Тема: «Понятие алгоритма и основные алгоритмические структуры»

# Алгоритм - это точное предписание исполнителю совершить указанную последовательность действий для получения решения задачи за конечное число шагов.

**Свойства алгоритмов**

К основным свойствам алгоритмов относятся следующие свойства:

1. Понятность для исполнителя, ─ исполнитель алгоритма должен понимать, как его выполнять. Иными словами, имея алгоритм и произвольный вариант исходных данных, исполнитель должен знать, как надо действовать для выполнения этого алгоритма.
2. Дискретность (прерывность, раздельность) - алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов (этапов).
3. 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.

1. Релевантность (или конечность) состоит в том, что за конечное число шагов алгоритм либо должен приводит к решению задачи, либо после конечного числа шагов останавливаться из-за невозможности получить решение с выдачей соответствующего сообщения, либо неограниченно продолжаться в течение времени, отведенного для исполнения алгоритма, с выдачей промежуточных результатов.
2. Массовость означает, что алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

# Правила построения алгоритма

Чтобы алгоритм выполнил свое предназначение, eго необходимо строить по определенным правилам. В этом смысле нужно говорить не о свойствах алгоритма, а о правилах построения алгоритма, или о требованиях, предъявляемых к алгоритму.

**Первое правило ─ при построении алгоритма, прежде всего, необходимо задать множество объектов, с которыми будет работать алгоритм.** Формализованное (закодированное) представление этих объектов носит название данных. **Алгоритм приступает к работе с некоторым набором данных, которые называются входными, и в результате своей работы выдает данные, которые**

**называются входными.** Таким образом, алгоритм преобразует входные данные в выходные.

Это правило позволяет сразу отделить алгоритмы от "методов" и “способов”. Пока мы не имеем формализованных входных данных, мы не можем построить алгоритм.

**Второе правило ─ для работы алгоритма требуется память**. В памяти размещаются входные данные, с которыми алгоритм начинает работать, промежуточные данные и выходные данные, которые являются результатом работы алгоритма. Память является дискретной, т.е. состоящей из отдельных ячеек. Поименованная ячейка памяти носит название переменной. В теории алгоритмов размеры памяти не ограничиваются, т. е. считается, что мы можем предоставить алгоритму любой необходимый для работы объем памяти.

**Третье правило ─ дискретность. Алгоритм строится из отдельных шагов** (действий, операций, команд). Множество шагов, из которых составлен алгоритм, конечно.

# Четвертое правило ─ детерменированность. После каждого шага необходимо указывать, какой шаг выполняется следующим, либо давать команду остановки.

**Пятое правило ─ сходимость (результативность).** Алгоритм должен завершать работу после конечного числа шагов. При этом необходимо указать, что считать результатом работы алгоритма.

Итак, алгоритм ─ неопределяемое понятие теории алгоритмов. Алгоритм каждому определенному набору входных данных ставит в соответствие некоторый набор выходных данных, т. е. вычисляет (реализует) функцию. При рассмотрении конкретных вопросов в теории алгоритмов всегда имеется в виду какая-то конкретная модель алгоритма.

# Формы записи алгоритма

Алгоритм, как последовательность шагов или инструкций, может быть представлен в различных формах*.*

На практике наиболее распространены следующие ***формы представления алгоритмов:***

* 1. словесная *(запись на естественном языке);*
	2. графическая *(изображения из графических символов);*
	3. псевдокоды (*полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);*
	4. программная *(тексты на языках программирования).*

# Словесная форма записи алгоритмов

Словесный способ записи алгоритмов представляет собой описание последовательных этапов обработки данных. Алгоритм задается в произвольном изложении на естественном языке.

Например. Записать алгоритм нахождения наибольшего общего делителя (НОД) двух натуральных чисел (алгоритм Эвклида).

Алгоритм может быть следующим:

1. задать два числа;
2. если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение

алгоритма;

* 1. определить большее из чисел;
	2. заменишь большее из чисел разностью большего и меньшего из чисел;
	3. повторить алгоритм с шага 2.

Описанный алгоритм, применим к любым натуральным числам и должен приводить к решению поставленной задачи.

Словесный способ не получил широкого распространения из-за сле- дующих ***недостатков***:

1. Строго не формализуем;
2. Страдает многословностью записей;
3. Допускает неоднозначность толкования отдельных предписаний.

# Графическая форма записи алгоритмов

Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным. ***При графическом представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий.***

***Такое графическое представление называется схемой алгоритма или блок- схемой.*** В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде блочного символа. Блочные символы соединяются линиями переходов, определяющими очередность выполнения действий. В приведены наиболее часто употребляемые блочные символы.

|  |  |  |
| --- | --- | --- |
| Название символа | Обозначения и пример заполнения | Пояснения |
| Процесс | https://studfile.net/html/2706/64/html_U2l5TOVswd.LsMX/img-Er2ocu.jpg | Вычислительное действие или последовательность действий |
| Решение | https://studfile.net/html/2706/64/html_U2l5TOVswd.LsMX/img-t_Me1Z.jpg | Проверка условий |
| Модификация | https://studfile.net/html/2706/64/html_U2l5TOVswd.LsMX/img-HQARF_.jpg | Начало цикла |
| Предопределенный процесс | https://studfile.net/html/2706/64/html_U2l5TOVswd.LsMX/img-Gp9eP8.jpg | Вычисления по подпрограмме, стандартной подпрограмме |
| Ввод-вывод | https://studfile.net/html/2706/64/html_U2l5TOVswd.LsMX/img-GcML86.jpg | Ввод-вывод в общем виде |
| Пуск-остановка | https://studfile.net/html/2706/64/html_U2l5TOVswd.LsMX/img-AtywvW.jpg | Начало, конец алгоритма, вход и выход в подпрограмму |
| Документ | https://studfile.net/html/2706/64/html_U2l5TOVswd.LsMX/img-JpT0PP.jpg | Вывод результатов на печать |

# Псевдокод

*Псевдокод представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов.*

Псевдокод занимает промежуточное место между естественным и формальным языками. С одной стороны, он близок к обычному естественному языку, поэтому алгоритмы могут па нем 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.

В псевдокоде не приняты строгие синтаксические правила для записи команд, присущие формальным языкам, что облегчает запись алгоритма на стадии его проектирования и дает возможность использовать более широкий набор команд, рассчитанный па абстрактного исполнителя.

Однако в псевдокоде обычно имеются некоторые конструкции, присущие формальным языкам, что облегчает переход от записи на псевдокоде к записи алгоритма на формальном языке. В частности, в псевдокоде, так же, как и в формальных языках, есть служебные слова, смысл которых определен раз и навсегда. Они выделяются в печатном тексте жирным шрифтом, а в рукописном тексте подчеркиваются.

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

# Программная форма записи алгоритма

Программный способ записи алгоритма представляет собой написанный на языке программирования код программы.

**Контрольные вопросы:**

1. Запишите определение алгоритма
2. Перечислите свойства алгоритма
3. Перечислите формы записи алгоритма
4. Что представляет из себя псевдокод?