

Принципы обработки информации компьютером. Арифметические и логические основы работы компьютера. Алгоритмы и способы их описания.

Принципы обработки информации компьютером.

Компьютер или ЭВМ (электронно-вычислительная машина)- это универсальное техническое средство для автоматической обработки информации.

Аппаратное обеспечение компьютера- это все устройства, входящие в его состав и обеспечивающие его исправную работу.

Несмотря на разнообразие компьютеров в современном мире, все они строятся по единой принципиальной схеме, основанной на фундаменте идеи программного управления [Чарльза Бэббиджа](#) (середина XIX в). Эта идея была реализована при создании первой ЭВМ ENIAC в 1946 году коллективом учёных и инженеров под руководством известного американского математика [Джона фон Неймана](#), сформулировавшего *концепцию ЭВМ с вводимыми в память программами и числами -программный принцип*.

Главные элементы концепции:

1. двоичное кодирование информации;
2. программное управление;
3. принцип хранимой программы;
4. принцип параллельной организации вычислений, согласно которому операции над числом проводятся по всем его разрядам одновременно.

Алгоритмы и способы их описания.

Компьютер как исполнитель команд. Программный принцип работы компьютера.

Алгебра логики (булева алгебра) – это раздел математики, возникший в XIX веке благодаря усилиям английского математика Дж. Буля. Поначалу булева алгебра не имела никакого практического значения. Однако уже в XX веке ее положения нашли применение в разработке различных электронных схем. Законы и аппарат алгебры логики стали использоваться при проектировании различных частей компьютеров (память, процессор).

Алгебра логики оперирует с высказываниями. Под **высказыванием** понимают повествовательное предложение, относительно которого имеет смысл говорить, истинно оно или ложно. Над высказываниями можно производить определенные логические операции, в результате которых получаются новые высказывания. Наиболее часто используются логические операции, выражаемые словами «не», «и», «или».

Логические операции удобно описывать так называемыми **таблицами истинности**, в которых отражают результаты вычислений сложных высказываний при различных

значениях исходных простых высказываний. Простые высказывания обозначаются переменными (например, А и В).

Конъюнкция (логическое умножение). Сложное высказывание $A \& B$ истинно только в том случае, когда истинны оба входящих в него высказывания. Истинность такого высказывания задается следующей таблицей:

Обозначим 0 – ложь, 1 – истина

A	B	A&B
0	0	0
0	1	0
1	0	0
1	1	1

Дизъюнкция (логическое сложение). Сложное высказывание $A \cup B$ истинно, если истинно хотя бы одно из входящих в него высказываний. Таблица истинности для логической суммы высказываний имеет вид:

A	B	A ∪ B
0	0	0
0	1	1
1	0	1
1	1	1

Инверсия (логическое отрицание). Присоединение частицы НЕ (NOT) к данному высказыванию называется операцией отрицания (инверсии). Она обозначается \bar{A} (или $\neg A$) и читается не А . Если высказывание А истинно, то В ложно, и наоборот. Таблица истинности в этом случае имеет вид

A	$\neg A$
false	true
true	false

Алгоритм – система точных и понятных предписаний (команд, инструкций, директив) о содержании и последовательности выполнения конечного числа действий, необходимых для

решения любой задачи данного типа. Как всякий объект, алгоритм имеет название (имя). Также алгоритм имеет начало и конец.

В качестве исполнителя алгоритмов можно рассматривать человека, любые технические устройства, среди которых особое место занимает компьютер. Компьютер может выполнять только точно определенные операции, в отличие от человека, получившего команду и имеющего возможность сориентироваться в ситуации.

Алгоритм обладает следующими *свойствами*.

1. Дискретность (от лат. *discretus* – разделенный, прерывистый) указывает, что любой алгоритм должен состоять из конкретных действий, следующих в определенном порядке.
2. Детерминированность (от лат. *determinate* – определенность, точность) указывает, что любое действие алгоритма должно быть строго и недвусмысленно определено в каждом случае.
3. Конечность определяет, что каждое действие в отдельности и алгоритм в целом должны иметь возможность завершения.
4. Результативность требует, чтобы в алгоритме не было ошибок, т.е. при точном исполнении всех команд процесс решения задачи должен прекратиться за конечное число шагов и при этом должен быть получен ответ.
5. Массовость заключается в возможности применения алгоритма к целому классу однотипных задач, различающихся конкретными значениями исходных данных (разработка в общем виде).

Способы описания алгоритмов

- словесный (на естественном языке);
- графический (с помощью стандартных графических объектов (геометрических фигур) – блок-схемы);
- программный (с помощью языков программирования)