

Алгоритмизация и программирование

Литература по программированию

- Вострикова З.П. и др. Программирование на языке Бейсик для персональных ЭВМ. 1993 г.
- Мельникова О.И., Бонюшкина А.Ю. Начала программирования на языке QBasic. 2000 г.
- Михайлов В.Ю., Степанников В.М. Современный Бейсик для IBM PC: среда, язык программирования. 1993 г.

Понятие алгоритма

- Алгоритм – это строго определенная последовательность действий, необходимых для решения данной задачи.
- Алгоритм вычисления площади квадрата
- Очистить экран
- Ввести длину стороны квадрата
- Вычислить площадь
- Вывести результат

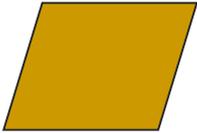
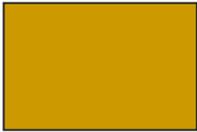
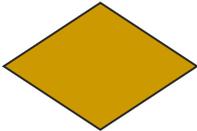
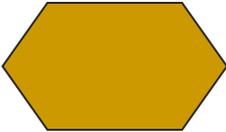
Свойства алгоритма

- Дискретность – алгоритм разбит на шаги (действия)
- Определенность (детерминированность) – однозначность результата при заданных исходных данных
- Результативность (конечность) – всегда приводит к результату за конечное число шагов
- Массовость – алгоритм применим для решения других задач того же типа, что исходная

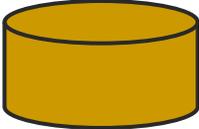
Способы описания алгоритма

- Словесно-формульное описание
- Графическое описание, блок-схема
- Описание с использованием псевдокода (на алгоритмическом языке)
- Запись на языке программирования

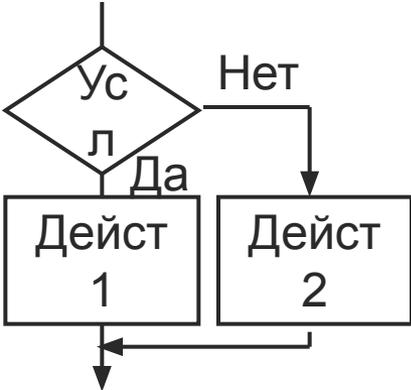
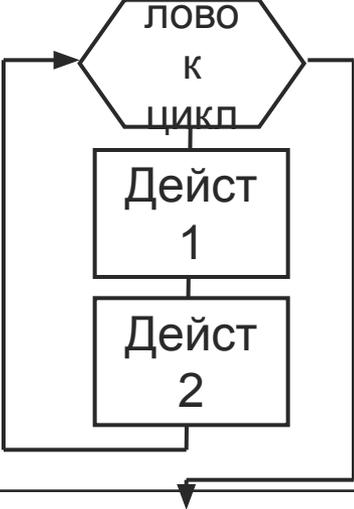
Основные графические символы блок-схем (ГОСТ 19.003-80)

	Блок «пуск, останов»	Начало-конец алгоритма
	Блок «ввод-вывод»	Ввод-вывод данных
	Блок «процесс»	Операции присваивания, объявления массивов
	Блок «решение»	Для организации разветвления в алгоритме (условие)
	Блок «модификация»	Для организации цикла

Основные графические символы блок-схем

	Блок «предопределенный процесс»	Обращение к подпрограмме
	Блок «соединитель»	Разрыв в блок-схеме Знак переноса на след. страницу
	Блок «магнитный барабан»	Запись на магнитный диск / чтение
	Блок «документ»	Ввод-вывод данных, носителем которых служит бумага

Основные типы алгоритмических структур

Линейная	Разветвленная	Циклическая
 <p>Последовательное выполнение действий</p>	 <p>Выбор действия по условию</p>	 <p>Неоднократное повторение одинаковых действий (каждый раз при новых значениях параметров)</p>

Язык программирования

- Язык программирования — формальная знаковая система, предназначенная для записи программ.
- Язык программирования - это набор символов (цифр, букв, специальных знаков) и система правил образования (синтаксис) и правил истолкования (семантика) конструкций из этих символов, с помощью которых описывается порядок выполнения алгоритма. Язык программирования имеет иерархическую структуру.
- Обычно в нем выделяют четыре уровня:
 - основные символы (алфавит);
 - ключевые слова;
 - выражения;
 - предложения (операторы)

Из истории языков программирования

Наиболее широко распространенным типом языков программирования высокого уровня являются процедурные языки. В таких языках широко используются управляющие конструкции (операторы), которые позволяют закодировать различные алгоритмические структуры (линейную, ветвление, цикл).

Одним из первых процедурных языков программирования был известный всем **Бейсик (Basic)**, созданный в **1964** году. В течение последующего времени Бейсик развивался, появлялись его различные версии (MSX-Basic, Бейсик-Агат, **QBasic** и др.). Другим широко распространенным языком программирования алгоритмического типа является **Pascal**. Basic (Бейсик) создавался в 60-х годах в качестве учебного языка и очень прост в изучении. По популярности занимает первое место в мире.

Введение в язык программирования QBasic

- Основными объектами обработки в языке Бейсик являются:
- Константы
- Переменные
- Функции
- Выражения
- Операторы

Константы

Ч И С Л О В Ы Е
К О Н С Т А Н Т Ы

1. К. целого типа – любое целое число
-32768 ÷ 32767
2. К. действительного типа
 - 2.1. Константы с фиксированной запятой (в естественной форме) **2.5** или **-259.167**
 - 2.2. Константы с плавающей запятой (в экспоненциальной форме) **1.7E-05** ($1.7 \cdot 10^{-5}$)
3. К. символьного типа – любая последовательность символов, заключенная в кавычки “08.09.2011г”

[Переменные]

Под переменной в языках программирования понимают программный объект (число, слово, часть слова, несколько слов, символы), имеющий имя и значение, которое может быть получено и изменено программой.

- **Переменная** - это имя физического участка в памяти, в котором в каждый момент времени может быть только одно значение.

Типы переменных

- Тип переменных соответствует типу констант.
- Имя переменной – это произвольный набор символов, который может содержать от 1 до 40 символов, причём **первый символ должен быть латинской буквой, а остальные – латинские буквы или цифры или допустимые символы.**
- Имя переменной может заканчиваться следующими символами:
 - % - целочисленная переменная
 - \$ - символьная переменная
 - # - числовая переменная с двойной точностью
- A=3: B\$="HELLO": C% = 10

Встроенные математические функции QBasic

Корень квадратный	SQR(X)
Модуль числа	ABS(X)
Синус	SIN(X)
Косинус	COS(X)
Тангенс	TAN(X)
Арктангенс	ATN(x)
Целая часть числа	INT(X)
Основание нат.логарифма в степени x	EXP(x)
Натуральный логарифм	LOG(X)

Арифметические операции

Операция	Обозначение	Пример	Результат
Сложение	+	2+5	7
Вычитание	-	10-8	2
Умножение	*	3*4	12
Деление		15/3	5
		15/4	3.75
Целочисленное деление	\	15\4	3
Возведение в степень	^	2^3	8
Остаток от деления	MOD	13 MOD 5	3

Выражения

- **Выражение** – это совокупность констант, переменных, функций, скобок и знаков арифметических операций.
- $Y = \text{SIN}(5 * \text{SQR}(x) / (x - 8)^{(1/3)})$



выражение

$$Y = \sin \frac{5\sqrt{x}}{\sqrt[3]{x-8}}$$

[Операторы]

- Оператор – специальное слово с параметрами, предназначено для выполнения какой-либо команды.
- Операторы ввода-вывода
- Операторы условного и безусловного перехода
- Операторы организации цикла
- Операторы работы с массивами
- Графические операторы
- Операторы организации файлов данных

Операторы ввода

Для задания значения переменной служит оператор присваивания. Он записывается так:

LET переменная = значение

или просто: переменная = значение

Пример:

LET a = 3

chislo1 = 15

[Операторы ввода]

- Оператор ввода с клавиатуры
INPUT

пример:

INPUT a

INPUT "Введите число a: ", a

Операторы ввода

- Оператор чтения блока данных

DATA перечень констант ч/з запятую

READ перечень переменных ч/з запятую

Пример:

DATA "ПРИВЕТ", 3, 0.5

READ A\$, C%, D

Операторы вывода

PRINT (вывод, печать) – оператор вывода.

Выводит информацию на экран.

LPRINT – оператор вывода. Выводит информацию на принтер.

пример:

```
PRINT "Привет!"
```

```
PRINT X
```

```
PRINT "Привет! " , " Меня зовут Саша."
```

(, - зонная печать)

```
PRINT "Площадь квадрата = "; s
```

(; - дает один пробел на экране)

Операторы вывода

PRINT USING – оператор вывода в заданном формате.

Выводит информацию на экран в соответствии с заданным шаблоном.

пример:

X =3.52: y = -25.61325

PRINT USING “#. # | +##.### ”; x, y

на экране:

3.5 | -25.613

[Операторы QBasic]

REM – оператор комментария.

Все что следует после этого оператора до конца строки игнорируется компилятором и предназначено исключительно для программиста.

пример:

REM Это комментарий

можно и так:

' Это тоже комментарий

[Операторы QBasic]

CLS - очистить экран.

Вся информация, которая была на экране, стирается.

пример:

CLS

[Операторы языка Qbasic]

END – оператор конца программы.

пример:

END

Запись алгоритма линейной структуры на языке QBasic

REM программа вычисления периметра и площади прямоугольника

CLS

INPUT "Введите a="; a

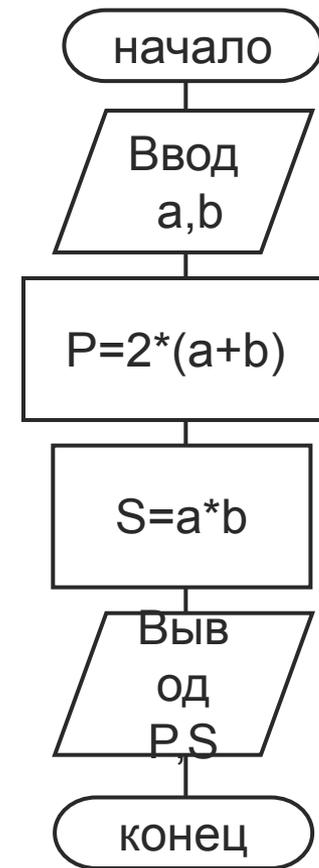
INPUT "Введите b="; b

p=2*(a+b)

s=a*b

PRINT "P="; p, "S="; s

END



Оператор условного перехода

- Оператор **IF** предназначен для управления ходом выполнения программы в зависимости от условия.
- **IF** выражение **THEN** дейст1 [**ELSE** дейст2]
- где **выражение** – любое числовое выражение
- **дейст** – оператор, либо последовательность операторов, разделенных двоеточием, либо номер строки

Если фраза ELSE отсутствует, выполняется следующий оператор

- **Пример:**
- `IF x>5 THEN Y=2*x ELSE Y=3*x`

Оператор безусловного перехода

- Оператор **GOTO** передает управление на строку с указанным номером
- Пример
- GOTO 10



Номер строки или метка

Операции отношения

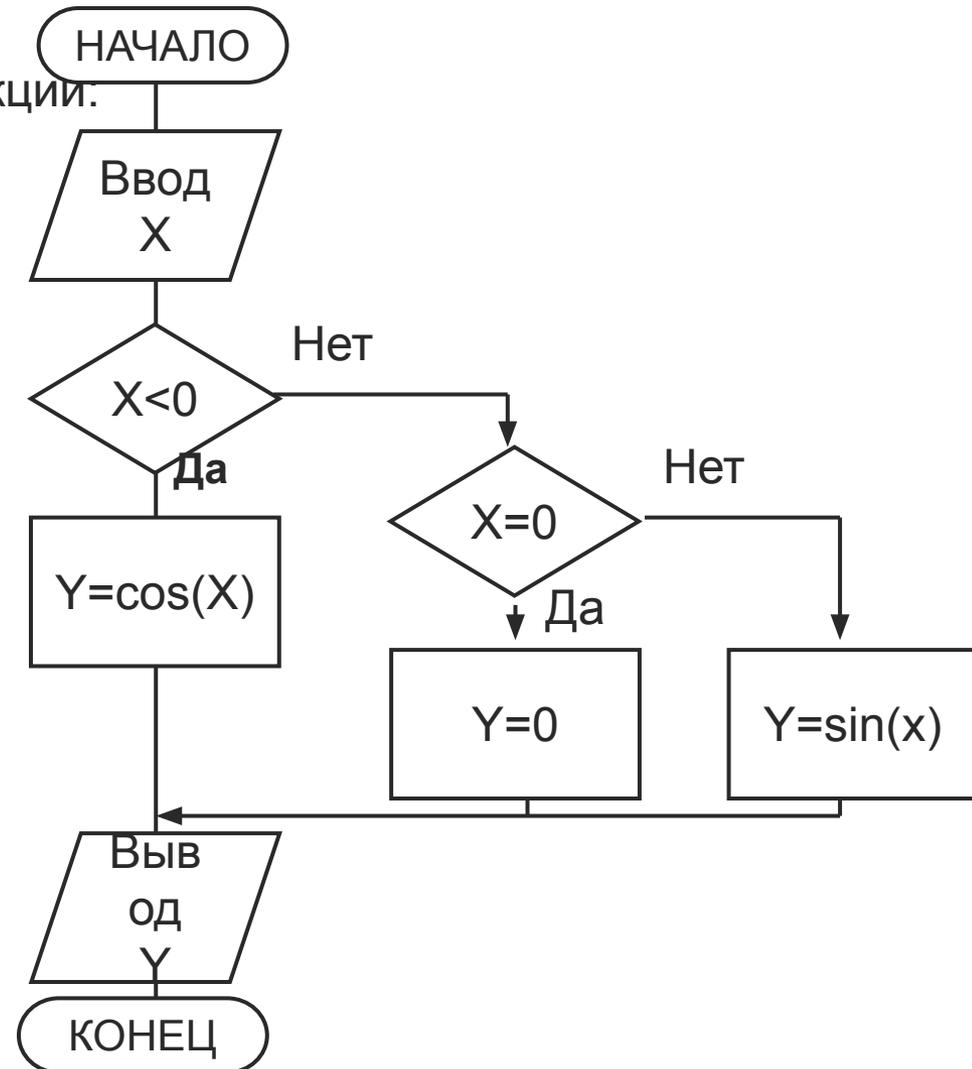
Значение операции	Знак операции	Выражение в Бейсике
Равенство	=	$x=y$
Неравенство	<>	$x<>y$
Меньше	<	$x<y$
Больше	>	$x>y$
Меньше или равно	<=	$x<=y$
Больше или равно	>=	$x>=y$

Запись алгоритма

разветвленной структуры на языке QBasic

Вычислить значение кусочной функции.

$$y = \begin{cases} \sin(x), & \text{если } x > 0 \\ \cos(x), & \text{если } x < 0 \\ 0, & \text{если } x = 0 \end{cases}$$



Программа разветвленной структуры

```
REM Вычисление кусочной функции
CLS
INPUT "x="; x
IF x<0 THEN y=cos(x): GOTO 10
IF x=0 THEN y=0 ELSE y=sin(x)
10 PRINT "y="; y
END
```

Оператор организации цикла с заданным числом повторений

- Оператор FOR-TO-NEXT

FOR переменная=нач.знач. **TO** кон.знач. [**STEP** шаг]

Операторы тела цикла

NEXT [переменная]

```
REM Табулирование функции  $y=\sin(x)$ 
```

```
DATA 0, 1, 0.1
```

```
READ xn, xk, h
```

```
PRINT " X | Y "
```

```
FOR X=xn TO xk STEP h
```

```
Y=sin(X)
```

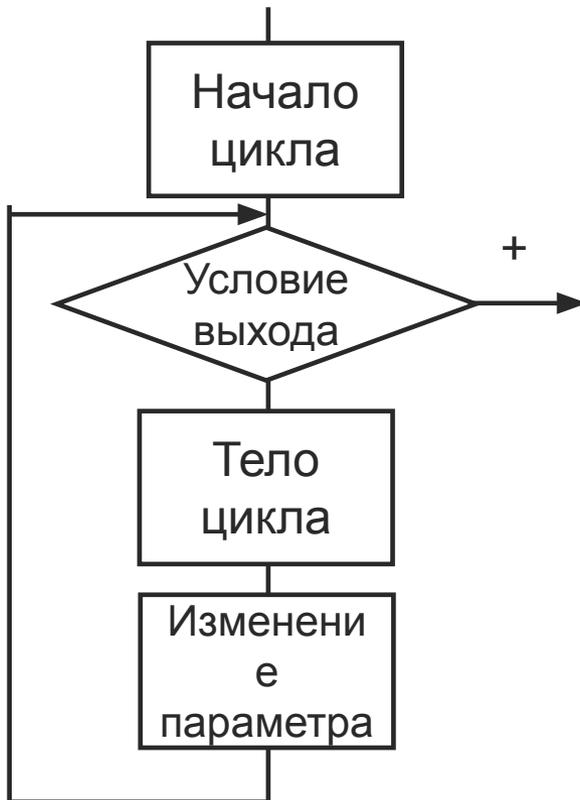
```
PRINT USING " #.# | ##.### "; X; Y
```

```
NEXT X
```

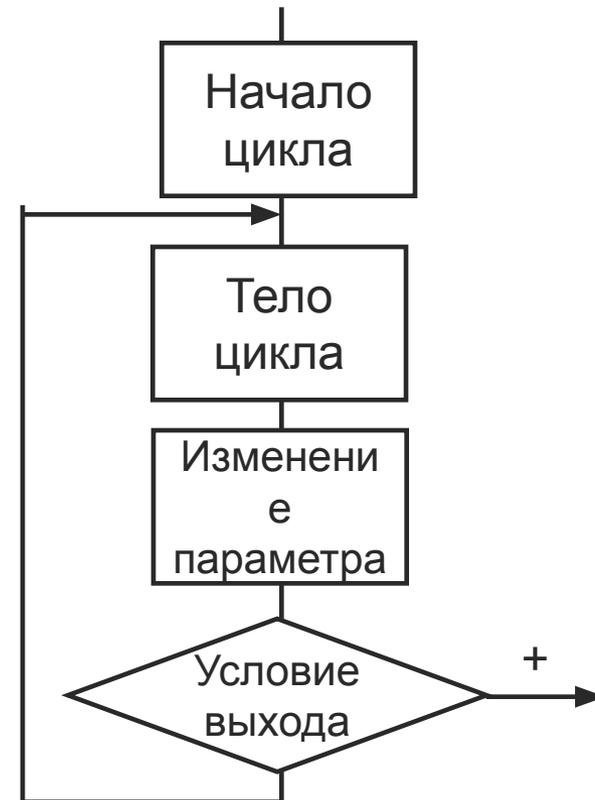
```
END
```

Организация циклов с неопределенным числом повторений

Цикл с предусловием



Цикл с постусловием



Сочетание операторов условного и безусловного перехода

Задача: сумма бесконечного ряда

- Вычислить сумму бесконечного ряда с заданной точностью (суммировать до тех пор, пока модуль текущего элемента ряда не станет меньше заданной точности) двумя способами:
- по общей формуле ряда;
- по рекуррентной формуле.

$$\sum_{k=0}^{\infty} (-1)^k \frac{(k+1)!}{(2k+1)!}$$

точность $\text{eps} = 0.001$

Проверка на сходимость ряда

- Проверка ряда на сходимость (при возрастании номера элемента сам элемент ряда должен уменьшаться)
- Подставим в формулу элемента ряда номера $k=0$, $k=1$, $k=2$.
- $K=0$
$$(-1)^0 \frac{(0+1)!}{(2*0+1)!} = 1$$
- $K=1$
$$(-1)^1 \frac{(1+1)!}{(2*1+1)!} = -\frac{2!}{3!} = -\frac{1}{3}$$
- $K=2$
$$(-1)^2 \frac{(2+1)!}{(2*2+1)!} = \frac{3!}{5!} = \frac{1*2*3}{1*2*3*4*5} = \frac{1}{20}$$
- Расчеты показали, что при увеличении номера элемента значение элемента ряда уменьшается. Ряд сходящийся.

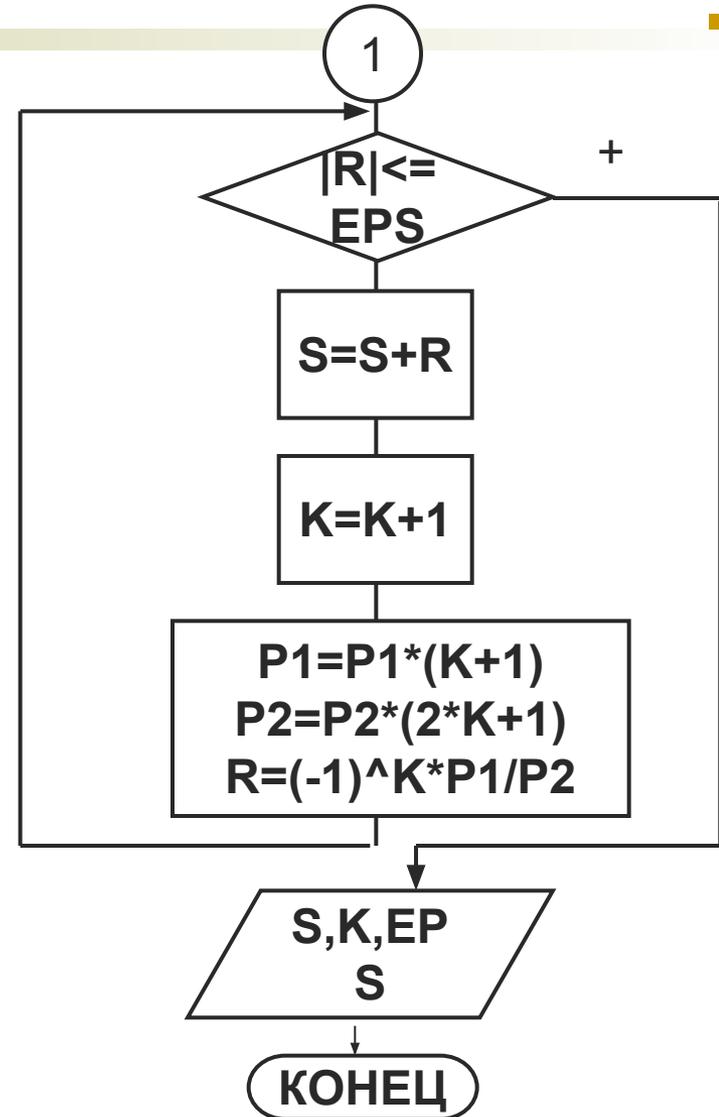
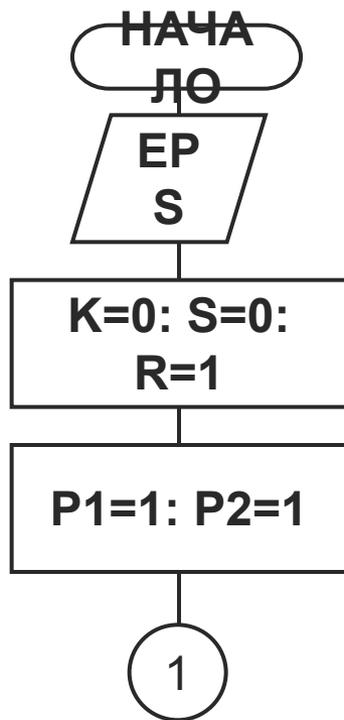
Расчет по общей формуле ряда

- **Постановка задачи**
- **Входные параметры**
- EPS – точность вычислений
- R – значение первого элемента ряда при начальном значении K – номере 1-го элемента ряда
- **Промежуточные параметры**
- P1 – переменная для накопления произведения $(k+1)!$
- P2 – переменная для накопления произведения $(2*k+1)!$
- K – номер итерации, совпадает с номером элемента ряда
- R – текущий (k-ый) элемент ряда
- **Выходные параметры**
- S – сумма элементов ряда

Программа: расчет по общей формуле ряда

- CLS
- INPUT "EPS="; EPS
- S=0
- K=0
- R=1
- P1=1
- P2=1
- 10 IF ABS(R)<=EPS THEN 20
- S=S+R
- K=K+1
- P1 = P1*(K+1)
- P2 = P2*(2*K+1)
- R=(-1)^K*P1 / P2
- GOTO 10
- 20 PRINT "S=";S, "EPS=";EPS, "K=";K
- END

Блок-схема: расчет по общей формуле ряда



Вывод рекуррентной формулы

- Общий вид рекуррентной формулы:

$$R_k = Q * R_{k-1}$$

где R_k – элемент ряда k -тый,

Q – коэффициент рекуррентной формулы.

$$Q = \frac{(-1)^k (k+1)!}{(2k+1)!} \frac{(2(k-1)+1)!}{(-1)^{k-1} ((k-1)+1)!} = \frac{(-1)^k (k+1)!}{(2k+1)!} \frac{(2k-1)!}{(-1)^{k-1} k!}$$

$$Q = (-1) \frac{1*2*...*k*(k+1)}{1*2*...*k} \frac{1*2*...*(2k-1)}{1*2*...*(2k-1)*2k*(2k+1)} = (-1) \frac{k+1}{2k*(2k+1)}$$

Программа: расчет по рекуррентной формуле

- CLS
- INPUT "EPS="; EPS
- S=0
- K=0
- R=1
- 10 IF ABS(R)<=EPS THEN 20
- S=S+R
- K=K+1
- $Q = - (K+1) / ((2*K)*(2*K+1))$
- R=R*Q
- GOTO 10
- 20 PRINT "S=";S, "EPS=";EPS, "K=";K
- END

Массивы

- **Массив** – это упорядоченная совокупность однородных элементов (переменных)
- Пример массива – любое пронумерованное множество элементов одного типа.
- A(1) = "Иванов"$
- A(2) = "Хорошева"$
- A(3) = "Печенкин"$
- A(4) = "Вахрушина"$
- A(5) = "Петров"$
- A(6) = "Васильков"$

Массивы

- Массив организуется с помощью индекса, который указывается после имени переменной в скобках. Индекс – это целое число, обозначающий порядковый номер элемента в массиве.

Классификация массивов

- **Одномерные и двумерные массивы**

Одномерный массив (вектор) – совокупность однородных элементов, имеющих один индекс.

Двумерный массив (матрица) – совокупность однородных элементов, имеющих два индекса (номер строки и номер столбца).

- **Статические и динамические массивы**

Статические массивы – массивы, размерность которых определяется константой

Динамические массивы – массивы, размерность которых определяется переменной

Оператор, объявляющий массив

- Оператор **DIM** объявляет массивы
- Задается имя массива, его тип и размерность. В памяти компьютера выделяется место для записи значений элементов массива и по умолчанию значения элементов массива определяются равными нулю.
- Пример:
- DIM A\$(6), B(10)