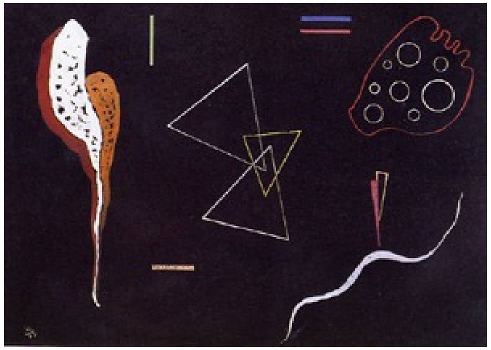
Василий Кандинский
(1866-1944)

(1898 г.)

(1911 г.)



(1938 г.)

http://wassilykandinsky.ru

Казимир Малевич

(1879-1935)

Лекция 1

Определение алгоритма. Представление алгоритма: псевдокод, блок-схема. Базовые алгоритмические конструкции.

Определение алгоритма.

Алгоритм – это ***упорядоченный*** набор ***конечного*** числа строго ***определенных выполнимых*** шагов для решения ***задачи определенного типа*.**

Конечность.

Задача. Даны два отрезка разной длины *a* и *b*. Построить *c* - наибольший из отрезков, укладывающихся целое число раз в данных отрезках.
Алгоритм(?). Пусть |*a*| > |*b*|.
Шаг 1. Отложим отрезок *b* на отрезке *a* наибольшее количество раз.
Шаг 2. Если *b* точно отложился на *a* целое число раз, то выполнение алгоритма прекращается, задача решена, искомый отрезок – это *b*.
Шаг 3. Принять в качестве отрезка *b* остаток отрезка *a*, куда не помещался отрезок *b*, а в качестве отрезка *a* отрезок *b* и перейти к Шагу 1.

Является ли данный набор шагов алгоритмом?

Если существует некий отрезок, пусть очень малой длины, укладывающийся целое число раз в отрезках a и b, то можно среди таких отрезков найти и наибольший, используя приведенный выше набор шагов. Но может не существовать такого отрезка, в этом случае говорят, что отрезки несоизмеримы (отношение их длин выражается бесконечной непериодической десятичной дробью). Тогда последовательность приведенных шагов становится бесконечной. Таким образом, в общем случае вышеприведенный набор шагов не является алгоритмом. Для соизмеримых отрезков этот набор является алгоритмом (геометрический аналог алгоритма Евклида).

Вопросы:

Является ли метод деления столбиком алгоритмом нахождения частного? (бесконечные периодические дроби)

Является ли метод деления столбиком алгоритмом нахождения частного с заданной точностью?

То же для вавилонского метода оценки квадратного корня x из целого числа y:

$x:=(x+y/x)/2.$

Определенность.

Задача. Найти длину гипотенузы прямоугольного треугольника, зная длину его катетов.

Шаг 1. Возвести в квадрат длину 1-го катета
Шаг 2. Возвести в квадрат длину 2-го катета
Шаг 3. Сложить полученные числа.
Шаг 4. Извлечь квадратный корень из полученного числа.

Является ли этот набор шагов алгоритмом?

Не определен шаг №4. При извлечении корня получается два числа и только одно из них положительное – арифметический квадратный корень. Необходимо детализировать этот шаг с тем, чтобы результат его выполнения был однозначным.

Выполнимость.

…

Шаг *N*. Умножить полученное число на сумму $x+y+z$, где (x,y,z) из N^3 является решением уравнения $x^4+y^4=z^4$ с наименьшим значением x.

…

Является ли этот набор шагов алгоритмом?

Во-первых, шаг *N* неоднозначен, при одном *x* может быть несколько решений с разными суммами. Во-вторых, и это главное, шаг *N* содержит невыполнимые действия. Дело в том, что в соответствие с уже доказанной *теоремой Ферма* таких решений вообще не существует.

[*алгоритмически неразрешимые задачи; 10-я проблема Гильберта; вычислимость* [*А. Тьюринг*]]

Представление алгоритма.

Виды представлений.

Псевдокод, *pidgin Pascal, C* и т.п. **Блок-схема**. [диаграмма активности в UML] Программы на языке высокого уровня. **Программа** – последовательность нулей и единиц.

Базовые алгоритмические конструкции.

Присвоение.

<имя переменной>:=<выражение>

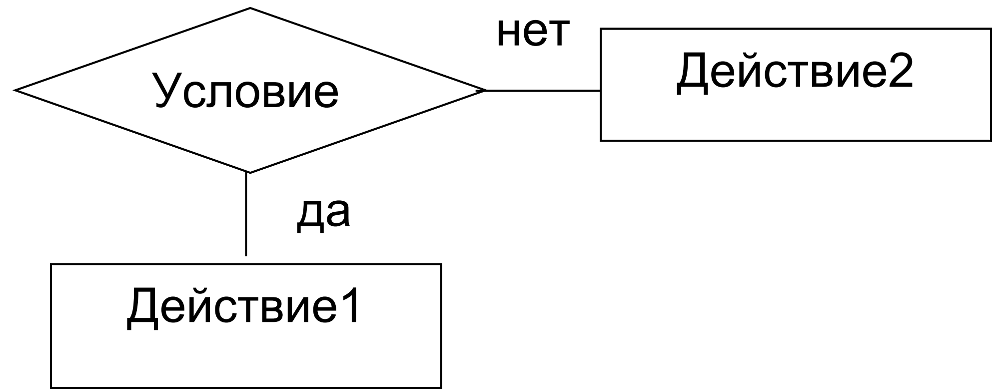
Следование.

<Действие 1>

<Действие 2>

Ветвление.

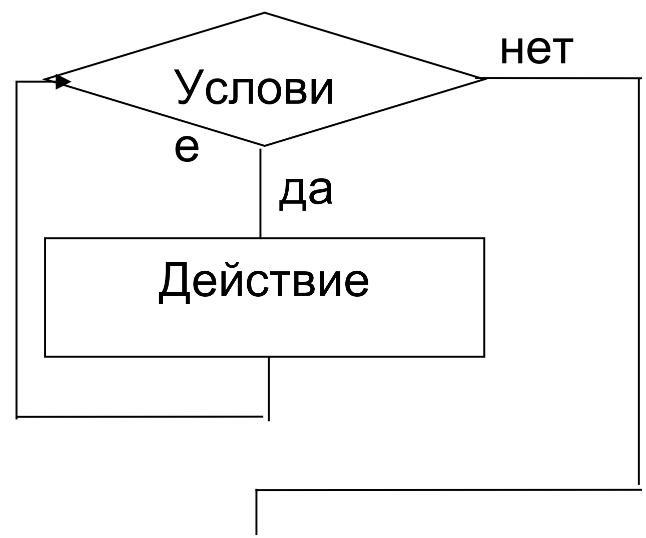
Если <условие>
то <действие 1>
иначе <действие 2>
Конец-если



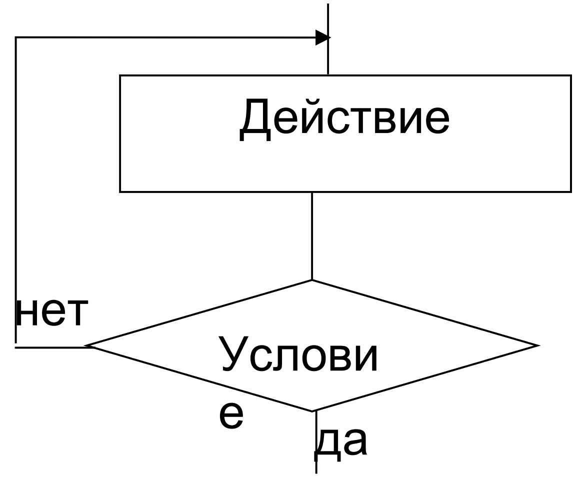
Цикл-пока.

Цикл-пока <условие>

<действие>
Конец-цикл

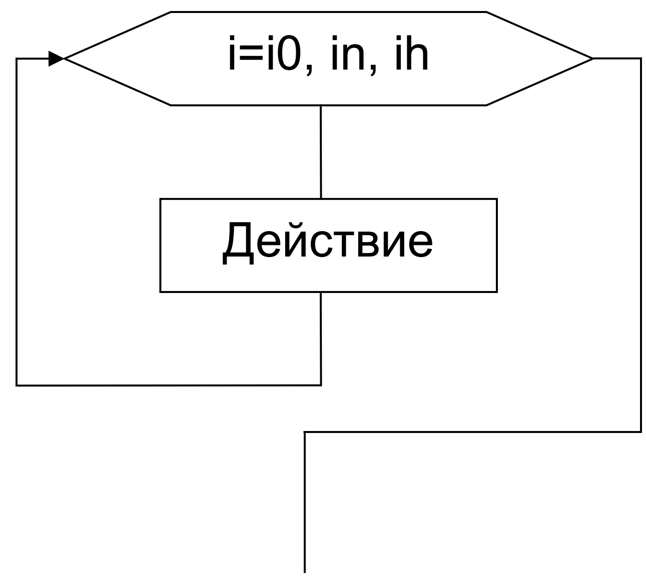


Цикл-до ***(****repeat****)***.



Выполнять
<Действие>
До <условие>

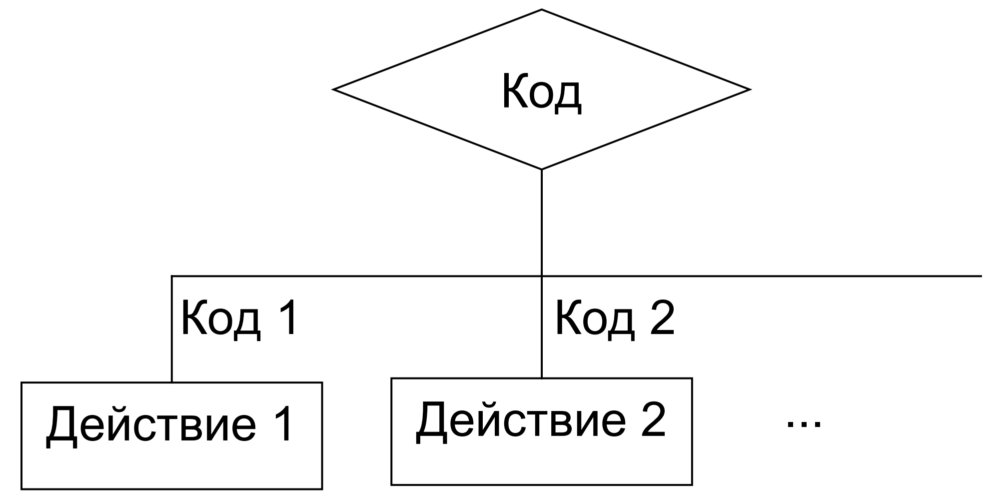
Счетный цикл



Для <индекс>=<i0, in, ih>
<Действие>
Конец-цикл

Выбор.

<u>Выбор</u> <код>
<код 1>: <действие 1>
<код 2>: <действие 2>
…
<u>Конец-выбор</u>



…

…

Процедура.

Описание процедуры:

<u>Процедура</u> <имя>(<список параметров>)

<действие 1>

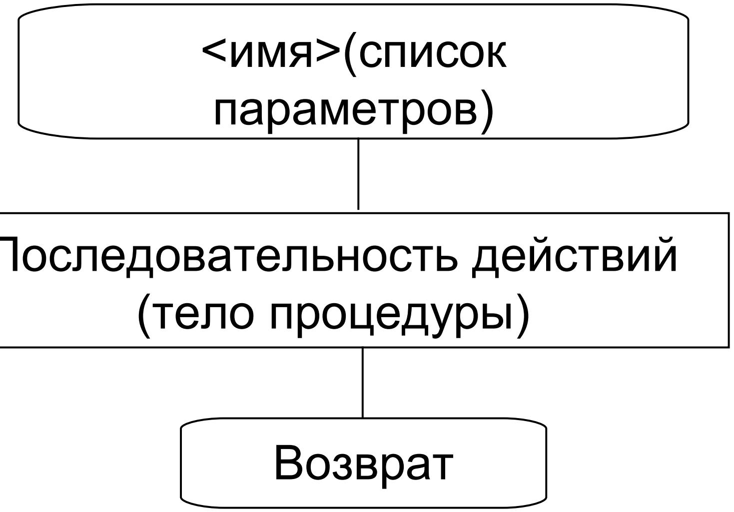
<действие 2>

……………..

[<имя>:=<выражение>]

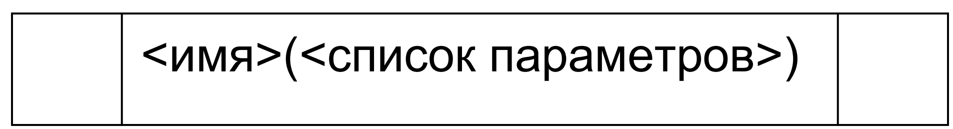
<u>Возврат</u>

<u>Конец</u>



Вызов процедуры:

<имя>(<список фактических параметров>)

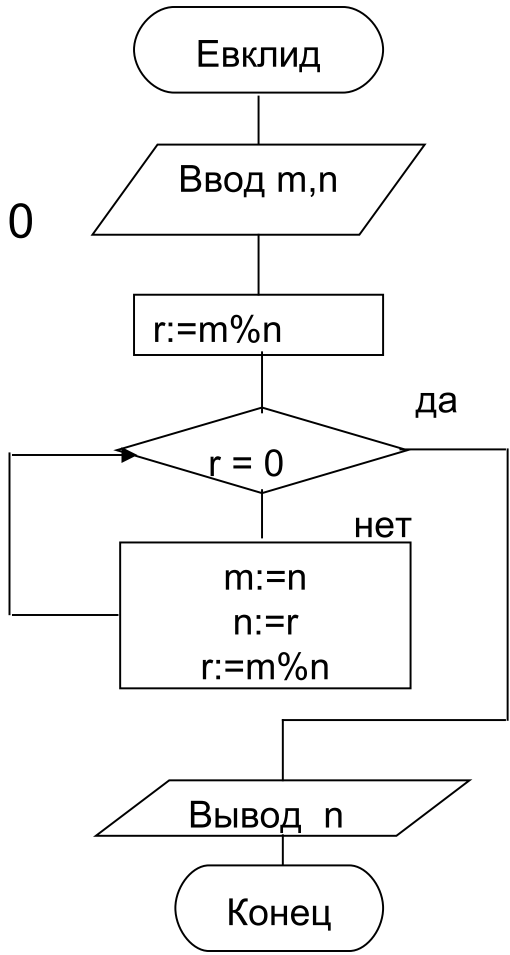


Упражнение

Представить алгоритм Евклида в виде псевдокода и блок-схемы.

Шаг1. Разделим m на n и пусть r – остаток.
Шаг2. Если r=0, то выполнение алгоритма прекращается; n – искомое значение.
Шаг3. Присвоить m:=n, n:=r и вернуться к шагу1.

<u>Программа Евклид</u>
  <u>Ввод</u> m, n
    r:=m%n
  <u>Цикл-пока</u> r <u>не равно</u> 0
    m:=n
    n:=r
    r:=m%n
  <u>Конец-цикл</u>
  <u>Вывод</u> n
<u>Конец</u>



Задание.

- Будет ли алгоритм работать, если *m*<*n*?
- Всегда ли за конечное число шагов переменная *r* получит значение 0
- Замените операцию деления по модулю - % (получение остатка) вычитанием.
- Оформите в виде процедуры совокупность действий, направленных на нахождение остатка.