ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В КОМПЬЮТЕРЕ:

СИСТЕМЫ СЧИСЛЕНИЯ, ПЕРЕВОД ЧИСЕЛ ИЗ ОДНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ

Представление информации в компьютере

- Аналоговое – каждый параметр имеет *непрерывное* множество возможных значений. Примеры:
  – аналоговые ЭВМ
  – аналоговый звуковой и видео сигнал

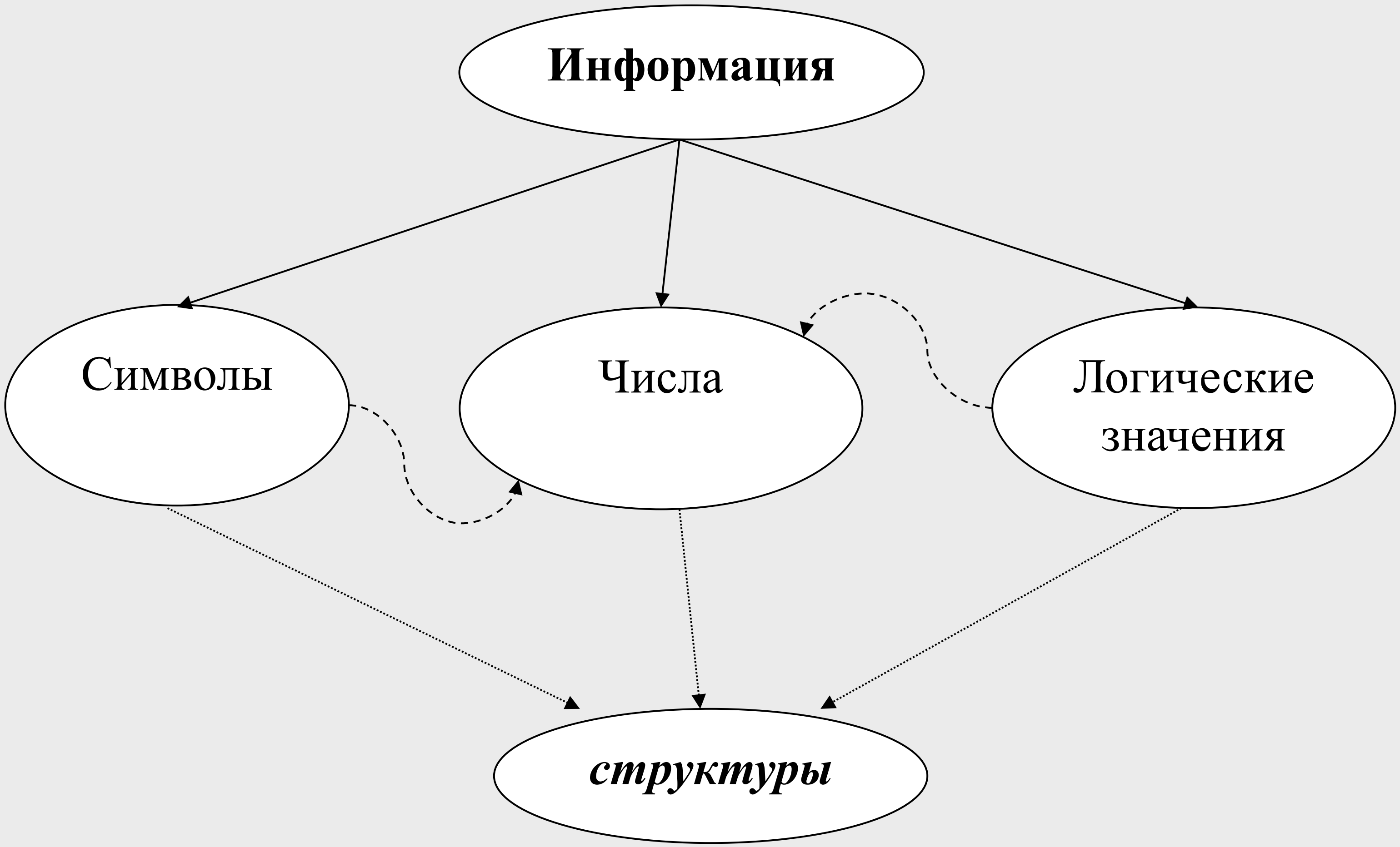
- Цифровое – каждый параметр имеет *конечное* множество возможных значений

Выполнение вычислений на основе алгоритмов предполагает **дискретность** обрабатываемой информации:

то есть любой информационный объект может принимать конечное число значений.

Таким образом, **основной вид представления данных** в компьютере – **цифровой**.

Основные способы представления цифровой информации в компьютере:



Числа, по сути, являются базовой единицей представления информации в компьютере.

Система счисления (с.с.) – это способ представления (записи) любого числа с помощью некоторого алфавита символов (цифр).

Кодом числа называют его представление в данной системе счисления (упорядоченная совокупность цифр).

Пусть число $\boldsymbol{N}$ представлено в некоторой системе счисления с основанием K. Тогда:

$a_l a_{l-1} \ldots a_1$ – код числа в с.с. с основанием K

$a_i \ \left(i = \overline{1,l}\right)$ – цифры алфавита

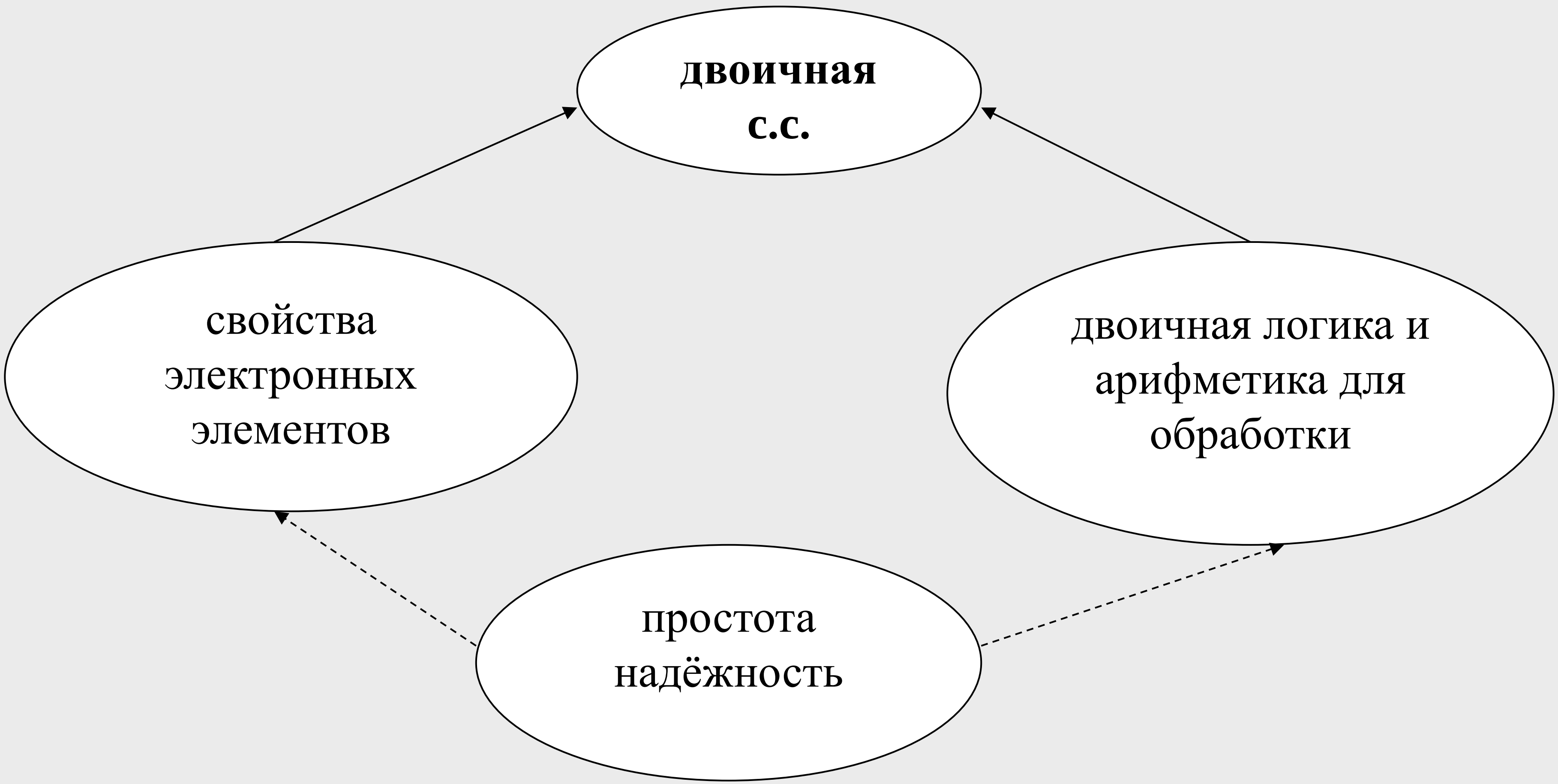
$$0 \le a_i \le K - 1$$

K^{i-1} – вес цифры (разряда)

$l = \log_K N$ – длина кода

$$N = \sum_{i=1}^{l} a_i K^{i-1}$$

В компьютерах в качестве базовой принята **двоичная система счисления**:



Базовой единицей **представления** и **хранения информации** в компьютере является ***бит*** – двоичная цифра, один разряд с.с. (1/0, да/нет, истина/ложь).

Также бит – единица измерения ***количества*** информации, которая может быть представлена одним разрядом двоичного числа

Для определения количества информации используются, также, **производные** единицы:

- Байт, byte – совокупность битов, обрабатываемая компьютером одновременно; во многих архитектурах – минимальный адресуемый объем памяти, **байт = 8 бит**;
- Ниббл, полубайт, nibble – совокупность из 4 битов
- Машинное слово – 2 и более байт, принято:

Слово = 2 байта (16 бит), Двойное слово = 4 байта

Порядок байт (endianness)

Определяет, как хранятся в памяти слова - многобайтовые числа

- от старшего к младшему, big-endian (младшие разряды по старшему адресу), network
  - используется в компьютерных сетях и некоторых процессорах (SPARC), многих кроссплатформенных форматах
  - аналогичен используемому человеком,

Порядок байт (endianness)

- <u>от младшего к старшему, little-endian</u> (младшие разряды по младшему адресу)
  - – используется в **архитектуре x86**
  - – также интерфейсе USB, таблице GUID

  Основной!

- <u>смешанный порядок, middle-endian</u>: машинные слова записаны одном порядке, а байты в них – в обратном

В описании числовой компьютерной информации совместно с двоичной с.с. иногда используются также **производные системы счисления -** восьмеричная с.с., шестнадцатеричная с.с.: основания таких систем связаны степенной зависимостью $K_{произв} = K^{p}_{исх}$, где p – целое число.

Восьмеричная, шестнадцатеричная с.с.

- Программисту приходится работать с двоичными числами.
  - проблема: в 2-й с.с. длинная запись, легко ошибиться
  - решение: использовать системы счисления с основанием, кратным 2
- Человеку более удобна запись в шестнадцатеричной с.с.
  - каждая цифра заменяет 4 двоичных разряда (тетраду)
  - в байте 2 цифры

Основные правила перевода чисел в новую систему счисления

Особенности:

•Различаются для целой и дробной части

•Простота для перевода в производную с.с.

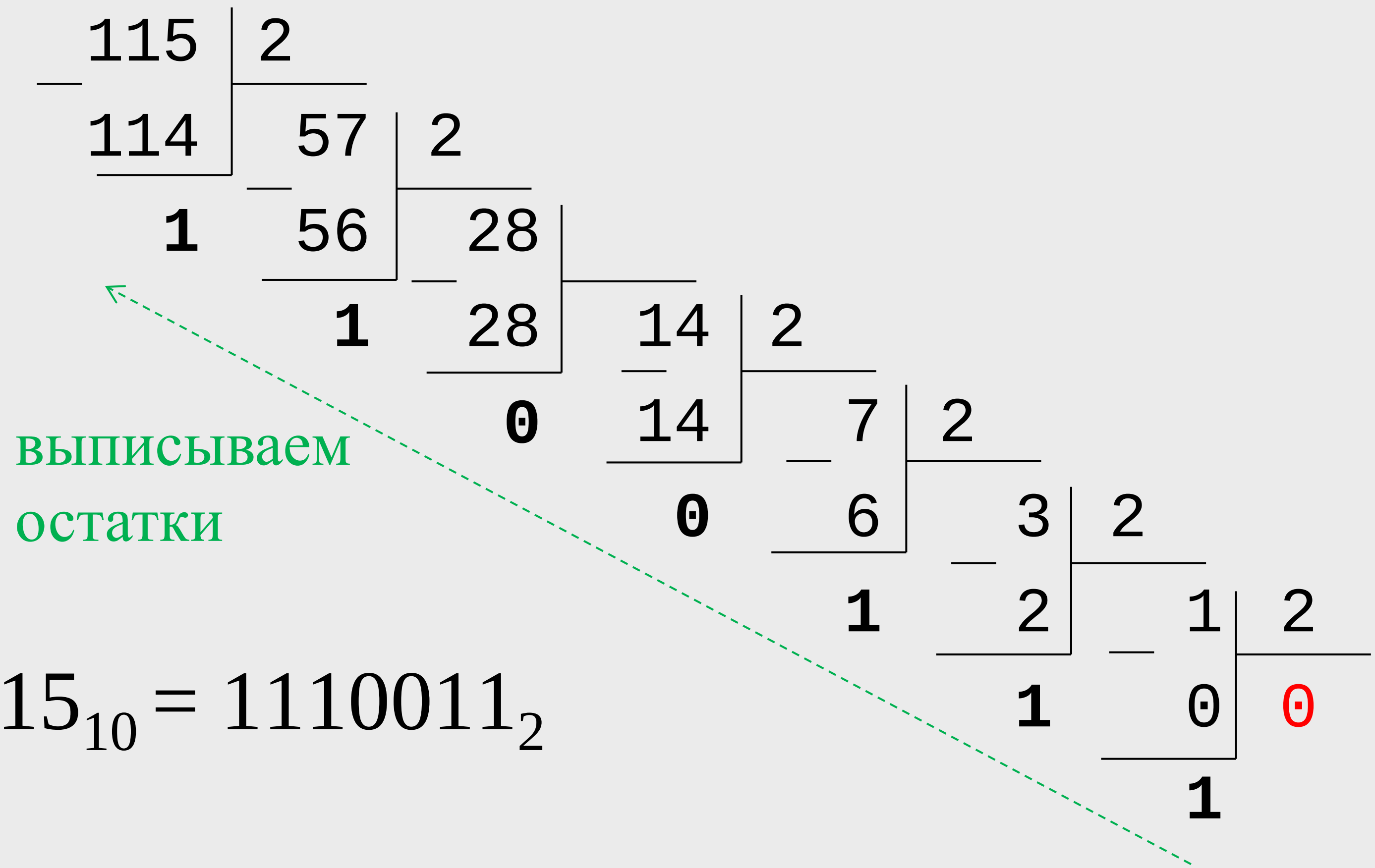
А) Правило перевода целой части числа:

Делим число **нацело** *на* ***основание новой системы счисления*** до тех пор, пока частное не станет равным 0.

Остаток от деления на каждом шаге и есть **цифра** числа в новой с.с..

Сборка (**запись**) цифр производится в **обратном** порядке (от конца – к началу).

Пример перевода десятичного 115 числа в 2-ю с.с.:



$$115_{10} = 1110011_2$$

Пример.

Выполним перевод числа 34_{10} в двоичную, восьмеричную и шестнадцатеричную с.с..

34	2					
-34	17	2				
0	-16	8	2			
	1	-8	4	2		
		0	-4	2	2	
			0	-2	1	2
				0	-0	0
					1	

$34_{10} = 100010_2$

34	8	
-32	4	8
2	-0	0
	4	

$34_{10} = 42_8$

34	16	
-32	2	16
2	-0	0
	2	

$34_{10} = 22_{16}$

Ниже приведена таблица соответствий цифр 10-й, 2-й, 8-й и 16-й с.с..

10 с.с.	**2 с.с.**	**8 с.с.**	**16 с.с.**
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Выполним обратный перевод:

$100010_2 = 1*2^5 + 1*2^1 = 32 + 2 = 34_{10}$

$42_8 = 4*8^1 + 2*8^0 = 32 + 2 = 34_{10}$

$22_{16} = 2*16^1 + 2*16^0 = 32 + 2 = 34_{10}$

Б) Правило перевода дробной части числа:

Умножаем число на **основание новой с.с.**.

Полученная ***целая часть*** произведения и является **цифрой** числа в новой с.с. (в дальнейших вычислениях на следующем шаге не принимает участия).

Сборка (**запись**) цифр производится **в прямом порядке** (по мере получения цифр).

Когда же следует прекратить процесс (в общем случае может понадобиться бесконечное число шагов)?

Для этого можно использовать 2 критерия:

1) – задана **точность** или **количество разрядов** (цифр).

В этом случае нужно получить ***заданное количество цифр***.

2) – точность **не задана**.

В этом случае количество цифр *определяется* точностью представления **исходного** числа: в новой системе счисления она должна быть ***не хуже***, чем в исходной – иначе появится дополнительная погрешность, обусловленная неправильным выбором количества цифр (хорошо видна при обратном переводе).

$$\delta_{исх} \geq \delta_{нов}; \Rightarrow \delta_{исх} \geq \frac{1}{K^l}; \Rightarrow l \geq \log_K \frac{1}{\delta_{исх}}$$

(причем, ***l*** нужно округлять до целого в большую сторону).

Пример.

Выполним перевод числа $0{,}34_{10}$ в двоичную, восьмеричную и шестнадцатеричную с.с..

Так как точность перевода не задана, пользуемся критерием 2: точность исходного числа – 0,01, поэтому при переводе нам понадобится:

$l_2 \geq \log_2 \dfrac{1}{0{,}01} = 7$ двоичных цифр, $l_8 \geq \log_8 \dfrac{1}{0{,}01} = 3$ восьмеричных цифр,

$l_{16} \geq \log_{16} \dfrac{1}{0{,}01} = 2$ шестнадцатеричных цифр

0,	34
x	2
0,	68
x	2
1,	36
x	2
0,	72
x	2
1,	44
x	2
0,	88
x	2
1,	76
x	2
1,	52

$0{,}34_{10} = 0{,}0101011_2$

0,	34
x	8
2,	72
x	8
5,	76
x	8
6,	08

$0{,}34_{10} = 0{,}256_8$

0,	34
x	16
5,	44
x	16
7,	04

$0{,}34_{10} = 0{,}57_{16}$

Выполним обратный перевод:

$0{,}0101011_2 = 1*2^{-2} + 1*2^{-4} + 1*2^{-6} + 1*2^{-7} = 0{,}3359.._{10}$

Проверка: $0{,}34 – 0{,}3359 = 0{,}0041 < 0{,}01$

$0{,}256_8 = 2*8^{-1} + 5*8^{-2} + 6*8^{-3} = 0{,}3398.._{10}$

Проверка: $0{,}34 – 0{,}3398 = 0{,}0002 < 0{,}01$

$0{,}57_{16} = 5*16^{-1} + 7*16^{-2} = 0{,}3398.._{10}$

Проверка: $0{,}34 – 0{,}3398 = 0{,}0002 < 0{,}01$

В) <u>Правило перевода смешанного числа</u>

(содержит и целую, и дробную часть) –

по сути, *нужно применять правила А) и Б)*

<u>отдельно</u> для целой и дробной частей, а затем

записать полученные цифры соответственно до и

после точки.

Г) Правила перевода в производную с.с.:

•*если основание новой (производной) с.с. больше исходной*, то используется разбиение исходного кода на группы цифр (от точки), причем количество цифр в группе определяется показателем степени ***p***, связывающим основания новой и исходной с.с. ($K_{нов} = K^{p}_{исх}$) , и дальнейшая запись этих групп цифрами новой с.с.

•*если основание новой с.с. меньше исходной* (т.е. фактически исходная с.с. является производной от новой), то используется запись каждой цифры исходной с.с. группой цифр новой с.с., причем количество цифр в группе определяется показателем степени ***p***, связывающим основания новой и исходной с.с..

*При этом можно пользоваться таблицей соответствия цифр 2-й, 8-й и 16-й с.с..

Пример.

Выполним перевод кодов чисел 34_{10} и $0{,}34_{10}$ из двоичной в восьмеричную и шестнадцатеричную с.с..

$34_{10} = 100010_2 = 100'010_2 = 42_8$

$34_{10} = 100010_2 = 0010'0010_2 = 22_{16}$

$0{,}34_{10} = 0{,}0101011_2 = 0{,}010'101'100_2 = 0{,}254_8$

$0{,}34_{10} = 0{,}0101011_2 = 0{,}0101'0110'_2 = 0{,}56_{16}$

Замечание:

При необходимости перевода кода числа из **8-й** с.с. в **16-ю** с.с. или наоборот, нужно выполнить перевод сначала в базовую **2-ю** с.с., а затем в производную от нее 8-ю или 16-ю с.с. (перевод через ***промежуточную с.с***.), т.к. 8-я и 16-я с.с. не являются непосредственно производными друг от друга, но обе являются производными от 2-ой с.с..

Перевод чисел между 2-й, 8-й и 16-й с.с.

- Пример:

$9C.23_{16} = X_8$

тетрады

$$9C.23_{16} = 10011100.00100011_{2}$$

триады

$$= 234.106_8$$

Пример.

Выполним перевод кодов чисел 34_{10} и $0{,}34_{10}$ из восьмеричной с.с. в шестнадцатеричную с.с.:

$34_{10} = 42_8 = 100'010_2 = 0010'0010_2 = 22_{16}$

$34_{10} = 22_{16} = 0010'0010_2 = 00'100'010_2 = 42_8$

$0{,}34_{10} = 0{,}254_8 = 0{,}010'101'100_2 = 0{,}0101'0110'0_2 = 0{,}56_{16}$

$0{,}34_{10} = 0{,}56_8 = 0{,}0101'0110'_2 =0{,}010'101'100_2 = 0{,}254_8$