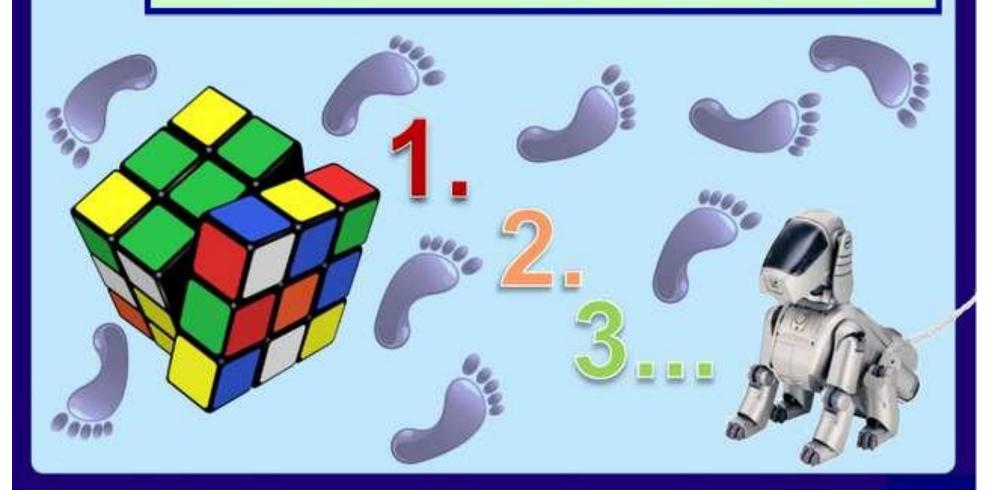
Составление алгоритмов для решения задач

ЧТО ТАКОЕ АЛГОРИТМ?



АЛГОРИТМ- это конечная последовательность шагов в решении задачи, приводящая от исходных данных к требуемому результату



Свойства алгоритма Детерминированность) Массовость Дискретность Алгоритм Конечность Формальность Результативность

Способы описания алгоритмов

- **«** словесное описание;
- описание алгоритма с помощью математических формул;
- ❖ графическое описание алгоритма в виде блоксхемы;
- описание алгоритма с помощью псевдокода программный;
- * комбинированный способ.

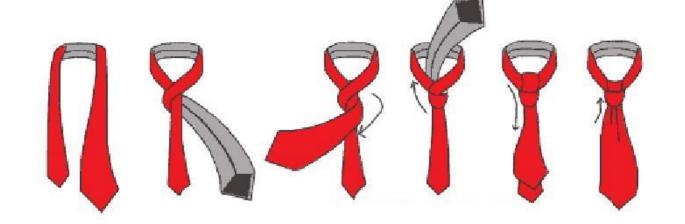
Словесное описание алгоритма

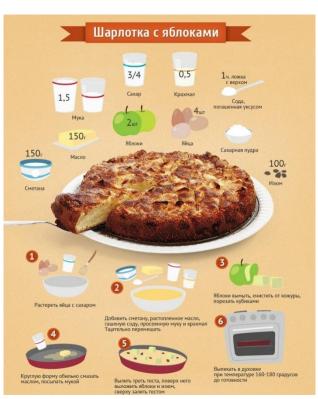
Словесное описание алгоритма представляет собой описание структуры алгоритма на естественном языке.

Достоинство: простота описания.

Недостаток: многословен и не имеет строгой формализации.

Примеры:





Описание алгоритма с помощью математических формул

$$Q = \frac{\pi}{4} \int_{0}^{l} d^{2} dl.$$

$$ctg 2\alpha = \frac{1 - tg^{2}\alpha}{2tg\alpha}$$

$$a^{2} + b^{2} = c^{2}$$

Описание алгоритма с помощью псевдокода — программный

Псевдокод основан на записи алгоритмов в формализованном представлении предписаний (спец. программы).

Достоинство: близость к языкам программирования.

Недостаток: сложность освоения и невозможность непосредственного ввода алгоритма для решения на ЭВМ.

```
Пример:
```

```
FLOYD-WARSHALL(M) D^0 = M \qquad // M - \text{матрица смежности} \\ n = rows[M] \qquad // \text{размер матрицы}  for k=1 to n do  for i=1 \text{ to } n \text{ do}  for j=1 to n do  d^{k}_{ij} = min(d^{k-1}_{ij}, d^{k-1}_{ik} + d^{k-1}_{kj})  return D^n
```

```
<u>ввод</u> а, b, c
D = b2 - 4ac
<u>если</u> D < 0 <u>то</u>
<u>вывод</u> Корней нет
<u>иначе</u>
<u>если</u> D > 0 <u>то</u>
х1 = ...
х2 = ...
<u>вывод</u> х1, х2
<u>иначе</u>
х1 = ...
<u>вывод</u> х1
```

Графическое описание алгоритма в виде блок-схемы

Графический способ предполагает, что для описания структуры алгоритма используется совокупность графических изображений (блоков), соединяемых линиями передачи управления.

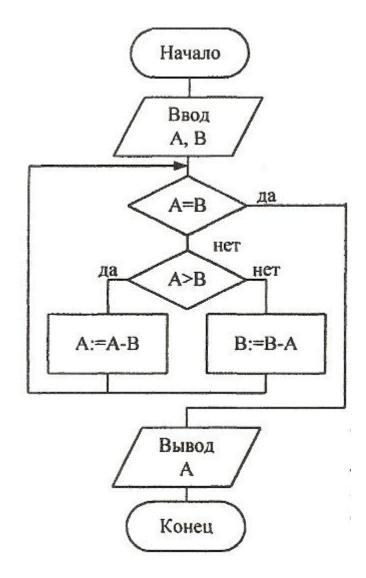
Примеры:





Блок-схема

Блок-схема алгоритма — это графическое представление хода решения задачи.



Название блока	Обозначение	Назначение блока
Терминатор	Действие	Начало, завершение программы или подпрограммы
Процесс	Действие	Обработка данных (вычисления, пересылки и т. п.)
Данные	Данные	Операции ввода-вывода
Решение	Условие	Ветвления, выбор, итерационные и поисковые циклы
Подготовка	Действия	Счетные циклы
Граница цикла	Начало Конец	Любые циклы
Предопределенный процесс	РММ	Вызов процедур
Соединитель	RMN	Маркировка разрывов линий
Комментарий	Комментарий	Пояснения к операциям



Виды алгоритмов



Циклический

Разветвляющийся

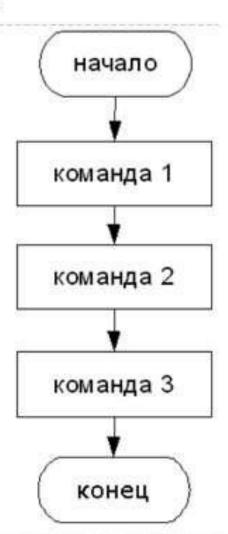




Линейный алгоритм

Линейный — алгоритм, в котором команды выполняются однократно, одна за другой в том порядке, в котором записаны в программе.





Пример

Разработать алгоритм вычисления гипотенузы прямоугольного треугольника по известным значениям длин его катетов а и b.

Этап 1. Математическое описание решения задачи.

Математическим рервением запани авпается известная формула:

$$c = \sqrt{a*a+b*b}$$

где с-длина гипотенузы, а, b – длины катетов.

Этап 2. Определение входных и выходных данных.

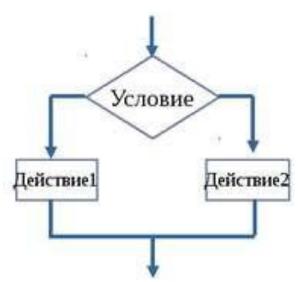
Входными данными являются значения катетов а и b. Выходными данными является длина гипотенузы – с.

Этап 3. Разработка алгоритма решения задачи.

Словесное описание алгоритма Запись алгоритма на языке блок-схем 1. Начало алгоритма. На данной схеме цифрами указаны номера элементов алгоритма, которые Ввод значений длин катетов а и b. соответствуют пунктов номерам 3. Вычисление длины гипотенузы с по словесного описания алгоритма. формуле $c = \sqrt{a * a + b * b}$ Начало 4. Вывод значения длины гипотенузы. 2 Ввода, в 5. Конец алгоритма $c = \sqrt{a*a+b*b}$ 3 4 Вывод с 5 Конец

Разветвляющийся алгоритм

Разветвляющийся — алгоритм, в котором последовательность действий зависит от результата выполнения условия



В разветвляющемся алгоритме имеются разные варианты решения задачи в зависимости от результата проверки какого-либо условия (расходятся в разные стороны как веточки на дереве, отсюда название – разветвленный). Например, алгоритм проведения выходного дня в зависимости от погоды. Если будет дождь – одни действия, если – нет, то планы будут другие.

Пример

Разработать алгоритм вычисления наибольшего числа из двух чисел х и у.

Этап 1. Математическое описание решения задачи.

Из курса математики известно, если x > y, то наибольшее число x, если x < y, то наибольшее число y, если x = y, то число x равно числу y.

Этап 2. Определение входных и выходных данных.

Входными данными являются значения чисел х и у.

Выходным данными являются:

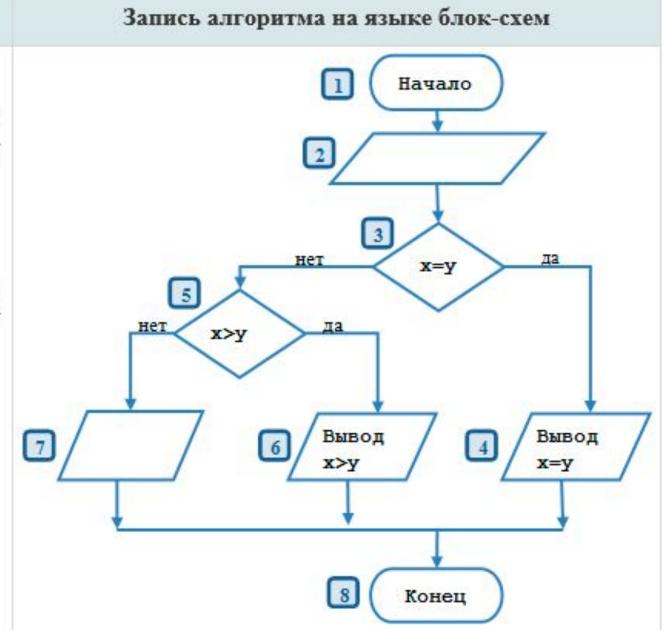
- •наибольшее число
- •любое из чисел, если числа равны

Для решения задачи нам необходимо знать значения х и у.

Этап 3. Разработка алгоритма решения задачи.

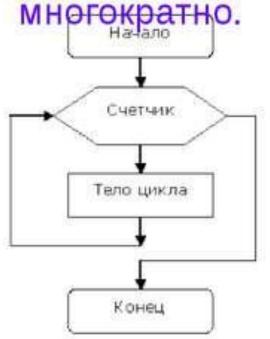
Словесное описание алгоритма

- 1. Начало алгоритма.
- 2. Ввод значений х и у.
- Сравниваем х и у. Если х = у, то переход к шагу 4, иначе к шагу 5.
- Вывод информации: числа х и у равны. Переход к шагу 8.
- Сравниваем х и у. Если х≥у, то переход к шагу 6, иначе к шагу 7.
- Вывод информации: число х больше у. Переход к шагу 8.
- Вывод информации: число у больше х. Переход к шагу 8.
- 8. Конец алгоритма.



Циклический алгоритм

Циклический — алгоритм, в котором определенный набор команд выполняется многократно



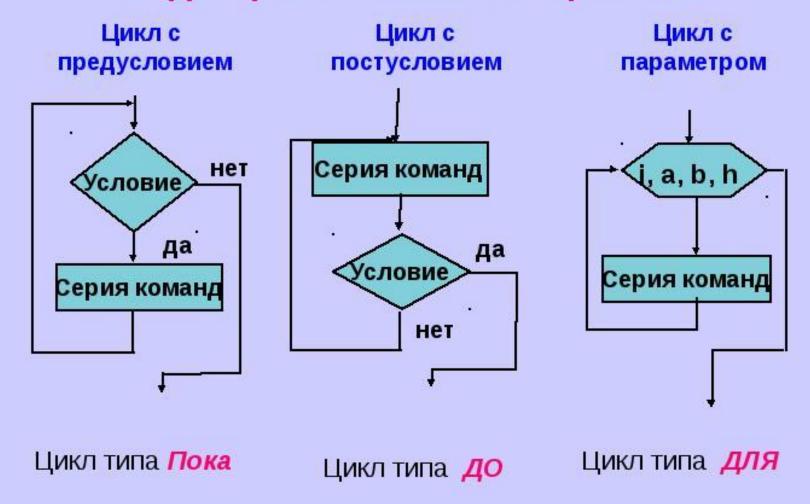


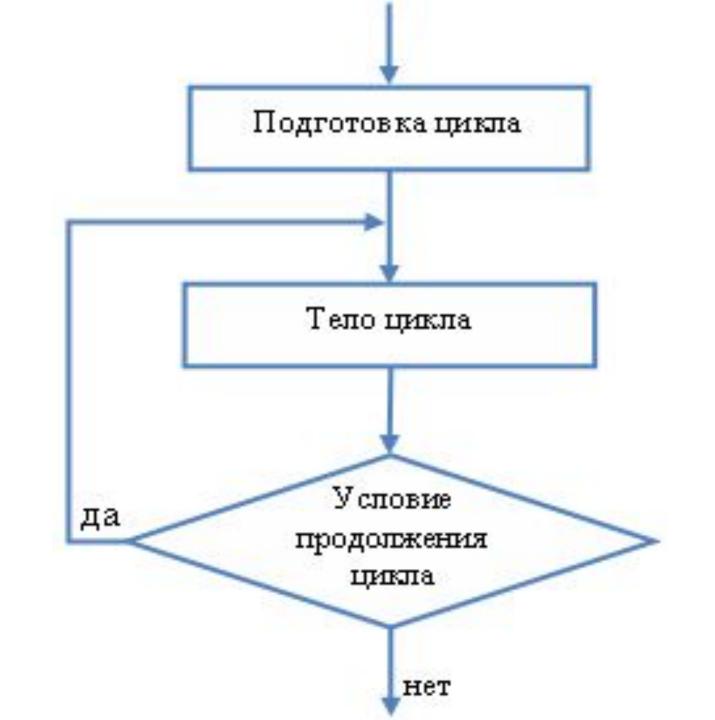
- это описание действий, которые должны повторяться указанное число раз или пока не выполнено заданное условие.

Циклическая структура

- •параметр цикла;
- •начальное и конечное значения параметров цикла;
- •шаг цикла.

Виды циклических алгоритмов





Пример

Разработать алгоритм вычисления суммы натуральных чисел от 1 до 100.

Этап 1. Математическое описание решения задачи.

Обозначим сумму натуральных чисел через S. Тогда формула вычисления суммы натуральных чисел от 1 до 100 может быть записана так:

$$S = 1 + 2 + 3 + ... + 97 + 98 + 99 + 100 = \sum_{i=1}^{n} X_i$$

где Xi — натуральное число X с номером i, который изменяется от 1 до n, n=100 — количество натуральных чисел.

Этап 2. Определение входных и выходных данных.

Входными данными являются натуральные числа: 1, 2, 3, 4, 5, ..., 98, 99, 100.

Выходные данные – значение суммы членов последовательности натуральных чисел.

Параметр цикла – величина, определяющая количество повторений цикла. В нашем случае i — номер натурального числа.

Подготовка цикла заключается в задании начального и конечного значений параметра цикла.

начальное значение параметра цикла равно 1,

конечное значение параметра цикла равно n,

шаг цикла равен 1.

Для корректного суммирования необходимо предварительно задать начальное значение суммы, равное 0.

Тело цикла. В теле цикла будет выполняться накопление значения суммы чисел, а также вычисляться следующее значение параметра цикла по формулам:

$$S=S+i;$$
 $I=I+1;$

Условие продолжения цикла: цикл должен повторяться до тех пор, пока не будет добавлен последний член последовательности натуральных чисел, т.е. пока параметр цикла будет меньше или равен конечному значению параметра цикла.

Этап 3. Разработка алгоритма решения задачи.

Сло	весное описание алгоритма	Запись алгоритма на языке блок-схем
4. 5.	Начало алгоритма. Подготовка цикла: S:=0; i=1; n= 100;	В схеме алгоритма решения задачи цифрами указаны номера элементо алгоритма. Номера элементов соответствуют номерам шагов словесног описания алгоритма.
7.	натуральных чисел – S. Конец алгоритма.	5 <u>j</u> :=i+1; 7 Конец

Задача 1

Алгоритм вычисления значения выражения K=3b+6a.

Ввод исх. данных; b, a

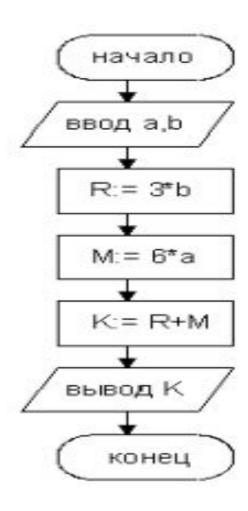
Вычисление R; = 3*b

Вычисление М 📜 6*а

Вычисление K;:= R+M

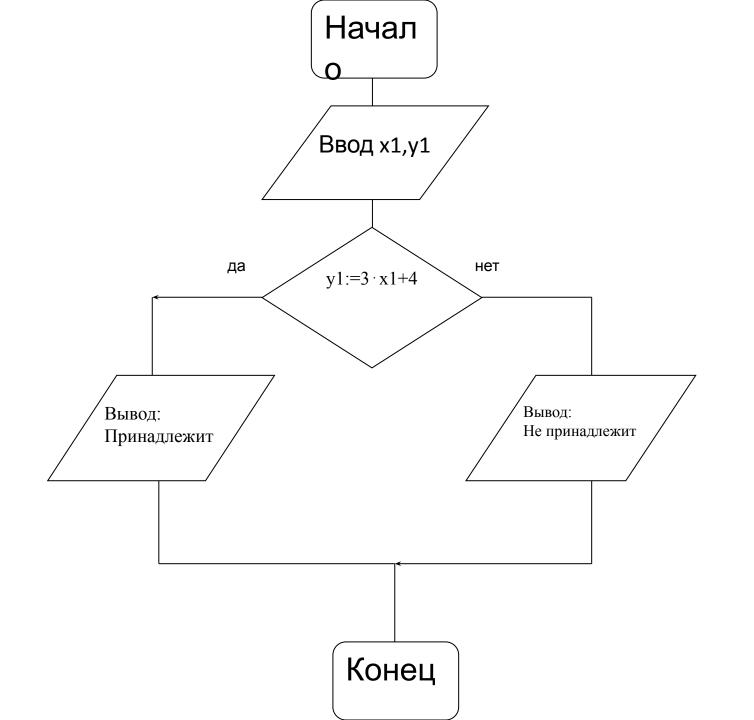
Вычисление K=3*b+6*a

Вывод результата: К



Задача 2

Алгоритм, определяющий, пройдет ли график функции y=3x+4 через точку с координатами x1,y1.



Задача 3

Алгоритм, определяющий факториал натурального числа n:

$$n! = 1*2*3*....*(n-1)*n$$

$$0!=1$$

