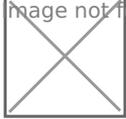


## Содержание:

image not found or type unknown



## Введение

Каждый из нас ежедневно решает задачи различной сложности: как быстрее добраться в школу или на работу в условиях нехватки времени; в каком порядке выполнить дела, намеченные на текущий день, и т.д. Некоторые задачи настолько сложны, что требуют длительных размышлений для нахождения решения, другие задачи мы решаем автоматически, так как выполняем их ежедневно на протяжении многих лет. Но в любом случае решение каждой задачи можно подразделить на простые этапы.

Целью реферата является раскрытие базовых знаний об элементах теории алгоритмов.

Для решения поставленной цели необходимо выполнить следующие задачи:

- Изучить и проанализировать литературу;
- Раскрыть базовые понятия элементов теории алгоритмов;
- Рассмотреть свойства и виды алгоритмов;
- Сформировать представление о способах записи алгоритмов.

Я выбрал данную тему из-за ее актуальности.

### Понятие алгоритма

Знакомство с понятием алгоритма я предлагаю начать с рассмотрения примера.

Предположим, вы хотите вылепить из пластилина дракона. Результат во многом будет зависеть от вашего умения и опыта. Однако достичь поставленной цели окажется гораздо легче, если вы предварительно наметите план действий, например следующий:

1. Изучить образ дракона по имеющейся картинке.
2. Вылепить голову.
3. Вылепить туловище.

4. Вылепить хвост.
5. Вылепить четыре ноги.
6. Сравнивая с картинкой, уточнить детали каждой вылепленной части дракона.

Следуя подготовленному плану, любой человек, даже не обладающий художественными способностями, но имеющий терпение, обязательно получит хороший результат. Подобный план с подробным описанием действий, необходимых для получения ожидаемого результата, получил название алгоритма.

Понятие алгоритма, является фундаментальным понятием математики и информатики, возникло задолго до появления вычислительных машин. Само же слово алгоритм появилось в средние века, когда европейцы познакомились со способами выполнения арифметических действий в десятичной системе, описанными узбекским математиком Мухаммедом бен Муса аль-Хорезми. Слово алгоритм – европеизированное произношение слов аль-Хорезми.

Первоначально под словом алгоритм понимали способ выполнения арифметических действий над десятичными числами. В дальнейшем это понятие стали использовать для обозначения любой последовательности действий, приводящих к решению поставленной задачи.

Область математики, известная как теория алгоритмов, посвящена исследованию свойств, способов записи, видов и сферы применения различных алгоритмов. Научное определение понятия алгоритма дал А. Черч в 1930 году. Позже и другие математики вносили свои уточнения в это определение. В школьном курсе информатики обычно пользуются следующими определениями:

- *Алгоритм – описание последовательности действий, строгое исполнение которых приводит к решению последовательной задачи за конечное число шагов.*
- *Алгоритм – Это конечная система правил, сформулированная на языке исполнителя, которая определяет последовательность перехода от допустимых исходных данных к конечному результату и которая обладает определенными свойствами.*
- *Алгоритмизация – процесс разработки алгоритма для решения задачи.*

Любой алгоритм существует не сам по себе, а предназначен для определенного исполнителя. Алгоритм описывается в командах исполнителя, который этот алгоритм будет выполнять. Объекты, над которыми исполнитель может совершать действия, образуют так называемую среду исполнителя. Исходные данные и

результаты любого алгоритма всегда принадлежат среде того исполнителя, для которого предназначен алгоритм.

Значение слова алгоритм очень схоже со значением слов рецепт, процесс, метод, способ. Однако любой алгоритм, в отличие от рецепта или способа обязательно обладает свойствами.

## **Свойства алгоритмов**

Мир алгоритмов очень разнообразен. Несмотря на это, удастся выделить общие свойства, которыми обладает любой алгоритм.

Обычно мы выполняем привычные действия не задумываясь, механически. Например, вы хорошо знаете, как открыть дверь ключом. Однако, чтобы научить этому малыша, придется четко разъяснить и самим действия, и порядок их выполнения:

1. Достать ключ из кармана.
2. Вставить ключ в замочную скважину.
3. Повернуть ключ два раза против часовой стрелки.
4. Вынуть ключ.

Представьте, что вас пригласили в гости и подробно объяснили, как добраться:

1. Выйти из дома.
2. Повернуть направо.
3. Пройти два квартала до остановки.
4. Сесть в автобус №5, идущий к центру города.
5. Проехать три остановки.
6. Выйти из автобуса.
7. Найти по указанному адресу дом и квартиру.

Это не что иное, как алгоритм. Внимательно анализируя эти примеры, можно найти много общего, несмотря на значительное различие в сути самих действий. Эти общие характеристики называют свойствами алгоритма. Рассмотрим их.

## **Дискретность**

Дискретность (от лат. discretus – разделенный, прерывистый). Это свойство указывает, что любой алгоритм должен состоять из конечных действий, следующих в определенном порядке. В приведенных выше алгоритмах общим является необходимость строгого соблюдения последовательности выполнения действий. Попробуем переставить в первом примере второе и третье действия. Вы, конечно, сможете выполнить и этот алгоритм, но дверь вряд ли откроется. А если поменять местами, предположим, пятое и второе действия во втором примере, алгоритм станет не выполнимым.

## Детерминированность

Детерминированность (от лат. determinate – определенность, точность). Это свойство указывает, что любое действие алгоритма должно быть строго и недвусмысленно определено в каждом случае. Например, если к остановке подходят автобусы разных маршрутов, то в алгоритме должен быть указан конкретный номер маршрута – 5. Кроме того, необходимо указать точное количество остановок, которое надо проехать, - скажем, три.

## Конечность

Это свойство определяет, что каждое действие в отдельности и алгоритм в целом должны иметь возможность завершения. В приведенных примерах каждое описанное действие реально и может быть выполнено. Поэтому и алгоритм имеет предел, то есть конечен.

## Массовость

Это свойство показывает, что один и тот же алгоритм можно использовать с разными исходными данными. Ниже описан алгоритм приготовления любого бутерброда.

1. Отрезать ломтик хлеба.
2. Намазать его маслом.
3. Отрезать кусок любого другого пищевого продукта (колбасы, сыра).
4. Наложить отрезанный кусок на ломтик хлеба.

# Результативность

Это свойство требует, чтобы в алгоритме не было ошибок. Например, рассмотрим алгоритм нахождения большего из двух заданных чисел А и В.

1. Из числа А вычесть число В.
2. Если получилось отрицательное значение, то сообщить, что число В больше.
3. Если получилось положительное значение, то сообщить, что число А больше.

При всей простоте и очевидности алгоритма, не каждый сразу поймет его ошибочность. Ведь если оба числа равны, то не получится никакого общения. Значит, надо обязательно предусмотреть это вариант, например:

1. Из числа А вычесть число В.
2. Если получилось отрицательное значение, то сообщить, что число В больше.
3. Если получилось положительное значение, то сообщить, что число А больше.
4. Если получилось ноль, то сообщить, что числа равны.

Таким образом, для любого алгоритма характерны следующие свойства: дискретность, детерминированность, конечность, массовость, результативность.

## Виды алгоритмов

Описание действий следует последовательно друг за другом. Однако очередность выполнения этих действий может быть изменена, если в алгоритме предусмотрен анализ некоторого условия. Путем включения условий создаются алгоритмы с различной структурой, в которой всегда можно выделить несколько типовых конструкций: линейную, циклическую, разветвляющуюся, вспомогательную.

Хочется заметить, что в 1969 году Эдсгер В. Дейкстра в статье «Структуры данных и алгоритмы» доказал, что для записи любого алгоритма достаточно трех основных алгоритмических конструкций: линейных, ветвящихся, циклических. Рассмотрим эти конструкции.

## Линейный алгоритм

Линейный (последовательный) алгоритм – описание действий, которые выполняются однократно в заданном порядке. Такими являются алгоритмы отпирания дверей, заваривания чая, приготовление одного бутерброда. Линейный алгоритм применяется при вычислении арифметического выражения, если в нем используются только действия сложения и вычитания.

## **Циклический алгоритм**

Многие процессы в окружающем мире основаны на многократном повторении одной и той же последовательности действий. Каждый год наступает весна, лето, осень и зима. Жизнь растений в течение года проходит одни и те же циклы. Подсчитывая число полных поворотов минутной или часовой стрелки, человек измеряет время.

Циклический алгоритм – описание действий, которые должны повторяться указанное число раз или пока не выполнено заданное условие.

Перечень повторяющихся действий называется телом цикла.

## **Разветвляющийся алгоритм**

Разветвляющийся алгоритм – алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий.

Вспомним сюжет из русской сказки. Царевич останавливается у развилки дороги и видит камень с надписью: «Направо пойдешь – коня потеряешь, налево пойдешь – сам пропадешь...»

Подобная ситуация, заставляющая принимать решение в зависимости от некоторого условия, постоянно встречается в повседневной жизни.

- Если пошел дождь, то надо открыть зонт.
- Если болит горло, то прогулку следует отменить.
- Если прозвенел будильник, то надо вставать и идти в школу и т.д.

Существует специальный раздел математики – формальная логика, которая объясняет, как выстраивать цепочку рассуждений, чтобы прийти к правильному выводу.

Логика учит правильно формулировать условие, под которым понимается предложение, начинающееся со слова «если» и заканчивающееся перед словом «то». Условие может принимать значение «истина», когда оно выполнено, или «ложь», когда оно не выполнено. От значения условия зависит наше дальнейшее действие.

## Вспомогательный алгоритм

Вспомогательный алгоритм – алгоритм, который можно использовать в других алгоритмах, указав только его имя.

Вспомогательному алгоритму должно быть присвоено имя.

Допустим, вы хотите научиться жонглировать двумя или даже тремя мячами. Если внимательно приглядеться к действиям профессионального артиста и попытаться понять, как это ему удается делать, то оказывается – секрет в том. Что надо научиться искусно выполнять несколько определенных движений, которым присвоим соответствующие названия:

- **Бросок левой** – подбросить мяч левой рукой.
- **Бросок правой** – подбросить мяч правой рукой.
- **Захват левой** – поймать мяч правой рукой.
- **Захват правой** – поймать мяч правой рукой.

Тогда алгоритм жонглирования можно записать с помощью вспомогательных алгоритмов выполнения отдельных действий в следующем виде:

1. Когда летящий шарик начинает поворачивать к правой руке, выполнить **Бросок правой** и **Захват правой**.
2. Когда летящий шарик начинает поворачивать к правой руке, выполнить **Бросок левой** и **Захват левой**.

Понятие вспомогательного алгоритма значительно упрощает процесс алгоритмизации задачи. Создавая алгоритм, вы описываете действие, результатом которого должно быть достижение поставленной цели. Этому алгоритму должно быть присвоено уникальное имя.

4. Способы описания алгоритмов

Любой сложный алгоритм можно составить, используя в разных комбинациях только типовые алгоритмические конструкции. Формы же представления этих алгоритмов могут быть разными, например:

- Текстовая форма записи (словесный метод).
- Запись в виде блок-схемы.
- Запись алгоритма на каком-либо алгоритмическом языке.
- Представление алгоритма в виде машины Тьюринга или машины Поста.

### Словесный способ

Способ описание на естественном языке, как делалось в предыдущих примерах. Он очень удобен, когда следует приближенно описать суть алгоритма.

### Блок-схемы

Для более наглядного представления алгоритма широко используется именно эта форма, которая составляется из стандартных графических объектов (таблица 1).

**Таблица 1** Стандартные графические объекты блок-схемы

<b>Вид стандартного графического объекта</b>	<b>Назначение</b>
<b>Начало</b>	Начало алгоритма
<b>Конец</b>	Конец алгоритма
<b>Гуляю</b>	Выполняемое действие записывается внутри прямоугольника
<b>Встречу?</b>	Условие выполнения действий записывается внутри ромба

Последовательность выполнения действий:

- влево и вверх – линия со стрелкой,
- вниз и вправо – линия без стрелки

На приведенных ниже рисунках 1–5 представлены блок-схемы типовых алгоритмических конструкций.

Начало

После школы иду гулять

Возвращаюсь домой

Делаю

уроки

Конец

*Рис. 1. Линейная алгоритмическая конструкция*

Начало

После школы иду гулять

Возвращаюсь домой

Делаю

уроки

Конец

*Рис. 1. Линейная алгоритмическая конструкция*

вход

Меньше полуночи?

да

Смотрю

TV

выход

*Рис. 2. Циклическая алгоритмическая конструкция, в которой условие поставлено в начале цикла*

нет

вход

Точить карандаш

Пустая коробка?

выход

да

нет

*Рис. 3. Циклическая алгоритмическая конструкция, в которой условие поставлено в конце цикла*

вход

Встречу?

Скажу

выход

*Рис. 4. Неполная форма разветвляющегося алгоритма*

вход

Встречу?

Скажу

выход

*Рис. 5. Полная форма разветвляющегося алгоритма*

Зайду сам

## **Заключение**

Понятие алгоритма является одним из фундаментальных понятий не только математики. Работы по формализации этого понятия привели к появлению нового научного направления – теории алгоритмов. В данном реферате вы познакомились с некоторыми элементами этой науки.

## **Литература**

1. *Аляев Ю. А.* Алгоритмизация и языки программирования Pascal, C++, Visual Basic / Ю. А. Аляев. – М., 2002 г.
2. *Макарова Н. В.* Информатика 7-9 класс Базовый курс. Теория / Н. В. Макарова. – М., 2003 г.
3. *Семакин И. Г.* Информатика Базовый курс 7-9 класс / И. Г. Семакин. – М., 1999 г.
4. *Стариченко Б. Е.* Теоретические основы информатики / Б. Е. Стариченко. – М., 2004 г.
5. *Фалина И. Н.* Элементы теории алгоритмов / И. Н. Фалина // Информатика. – 2004 г. – №3 (435). – 2-11.