
*Принципы обработки информации
компьютером.*

*Арифметические и логические
основы работы компьютера*

Компьютер – это техническое средство преобразования информации, в основу работы которого заложены те же принципы обработки электрических сигналов, что и в любом электронном устройстве:

1. входная информация, представленная различными физическими процессами, как электрической, так и неэлектрической природы (буквами, цифрами, звуковыми сигналами и т.д.), преобразуется в электрический сигнал;
2. сигналы обрабатываются в блоке обработки;
3. с помощью преобразователя выходных сигналов обработанные сигналы преобразуются в неэлектрические сигналы (изображения на экране).

С позиции функционального назначения компьютер – это система, состоящая из 4-х основных устройств, выполняющих определенные функции: запоминающего устройства или памяти, которая разделяется на оперативную и постоянную, арифметико-логического устройства (АЛУ), устройства управления (УУ) и устройства ввода-вывода (УВВ). Рассмотрим их роль и назначение.

Для характеристики памяти используются следующие параметры:

1. емкость памяти – максимальное количество хранимой информации в байтах;
2. быстродействие памяти – время обращения к памяти, определяемое временем считывания или временем записи информации.

Арифметико-логическое устройство (АЛУ). Производит арифметические и логические действия.

Следует отметить, что любую арифметическую операцию можно реализовать с использованием операции сложения.

Сложная логическая задача раскладывается на более простые задачи, где достаточно анализировать только два уровня: ДА и НЕТ.

Устройство управления (УУ) управляет всем ходом вычислительного и логического процесса в компьютере, т.е. выполняет функции "регулирующего движения" информации. УУ читает команду, расшифровывает ее и подключает необходимые цепи для ее выполнения. Считывание следующей команды происходит автоматически.

Фактически УУ выполняет следующий цикл действий:

1. формирование адреса очередной команды;
2. чтение команды из памяти и ее расшифровка;
3. выполнение команды.

В современных компьютерах функции УУ и АЛУ выполняет одно устройство, называемое центральным процессором.

Принципы обработки информации компьютером

Между алгеброй логики и двоичным кодированием существует следующая связь: математический аппарат алгебры логики очень удобен для описания того, как функционируют аппаратные средства компьютера, поскольку основной системой счисления в компьютере является двоичная, в которой используются цифры 1 и 0, а значений логических переменных тоже два: “1” и “0”.

Принципы обработки информации компьютером

Выводы:

- одни и те же устройства компьютера могут применяться для обработки и хранения как числовой информации, представленной в двоичной системе счисления, так и логических переменных;
- на этапе конструирования аппаратных средств алгебра логики позволяет значительно упростить логические функции, описывающие функционирование схем компьютера, и, следовательно, уменьшить число основных узлов компьютера.

Логические операции

1. Логическое умножение (конъюнкция) «И»

A И B или **A&B** или **A∧B**

A И B истинно тогда и только тогда,
когда оба высказывания **A** и **B**
истинны.

Примеры: **0 и 0=0** **0 и 1=0**

1 и 0=0 **1 и 1=1**

Техническая реализация И

- два последовательно соединенных ключа:

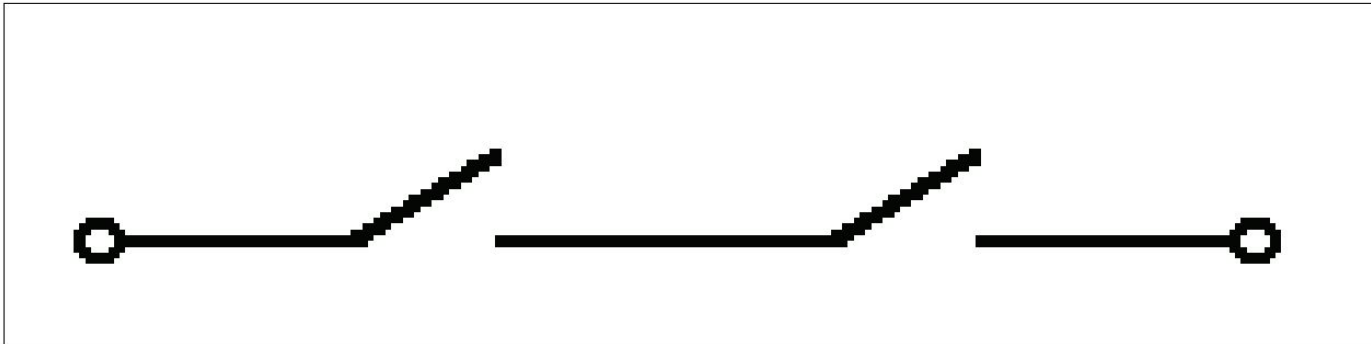


Таблица истинности

A	B	A И B			
0	0	0			
0	1	0			
1	0	0			
1	1	1			

Логические операции

2. Логическое сложение «ИЛИ»

A ИЛИ B или $A \vee B$ или $A+B$

A **ИЛИ** B ложно тогда и только тогда,
когда оба высказывания A и B ложны.

Примеры:

0 или **1=1** **1** или **0=1**

0 или **0=0** **1** или **1=1**

Техническая реализация ИЛИ

- два параллельно соединенных ключа:

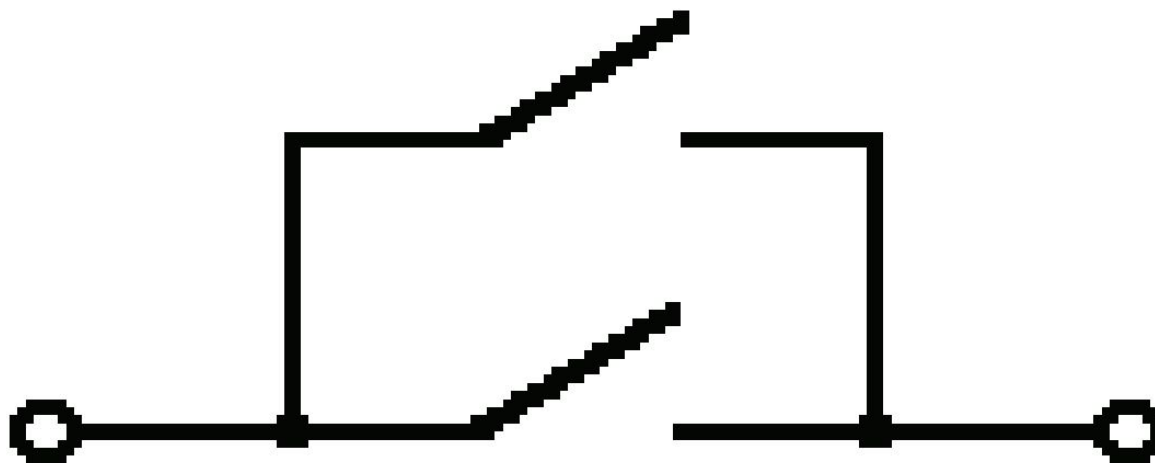


Таблица истинности

A	B	A И B	A ИЛИ B		
0	0	0	0		
0	1	0	1		
1	0	0	1		
1	1	1	1		

Логические операции

3. Логическое отрицание (инверсия) «НЕ»

НЕ A или $\neg A$

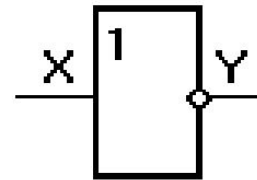
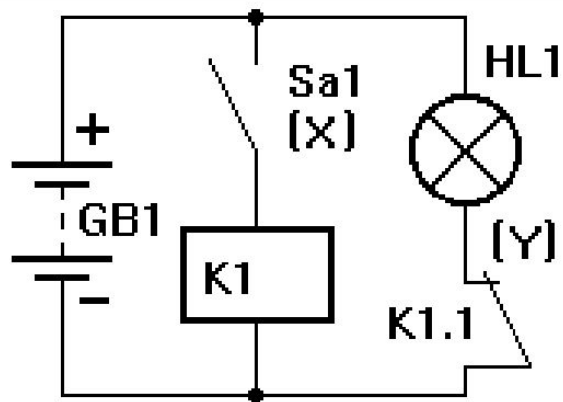
Логическое отрицание (инверсия) делает истинное выражение ложным и, наоборот, ложное – истинным.

Таблица истинности

A	B	A И B	A ИЛИ B	НЕ A	
0	0	0	0	1	
0	1	0	1	1	
1	0	0	1	0	
1	1	1	1	0	

Техническая реализация НЕ

- при отсутствии электрического тока через обмотку реле контакты реле замкнуты,
- при протекании достаточного тока через обмотку реле контакты реле разомкнуты:



X	Y
0	1
1	0

Логические операции

4. Логическое следование (импликация)

$$A \rightarrow B$$

$A \rightarrow B$ ложно только тогда, когда A истинно, а B ложно.

Импликация выражается через дизъюнкцию и отрицание:

$$A \rightarrow B = \text{НЕ } A \text{ ИЛИ } B$$

Таблица истинности

A	B	A И B	A ИЛИ B	НЕ A	A→B
0	0	0	0	1	1
0	1	0	1	1	1
1	0	0	1	0	0
1	1	1	1	0	1

Приоритет логических операций

- 1) инверсия НЕ
 - 2) конъюнкция И
 - 3) дизъюнкция ИЛИ
 - 4) импликация \rightarrow
-

Алгоритм

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.

Понятие алгоритма в информатике является фундаментальным, т. е. таким, которое не определяется через другие, более простые понятия.

ИСПОЛНИТЕЛЬ АЛГОРИТМОВ.

Задача составления алгоритма не имеет смысла, если не известны или не учитываются возможности его исполнителя, ведь выполнимость алгоритма зависит от того, какие действия может совершить исполнитель (СКИ – система команд исполнителя).

Например, прочесть алгоритм решения уравнения сможет и первоклассник, а выполнить его, конечно же, нет.

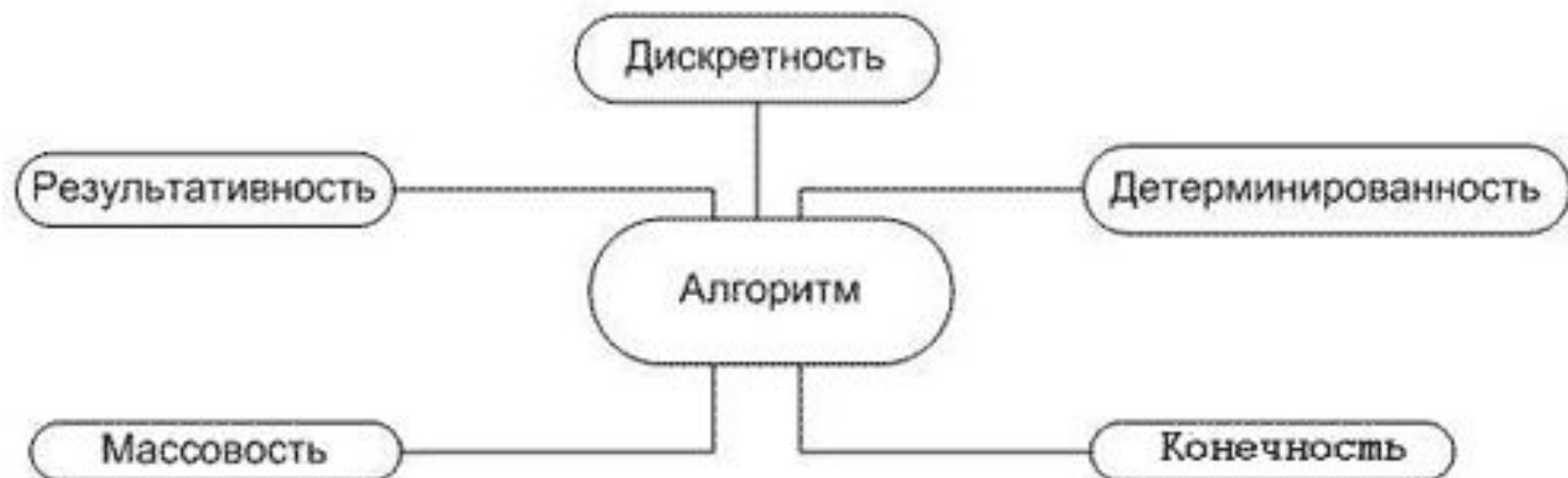
С другой стороны, малыш трех лет не сможет прочесть правила (алгоритм) поведения за столом во время еды, но выполнить их сможет, если ему о них рассказать и показать, что они обозначают.

Команда алгоритма правильна, если исполнитель ее понял и умеет выполнить.

Кто может являться исполнителем алгоритмов?

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: «Купи чего-нибудь вкусенького» и имеющего возможность сориентироваться в ситуации.

Алгоритм обладает следующими свойствами:



СВОЙСТВА:

- **Дискретность** (от лат. discretus – разделенный, прерывистый) указывает, что любой алгоритм должен состоять из конкретных действий, следующих в определенном порядке. Образованная структура алгоритма оказывается дискретной: только выполнив одну команду, исполнитель сможет приступить к выполнению следующей.
- **Детерминированность** (от лат. determinate – определенность, точность) указывает, что любое действие алгоритма должно быть строго и недвусмысленно определено в каждом случае. При этом каждая команда алгоритма входит в состав системы команд исполнителя.

- **Конечность** определяет, что каждое действие в отдельности и алгоритм в целом должны иметь возможность завершения.
- **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 определенный постановкой задачи результат (ответ).
- **Массовость.** Это свойство показывает, что один и тот же алгоритм можно использовать с разными исходными данными, т.е. применять при решении всего класса задач данного типа, отвечающих общей постановке задачи. Пример: алгоритмы «Решение квадратного уравнения», «Приготовить бутерброд».

Алгоритмом также называется информационный процесс, обладающий следующими свойствами:

- Наличие исполнителя преобразований (с его системой команд).
- Разбиение всего процесса преобразования на отдельные команды (понятные исполнителю).
- Определено начальное состояние объекта (над которым производится преобразование) и его требуемое конечное состояние (цель преобразования).

Тип алгоритма определяется характером решаемой (в соответствии с его командами) задачи.

Типовые конструкции алгоритмов:

- Линейная.
- Циклическая.
- Разветвляющаяся.
- Вспомогательная.

Линейный (последовательный) алгоритм – описание действий, которые выполняются однократно в заданном порядке.

Циклический – описание действий или группы действий, которые должны повторяться указанное число раз или пока не выполнено заданное условие. Совокупность повторяющихся действий – тело цикла.

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

Условие – выражение, находящееся между словом «если» и словом «то» и принимающее значение «истина» (ветвь «да») или «ложь» (ветвь «нет»). Возможна полная и неполная форма ветвления.

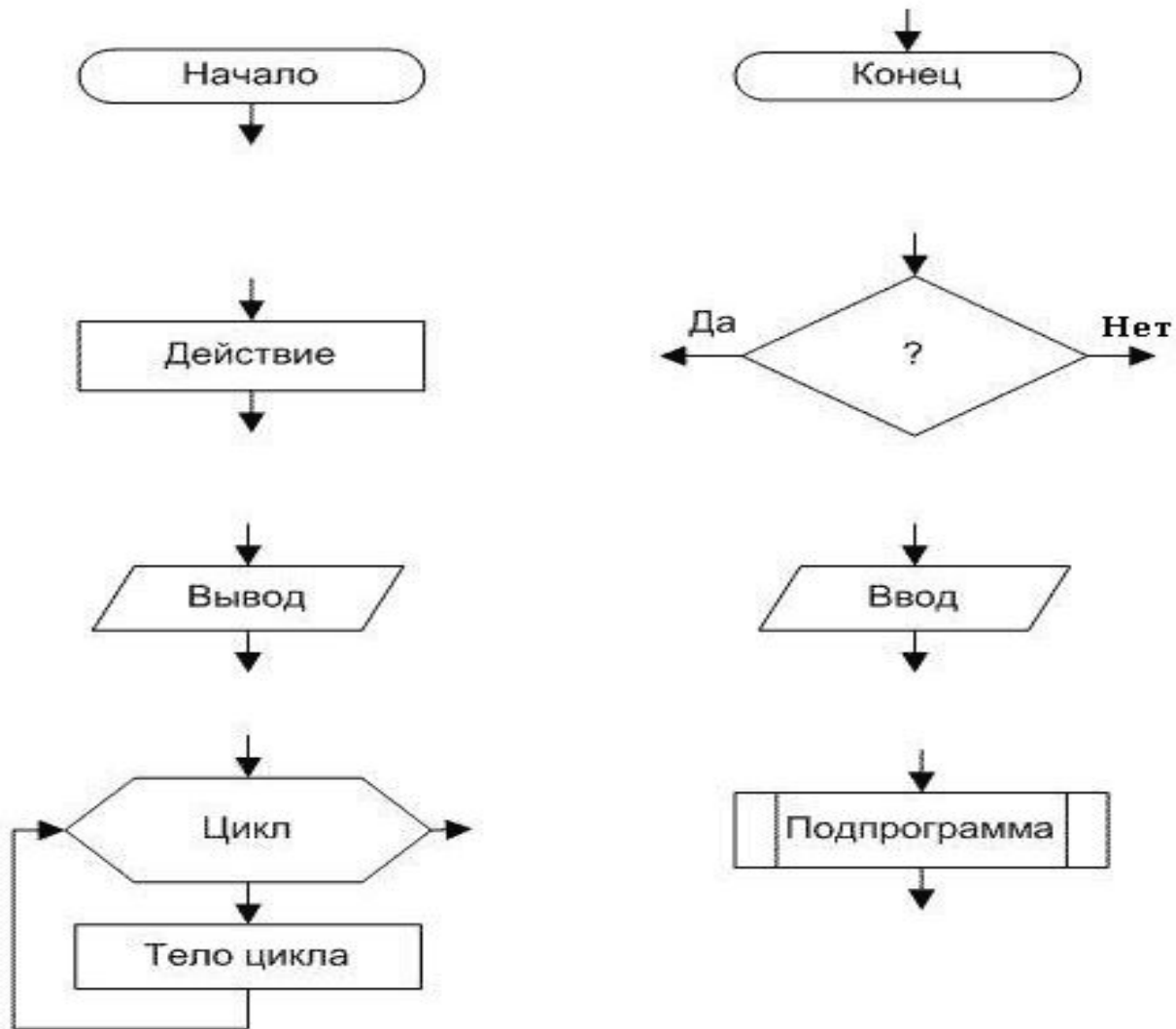
Вспомогательный – алгоритм, который можно использовать в других алгоритмах, указав только его имя. Вспомогательному алгоритму должно быть присвоено имя.

Способы описания алгоритмов.

- на естественном языке;
- на специальном (формальном) языке;
- с помощью формул, рисунков, таблиц;
- с помощью стандартных графических объектов (геометрических фигур) – блок-схемы.

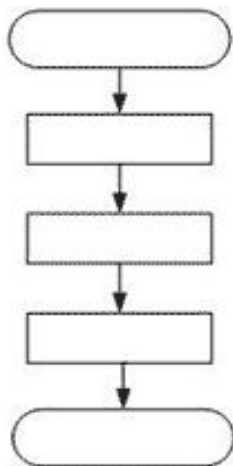
Текстовый процессор Word из офисного пакета Microsoft Office позволяет создавать блок-схемы для графического описания алгоритмов.

Основные элементы блок-схемы

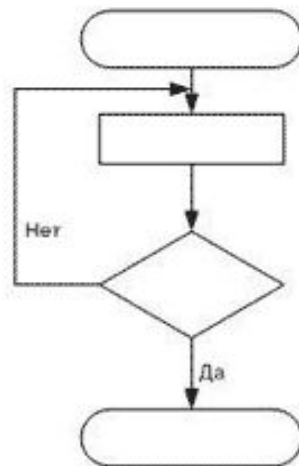


Составить алгоритмы по заготовке.

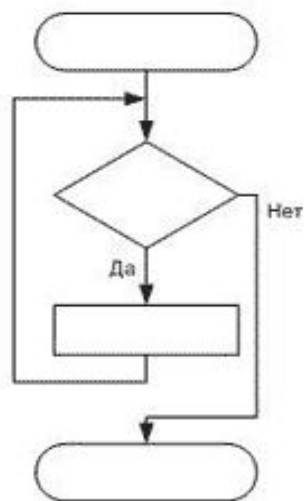
Линейный



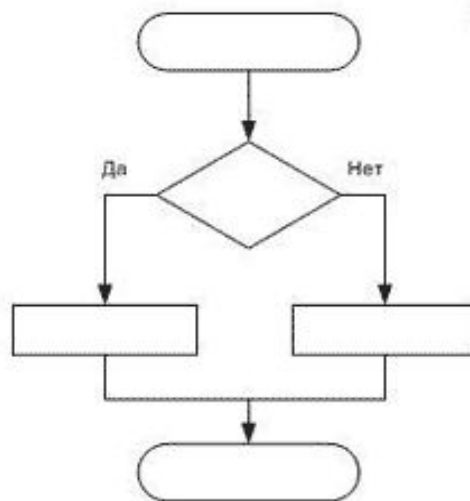
Ветвящийся



Неполное ветвление



Полное ветвление



Законы алгебры логики (доп.)

Закон	для «ИЛИ»	для «И»
<p>Коммутативный (переместительный): Логические переменные можно менять местами.</p>	$A \vee B = B \vee A$	$A \wedge B = B \wedge A$
<p>Ассоциативный (сочетательный): Логические переменные в дизъюнкциях и конъюнкциях можно объединять в группы.</p>	$(A \vee B) \vee C = A \vee (B \vee C)$	$(A \wedge B) \wedge C = A \wedge (B \wedge C)$
<p>Дистрибутивный (распределительный) Одинаковые переменные в дизъюнкциях и конъюнкциях можно выносить за скобки. (В отличие от обычной алгебры, в алгебре высказываний можно выносить за скобки как общие множители, так и общие слагаемые.)</p>	$(A \wedge B) \vee (A \wedge C) = A \wedge (B \vee C)$	$(A \vee B) \wedge (A \vee C) = A \vee (B \wedge C)$
<p>Закон непротиворечия: Высказывание не может быть одновременно истинным и ложным. (Если высказывание A истинно, то его отрицание $\neg A$ должно быть ложным. Логическое произведение высказывания и его отрицания ложно.)</p>		$A \wedge \neg A = 0$

Законы алгебры логики (доп.)

Закон	для «ИЛИ»	для «И»
Закон исключенного третьего: Высказывание может быть только истинным или ложным, третьего не дано. (Результат логического сложения высказывания и его отрицания всегда принимает значение «истина».)	$A \vee \neg A = 1$	
Законы де Моргана (законы общей инверсии): Общая инверсия двух логических слагаемых равносильна логическому умножению инвертированных переменных. Общая инверсия двух логических сомножителей равносильна логическому сложению инвертированных переменных.	$\neg(A \vee B) = \neg A \wedge \neg B$	$\neg(A \wedge B) = \neg A \vee \neg B$
Закон двойного отрицания: Если дважды отрицать некоторое высказывание, то в результате мы получим исходное высказывание.	$\neg\neg A = A$	
Идемпотентности	$A \vee A = A$	$A \wedge A = A$

Законы алгебры логики (доп.)

Закон	для «ИЛИ»	для «И»
Контрапозиции	$A \rightarrow B = \neg B \rightarrow \neg A$	
Законы склеивания	$(A \wedge B) \vee (\neg A \wedge B) = B$	$(A \vee B) \wedge (\neg A \vee B) = B$
Исключение констант	$A \vee 0 = A, A \vee 1 = 1$	$A \wedge 0 = 0, A \wedge 1 = A$
Снятие импликации	$A \rightarrow B = \neg A \vee B$	
Снятие эквивалентности	$A \leftrightarrow B = (A \wedge B) \vee (\neg A \wedge \neg B)$	
Закон поглощения	$A \vee (A \wedge C) = A$	$A \wedge (A \vee C) = A$

Задачи

1. Для какого из указанных значений числа X истинно высказывание:

НЕ $((X > 3) \rightarrow (X > 4))$?

1) 2

2) 3

3) 4

4) 5

Решение (способ 1)

A	B	$\neg A$	$A \vee B$	$A \wedge B$	$A \rightarrow B$
0	0	1	0	0	1
0	1	1	1	0	1
1	0	0	1	0	0
1	1	0	1	1	1

По таблице истинности импликации видим, что она будет ложной в одном в одном-единственном случае: когда первое высказывание истинно, а второе ложно.

Первое высказывание ($X > 3$) может быть истинно только при $X=4$ или $X=5$. Второе высказывание принимает значение ЛОЖЬ только при $X=4$.

Ответ: 3) 4

Решение (способ 2)

Метод последовательной подстановки

1) $X=2$ НЕ $((2>3) \rightarrow (2>4)) = \neg(0 \rightarrow 0) = \neg(1) = 0$

2) $X=3$ НЕ $((3>3) \rightarrow (3>4)) = \neg(0 \rightarrow 0) = \neg(1) = 0$

3) $X=4$ НЕ $((4>3) \rightarrow (4>4)) = \neg(1 \rightarrow 0) = \neg(0) = \underline{1}$

4) $X=5$ НЕ $((5>3) \rightarrow (5>4)) = \neg(1 \rightarrow 1) = \neg(1) = 0$

Ответ: 3) 4

Задачи

2. Для какого из указанных значений числа X ложно выражение

$(X > 2)$ ИЛИ НЕ $(X > 1)$?

1) 1

2) 2

3) 3

4) 4

Ответ: **2) 2**

Задачи

3. Для какого из указанных значений числа X истинно выражение

НЕ ($X > 3$) И ($X > 2$)?

1) 1

2) 2

3) 3

4) 4

Ответ: **3) 3**

Контрольные вопросы

1. Операция логического умножения и ее свойства.
2. Операция логического сложения и ее свойства.
3. Операция логического отрицания и ее свойства.
4. Операция логического следования и ее свойства.
5. Приоритет логических операций.