




Понятие алгоритма и его  
свойства.

Блок-схема алгоритма



Слово "Алгоритм" происходит от algorithmi - латинского написания имени аль-Хорезми, под которым в средневековой Европе знали величайшего математика из Хорезма (город в современном Узбекистане) Мухаммеда бен Мусу, жившего в 783-850 г.г.

В своей книге "Об индийском счете" он сформулировал правила записи натуральных чисел с помощью арабских цифр и правила действий над ними столбиком.

## Эволюция значения слова "Алгоритм"


IX век

Правила  
арифметических  
действий



XX век

Последовательность  
действий для  
решения различных  
задач




*В дальнейшем алгоритмом стали называть точное предписание, определяющее последовательность действий, обеспечивающую получение требуемого результата из исходных данных.*


Алгоритм может быть предназначен для выполнения его человеком или автоматическим устройством.

Создание алгоритма, пусть даже самого простого, - процесс творческий.


Он доступен исключительно живым существам, а долгое время считалось, что только человеку.



Другое дело - реализация уже имеющегося алгоритма. Ее можно поручить субъекту или объекту, который не обязан вникать в существо дела, а возможно, и не способен его понять. Такой субъект или объект принято называть формальным исполнителем. Примером формального исполнителя может служить стиральная машина-автомат, которая неукоснительно исполняет предписанные ей действия, даже если вы забыли положить в нее порошок.



Каждый алгоритм создается в расчете на вполне конкретного исполнителя. Те действия, которые может совершать исполнитель, называются его допустимыми действиями. Совокупность допустимых действий образует систему команд исполнителя. Алгоритм должен содержать только те действия, которые допустимы для данного исполнителя.



Объекты, над которыми исполнитель может совершать действия, образуют так называемую среду исполнителя.

Для алгоритмов, встречающихся в математике, средой того или иного исполнителя могут быть числа разной природы - натуральные, действительные и т.п., буквы, буквенные выражения, уравнения, тождества и т.п.


# Свойствами алгоритма являются:

- Дискретность (прерывность, раздельность) – алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов. Каждое действие, предусмотренное алгоритмом, исполняется только после того, как закончилось исполнение предыдущего.
- Определенность – каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.



# Свойствами алгоритма являются:

- Результативность (конечность) – алгоритм должен приводить к решению задачи за конечное число шагов.
- Массовость – алгоритм решения задачи разрабатывается в общем виде, то есть, он должен быть применим для некоторого класса задач, различающихся только исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.




Алгоритм – это последовательность математических, логических или вместе взятых операций, отличающихся детерминированностью, массовостью, направленностью и приводящая к решению всех задач данного класса за конечное число шагов.

Такая трактовка понятия “алгоритм” является неполной и неточной.

Во-первых, неверно связывать алгоритм с решением какой-либо задачи. Алгоритм вообще может не решать никакой задачи.

Во-вторых, понятие

“массовость” относится не к алгоритмам как к таковым, а к математическим методам в целом.




Алгоритм применительно к вычислительной машине – точное предписание, т.е. набор операций и правил их чередования, при помощи которого, начиная с некоторых исходных данных, можно решить любую задачу фиксированного типа.

Алгоритмы в зависимости от цели, начальных условий задачи, путей ее решения, определения действий исполнителя подразделяются следующим образом:

**Механические алгоритмы**, или иначе детерминированные, жесткие (например алгоритм работы машины, двигателя и т.п.);  
**Гибкие алгоритмы**, например стохастические, т.е. вероятностные и эвристические.

Эвристический алгоритм (от греческого слова “эврика”) – это такой алгоритм, в котором достижение конечного результата программы действий однозначно не predetermined, так же как не обозначена вся последовательность действий, не выявлены все действия исполнителя. К эвристическим алгоритмам относят, например, инструкции и предписания. В этих алгоритмах используются универсальные логические процедуры и способы принятия решений, основанные на аналогиях, ассоциациях и прошлом опыте решения схожих задач.



Механический алгоритм задает определенные действия, обозначая их в единственной и достоверной последовательности, обеспечивая тем самым однозначный требуемый или искомый результат, если выполняются те условия процесса, задачи, для которых разработан алгоритм.

Вероятностный (стохастический) алгоритм дает программу решения задачи несколькими путями или способами, приводящими к вероятному достижению результата.

Алгоритмы в зависимости от цели, начальных условий задачи, путей ее решения, определения действий исполнителя подразделяются следующим образом:

- Линейный алгоритм – набор команд (указаний), выполняемых последовательно во времени друг за другом.
- Разветвляющийся алгоритм – алгоритм, содержащий хотя бы одно условие, в результате проверки которого ЭВМ обеспечивает переход на один из двух возможных шагов.
- Циклический алгоритм – алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными. К циклическим алгоритмам сводится большинство методов вычислений, перебора вариантов.

На языке блок-схем каждый шаг алгоритма описывается с помощью соответствующей фигуры, а последовательность выполнения шагов определяется линиями-связями.



# Алгоритмические структуры:

- **Следование.** Предполагает последовательное выполнение команд сверху вниз. Если алгоритм состоит только из структур следования, то он является линейным.
- **Ветвление.** Выполнение программы идет по одной из двух, нескольких или множества ветвей. Выбор ветви зависит от условия на входе ветвления и поступивших сюда данных.
- **Цикл.** Предполагает возможность многократного повторения определенных действий. Количество повторений зависит от условия цикла.
- **Функция (подпрограмма).** Команды, отделенные от основной программы, выполняются лишь в случае их вызова из основной программы (из любого ее места). Одна и та же функция может вызываться из основной программы сколько угодно раз.



# Описание различных алгоритмических структур на языке блок-схем



## Ветвление if-else

Если выражение-условие возвращает true (правда), то выполнение алгоритма идет по ветке «Да», если условие не выполняется (false), то выполнение идет по ветке «Нет». При любом результате выражения-условия нельзя вернуться в основную ветку программы, минуя дополнительные действия.

# Описание различных алгоритмических структур на языке блок-схем



## Ветвление if-elif-else

Количество условий может быть различно. Если выполняется первое, то после выполнения действий, программа переходит к основной ветке, не проверяя дальнейшие условия. Если первое условие возвращает ложь, то проверяется второе условие. Если второе условие возвращает правду, то выполняются действия, включенные в вторую ветку конструкции. Последнее условие проверяется лишь в том случае, если ни одно до него не дало в результате true. Данную алгоритмическую конструкцию (if – elif – else) не следует путать с алгоритмической конструкцией «Выбор».

# Описание различных алгоритмических структур на языке блок-схем



## Цикл while

Пока условие выполняется (результат логического выражения дает true), будут выполняться действия тела цикла. После очередного выполнения вложенных действий условие снова проверяется. Для того чтобы выполнение алгоритма не зациклилось, в теле цикла (помимо прочих действий) должно быть выражение, в результате выполнения которого будет изменяться переменная, используемая в условии. Тело цикла может ни разу не выполниться, если условие с самого начала давало false.

# Описание различных алгоритмических структур на языке блок-схем



## Цикл do

В этом цикле первый раз условие проверяется лишь после выполнения действий тела цикла. Если условие возвращает true, то выражения-действия повторяются снова. Каким бы ни было условие, тело данного цикла хотя бы раз, но выполнится.

# Описание различных алгоритмических структур на языке блок-схем



## Цикл for

Данный цикл также называют циклом «Для» (for). В его заголовке указывается три параметра: начальное значение переменной (от), конечно значение (до) и ее изменение с помощью арифметической операции на каждом «обороте» цикла (шаг).

# Самостоятельная работа №3

1. Что такое база данных?
2. Перечислите свойства алгоритма.
3. Укажите основные способы представления алгоритмов.