Алгоритм. Понятие алгоритма.

Алгоритм - понятное и точное предписание (указание) исполнителю совершить определенную последовательность действий для достижения указанной цели или решения поставленной задачи.

Алгоритм - список команд, набор инструкций, выполнив которые можно получить определенный результат.

Сборником алгоритмов можно назвать книгу кулинарных рецептов. Рассмотрим простейший алгоритм.

Пр. 1. Алгоритм заварки чая.

1. Подготовить исходные величины - заварку, воду, чайник, заварник
2. Налить в чайник воду.
3. Насыпать в заварник заварку.
4. Довести воду до кипения.
5. Налить в заварник кипяток и подождать 3 минуты.
6. Заварка готова.

I. Свойства алгоритма.

Не каждый набор команд можно назвать алгоритмом. Алгоритм обладает определенными свойствами:

1. *Конечность*. Суть свойства: алгоритм не может быть бесконечным, он должен закончиться за конечное число шагов.

2. *Результативность*. Суть свойства: выполнив алгоритм, должны получить результат. Установление факта, что задача решения не имеет, является тоже результатом исполнения алгоритма.

3. *Дискретность* (прерывистость). Суть свойства: алгоритм разбивается на отдельные шаги (команды), которые выполняются одна за другой.

4. *Понятность*. Суть свойства: команды алгоритма должны быть понятны исполнителю. В алгоритме используются только команды из системы команд исполнителя.

5. *Определенность*. Суть свойства: каждая команда однозначно определяет действия исполнителя.

6. *Массовость*. Суть свойства: алгоритм должен обеспечивать решение не одной конкретной задачи, а класса задач данного типа.

7. *Эффективность*. Суть свойства: каждый шаг алгоритма должен быть выполнен точно и за конечное время, а, значит, весь алгоритм должен быть выполнен за разумно конечное (эффективное) время.

II. Система команд исполнителя

Исполнитель - это тот, кто будет исполнять алгоритм.

Совокупность команд, которые могут быть выполнены конкретным исполнителем, называется **системой команд исполнителя**.

III. Формальное исполнение алгоритма

Исполнитель может не иметь представления о цели выполнения алгоритма. Он должен строго и точно выполнять действия, предписанные алгоритмом, не понимая, зачем и почему это надо делать. Такое исполнение называется **формальным исполнением алгоритма**, что позволяет передать исполнение алгоритма автомату.

IV. Способы записи алгоритмов.

1. Естественный язык (словесная запись алгоритма)

Обычно используется для алгоритмов, ориентированных на исполнителя – человека. Команды алгоритма нумеруют, чтобы иметь возможность на них ссылаться.

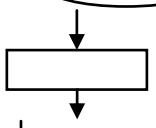
Словесная запись алгоритма была использована выше для составления алгоритма заварки чая (см. Пр. 1, стр. 2)

2. Язык блок-схем (графическая запись алгоритмов).

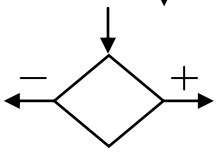
Команды алгоритмов помещаются внутрь блоков, соединенных стрелками, показывающими очередность выполнения команд алгоритма.

- Овал обозначает начало и конец алгоритма (блок начало и блок конец).

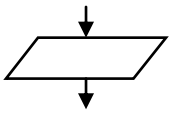
- Команды обработки информации помещают в блоках имеющих вид прямоугольников (блок арифметических выражений, блок присваиваний).



- Проверка условий - ромб. В результате проверки условия возникают два возможных пути для продолжения алгоритма. Эти пути изображаются стрелками со знаками "+" и "–" (иногда пишут "да" и "нет"). Переход по стрелке со знаком "+" происходит если условие соблюдено а переход по стрелке "–", если условие не выполняется.



- Операции ввода и вывода помещают в блоки, имеющие вид параллелограммов (блок ввода/вывода).



Для записи команд внутри блоков используется естественный язык с элементами математической символики.

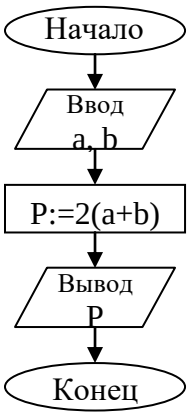
Пр. 2. *Задача*. Даны длина и ширина прямоугольника. Определить периметр этого прямоугольника.

Решение. Выделяем исходные данные и результаты.

Исходные данные: a – длина, b – ширина прямоугольника.

Результат: P – периметр прямоугольника.

Составим алгоритм решения задачи и запишем его на языке блок-схем. (см. рис.)



3. Алгоритмический язык (псевдокоды).

Псевдокод представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов. Он занимает промежуточное место между естественным и формальным языком.

С одной стороны он близок к обычному естественному языку, поэтому алгоритмы на нем могут записываться и читаться как обычный текст. С другой стороны, в псевдокоде используются некоторые формальные конструкции и математическая символика, что приближает запись алгоритма к общепринятой математической записи.

Единого или формального определения псевдокода не существует, поэтому возможны различные псевдокоды, отличающиеся набором служебных слов и основных (базовых) конструкций.

Запишем алгоритм нахождения периметра прямоу-гольника (см. Пр. 2), на алгоритмическом языке:

алг периметр прямоугольника
нач ввод a, b
  P := 2·(a + b)
  вывод P
кон

4. Формальный язык (язык программирования).

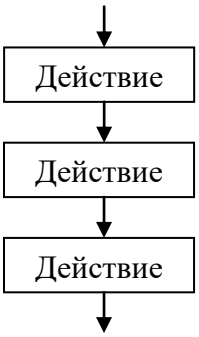
Обычно используется для алгоритмов, ориентированных на исполнителя – ЭВМ. Алгоритм, записанный на языке программирования - программа

V. Структуры алгоритмов.

Из простых команд и проверки условий образуются составные команды (структуры).

Любой алгоритм может быть построен из базовых структур: следование, ветвление, цикл.

Структура следование



Эта структура образуется из последовательности команд, следующих одна за другой. При исполнении алгоритма команды выполняются одна за другой в том порядке, как они записаны.

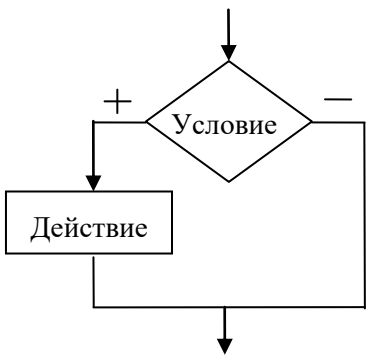
Под действием понимается либо простая, либо составная команда.

Линейным называется алгоритм, в котором все этапы решения задачи выполняются строго последовательно. Алгоритмы заварки чая, нахождения периметра прямоугольника, приведенные выше, являются линейными.

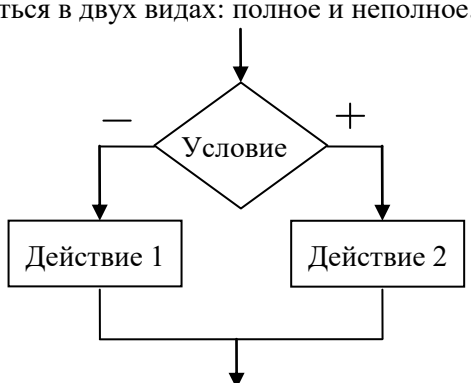
Структура ветвление (развилка).

Выбор одного из двух возможных действий, в зависимости от результата проверки условия, осуществляется с помощью **развилки (ветвления**).

Ветвление может использоваться в двух видах: полное и неполное.



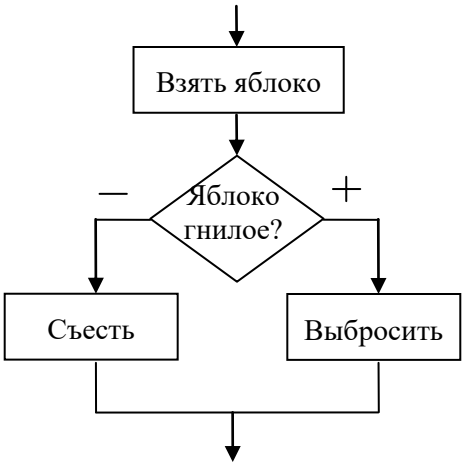
Блок-схема неполной развилки



Блок-схема полной развилки

Рассмотрим ветвление на конкретных примерах.

Пр. 3. Фрагмент алгоритма «Поедание яблока»



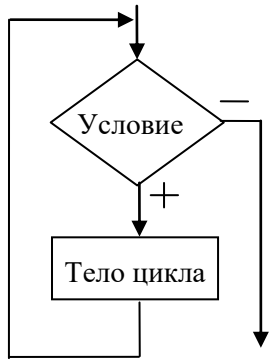
Пр. 4. Фрагмент алгоритма «Покупка билетов»

Структура повторение (цикл)

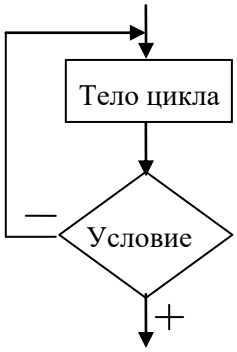
Цикл – алгоритмическая структура, организующая многократное повторение действий.

Действия, которые повторяются в цикле, называют **телом цикла.**

Циклы бывают двух видов: цикл «До» и цикл «Пока».



Блок-схема цикла «Пока»



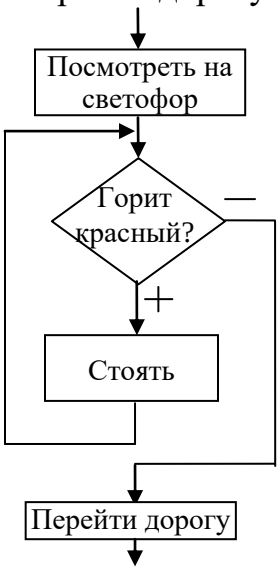
Блок-схема цикла «До»

Цикл «Пока» - цикл с предусловием (сначала проверяется условие, потом выполняется тело цикла). В цикле «Пока» тело цикла выполняется, пока выполняется условие.

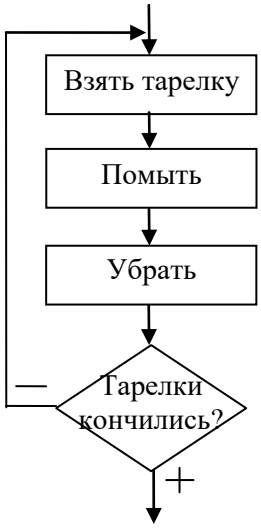
Цикл «До» - цикл с постусловием (сначала выполняется тело цикла, потом проверяется условие). В цикле «До» тело цикла выполняется до тех пор, пока не выполнится условие.

Рассмотрим циклы на конкретных примерах.

Пр. 5. Фрагмент алгоритма «Перейти дорогу»



Пр. 6. Фрагмент алгоритма «Помыть тарелки»

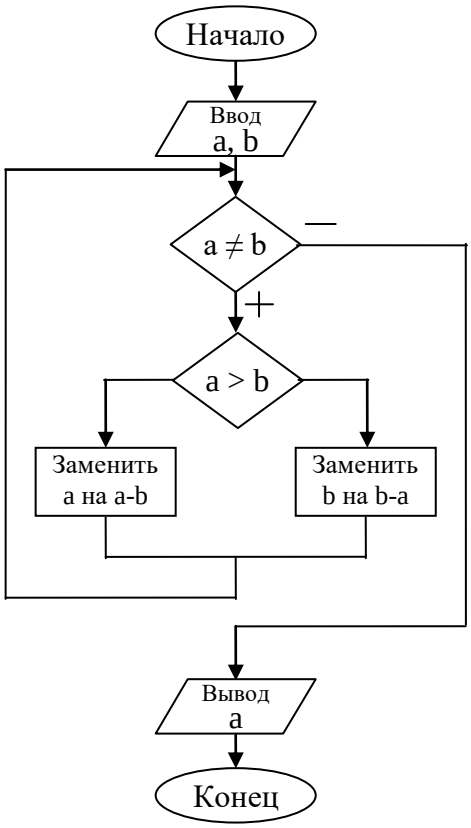


Рассмотрим пример алгоритма, в котором внутри одной составной структуры находится другая.

Пр. 7 ***Алгоритм Евклида*** для нахождения наибольшего общего делителя (НОД) двух натуральных чисел:

1. Если числа равны, то взять первое число в качестве ответа и закончить исполнение алгоритма, иначе перейти к п.2
2. Определить большее из двух чисел.
3. Заменить большее число на разность большего и меньшего чисел.
4. Перейти к п.1

Блок-схема алгоритма Евклида:



Например, a = 32, b = 24

Трассировочная таблица:

шаг	Операция	a	b	Условие
1.	Ввод a	32		
2.	Ввод b		24	
3.	a ≠ b			32 ≠ 24, да
4.	a > b			32 > 24, да
5.	a на a-b	8		
6.	a ≠ b			8 ≠ 24, да
7.	a > b			8 > 24, нет
8.	b на b-a		16	
9.	a ≠ b			8 ≠ 16, да
10.	a > b			8 > 16, нет
11.	b на b-a		8	
12.	a ≠ b			8 ≠ 8, нет
13.	Вывод a			
14.	Конец			

В данном алгоритме ветвление находится внутри цикла. Алгоритм представляет собой структуру следование.