Алгоритм
и его свойства

Возникновение понятия "алгоритм"

Слово **"алгоритм"** происходит от имени выдающегося математика средневекового Востока **Мухаммеда аль-Хорезми** (787-850). Около 825 года он написал книгу, в которой им были предложены приёмы выполнения арифметических вычислений с многозначными числами.

В первой половине XII века книга аль-Хорезми в латинском переводе проникла в Европу. Переводчик (имя его неизвестно) дал ей название **Algoritmi de numero Indorum** («Алгоритми о счёте индийском»). Слово **algorism** (или algorismus) обрело значение способа выполнения арифметических действий посредством арабских цифр, то есть на бумаге, без использования абака. Именно в таком значении оно вошло во многие европейские языки

Таким образом, сочинения по искусству счёта стали называть алгоритмами.

Алгоритм - это понятное и точное предписание исполнителю выполнить конечную последовательность команд, приводящих от исходных данных к искомому результату

Каждый человек в повседневной жизни выполняет огромное количество алгоритмов. Например, процесс приготовления чая можно описать следующим алгоритмом:

1. вскипятить воду в чайнике,
2. положить в пустую чайную чашку пакетик чая,
3. залить чашку горячей водой,
4. подождать 1 минуту,
5. вытащить пакетик,
6. положить в чашку 2 чайных ложки сахара,
7. размешать сахар.

Свойства алгоритма

Любой алгоритм должен удовлетворять четырём основным свойствам:

Конечность алгоритма означает, что за конечное число шагов должен быть получен результат. Поэтому иногда это свойство называют **результативностью**.

Пример:

Пусть имеется последовательность команд:

1. Взять книгу,

2. Открыть первую страницу.

3. Пока не конец книги выполнять следующие действия:

3.1 Прочитать текст

3.2 Перелистнуть книгу на следующую страницу

3.3 Прочитать текст

3.4 Открыть первую страницу

Легко догадаться, что данная последовательность команд будет выполняться бесконечно и поэтому алгоритмом не является.

ПОДСКАЗКА Что надо изменить в алгоритме, что бы он стал конечным?

Конечность алгоритма означает, что за конечное число шагов должен быть получен результат. Поэтому иногда это свойство называют **результативностью**.

Пример:

Пусть имеется последовательность команд:

1. Взять книгу,

2. Открыть первую страницу.

3. Пока не конец книги выполнять следующие действия:

3.1 Прочитать текст

3.2 Перелистнуть книгу на следующую страницу

3.3 Прочитать текст

3.4 Открыть первую страницу

Легко догадаться, что данная последовательность команд будет выполняться бесконечно и поэтому алгоритмом не является.

Надо исключить из алгоритма пункты 3.3 и 3.4

ПОДСКАЗКА Что надо изменить в алгоритме, что бы он стал конечным?

Дискретность означает, что алгоритм должен быть разбит на последовательность отдельно выполняемых шагов.

Пусть необходимо решить следующий пример: **(80+10)-5*(3+5)=**

Запишем алгоритм решения примера, разбив его на шаги:

1. **Вычислить 80+10**
2. **Вычислить 3+5**
3. **Умножить 5 на результат предыдущего действия**
4. **Вычесть из результата 1-го действия результат 3-го действия**

В результате выполнения алгоритма получим 50.

Если в данном алгоритме начать, например, выполнять четвёртое действие, не дожидаясь окончания выполнения третьего, то результат не может быть получен.

Понятность алгоритма означает, что алгоритм должен содержать только те команды, которые входят в СКИ.

Рассмотрим алгоритм:

1. **Пойти на кухню**
2. **Вскипятить чайник**
3. **Насыпать в чашку 1 чайную ложку кофе**
4. **Положить в чашку 3 чайных ложки сахара**
5. **Налить полную чашку кипячёной воды**

Очевидно, что он легко может быть выполнен 10-летней девочкой, которая понимает все команды, входящие в данный алгоритм. Однако, для 10-месячного малыша данный алгоритм будет непонятен.

Точность алгоритма означает, что любая его команда должна определять однозначное действие исполнителя. Иными словами, алгоритм не должен быть рассчитан на принятие каких-либо самостоятельных решений исполнителем.

Рассмотрим следующий алгоритм, описывающий, как добраться до стадиона :

1. **Идти прямо**
2. **Повернуть**
3. **Идти прямо**
4. **Сесть на автобус**
5. **Доехать до остановки "Стадион"**

Данный алгоритм не уточняет, какое расстояние нужно пройти прямо, в какую сторону повернуть, на какой автобус сесть, поэтому разные исполнители будут выполнять его по-разному и цель вряд ли будет достигнута.

Для успешной работы алгоритма необходимо также, чтобы имелся **полный набор исходных данных**, необходимый для его выполнения. Если исходные данные неполные, то либо задачу вообще нельзя решить, либо она будет иметь неоднозначное решение.

Пусть вы пришли в магазин самообслуживания и решили подсчитать стоимость предполагаемых покупок, чтобы узнать, хватит ли вам денег. Вам нужно купить 2 кг сахарного песка, 3 кг муки и 2 батона хлеба. Тогда для вычисления общей стоимости вам надо:

1. **Умножить стоимость 1 кг сахарного песка на 2**
2. **Умножить стоимость 1 кг муки на 3**
3. **Умножить стоимость 1 батона на 2**
4. **Сложить все полученные результаты**

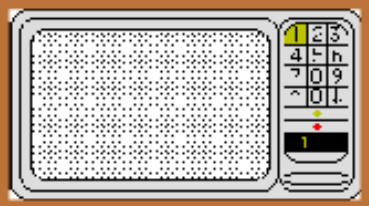
Если по какой-либо причине не будет вывешен ценник хотя бы для одного из продуктов, то данная задача не сможет быть решена.

Исполнители алгоритмов

Исполнитель алгоритма - это объект, способный выполнять определённый набор команд.

Исполнителями алгоритмов могут быть как живые существа, так и технические устройства.

Примеры исполнителей алгоритмов:



микроволновая печь

телевизор

дрессированный лев

солдат

робот

автомобиль

лошадь

Система команд исполнителя алгоритмов

Любой исполнитель алгоритмов умеет выполнять лишь ограниченный набор команд. Этот набор называется системой команд исполнителя.

Система команд исполнителя (СКИ) - это набор команд, которые может выполнить конкретный исполнитель.

Пример некоторых команд из СКИ исполнителя "DVD-плеер":

- **Начать воспроизведение**
- **Пауза**
- **Остановить воспроизведение**
- **Увеличить громкость**
- **Уменьшить громкость**
- **Ускоренное воспроизведение назад**
- **Ускоренное воспроизведение вперед**
- **Покадровое воспроизведение**
- **Выбор языка**

Задание на дом

п. 3 (устно ответить на вопросы после параграфа)