ПРЕЗЕНТАЦИЯ

ПРОГРАМНОЕ УПРАВЛЕНИЕ РАБОТОЙ КОМПЬЮТЕРА

ВЫПОЛНИЛ: КРАШАКОВ ДЕНИС
ПРОВЕРИЛА: ПОПОВА Е.А

ЧТО ТАКОЕ ПРОГРАММИРОВАНИЕ

- КТО ТАКИЕ ПРОГРАММИСТЫ

Назначение *ПРОГРАММИРОВАНИЯ* – разработка программ управления компьютером с целью решения различных информационных задач.

ПРОГРАММИСТЫ-специалисты, профессионально занимающиеся программированием. Люди, работающие на компьютерах, разделились на пользователей и программистов. В настоящее время пользователей больше чем программистов.

- Системные программисты занимаются разработкой системного программного обеспечения: операционных систем, утилит, и пр., а также систем программирования.
- *Прикладные программисты* создают прикладные программы: редакторы, табличные процессоры, игры, обучающие программы и многие другие.

ЧТО ТАКОЕ ЯЗЫК ПРОГРАММИРОВАНИЯ

- Для составления программ существуют разнообразные *языки программирования*.

ЯЗЫК ПРОГРАММИРОВАНИЯ – это фиксированная система обозначений для описания алгоритмов и структур данных.

Универсальные языки программирования – *ПАСКАЛЬ, БЕЙСИК, СИ, ФОРТРАН.*

ЧТО ТАКОЕ СИСТЕМА ПРОГРАММИРОВАНИЯ

- Для создания и исполнения на компьютере программы, написанной на языке программирования, используются *системы программирования.*
- *СИСТЕМА ПРОГРАМИРОВАНИЯ* – это программное обеспечение компьютера, предназначенное для разработки, отладки и исполнения программ, записанных на определенном языке программирования.

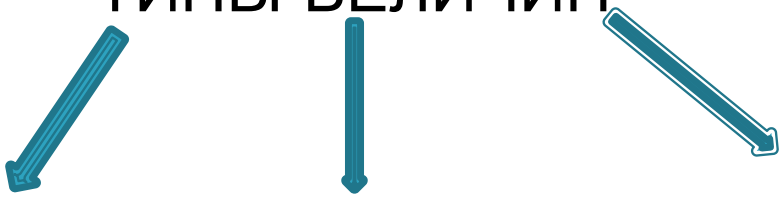
АЛГОРИТМЫ РАБОТЫ С ВЕЛИЧИНАМИ

- КОМПЬЮТЕР КАК ИСПОЛНИТЕЛЬ АЛГОРИТМОВ

ВЕЛИЧИНЫ: КОНСТАНТЫ И ПЕРЕМЕННЫЕ

- *ВЕЛИЧИНА* - отдельный информационный объект (число, символ, строка, таблица и пр.)
- *Всякая обрабатываемая программой величина занимает свое место в памяти компьютера.*
- *ЗНАЧЕНИЕ ВЕЛИЧИНЫ* – это информация, хранимая в этом поле памяти.

ТИПЫ ВЕЛИЧИН



ЧИСЛОВОЙ *СИМВОЛЬНЫЙ* *ЛОГИЧЕСКИЙ*

- *КОНСТАНТЫ* записываются в алгоритмах своими десятичными значениями, например: 23, 3.5, 34. Значение константы хранится в выделенной под нее ячейки памяти и остается неизменным в течении работы программы.
- *ПЕРЕМЕННЫЕ* в программировании, как и в математике, обозначаются символическими именами. Эти имена называют – *идентификаторами.* Примеры идентификаторов: A, X, B3, prim, r25 и т.п.

- Алгоритм решения любой задачи на компьютере составляется из следующих команд: *присваивания; ввода; вывода; обращения к вспомогательному алгоритму; цикла; ветвления.*

- *КОМАНДА ПРИСВАИВАНИЯ*

< переменная> := < выражение >

Значок «:=» читается «присвоить». Например:
Z := X+Y

Если слева от знака присваивания стоит числовая переменная, а справа математическое выражение, то такую команду называют *арифметической командой присваивания.*

- *КОМАНДА ВВОДА*

Значения переменных, являющихся исходными данными решаемой задачи, как правило, задаются вводом.

Команда ввода в описаниях алгоритмов будет выглядеть так:

ввод <список переменных>

Например: ввод A, B, C

Переменные величины получают конкретные значения в результате выполнения команды присваивания или команды ввода.

- *КОМАНДА ВЫВОДА*

Результаты решения задачи сообщаются компьютером пользователю путем выполнения *команды вывода.*

Команда вывода в алгоритмах будет записываться так:

вывод <список вывода>

Например: X1, X2

ЛИНЕЙНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ АЛГОРИТМЫ

- *ПРИСВАИВАНИЕ; СВОЙСТВА ПРИСВАИВАНИЯ*
- Переменная величина получает значение в результате присваивания.

ТРАССИРОВОЧНАЯ ТАБЛИЦА

команда	a	b
a:= 1	1	-
b:= 2 x a	1	2
a:= b	2	2
b:= a + b	2	4

процесс ее заполнения- *трассировочный*

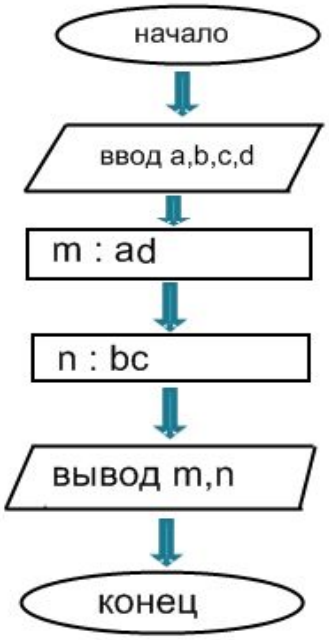
ОБМЕН ЗНАЧЕНИЯ ДВУХ ПЕРЕМЕННЫХ

Команда	X	Y	Z
Ввод X, Y	1	2	-
Z:= X	1	2	1
X:= Y	2	2	1
Y:= Z	2	1	1
вывод X, Y	2	1	1

ОПИСАНИЕ ЛИНЕЙНОГО ВЫЧЕСЛИТЕЛЬНОГО АЛГОРИТМА

- Алгоритм деления дробей

В матем. форме: $\frac{a}{b} : \frac{a}{b} = \frac{ab}{bc} = \frac{m}{n}$



алг Деление дробей
цел a,b,c,d,m,n

нач ввод a,b,c,d
m: ad
n: bc
вывод m,n
кон

ЗНАКОМСТВО С ЯЗЫКОМ ПАСКАЛЬ

- ВОЗНИКНОВЕНИЕ И НАЗНАЧЕНИЕ ПАСКАЛЯ
- *ПАСКАЛЬ* – это универсальный язык программирования, позволяющий решать самые разнообразные задачи обработки информации. Язык разработан 1971году, назван в честь Блеза Паскаля.
- Команду алгоритма, записанную на языке программирования, принято называть *оператором.*

```
алг   Деление дробей
цел a, b, c, d, m, n
нач
 ввод a, b, c, d
m: ad
n: bc
вывод m, n
кон
```

Алгоритм реш. дробей

```
Program Division
 var a, b, c, d, m, n: integer;
 begin
readln (a, b, c, d); {ввод}
 m: a*d; {числитель}
n: b*c; {знаменатель}
 write (m, n) {вывод}
 end
```

Программа на паскале

СТРУКТУРА ПРОГРАММЫ НА ПАСКАЛЕ

- Program <имя программы>
- В стандарте языка Паскаль существуют два числовых типа величин: *вещественный* и *целый. Integer – целый тип, real – вещественный тип. Begin – начало, end – конец.*
- *begin*

<операторы>

end

ОПЕРАТОРЫ ВВОДА, ВЫВОДА, ПРИСВАИВАНИЯ

- read (<список переменных>) или readln (< список переменных >) В конце нажимается клавиша <ВВОД> (<Enter>)
- Write(<список вывода>) или writeln (<список вывода>)

+ сложение

- вычитание

* умножение

/ деление

ПРАВИЛА ЗАПИСИ АРИФМЕТИЧЕСКИХ ВЫРАЖЕНИЙ

- Матем. выраж. $A^2 + B^2 - 12C$

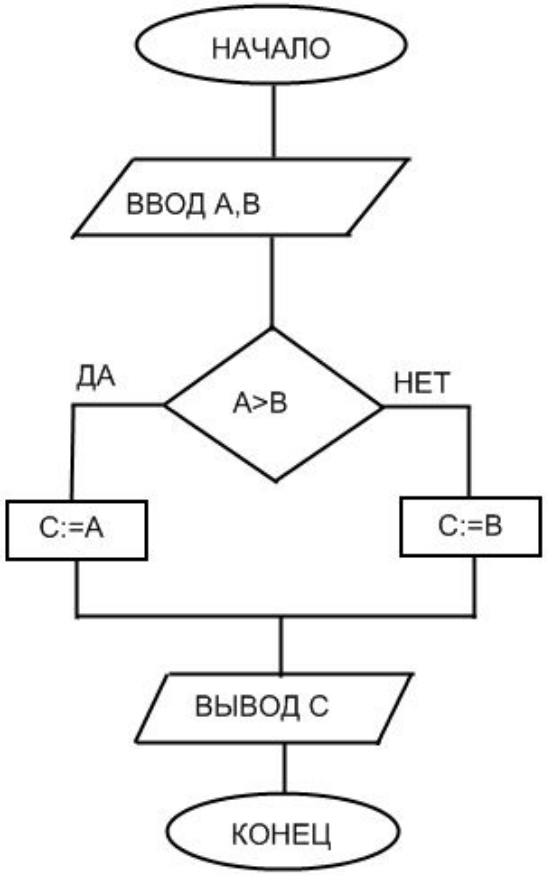
На Паскале A*A + B*B – 12*C или

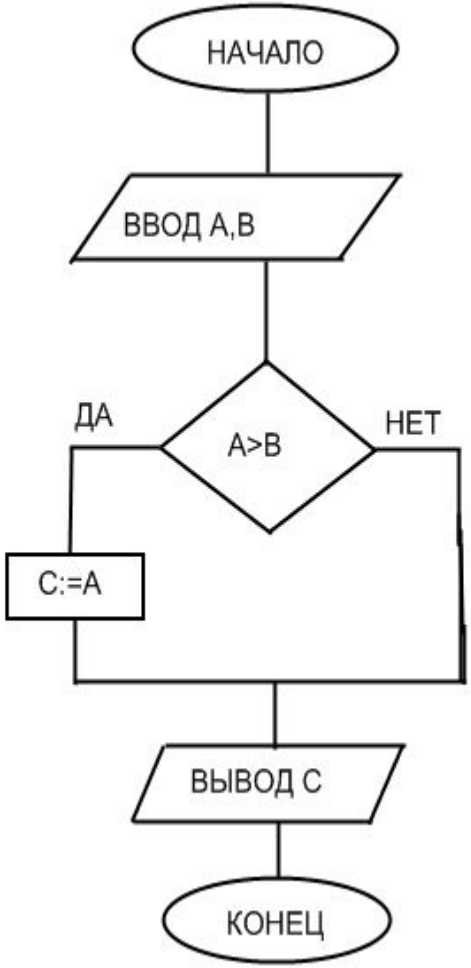
SQR (A) + SQR (B) – 12*C

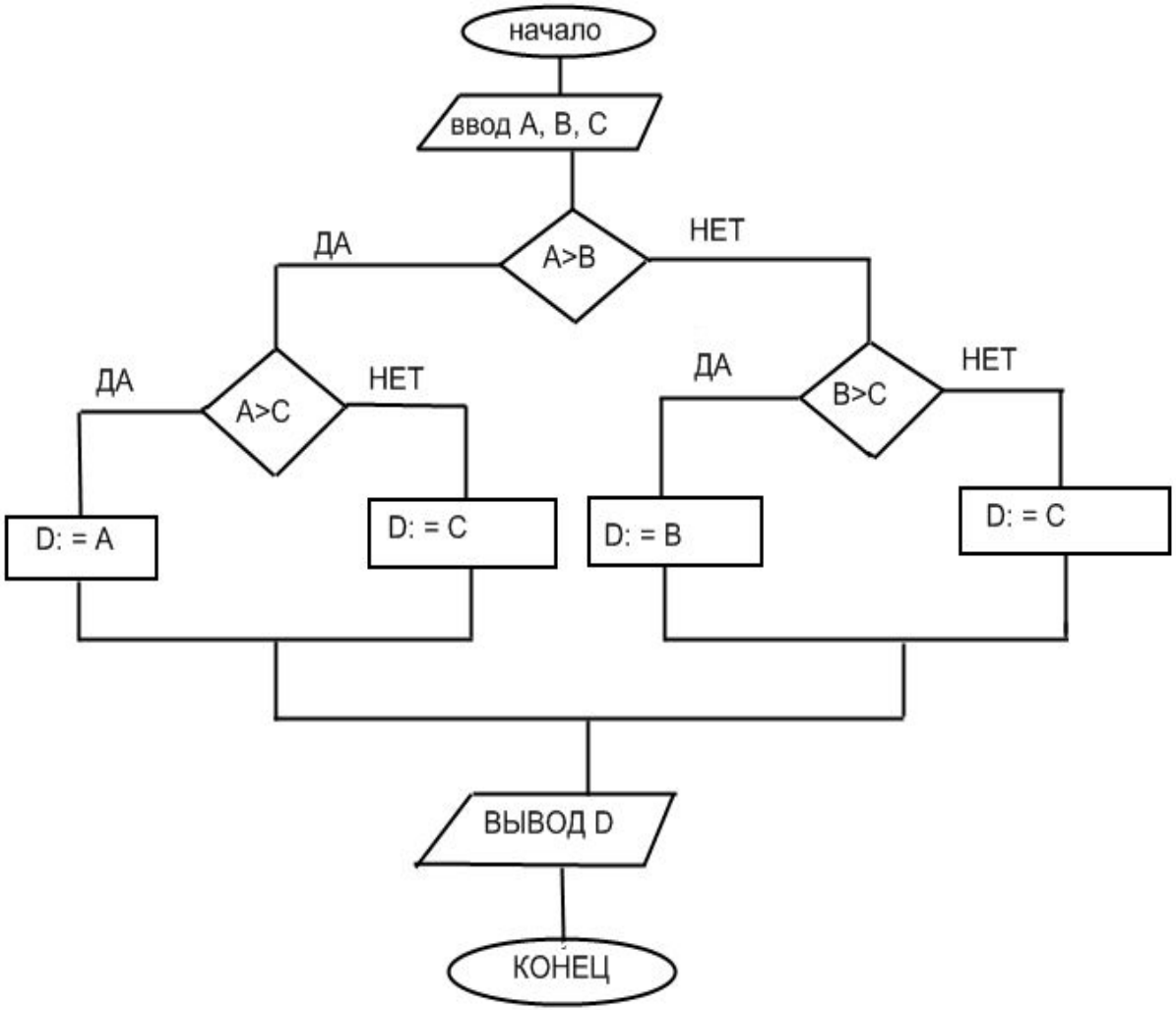
Последовательность выполнения операций определяется по их *приоритетам* (старшинству).

Необходимо строгое соблюдение правописания (синтаксиса) программы.

АЛГОРИТМЫ С ВЕТВЯЩЕЙСЯ СТРУКТУРОЙ






начало
ввод A, B, C
A>B
ДА
НЕТ
A>C
ДА
НЕТ
B>C
ДА
НЕТ
D: = A
D: = C
D: = B
D: = C
ВЫВОД D
КОНЕЦ

- В команде ветвления в качестве условия может использоваться отношение неравенства между величинами.
- Числовые величины, которые могут принимать любые значения(целые и дробные), имеют вещественный тип.
- Для решения одной и той же задачи можно построить несколько вариантов алгоритмов.
- Несколько ветвлений в одном алгоритме могут быть *последовательными и вложенными.*

- На языке Паскаль имеется *оператор ветвления - условный оператор.*

Который имеет вид:

If <логическое выражение>

then<оператор1> else<оператор2>

На ветвях условного оператора могут находиться простые или составные операторы. *Составной оператор* – это последовательность операторов, заключенная между служебными словами begin и end.

В сложных логических выражениях используются логические операции: and, or, not.

ПРОГРАММИРОВАНИЕ ДИАЛОГА С КОМПЬЮТЕРОМ

- Любую программу составлять нужно так, чтобы ее исполнение имитировало диалог между компьютером и пользователем в понятной для человека форме.
- Прежде чем начать составление программы, нужно продумать *сценарий.*
- Например сценарий вычисляющий сумму двух целых чисел: Введите первое слагаемое: A=237 ; Введите второе слагаемое: B=658 ; A+B = 895

Пока!

```
Program Summa;
var A, B: integer;
begin write (введите первое слагаемое: A=‘);
readl (A);
write (введите второе слагаемое: B=‘);
readl (B);
writeln;
writeln (A+B=‘ , A+B);
writeln (пока!)
end.
```

Программирование циклов

- Этапы решения расчетной задачи на компьютере

1)Постановка задачи

2)Математическая формализация

3)Построение алгоритма

4)Составление программы на языке программирования

5)Отладка и тестирование программы

6)Проведение расчетов и анализ полученных результатов

Эту последовательность называют *технологией решения задачи на компьютере.*

- Количество различных комбинаций из *N* предметов, получаемых изменением их порядка, называется – *числом* перестановок. Число перестановок равно *N!*

(N-факториал): N!= 1*2*…*N.

Любой циклический алгоритм может быть построен с помощью команды «цикл-пока» (цикл с предусловием).

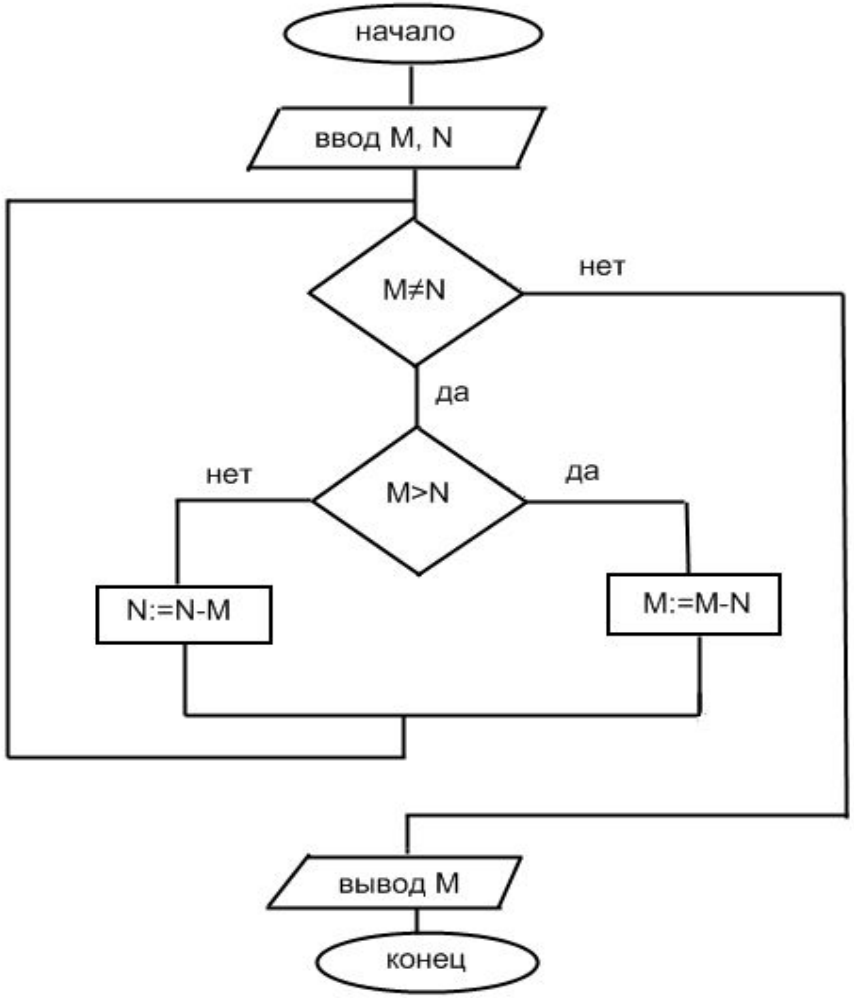
Цикл – команда исполнителю многократно повторить указанную последовательность команд.

Оператор цикла с предусловием в Паскале:

while <логическое выражение> do <оператор>;

Оператор составляющий тело цикла, может быть *простым* или *составным.*

Алгоритм Евклида
Блок схема алгоритма Евклида



- Алгоритм Евклида предназначен для получения наибольшего общего делителя двух натуральных чисел. Структура алгоритма Евклида-цикл с вложенным ветвлением.
- Ручная трассировка может использоваться для проверки правильности лишь сравнительно простых алгоритмов. Правильность программ проверяется путем тестирования на компьютере.

ТАБЛИЦЫ И МАССИВЫ

- МАССИВ – это пронумерованная конечная последовательность однотипных величин.

Представление таблицы в языках программирования называется *массивом.*

Месяц	1	2	3	4	5	6	7	8	9	10	11	12
температура	-24	-18	-7,5	5,6	10	18	22,2	24	17	5,4	-7	-18

Линейная таблица- одномерный массив

В алгоритмах, связанных с перебором элементов массива, удобно использовать структуру «цикл с параметром».

МАССИВЫ В ПАСКАЛЕ

- Описание и обработка массива на Паскале

Общая форма описания одновременного массива на Паскале такая:

var <имя массива>: array[< нижняя граница индекса.. верхняя граница индекса>] of <тип массива>

- Описание массива t будет следующим:

```
var T: array [1..12] of real;
```

Цикл с параметром на Паскале

Рассмотрим полный текст программы на Паскале.

```
Program Temperature;
 var T: array [1..12] of real;
I: integer; Tsred: real;
begin
(цикл ввода)
for I:=1 to 12 do
begin
write('T [',I:2,']=');
readln (T[I])
end;
(цикл суммирования)
Tsred:=0;
For I:=1 to 12 do
Tsred:= Tsred:+ [I];
(вычисление среднего)
Tsred:= Tsred/12;
Writeln('среднегодовая t=',Tsred: 6: 2, ' градусов')
end.
```

Одна задача обработки массива

- Случайные числа – результаты случайного выбора из конечного множества значений (игровой кубик, жребий и Т.Д)
- Функция random (x) – датчик случайных чисел в диапазоне от 0 до X на Паскале.
- Для почёта количества искомых величин используется переменная – счётчик.