СОДЕРЖАНИЕ

ИНФОРМАЦИЯ

сведение, разъяснение, ознакомление (лат. informatio) .

Базовые понятия:
точка, прямая, плоскость в геометрии, информация в информатике.

Определение базовых понятий невозможно выразить через другие, более простые понятия.

Содержание базовых понятий поясняется на примерах или выявляется путем их сопоставления с содержанием других понятий.

Забуга А.А. Теоретические основы информатики. Стр. 4-23

ИНФОРМАЦИЯ

Философский подход. «Информация» –взаимодействие, отражение, познание.

Кибернетический подход. «Информация» соотносится – с процессами управления в сложных системах (живых организмах или технических устройствах) это характеристики управляющего сигнала, передаваемого по линии связи

В ***физике*** информация рассматривается как мера сложности и упорядоченности системы, энтропия с обратным знаком

Информация в биологии. Понятие «информация» связывается с целесообразным поведением живых организмов.

Используется в связи с исследованиями механизмов наследственности

Забуга А.А. Теоретические основы информатики. Стр. 4-23

ИНФОРМАЦИЯ

В **информатике (традиционный подход)** информация – это сведения, знания, сообщения о положении дел, которые человек воспринимает из окружающего мира с помощью органов чувств (зрения, слуха, вкуса, обоняния, осязания).

В **теории информации (вероятностный подход)** информация – это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределённости и неполноты знаний.

С **обыденной точки зрения для человека** информация – это знания, которые он получает из различных источников с помощью органов чувств.
Например, декларативные («знаю, что …») или процедурные («знаю, как …»).

ИНФОРМАЦИЯ

По способам восприятия: визуальная, аудиальная, тактильная, обонятельная, вкусовая.

По формам представления: текстовая, числовая, графическая, музыкальная, комбинированная и тд.

По общественному значению:
Массовая - обыденная, общественно-политическая, эстетическая.
Специальная - научная, техническая, управленческая, производственная.
Личная – знания, умения, интуиция.

СВОЙСТВА ИНФОРМАЦИИ

Объективность – не зависит от чьего-либо мнения.

Достоверность – отражает истинное положение дел.

Полнота – достаточна для понимания и принятия решения.

Актуальность – важна и существенна для настоящего времени.

Ценность (полезность, значимость)- обеспечивает решение поставленной задачи, нужна для того чтобы принимать правильные решения.

Понятность (ясность)– выражена на языке, доступном получателю.

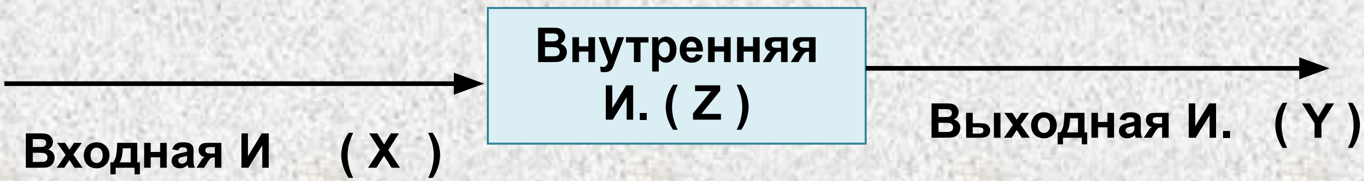
СВОЙСТВА ИНФОРМАЦИИ

Атрибутивные свойства: дискретность (информация состоит из отдельных частей, знаков) и непрерывность (возможность накапливать информацию).

Динамические свойства связаны с изменением информации во времени:
- копирование – размножение информации,
- передача от источника к потребителю,
- перевод с одного языка на другой,
- перенос на другой носитель,
- старение (физическое – носителя, моральное – ценностное).

Практические свойства - информационный объем и плотность.

ТЕОРИЯ ИНФОРМАЦИИ



Под **информацией** в кибернетике понимается любая совокупность сигналов, которую некоторая система **воспринимает**, **хранит** и **выдает** в окружающую среду.

Пространство и время являются мерой для формирования информации, фиксации объектов и событий.

ТЕОРИЯ ИНФОРМАЦИИ

Все находится в движении, в колебательном состоянии. Колебания проявляются в виде многообразия сигналов. Человек воспринимает лишь 5 видов сигналов с помощью органов чувств. **Сигнал** – это физический процесс, чувствительный для приемника.

Объекты - фрагменты материи, генерирующие сигналы, которые можно воспринимать как некоторую целостную совокупность. **Сообщение** - совокупность сигналов, связанная с объектами.

Объекты и их взаимодействие порождают события. **Событие** – это процесс или явление, связанный с поведением объектов во времени. Объекты и события являются источниками сигналов и сообщений.

ТЕОРИЯ ИНФОРМАЦИИ

Тезаурус - совокупность образов объектов и событий сформированных органами чувств и отраженных на основе принятой человеком системы метрик (меры).

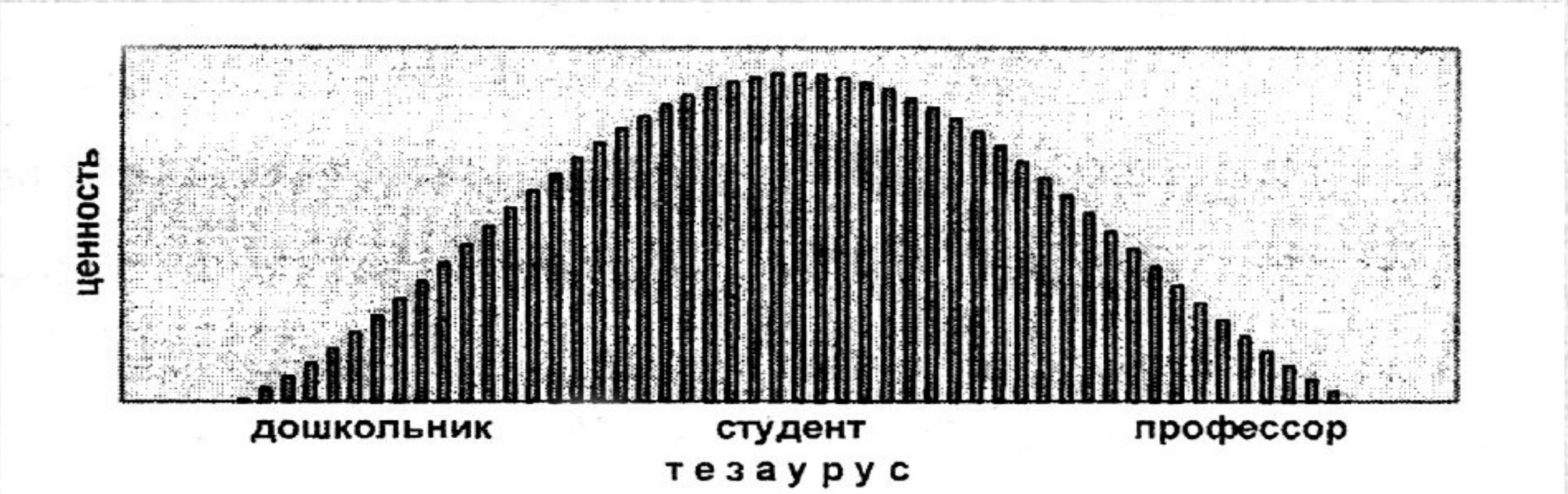
Информация - интерпретация сообщения.

Информационный процесс - форма существования информации. Для любого информационного процесса характерны

- возникновение,
- восприятие,
- запоминание,
- воспроизведение,
- хранение,
- передача и
- обработка информации.

Одну и ту же информацию разные люди могут оценить по

Восприятие информации из сообщения зависит от объема тезауруса приемника. Если соответствующие понятия отсутствуют в тезаурусе, человек не в состоянии извлечь информацию из сообщения. Если объем тезауруса велик настолько, что сообщение не вносит в него никаких изменений, человек считает информацию банальной, т.е. для него сообщение



Ценность информации: $W = \log_2 (p' / p)$, где p и p' — вероятности достижения цели до и после получения информации.

ТЕОРИЯ ИНФОРМАЦИИ

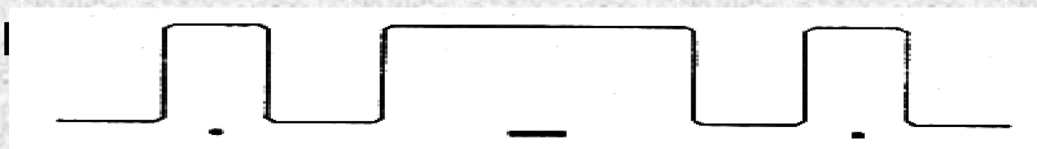
Запоминание. Информация «понятная» приемнику, отобранная им, запоминается. «Запоминание характеризуется как «перекодирование» сигнала из алфавита процессов, протекающих во времени, в алфавит состояний объектов в пространстве и наоборот».

Хранение можно рассматривать как частный случай передачи информации, а именно, передачу во времени.

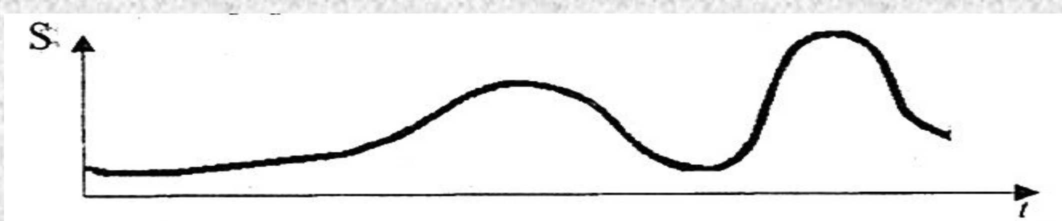
НОСИТЕЛЬ ИНФОРМАЦИИ - СИГНАЛ

Сигнал - это изменяющийся во времени физический процесс. Та из характеристик, которая используется для представления сообщений, называется **параметром сигнала.**

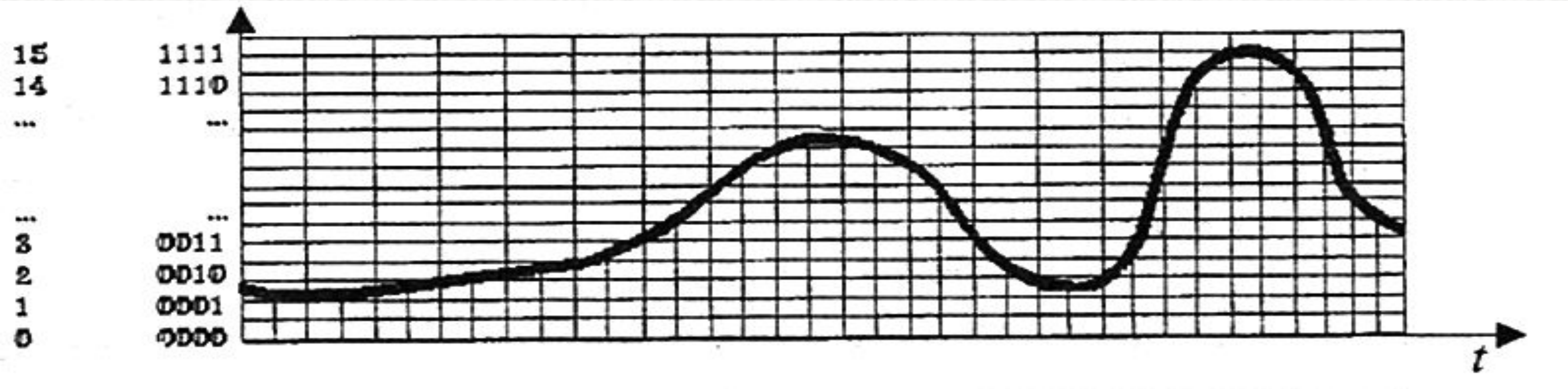
Дискретный сигнал - информативный параметр сигнала принимает последовательное во времени конечное число значений (при этом все они могут быть прону

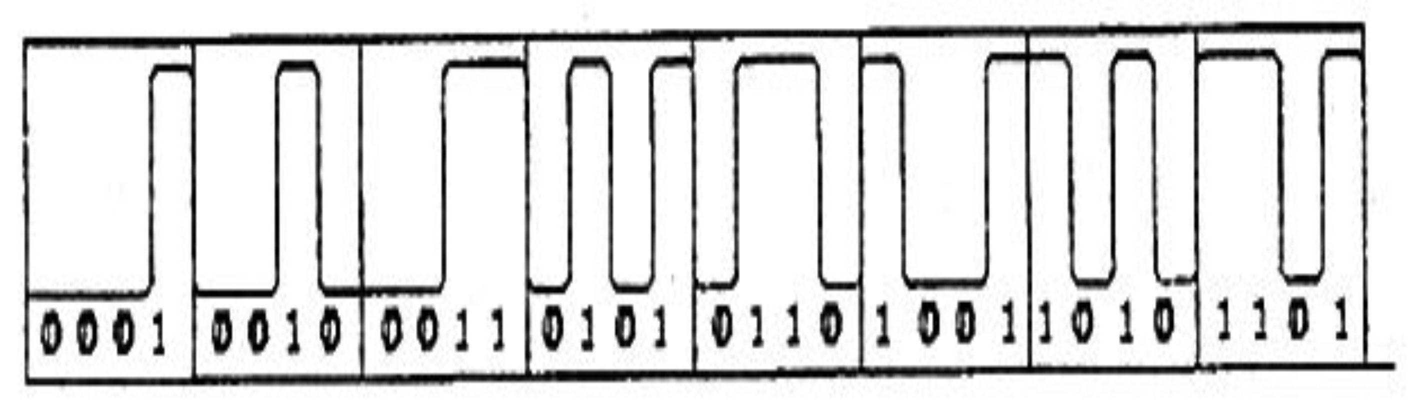


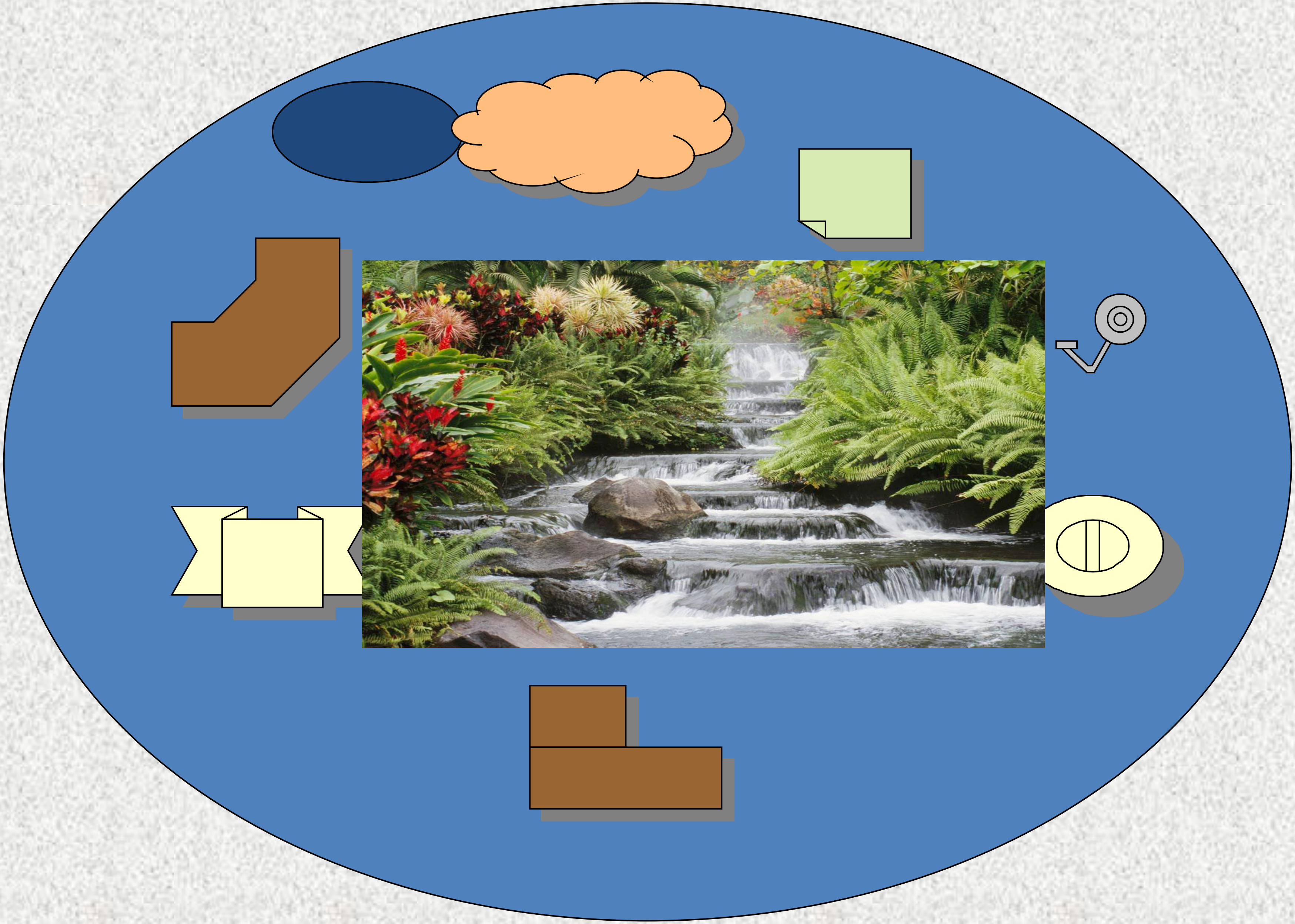
Непрерывный сигнал - информативный параметр сигнала - непрерывная функция от времени.



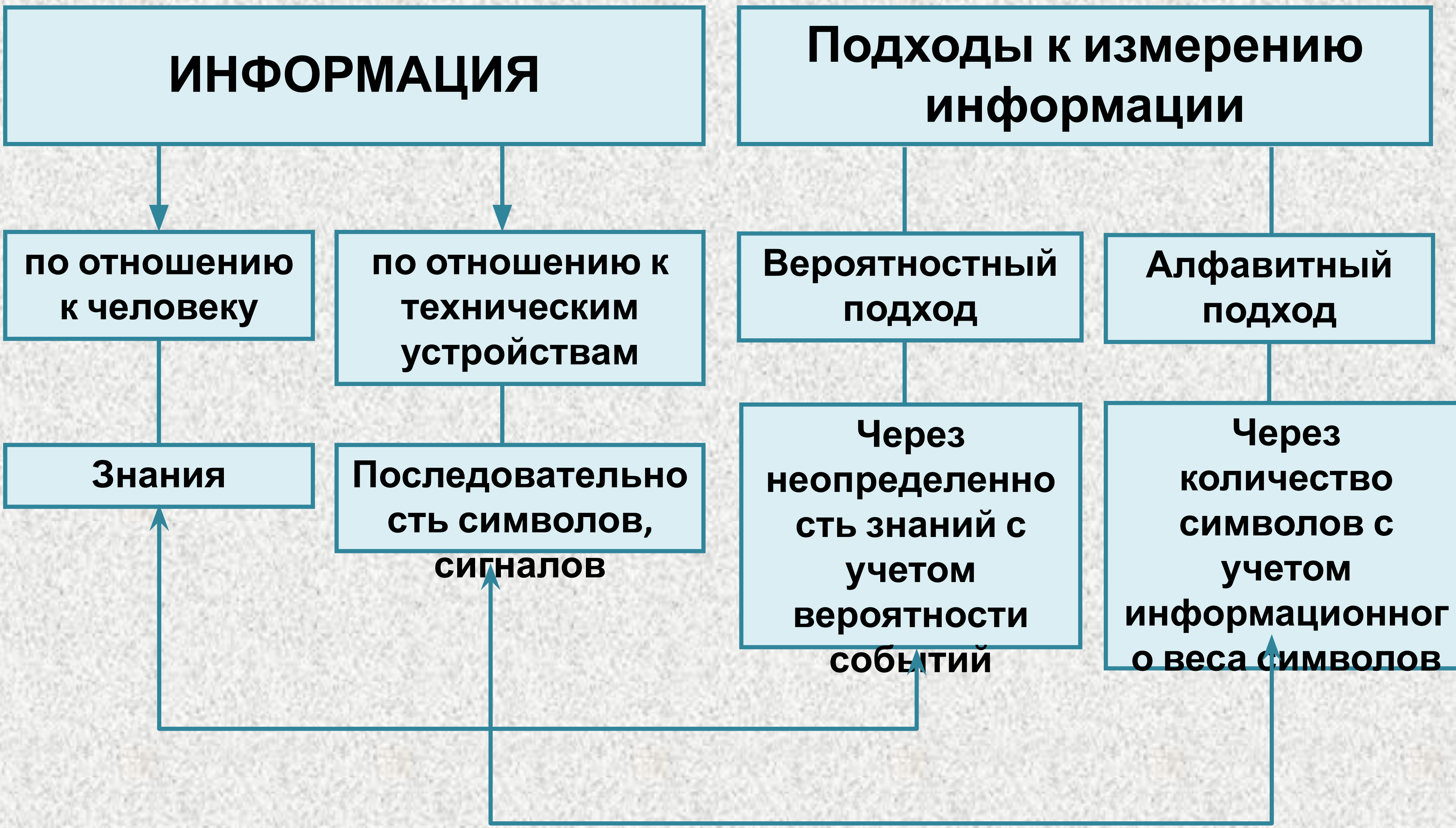
Пример квантования и оцифровки







ИЗМЕРЕНИЕ ИНФОРМАЦИИ



Единицы измерения информации

ГОСТ 8.417-2002 :

1 Кбайт = 2^{10} байт,
1 Мбайт= 2^{10} Кбайт,
1 Гбайт = 2^{10} Мбайт,
и т. д.

ГОСТ 8.417-2002, приложение А: “с наименованием «байт» некорректно (вместо 1000 = 10^3 принято 1024 = 2^{10}) использовали (и используют) приставки СИ:
1 Кбайт = 1024 байт, 1 Мбайт = 1024 Кбайт, 1 Гбайт = 1024 Мбайт и т. д.
При этом обозначение Кбайт начинают с прописной буквы в отличие от строчной буквы «к» для обозначения множителя 10^3”.

1 килобайт = 10^3 байтов,
1 мегабайт = 10^3 килобайтов,
1 гигабайт = 10^3 мегабайтов

1 *бит* - объем данных, состоящий из одного символа двоичного алфавита (0 или 1);
1 *байт* = 8 битов

Единицы измерения информации

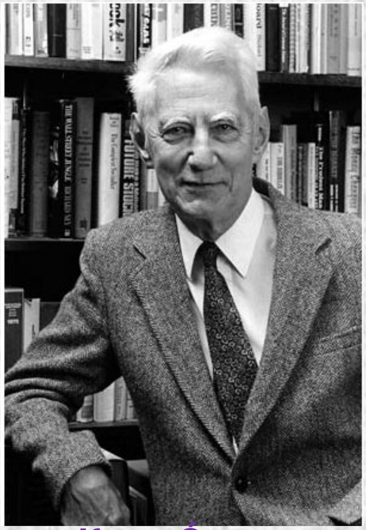
байт	B	10^0
килобайт	kB	10^3
мегабайт	MB	10^6
гигабайт	GB	10^9
терабайт	TB	10^{12}
петабайт	PB	10^{15}
эксабайт	EB	10^{18}
зеттабайт	ZB	10^{21}
йоттабай		[illegible]

МЭК 60027-2	ГОСТ 8.417-2002	
B	байт	2^0
KiB	Кбайт	2^{10}
MiB	Мбайт	2^{20}
GiB	Гбайт	2^{30}
TiB	Тбайт	2^{40}
PiB	Пбайт	2^{50}
EiB	Эбайт	2^{60}
ZiB	Збайт	2^{70}
YiB	Йбайт	2^{80}

Международная некоммерческая организация по стандартизации в области электрических, электронных и смежных технологий (МЭК) – *International Electrotechnical Commission* (IEC)

KiB - «кибибайт», MiB – «мебибайт», и т. д.

КОДИРОВАНИЕ ИНФОРМАЦИИ



Клод Э́лвуд Ше́ннон (1916-2001)

Шенноновский источник сообщений:

1) $A=\{a_1, a_2, \ldots, a_N\}$ - алфавит источника,
   N - мощность алфавита,

2) $p(a_i)$ $i=1, 2, \ldots, N$ - частота

Ральф Винтон Лайон Хартли

я a_i

м сообщении источника,

$(a_2) + \ldots + p(a_N) =1$

Количество информации

Объёмный подход, при $p(a_1)= p(a_2)=\ldots= p(a_N)$:

$h=\log_2 N$

Вероятностный подход, с учетом частоты $p(a_i)$
появления a_i в сообщениях:

$h_i=-\log_2 p(a_i), i=1,2, \ldots, N$

КОДИРОВАНИЕ ИНФОРМАЦИИ

Энтропия источника сообщений –
среднее количество информации на один символ источника
с учётом частоты его появления в сообщении

$$H = -\sum_{i=1}^{N} p(a_i)\log_2 p(a_i)$$

К. Шеннон (1948 г.)

КОДИРОВАНИЕ ИНФОРМАЦИИ. Равномерный код

Равномерный код. Все кодовые комбинации содержат одинаковое количество символов.

Длина равномерного кода: $l = \rbrack \log_2 N \lbrack$

N – мощность алфавита источника

Пример сообщения источника:

ОТ_ТОПОТА_ТОПОТА_РОПОТА

Названия строк	Количество по полю Собщение источника
О	7
Т	6
ПРОБЕЛ	3
А	3
П	3
Р	1
Общий итог	23

Названия строк	Равномерный код
О	000
Т	111
ПРОБЕЛ	010
А	011
П	100
Р	101

КОДИРОВАНИЕ ИНФОРМАЦИИ

Неравномерные коды. Код Шеннона-Фано

Общая схема построения кодовых слов кода Шеннона-Фано:

1. Символы исходного алфавита располагаются в порядке убывания их вероятностей.

2. Все множество символов разбивается на две группы так, чтобы суммарные вероятности сообщений каждой из групп были как можно более близки друг к другу. Одной группе приписывается кодовый знак «0», а другой – «1». Это первый знак кодового слова.

3. По тому же принципу каждая из полученных групп снова разбивается на две части. Каждой новой группе приписывается следующий знак кодового слова («0» или «1»).

4. Процедура разбиения группы на подгруппы продолжается до тех пор, пока в ней содержится более одного символа.

6. В результате каждому символу исходного алфавита будет сопоставлено кодовое слово из нулей и единиц.

Алгоритм завершается, когда каждый символ исходного алфавита выделен в «самостоятельную» группу. Код Шеннона-Фано – префиксный.

В алгоритме построения кода Шеннона-Фано те символы исходного алфавита, у которых вероятность больше, быстрее образуют «самостоятельную» группу, их кодовое слово будет более коротким. Поэтому код Шеннона-Фано называют «экономным».

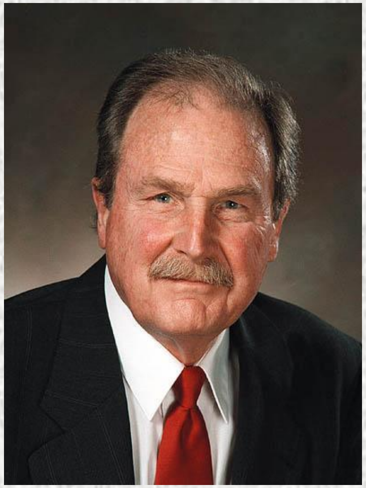
КОДИРОВАНИЕ ИНФОРМАЦИИ. Неравномерные коды

Неравномерные коды. Код Шеннона-Фано

Сообщение:
ОТ_ТОПОТА_ТОПОТА_РОПОТА

Названия стр	Количество по полю Собщение источника
О	7
Т	6
ПРОБЕЛ	3
А	3
П	3
Р	1
Общий итог	23

Код Шеннона-Фано						
Частота встречаемости символов					Кодовые слова	Длина кодового слова
0,30	0	0			00	2
0,26		1			01	2
0,13	1	0	0		100	3
0,13			1		101	3
0,13		1	0		110	3
0,04			1		111	3



Кодирование информации

Неравномерные коды. Код Хаффмана

Метод кодирования состоит из двух основных этапов:
– этап сжатия, на котором происходит построение оптимального кодового дерева,
– этап расщепления, на этом этапе для каждого символа исходного алфавита создается кодовое слово.

Этап сжатия. Каждый символ исходного алфавита приписывается оконечному узлу кодового дерева, вероятность появления символа считается весом узла. Все узлы объявляются свободными.
1. Среди всех свободных узлов дерева отыскиваются два узла с наименьшими вероятностями. Вероятности найденных свободных узлов складываются.
2. В дереве создается новый свободный узел, ему приписывается вес – полученная сумма. Этап сжатия завершается, когда образуется корневой узел кодового дерева, его вес равен единице.
Этап расщепления. Каждому ребру кодового дерева, сформированного на этапе сжатия, приписывается кодовый знак.
Затем для каждого символа исходного алфавита формируется кодовое слово. Начиная от **корня дерева**, последовательно выписываются кодовые знаки («0» или «1») ребер, соединяющих корень дерева и оконечную вершину дерева, где расположен символ исходного алфавита. Таким образом, для каждого символа исходного алфавита будет создано кодовое слово.
Алгоритм Хаффмана обеспечивает оптимальное префиксное кодирование символов алфавита шенноновского источника.

Кодирование информации

Неравномерные коды. Код Хаффмана

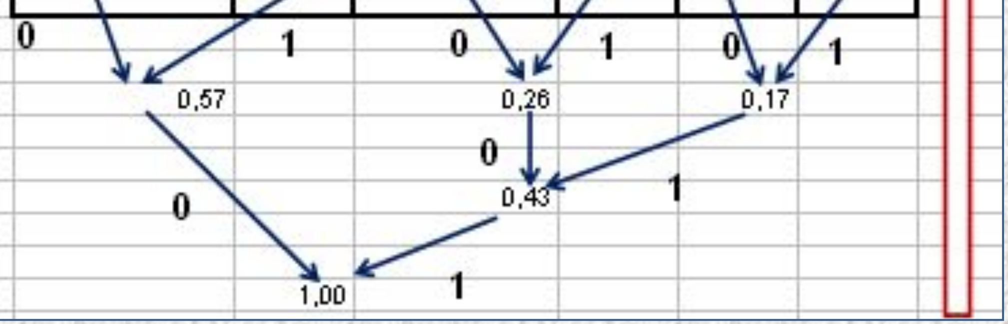
Сообщение:

ОТ_ТОПОТА_ТОПОТА_РОПОТА

Названия строк	Количество по полю Собщение источника
О	7
Т	6
ПРОБЕЛ	3
А	3
П	3
Р	1
Общий итог	23

Код Хаффмана

Длина кодового слова	2	2	3	3	3	3
Кодовые слова	00	01	100	101	110	111
Названия строк	О	Т	ПРОБЕЛ	А	П	Р
Частота встречаемости символов	0,30	0,26	0,13	0,13	0,13	0,04



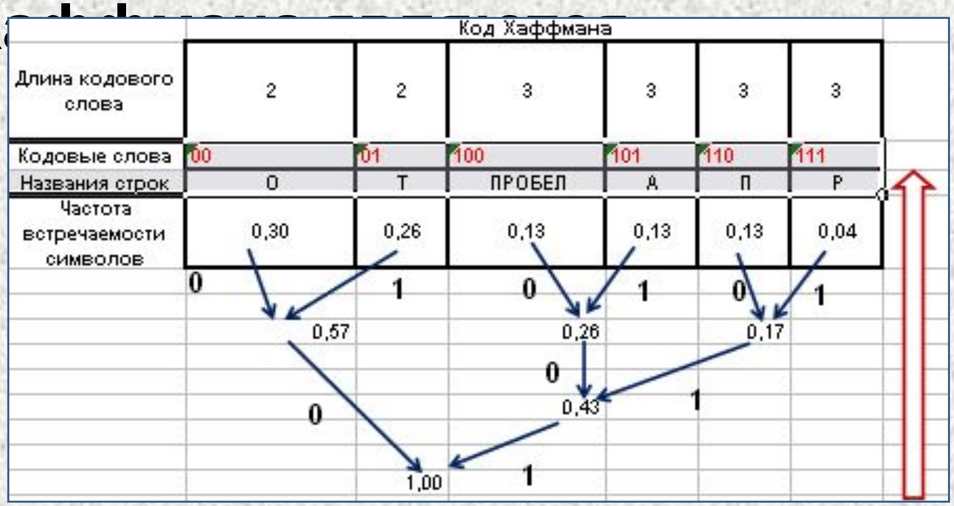
Кодирование информации

Условие Фано (декодируемости неравномерного кода):

в префиксном коде ни одно кодовое не является началом другого кодового слова.

Коды Шеннона-Фано и Ха

Названия строк	Количество по полю Собщение источника
О	7
Т	6
ПРОБЕЛ	3
А	3
П	3
Р	1
Общий итог	23



Код Шеннона-Фано

Частота встречаемости символов					Кодовые слова	Длина кодового слова
0,30	0	0			00	2
0,26		1			01	2
0,13	1	0	0		100	3
0,13			1		101	3
0,13		1	0		110	3
0,04			1		111	3

Кодирование информации

Неравномерные коды. Средняя длина кодового слова

$A=\{a_1, a_2, \dots , a_N\}$, $p(a_i)$, $i=1, 2, \dots , N$

Для неравномерного бинарного кода средняя длина кодового слова рассчитывается по формуле:

$$\bar{l} = \sum_{i=1}^{N} l_i \, p(a_i)$$

l_i – длина кодового слова, $i=1, 2, \dots , N$

КОДИРОВАНИЕ ИНФОРМАЦИИ

Теорема кодирования Шеннона *)

1. Для любого источника сообщений и для любого способа кодирования справедливо неравенство: **H ≤ L.**
2. Алфавит любого источника сообщений можно закодировать таким образом, что разность **L – H** будет сколь угодно мала.

$$l = \,]\log_2 N[$$

$$H = -\sum_{i=1}^{N} p(a_i)\log_2 p(a_i)$$

$$\bar{l} = \sum_{i=1}^{N} l_i\, p(a_i)$$

*) Бауэр Ф.Л., Гооз Г. Информатика. Вводный курс. – М.:Мир, 1976. – 489 с.

КОДИРОВАНИЕ ИНФОРМАЦИИ

Относительная избыточность кода определяет количество избыточных знаков (нулей или единиц) на каждые 100 знаков передаваемой цепочки символов.

Для источника сообщений, энтропия которого равна *H*, формулы расчета относительной избыточности (*r*):

– для неравномерного кода:

$$r = \left(1 - \frac{H}{\bar{l}}\right) \cdot 100\%$$

– для равномерного кода:

$$r = \left(1 - \frac{H}{l}\right) \cdot 100\%$$

	Избыточность (бит)	Относительная избыточность (процент)
Для кода Шеннона-Фано	0,06	2,5
Для кода Хаффмана	0,06	2,5
Для равномерного кода	0,63	20,8

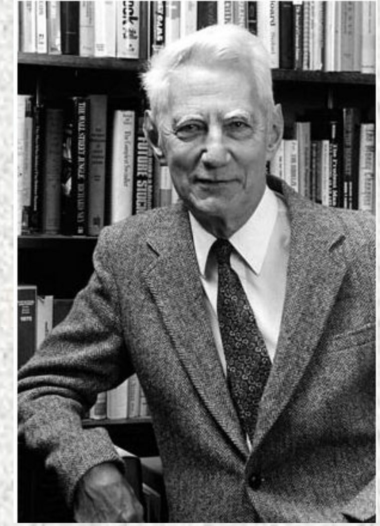
94НН03 С006ЩЗНN3 П0К4ЗЫ8437,
К4КN3 У9N8N73ЛЬНЫ3 83ЩN М0Ж37 93Л47Ь
Н4Ш Р4ЗУМ!
8П3Ч47ЛЯЮЩN3 83ЩN!

СН4Ч4Л4 Э70 6ЫЛ0 7РУ9Н0, Н0 С3ЙЧ4С,
Н4 Э70Й С7Р0К3 84Ш Р4ЗУМ ЧN7437 Э70
4870М47NЧ3СКN, Н3 349УМЫ84ЯСЬ 06
Э70М.
Г0Р9NСЬ. ЛNШЬ 0ПР393Л3ННЫ3 ЛЮ9N
М0ГУ7 ПР0ЧN747Ь Э70.

Помехоустойчивое кодирование

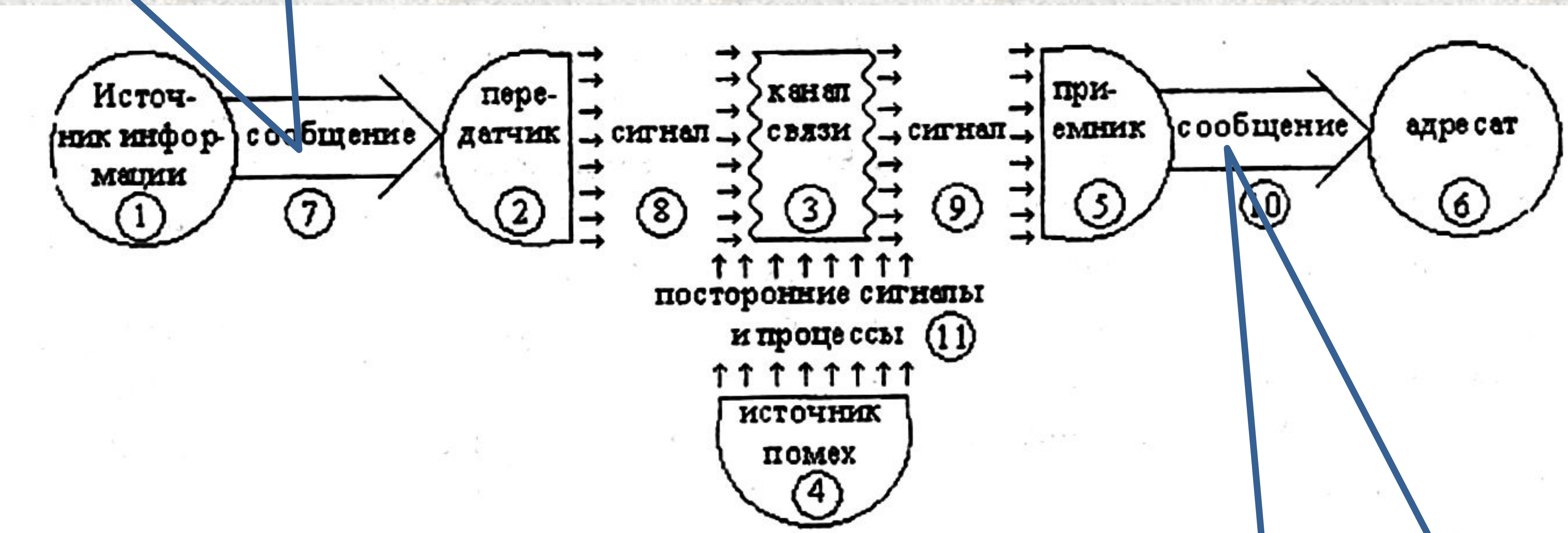
Самоконтролирующиеся и самокорректирующиеся коды

Классическая схема передачи информации (по К.Э.Шеннону)



Клод Э́лвуд Ше́ннон
(1916-2001)

1100110011001100



1100110011001100
1110110011001100

ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ

Самоконтролирующийся код **– код, позволяющий автоматически обнаруживать наиболее вероятные ошибки.**

Самокорректирующийся код **- код, позволяющий автоматически исправлять ошибки.**

Самоконтролирующиеся коды

1. Код с битом контроля чётности

1 0 0 1 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0 1 0 1…

1 0 0 1 1 1 0 1 0 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0 0 0 1 0 1…

1 0 0 0 1 1 0 1 0

Одиночная ошибка

1 0 0 1 0 1 0 1 1

Двойная ошибка

1 1 0 1 1 1 0 1 0

Тройная ошибка

0 1 0 1…

Если число единиц в 8-ми битовом блоке чётно, проверочный бит равен 1, иначе – 0.

Код обнаруживает одиночные ошибки и групповые с нечетной кратностью.

Примеры:

Кодируемое (исходное) слово -10111101

Оно содержит 6 единиц, бит чётности для него равен 1.

Слово кода с проверкой четности для него: 011110111

Кодируемое (исходное) слово 01110011

Оно содержит 5 единиц, бит чётности для него равен 0.

Слово кода с проверкой четности для него: 111001100

2. Код с контрольной суммой

Алгоритм:

1. Сложить все цифры, которые стоят на четных местах: 6+1+4+0+1+9=21.
2. Полученную сумму умножить на 3: 21x3=63.
3. Сложить все цифры, которые стоят на нечетных местах, без контрольной цифры: 4+0+5+6+2+2=19.
4. Сложить числа, полученные в пунктах 2 и 3: 63+19=82.
5. От полученной суммы отбросить десятки: получим 2.
6. Из 10 вычесть полученное в пункте 5 число: 10-2=8.

Если полученная в результате расчета цифра совпадает с контрольной цифрой в штрих-коде – товар произведён легально.

Самокорректирующиеся коды

Код с повторением

1 0 0 1 1 1 0 1 0 1 0 1 …

111 000 000 111 111 111 000 111 000 111 000 111 …

101 010 000 101 110 111 000 111 010 111 000 111 …

1 0 0 1 1 1 0 1 0 1 0 1 …

Алгоритм

Кодирование: каждый символ исходного слова заменяется блоком из n
(n-нечетное) точно таких символов.

При декодировании n-знаковый блок заменяется на символ 1 или 0 решением «большинства символов» блока.

Код Хемминга

Определение 1.
На множестве двоичных слов равной длины <u>расстоянием Хэмминга</u> ρ(a, b) между словами a и b называют число несовпадающих позиций этих слов.

Примеры:

1) a~ <0000>, b~ <0011>, ρ(a,b)= 2

2) a~ <01011>, b~ <11000>, ρ(a,b)= 3

Определение 2.

<u>Кодовое расстояние</u> (d) – это минимальное значение расстояния Хемминга между двумя любыми словами алфавита кода.

Примеры:

1) A1={0000; 0011; 0101; 1001}, d(A1)=2

1) A2={0000; 0111; 0010}, d(A2)=1

Теорема Хэмминга

Ричард Хэмминг
(1915-1998)

Для того чтобы код позволял <u>обнаруживать</u> ошибки в t (или менее) позициях, необходимо и достаточно, чтобы его кодовое расстояние (d) было больше или равно (t +1), $d \geq t+1$.

Для того чтобы код позволял <u>исправлять</u> ошибки в t (или менее) позициях, необходимо и достаточно, чтобы его кодовое расстояние было больше или равно (2t + 1): $d \geq 2t+1$.

Пример 1.
A1={0000; 0111}, d(A1)=3 ⇒
код обнаруживает 2 ошибки, исправляет (локализует) одну.

1. Пусть принято слово $\Omega = 0011$, $\Omega \notin A1$.
Расстояния Хемминга между Ω и каждым словом из A1:
$\rho(\Omega, 0000) = \rho(0011, 0000) = 2$
$\rho(\Omega, 0111) = \rho(0011, 0111) = 1$
Ближайшее к Ω кодовое слово из алфавита A1 (в смысле расстояния Хемминга) – 0111, это слово принимается за исходное, которое было искажено при передаче.
Одиночная ошибка исправлена.

2. Пусть $\Omega = 1010$, $\Omega \notin A1$.
Расстояния Хемминга между Ω и каждым словом из A1:
$\rho(\Omega, 0000) = \rho(1010, 0000) = 2$
$\rho(\Omega, 0111) = \rho(1010, 0111) = 2$
Выбрать ближайшее к Ω кодовое слово из A1 нельзя, т.к. не задан критерий выбора для двух и более равноотстоящих слов.

Пример 2.

A2={0000000000; 0000011111; 1111100000; 1111111111},
d(A2) =5
Код обнаруживает четыре одиночные ошибки, локализует две.

Пусть принято слово Ω= 0100111111, *Ω* ∉ A2.
***ρ (Ω,* 0000000000*)=* ρ(0100111111, 0000000000)=7,**
***ρ (Ω,* 0000011111*)=* ρ(0100111111, 0000011111)=2,**
***ρ (Ω,* 1111100000*)=* ρ(0100111111, 1111100000)=8,**
***ρ (Ω,* 1111111111*)=* ρ(0100111111, 1111111111)=3.**
В алфавите A2 ближайшее к Ω слово – 0000011111*.*

Слово – 0000011111 принимается за исходное слово, в котором при передаче были искажены две позиции.

Пример 3.

А3={0000; 0001; 0010; 0011; 0100; 0101; 0110; 0111; 1000; 1001; 1010; 1011; 1100; 1101; 1110; 1111}

D(А3)=1

**Код не позволяет исправить или обнаружить ошибки,
он не относится к помехоустойчивым кодам.**

Алгоритм построения кода Хемминга

Обозначения:

Hem(n,m), n , m* – целые числа, *n = m+r

***n* – число позиций в кодовом слове,**

***m* – число информационных разрядов**

***r* – число контрольных разрядов.**

Проверочные биты – на позициях, номера - степени двойки:

$2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^5, \ldots$

Величины m, n, r связаны соотношением: $m+r+1 \leq 2^r$

n	3	4	5	6	7	8	9	10	11	12
m	1	1	2	3	4	4	5	6	7	8
r	2	3	3	3	3	4	4	4	4	4

Контрольные и информационные позиции кодового слова Hem(7,4)

Позиция	Какими битами контролируется	Комментарии
1 (A)	$2^0 = 1$	Это контрольный бит
2 (B)	$2^1 = 2$	Это контрольный бит
3	$2^0+2^1 = 1 + 2 = 3$	Контролируется 1 и 2 битами
4 (C)	$2^2 = 4$	Это контрольный бит
5	$2^0+2^2 = 1 + 4 = 5$	Контролируется 1 и 4 битами
6	$2^1+2^2 = 2 + 4 = 6$	Контролируется 2 и 4 битами
7	$2^0+2^1+2^2 = 1 + 2 + 4 = 7$	Контролируется 1, 2 и 4 битами

Для *Hem*(*7,4*) номера контрольных битов: 1, 2 и 4, остальные - информационные.
Значения контрольных битов вычисляются по формулам:
бит 1 (A): (значение бита 3 + значение бита 5 + значение бита 7) (mod 2),
бит 2 (B): (значение бита 3 + значение бита 6 + значение бита 7) (mod 2),

Исходное слово: 0100

Номера разрядов	1	2	3	4	5	6	7
Исходное слово			0		1	0	0
Кодовое слово	1	0	0	1	1	0	0

Принято кодовое слово:

1001000

Номера разрядов	1	2	3	4	5	6	7
Принятое слово	1	0	0	1	0	0	0

Исходное слово: 1101

Номера разрядов	1	2	3	4	5	6	7
Исходное слово			1		1	0	1
Кодовое слово	1	0	1	0	1	0	1

Принятое слов

Номера разрядов	1	2	3	4	5	6	7	
Принятое слово	1	0	1	0	1	0	1	

) 1.

ТРЕНИНГ

1. Семибитовое кодовое слово Хемминга с тремя проверочными символами - 1101110, номер искаженного разряда в этом слове …

Эталон: 4

2. Семибитовое кодовое слово Хемминга с тремя проверочными символами - 1101110, исходная последовательность символов для него …

а) 0010 б) 0110 в) 1100 г) 0111

Эталон: б)

3. Кодовое расстояние между двумя кодовыми словами – …

а) номер первой единицы в кодовом слове
б) количество единиц в кодовом слове
в) общее число позиций в кодовых словах
г) число разрядов с неодинаковыми значениями

Эталон: г)

4. Для кода A={0000; 0011; 0101; 0110} кодовое расстояние равно…

Эталон: 2

Введение в теорию алгоритмов

Введение в теорию алгоритмов

Алгоритм – предписание, точный набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное время.
Алгоритм описывает **процесс преобразования исходных данных в результаты.**

Алгоритм описывается в командах исполнителя, который это алгоритм будет выполнять.
Исходные данные и результаты любого алгоритма всегда принадлежат среде того исполнителя, для которого предназначен алгоритм.

Свойства алгоритма

1. *Дискретность* – алгоритм представляет процесс решения задачи как последовательное выполнение некоторых простых шагов. При этом для выполнения каждого шага алгоритма требуется конечный отрезок времени, преобразование исходных данных в результат осуществляется во времени дискретно.

2. **Элементарность шагов** означает, что объем работы, выполняемой на любом шаге, мажорируется некоторой константой, зависящей от характеристик исполнителя алгоритмов, но не зависящей от входных данных и промежуточных значений, получаемых алгоритмом.

3. *Детерминированность* - определённость. В каждый момент времени следующий шаг работы алгоритма однозначно определяется состоянием системы: алгоритм выдаёт один и тот же результат для одних и тех же исходных данных.

Свойства алгоритма

4. Понятность - алгоритм для исполнителя должен включать только те команды, которые ему (исполнителю) доступны, которые входят в его систему команд.

5. Результативность - завершение алгоритма определенными результатами.

Если же входные данные уникальны, то алгоритм в силу свойства определенности (детерминированности) будет давать всегда один и тот же результат и само построение алгоритма теряет смысл.

6. Конечность (финитность) алгоритма означает, что для получения результата нужно выполнить конечное число шагов, т. е. исполнитель в некоторый момент времени останавливается. Требуемое число шагов зависит от входных данных алгоритма и его структуры. Для вероятностных алгоритмов в результаты работы включают их вероятность.

7. Массовость - алгоритм должен быть применим к разным, допустимым условием задачи, наборам исходных данных.

Алгоритм всегда рассчитан на выполнение «неразмышляющим» исполнителем.

Алгоритм не содержит ошибок, если он дает правильные результаты для любых допустимых исходных данных.

Алгоритм содержит ошибки, если приводит к получению неправильных результатов либо не дает результатов вовсе.

Виды алгоритмов

- Механические алгоритмы - детерминированные, жесткие
- Гибкие алгоритмы
- Вероятностный (стохастический) алгоритм
- Эвристический алгоритм
- Линейный алгоритм
- Разветвляющийся алгоритм
- Циклический алгоритм
- Вспомогательный (*подчиненный*) алгоритм

Разработка алгоритма решения задачи - это разбиение задачи на дискретные этапы (последовательные или параллельные), причем результаты выполнения предыдущих этапов могут использоваться при выполнении последующих.

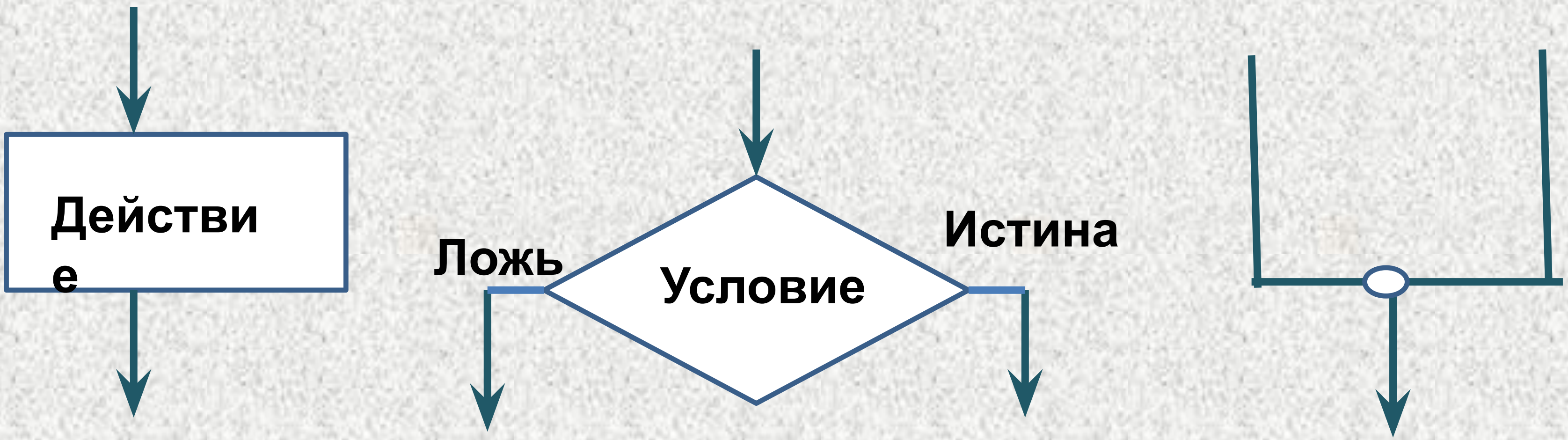
Разработанный алгоритм можно записать несколькими способами:

- на естественном языке;
- в виде блок-схемы;
- на языке разработки приложений.

БЛОК-СХЕМА

ориентированный граф, указывающий порядок исполнения команд алгоритма.

Типы вершин:



функциональн
ая вершина

предикатная
вершина

вершина
слияния

Алгоритмические структуры

Следование предполагает последовательное выполнение команд сверху вниз.

Если алгоритм состоит только из структур следования, то он является **линейным.**

Алгоритмические структуры

Ветвление

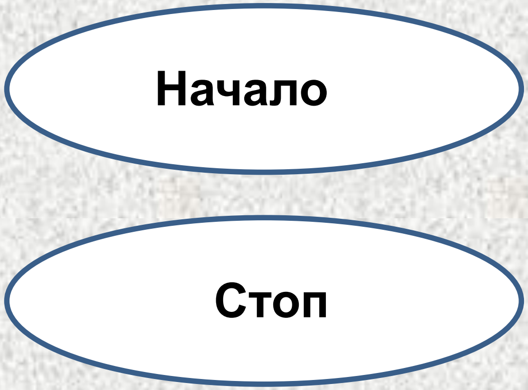
Выполнение программы идет по одной из двух, нескольких или множества ветвей.

Выбор ветви зависит от условия на входе ветвления и поступивших сюда данных.

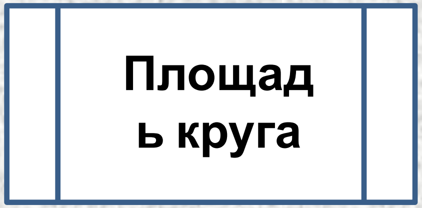
Неполное ветвление

Алгоритмические структуры

Цикл. Предполагает возможность многократного повторения определенных действий. Количество повторений зависит от условия цикла.

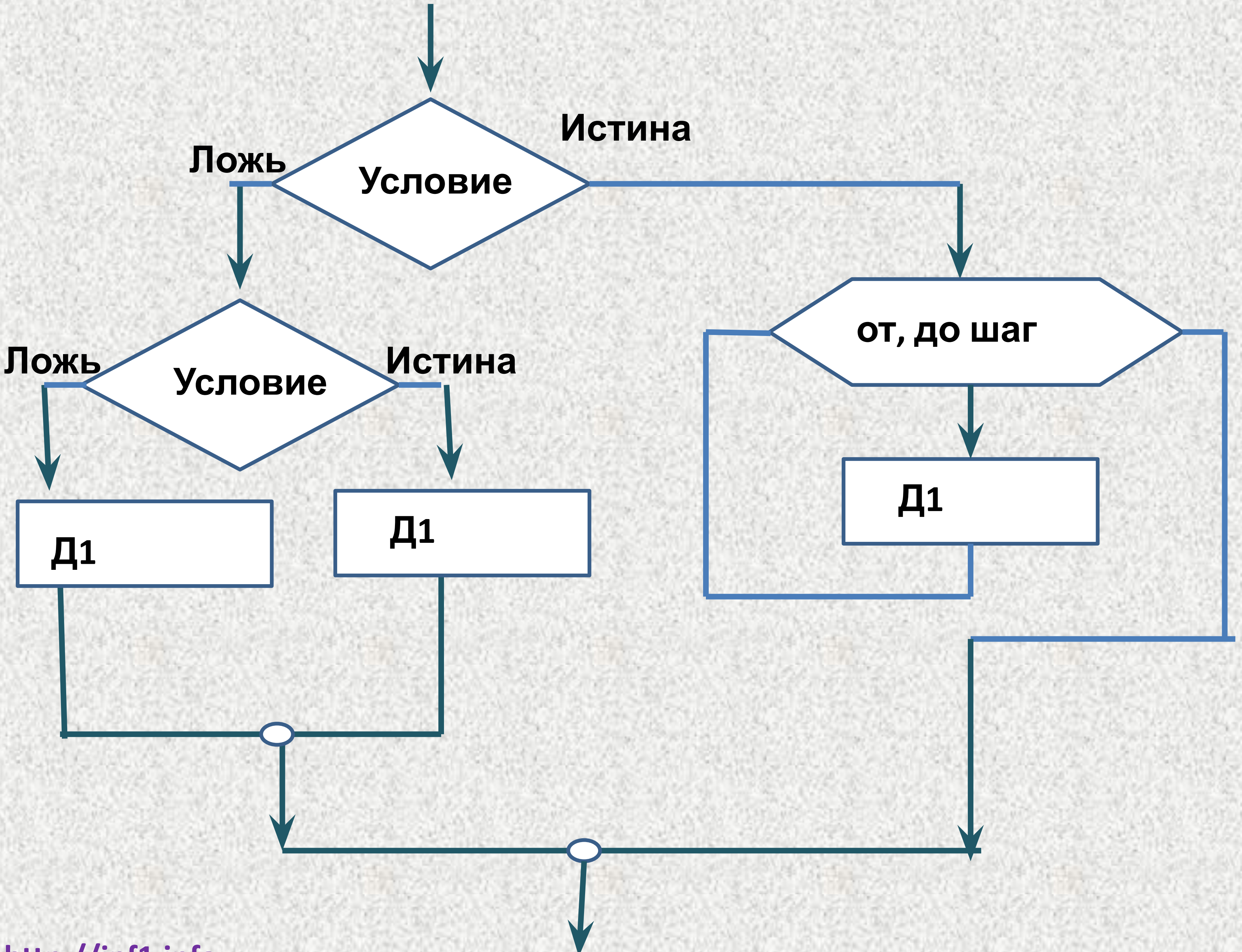


начало и завершение алгоритма



функция (подпрограмма). Команды, отделенные от основной программы, выполняются в случае их вызова из основной программы.

операции ввода и вывода данных

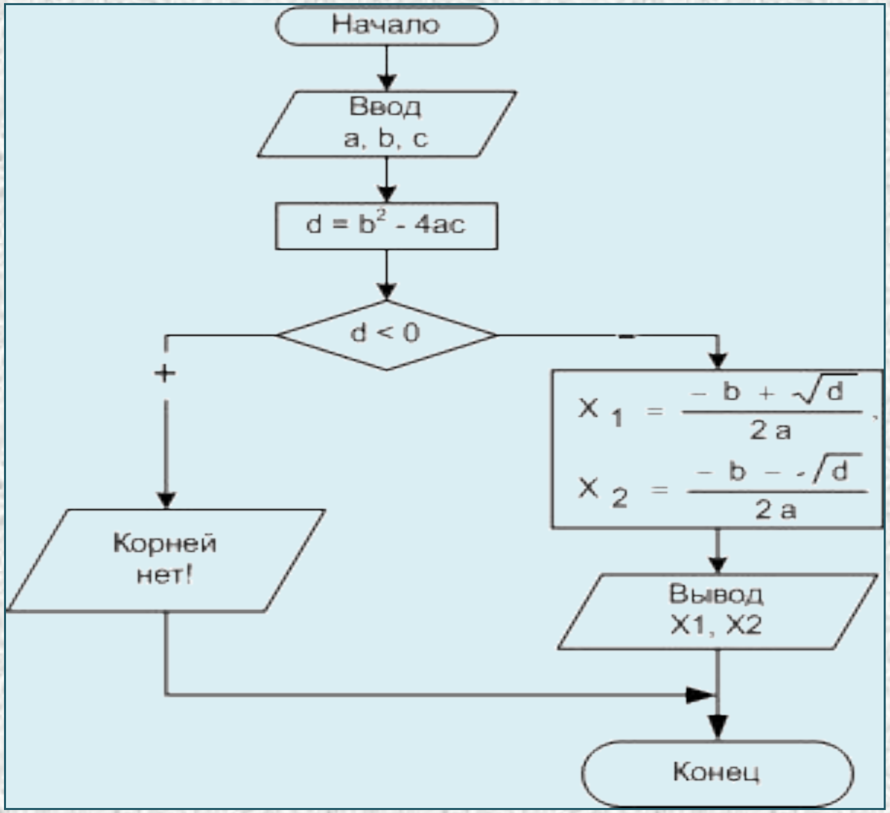


http://inf1.info

ТРЕНИНГ

Пример 1

Блок-схема алгоритма решения квадратного уравнения

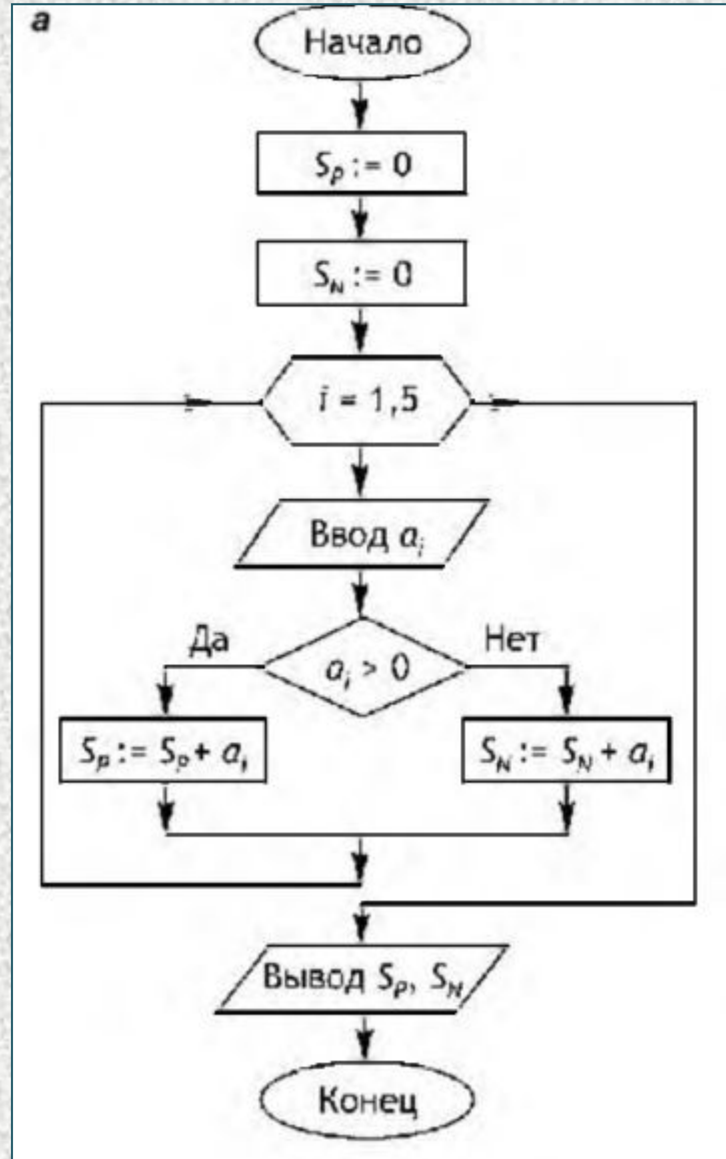


ТРЕНИНГ

Пример 2

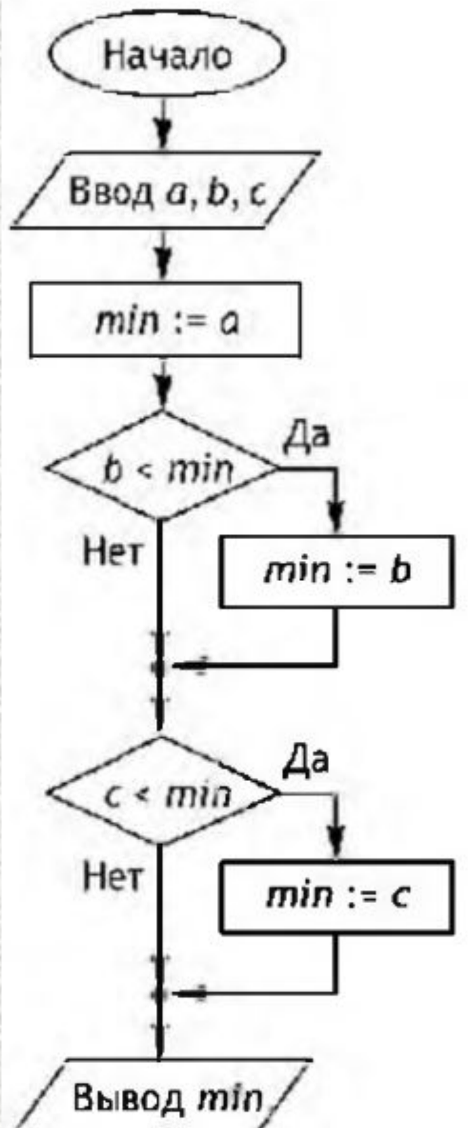
**Определить сумму положительных (Sp)
и сумму отрицательных и нулевых (Sn)
элементов массива а (1:5).**

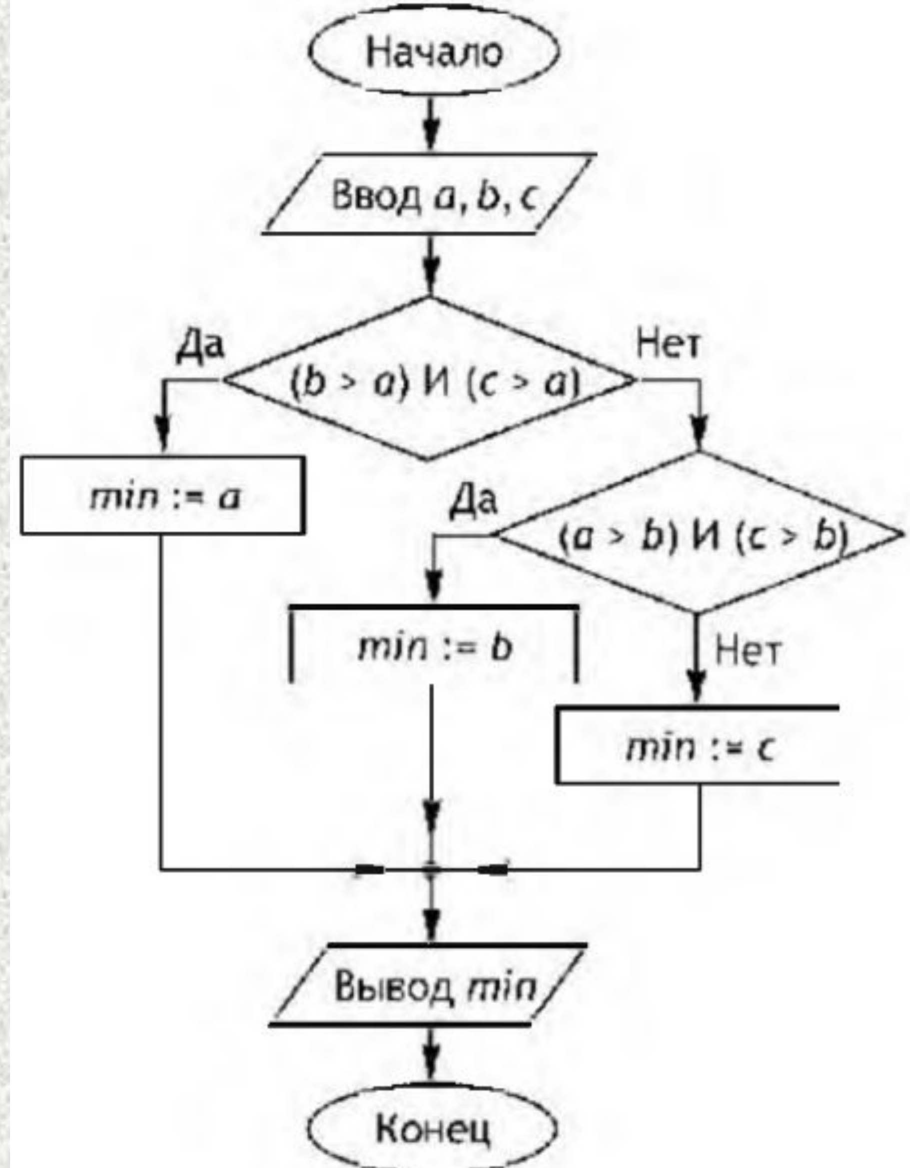
```
Sp:=0
SN:=0
нц для i от 1 до 5
   Ввод ai
   если ai>0
   то Sp:=Sp+ai
   иначе SN:=SN+ai
   всё
кц i
Вывод Sp, SN
```



ТРЕНИНГ

Пример 3





Примем за min одно из чисел, например, min=a. Затем определяем наименьшее попарным сра... и b, min и c.

Вычисляем соотношения: если (b>a & c>a), то min=a, иначе (если (a>b & c>b), то min=b,

Алгоритмы определяют наименьшее из чисел: a, b,

ТРЕНИНГ

Пример 4

Алгоритм выполняет …

а) попарную перестановку значений переменных

A ⇔ B и C ⇔ D

б) циклическое перемещение вправо значений между переменными A, B, C, D по схеме:

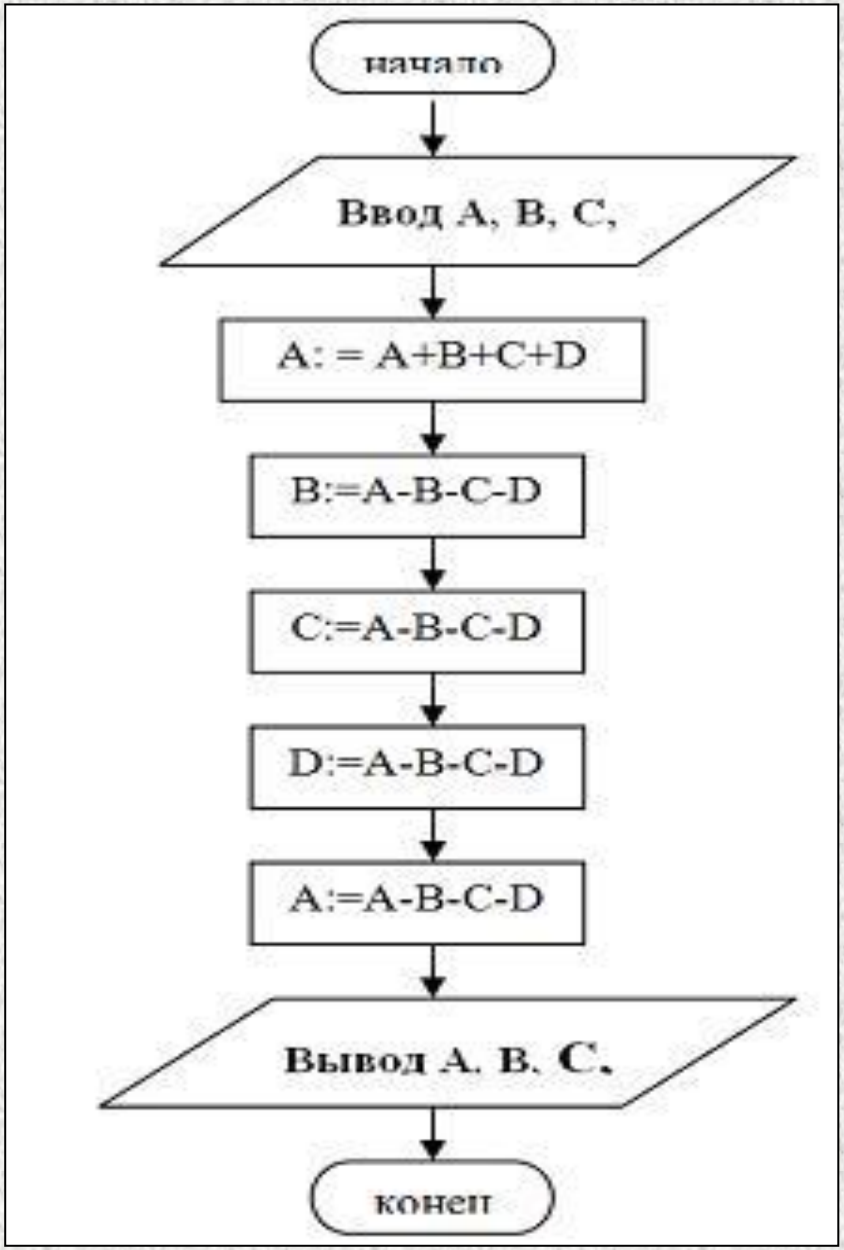
A ⇒ B ⇒ C ⇒ D ⇒ A

в) циклическое перемещение влево значений между переменными A, B, C, D по схеме:

A ⇐ B ⇐ C ⇐ D ⇐ A

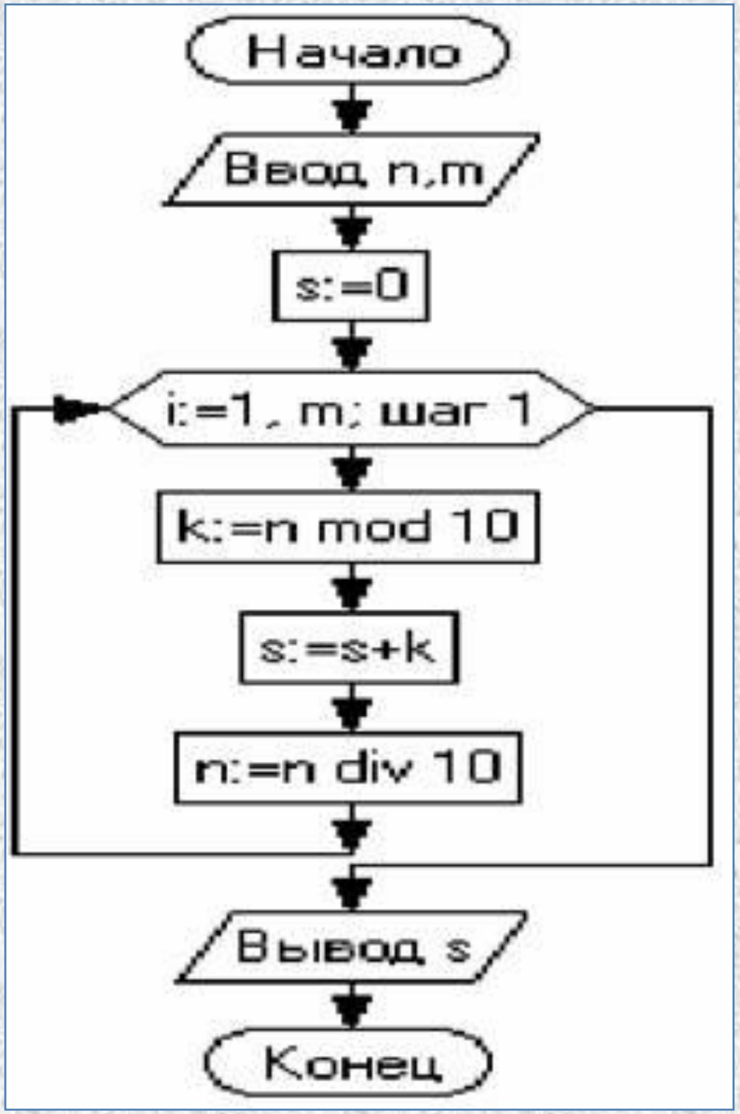
г) попарную перестановку значений переменных:

A ⇔ D и C ⇔ B



ТРЕНИНГ

Пример 5



Операция ***a mod b*** вычисляет остаток от деления числа ***a*** на ***b***.

Операция ***a div b*** определяет целую часть от деления числа *a* на *b*.

В результате выполнения алгоритма при входных данных ***n=8975, M=4*** значение переменной ***S*** будет равно …

14 **29** **2520** **5798**