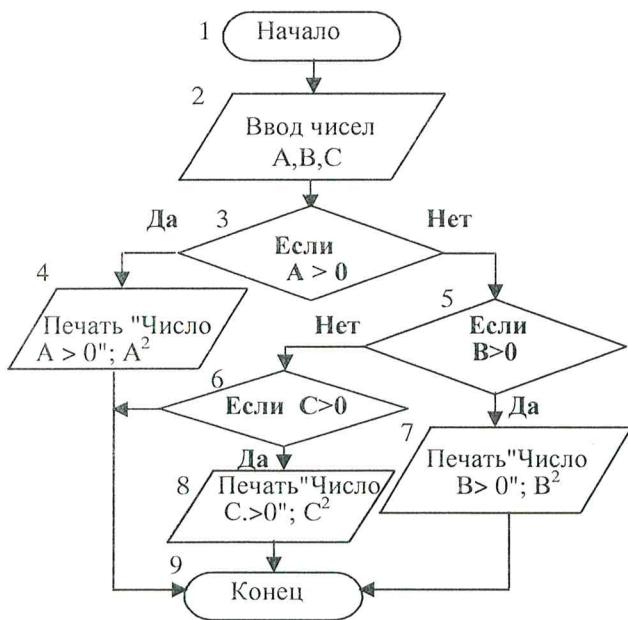
→ Третье условие мы не рассматриваем потому, что если не выполняются первые два, то очевидно, X находится в третьем интервале, поскольку запись в условиях охватывает всю числовую ось.

Задания на дом:

Пример1: С клавиатуры вводятся три числа A,B,C. Составить программу, которая возводит в квадрат те из них, которые не отрицательны.

Решение задачи:

1. Составим блок-схему программы.
2. Написать программу на языке QBasic.



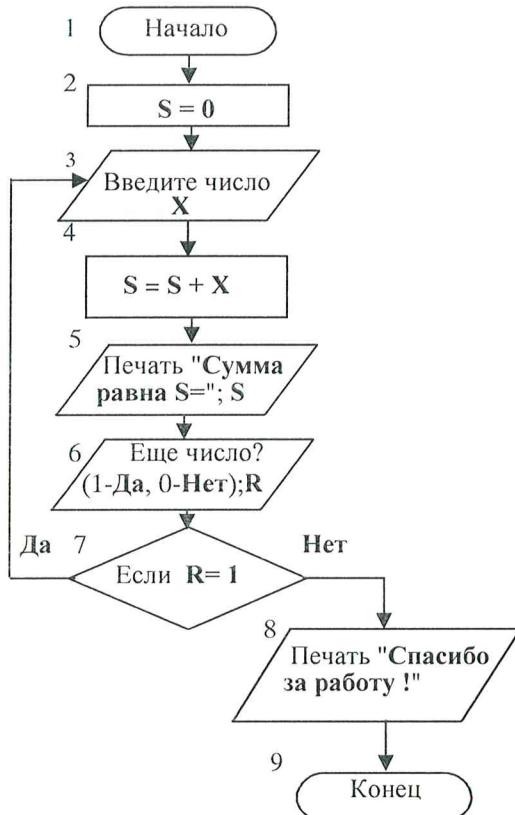
```

10 CLS
20 INPUT "Введите числа A,B,C";A,B,C
30 IF A>0 THEN PRINT "Число A
больше";A*A
40 IF B>0 THEN PRINT "Число B
больше";B*B
50 IF C>0 THEN PRINT "Число C
больше";C*C
60 END
  
```

Пример 2: Составить программу-сумматор, которая по мере ввода чисел с клавиатуры вычисляла бы их сумму, пока мы не захотели бы прекратить этот процесс.

Решение задачи:

1. Составим блок-схему программы.
2. Написать программу на языке QBasic.



```

10 CLS
20 S=0
30 INPUT "Введите число ";X
40 S=S+X
50 PRINT "Сумма равна S =";S
60 INPUT "Еще число?(1-Да, 0-Нет)";R
70 IF R=1 THEN GOTO 30 ELSE PRINT
"Спасибо за работу!"
80 END
    
```

3.7. Операторы цикла

Для выполнения циклических алгоритмов, в которых изменяется только какая-либо величина (*параметр*), а неизменяемая часть (действия, команды или серия команд) называемых телом цикла выполняется многократно, в QBasic используется оператор цикла.

Формат оператора:

[n] FOR p=p₁ TO p₂ STEP h → заголовок цикла
<последовательность операторов> → тело цикла
[m] NEXT p → конец цикла

где:

FOR(для), **TO**(до), **STEP**(шаг), **NEXT**(следующий) – ключевые слова;

p – управляющая переменная цикла(параметр);

p₁, p₂ – начальное и конечное значение переменной цикла;

h - шаг приращения параметра цикла;

[n], [m] – порядковые номера операторов начала и конца цикла.

Работа оператора цикла:

При выполнении оператора **FOR** сравнивается значение начального и конечного значения управляющей переменной(параметра), если начальное значение меньше, то выполняются операторы, образующие тело цикла. При выполнении оператора **NEXT** значение параметра цикла увеличивается на величину шага h т.е. p = p₁ + h и управление передается в начало цикла т.е. оператору **FOR**, где снова происходит сравнение нового значения управляющего параметра с конечным значением p₂. Если в результате сравнения окажется, что значение управляющей переменной больше конечного значения, то цикл на этом заканчивается и управление передается на оператор, следующий за оператором **NEXT**. В противном случае выполнение цикла повторяется снова см. Рис1.

Примечание Если шаг цикла равен 1, то ключевое слово **STEP** и арифметическое выражение задающее значение шага h , можно опустить.

Пример 1: Составить программу вычисляющую сумму десяти первых натуральных чисел.

Решение задачи:

1. Составим блок-схему программы.
2. Написать программу на языке QBasic.

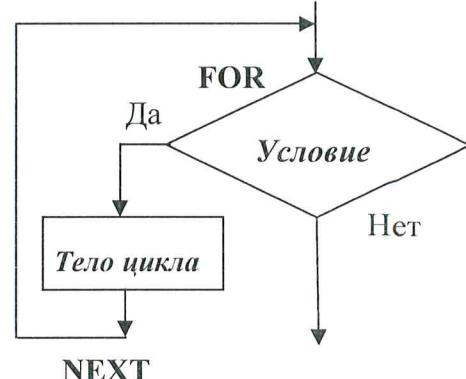
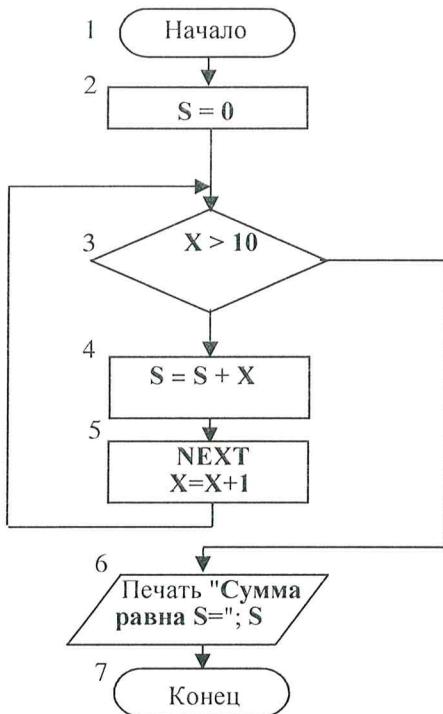


Рис.1



```

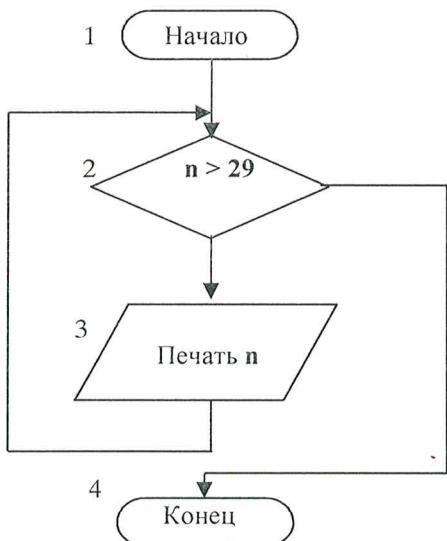
10 CLS
20 S=0
30 FOR X=1 TO 10
40 S=S+X
50 NEXT X
60 PRINT "Сумма равна S=";S
70 END

```

Пример 2. Составить программу, которая выводит на экран дисплея в строку первые 15 нечетных натуральных чисел.

Решение задачи:

1. Составим блок-схему программы.
2. Написать программу на языке QBasic.

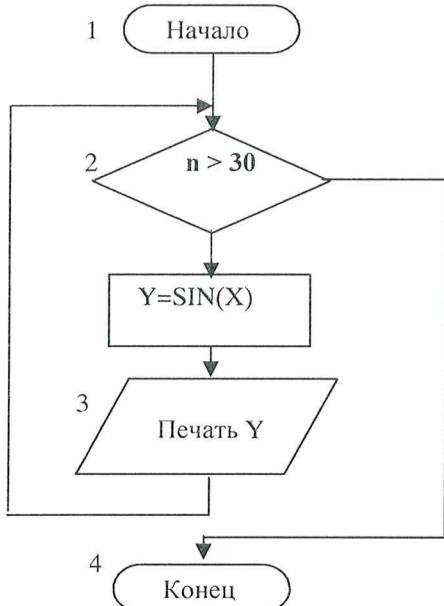


10 CLS
20 FOR n=1 TO 29 STEP 2
30 PRINT n;
40 NEXT n
50 END

Пример 3. Составить программу вычисления и печати значений функции $y = \sin(x)$ в интервале $[-30^0; +30^0]$ с шагом 5^0 (табулирование функции).

Решение задачи:

1. Составим блок-схему программы.
2. Написать программу на языке QBasic.



```

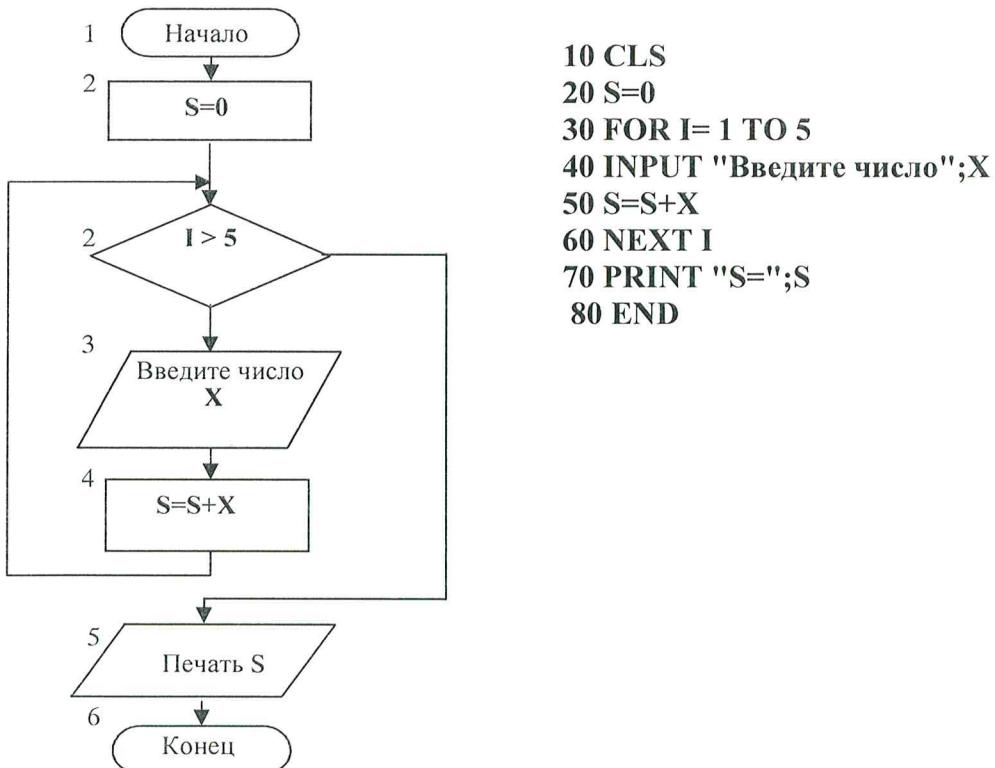
10 CLS
20 FOR X=-30 TO 30 STEP 5
30 Y=SIN(X*3.14/180)
40 PRINT "Y=";Y
50 NEXT X
60 END
  
```

Примечание: Переход от градусов к радианам по формуле: $\alpha(\text{радианов}) = \frac{\pi}{180} \alpha^0$

Пример 4. Составить программу нахождения суммы любых 5 чисел, введенных с клавиатуры.

Решение задачи:

1. Составим блок-схему программы.
2. Написать программу на языке QBasic.



```

10 CLS
20 S=0
30 FOR I= 1 TO 5
40 INPUT "Введите число";X
50 S=S+X
60 NEXT I
70 PRINT "S=";S
80 END
  
```

Вложенные циклы.

Пример 5. Составить программу "бегущей" вдоль строки дисплея (длина строки 80 символов).

Решение задачи:

1. Составим блок-схему программы.
2. Написать программу на языке QBasic.



Пояснение к программе:

1 вариант

Используем 2 цикла- один по **I** печатает звездочку, а 2 по **X** стирает звездочку в позиции табуляции в строке

2 вариант

Скорость выполнения программы велика, поэтому мы не видим как звездочка "бежит" по экрану. Для замедления процесса введем фиктивный цикл по **P** в строках 55 и 56 ,величина конечного интервала подбирается экспериментально.

3 вариант

В этом варианте "бегущая" звездочка окрашивается в разные цвета. Добавим оператор датчика случайных чисел оператор 35 который будет случайным образом вырабатывать случайные числа в интервале от 1 до 10. Известно ,что функция **RND(1)** вырабатывает случайные числа в интервале от 0 до 1 ,чтобы получать случайные числа в интервале от 1 до 10, умножим их на 10 и будем брать целую часть от произведения (функция **INT(X)**), а добавленная 1 позволят получать числа именно от 0 до 10. Полученные значения переменной **T** мы используем в операторе изменения цвета звездочки 36 **COLOR(T)**.

1 Вариант

```
10 CLS  
20 FOR I=1 TO 80  
30 LOCATE(15)  
40 PRINT TAB(I);"*"  
50 FOR X=1 TO 80  
60 LOCATE(15)  
70 PRINT TAB(X);" "  
80 NEXT X  
90 NEXT I  
100 END
```

2 Вариант

```
10 CLS  
20 FOR I=1 TO 80  
30 LOCATE(15)  
40 PRINT TAB(I);"*"  
50 FOR X=1 TO 80  
55 FOR P=1 TO 200  
56 NEXT P  
60 LOCATE(15)  
70 PRINT TAB(X);" "  
80 NEXT X  
90 NEXT I  
100 END
```

3 Вариант

```
10 CLS  
20 FOR I=1 TO 80  
30 LOCATE(15)  
35 T=INT(RND(1)*10)+1  
36 COLOR(T)  
40 PRINT TAB(I);"*"  
50 FOR X=1 TO 80  
55 FOR P=1 TO 500  
56 NEXT P  
60 LOCATE(15)  
70 PRINT TAB(X);" "  
80 NEXT X  
90 NEXT I  
100 END
```

Список используемых источников :

1. Угринович Н. "Информатика и информационные технологии", "Бином", 2014 г.
2. Михеева Е.В., Титова О.И. "Информатика", "Академия", 2014 г.
3. Шифрин Ю. "Основы компьютерной технологии", М.,"ABC", 2013 г.
4. Федоренко Ю."Алгоритмы и программы на QBasic", "Питер", 2015 г.
5. Михайлов В.Ю., Степанков В.М."Современный Бейсик для IBM PC", М."Академия",2012 г.
6. Сафонов И. "Бейсик в примерах и задачах", "Питер", 2015 г.
7. Бобровский С. "Программирование на QBasic", М."ABC", 2014 г.

Содержание

Введение	3
1. Языки парограммирования.	4
2 .Общие сведения о языке программирования QBasic.	4
3. Операторы языка QBasic.	9
Список используемых источников	23