

Соловьев С.Ю.
soloviev@glossary.ru

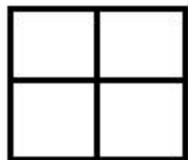
Алгоритмы и **А**лгоритмические языки

www.park.glossary.ru/pascal/

Лекция No.3

2022

Напоминание



Задача



Алгоритм



человек



Исполнитель



Компьютер



Машина Тьюринга



Алгоритмически
неразрешимые задачи



НАМ

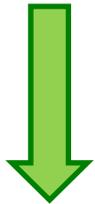
Компьютеры / ЭВМ

исполнитель алгоритмов

ЭНИАК 1946 (США)

МЭСМ 1950 (СССР)

< ПАМЯТЬ, ПРОЦЕССОР, ВНЕШНИЕ УСТРОЙСТВА >



Процессор выполняет команды.
Команды и данные хранятся в памяти.

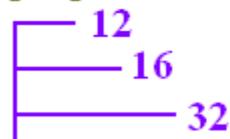
1960-е

нумерованная последовательность ячеек

Я ч е й к а



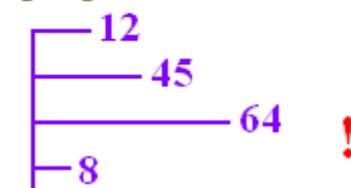
разрядность



Адрес: Содержимое



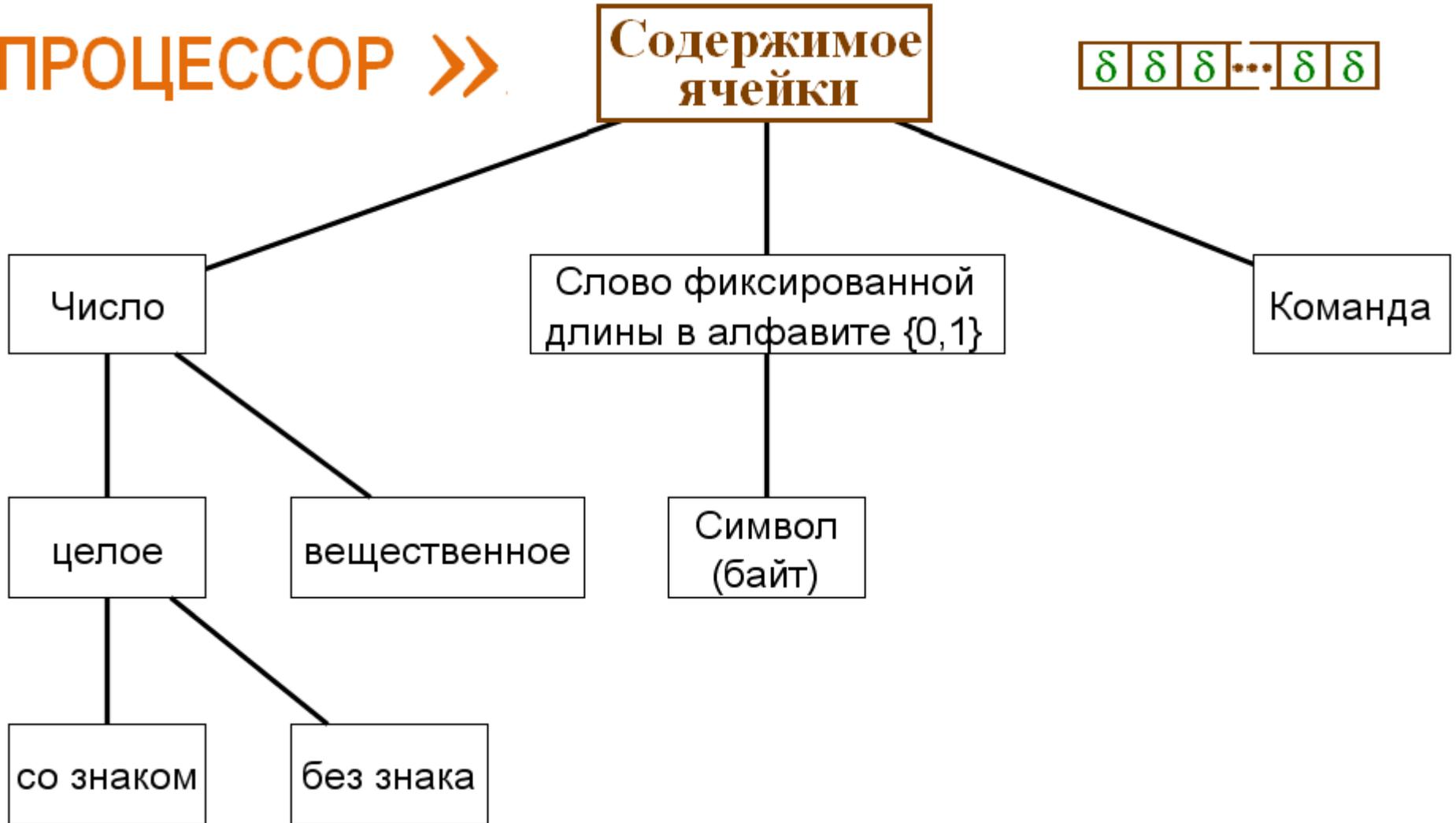
разрядность



12, 12 — PDP 8
12, 45 — БЭСМ-4
24, 8 — IBM/360 (1964)
16, 8 — Intel 286

ПАМЯТЬ

ПРОЦЕССОР >>



Целые

без знака	n = 8	со знаком	n = 8
δδδδδδδδδ		δδδδδδδδδ	
00000000	0	00000000	0
00000001	1	00000001	1
00000010	2	00000010	2
.....		
01111111	127	01111111	127
10000000	128	10000000	-128
10000001	129	10000001	-127
.....		
11111111	255	11111111	-1
$0..2^n-1$		$-2^{n-1}..2^{n-1}-1$	

N – целое, получить -N
 Алгоритм: (a) инверсия N
 (б) +1

Пример: N=11

$$11_{10} = 00001011 \quad (a)$$

$$11110100 \quad (б)$$

$$11110101 = -11_{10}$$

Проверка 11+(-11)

$$\begin{array}{r} 00001011 \\ + 11110101 \\ \hline 100000000 \end{array}$$

С

Размер(фильм.avi) > 2³¹

$$\begin{array}{r} 01111111 \\ + 00000001 \\ \hline 10000000 \end{array} \quad \boxed{127+1 \neq 128}$$

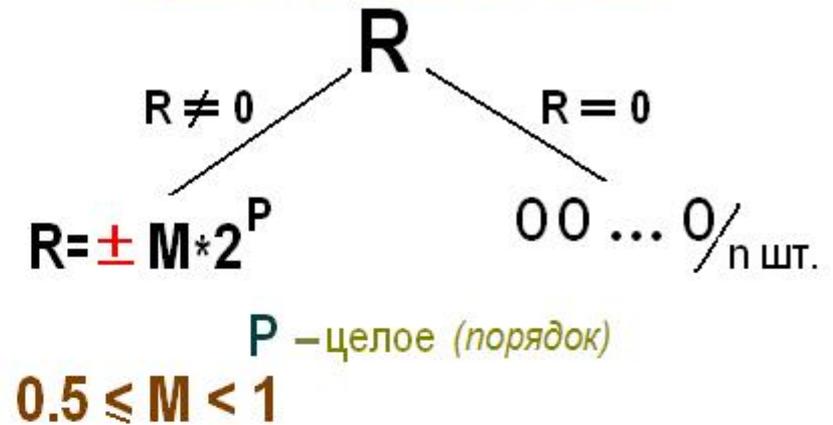
-?-ABOCT

Вещественные

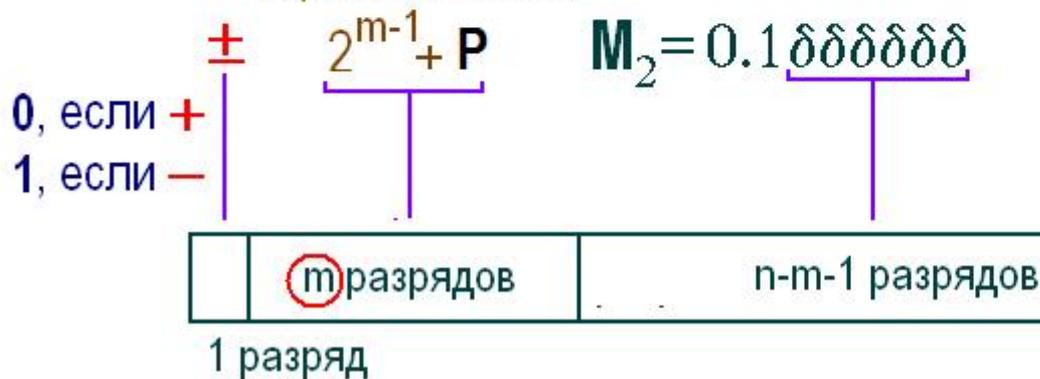
числа с плавающей запятой; ячейка – n разрядов



нормализованные числа



характеристика



float: $n = 32, m = 8$
double: $n = 64, m = 11$

Операции с нормализованными числами

$$1/4 - 1/16 =$$

$$= 0.1_2 * 2^{-1} - 0.1_2 * 2^{-3} = \text{выравнивание порядков}$$

$$= 0.1_2 * 2^{-1} - 0.001_2 * 2^{-1} = \text{вычитание}$$

$$= 0.011_2 * 2^{-1} = \text{нормализация}$$

$$= 0.11_2 * 2^{-2} \quad \text{результат } 3/16$$

Восьмеричные и шестнадцатиричные числа

Восьмеричные

000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Шестнадцатиричные

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

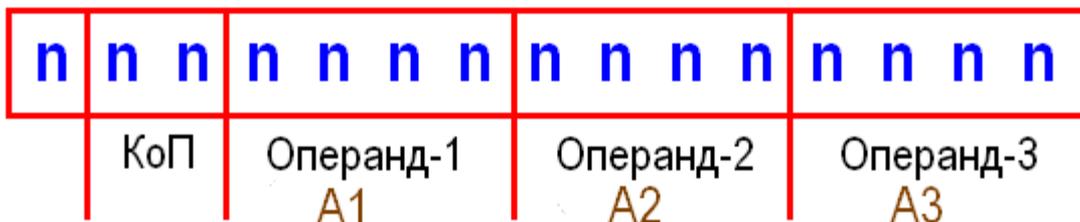
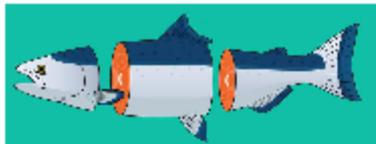
1000011101 = 2035₈

1000011101 = 41D₁₆

Команды

(на примере БЭСМ-4)

Содержимое ячейки: n n БЭСМ-4, 45 разрядов



Команды

[A] – содержимое ячейки с адресом A

[0000] \equiv
0000000000000000

		$\omega = 1$
00	[A1] переслать в A3	
01	вещ.[A1]+вещ.[A2] переслать в A3	Рез-т < 0
02	вещ.[A1]-вещ.[A2] переслать в A3	Рез-т < 0
04	вещ.[A1]/вещ.[A2] переслать в A3	Рез-т < 1
36	если $\omega = 1$, то след.команда из A2	
56	след.команда из A2	
77	СТОП	

ПРОЦЕССОР (регистры)

КРА – командный регистр адреса;	12 разрядов
РК – регистр команд;	45 разрядов
АР1 – арифметический регистр 1;	45 разрядов
АР2 – арифметический регистр 2;	45 разрядов
АР3 – арифметический регистр 3;	45 разрядов



ПРОЦЕССОР (исполнение команд)

Старт. КРА содержит адрес M

1. [КРА] поместить в РК: КоП A1 A2 A3

2. Увеличить КРА на 1

3. В зависимости от КоП выполнить:

01: • [A1] занести в AP1

56: • A2 занести в КРА

• [A2] занести в AP2

• AP1 + AP2 занести в AP3

• AP3 записать в [A3]

• сформировать ω

4. Перейти к 1.

Программирование в кодах

$$S = \sum_{i=1}^n \frac{1}{i}$$

	КоП	операнды			комментарий
0020					аргумент n
0021					результат S
старт 0022	0 00	0000	0000	0021	0 → S
3	0 00	0000	0000	I	0 → I
4	0 01	I	1	I	I+1 → I ← ЦИКЛ
5	0 04	1	I	R	1/I → R
6	0 01	0021	R	0021	S+R → S
7	0 02	I	0020	R	I-N → R
0030	0 36	0000	0024	0000	если ω = 1, то
1	0 77	0000	0000	0000	СТОП

Программирование в кодах - 2

	КоП	операнды			комментарий
0020					аргумент n
0021					результат S
старт 0022	0 00	0000	0000	0021	0 → S
3	0 00	0000	0000	0033	0 → I
4	0 01	0033	0032	0033	I+1 → I ← ЦИКЛ
5	0 04	0032	0033	0034	1/I → R
6	0 01	0021	0034	0021	S+R → S
7	0 02	0033	0020	0034	I-N → R
0030	0 36	0000	0024	0000	если $\omega = 1$, то
1	0 77	0000	0000	0000	СТОП
2	1 01	4000	0000	0000	конст. 1 вещ.
3					I
4					R

О количестве операндов

00	0000	<i>не исп.</i>	0021
00	0000	<i>не исп.</i>	0033
01	0033	0032	0033
04	0032	0033	0034
01	0021	0034	0021
02	0033	0020	<i>не исп.</i>
36	<i>не исп.</i>	0024	<i>не исп.</i>
77	<i>не исп.</i>	<i>не исп.</i>	<i>не исп.</i>

0 операндов -- 1 оператор
1 операнд -- 1 оператор
2 операнда -- 3 оператора
3 операнда -- 3 оператора

↑
37.5%

- 01:**
- [A1] занести в AP1
 - [A2] занести в AP2
 - AP1 + AP2 занести в AP3
 - AP3 записать в [A3]
 - сформировать ω

Эволюция-1

I. Коды операций: 000 \Rightarrow =, 001 \Rightarrow +, 002 \Rightarrow -,
 004 \Rightarrow /, 036 \Rightarrow if ω , 077 \Rightarrow fin

II. Символические адреса: C, I, N, S, R – ячейки,
 α – начало загрузки, β – старт программы,
 η – метка цикла

III. Секция кода

β : =	0	■	S	0 \rightarrow S
=	0	■	I	0 \rightarrow I
η : +	I	C	I	I+1 \rightarrow I
+	C	I	R	1/I \rightarrow R
+	S	R	S	S+R \rightarrow S
-	I	N	R	I-N \rightarrow R
if ω	■	η	■	если $\omega = 1$, то
fin	■	■	■	СТОП

Секция данных

C	=	1	константа
I			переменная
N	=		конст./число
S			переменная
R			переменная

Эволюция-2

Секция данных

```
C = 1
I
N =
S
R
```

Секция кода

```
β: = 0 █ S
    = 0 █ I
η: + I C I
    + C I R
    + S R S
    - I N R
if ω █ η █
fin █ █ █
```

```
β: 0 → S
    0 → I
η: I+1 → I
    1/I → R
S+R → S
I-N → R
if ω = 1 goto η
fin
```

Программа

```
0020
0021
0022  0 00 00000 00000 0021
      3 0 00 00000 00000 0033
      4 0 01 0033 0032 0033
      5 0 04 0032 0033 0034
      6 0 01 0021 0034 0021
      7 0 02 0033 0020 0034
0030  0 36 00000 0024 00000
      1 0 77 00000 00000 00000
      2 1 01 40000 00000 00000
```



Эволюция-3

Секция данных

```

I
N =
S
    
```

```

β: 0 → S
    0 → I
η: I+1 → I
    1/I → R
    S+R → S
    I-N → R
    if ω = 1 goto η
    fin
    
```

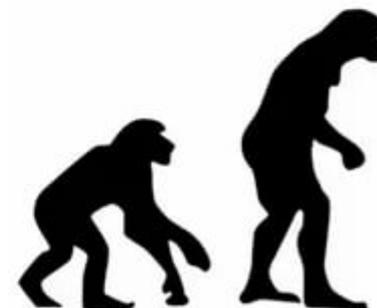
```

β: S = 0
    I = 0
η: I = I + 1
    S = S + 1/I
    if I < N goto η
    fin
    
```

Программа

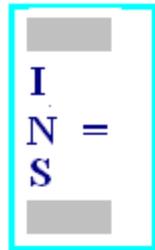
```

0020
0021
0022 0 00 00000 00000 0021
      3 0 00 00000 00000 0033
      4 0 01 0033 0032 0033
      5 0 04 0032 0033 0034
      6 0 01 0021 0034 0021
      7 0 02 0033 0020 0034
0030 0 36 0000 0024 0000
      1 0 77 0000 0000 0000
      2 1 01 4000 0000 0000
    
```



Эволюция-4

Секция данных

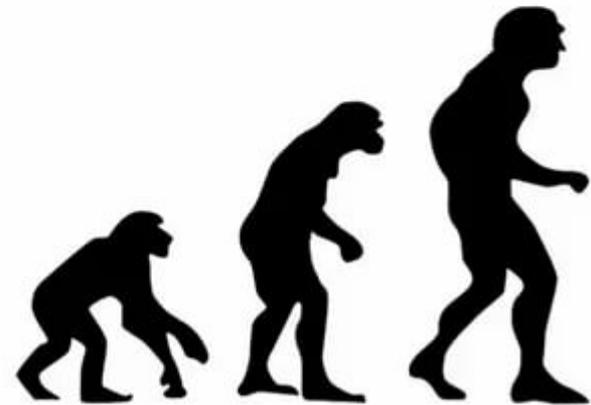


```
β: S = 0  
I = 0  
η: I = I + 1  
S = S + 1/I  
if I < N goto η  
fin
```

```
S = 0  
I = 0  
ПОВТОРЯТЬ  
I = I + 1  
S = S + 1/I  
пока I < N
```

Программа

```
0020  
0021  
0022 0 00 00000 00000 0021  
3 0 00 00000 00000 0033  
4 0 01 0033 0032 0033  
5 0 04 0032 0033 0034  
6 0 01 0021 0034 0021  
7 0 02 0033 0020 0034  
0030 0 36 00000 0024 00000  
1 0 77 00000 00000 00000  
2 1 01 40000 00000 00000
```



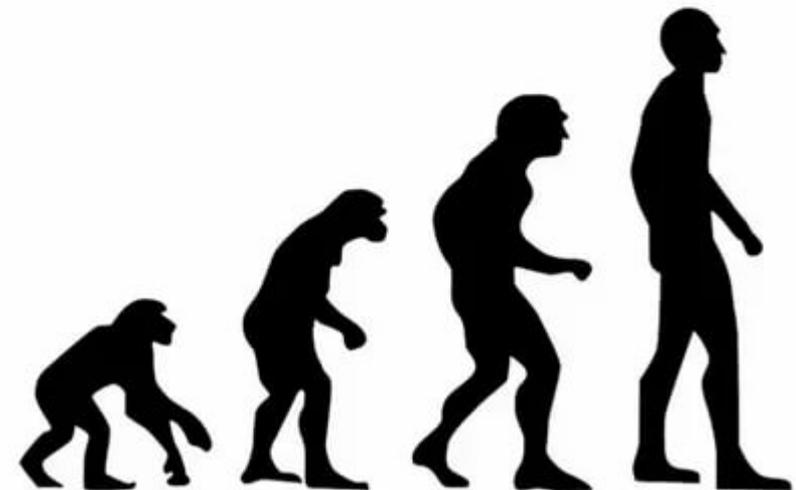
Эволюция-5

числа I, N, S
S = 0
I = 0
ПОВТОРЯТЬ
 I = I + 1
 S = S + 1/I
пока I < N

**Язык
высокого
уровня**

Компилятор
(транслятор)

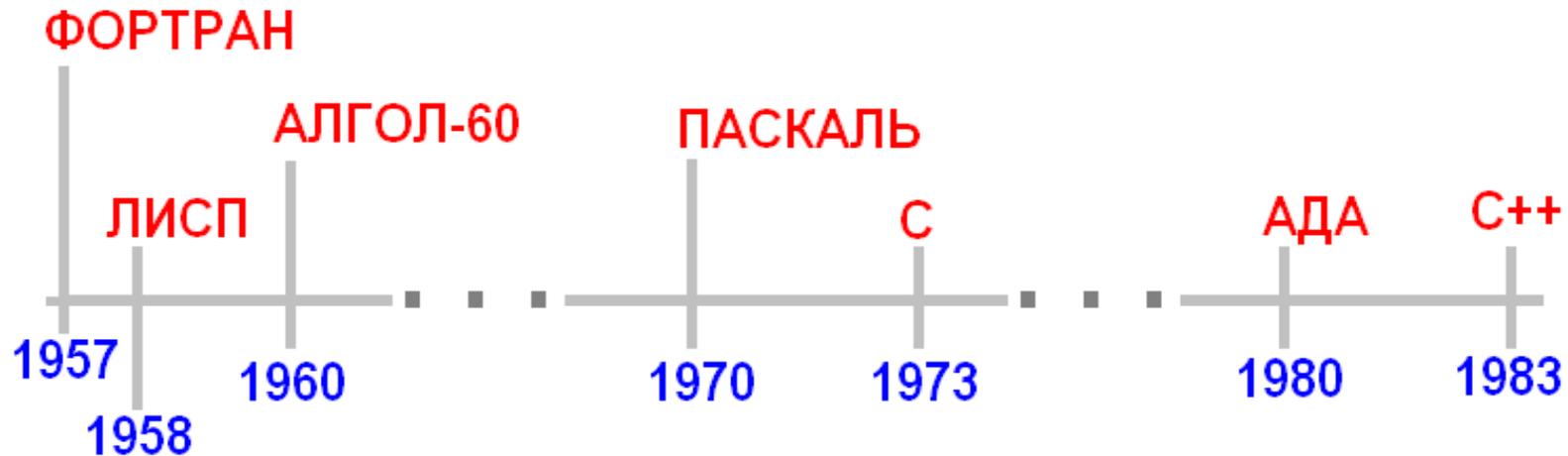
```
0020
0021
0022 0 00 0000 0000 0021
      3 0 00 0000 0000 0033
      4 0 01 00 0032 0033
      5 0 04 0 0033 0034
      6 0 01 0 0034 0021
      7 0 02 0 0020 0034
0030 0 36 00 0024 0000
      1 0 77 0000 0000 0000
      2 1 01 4000 0000 0000
```



Алгоритмические языки (высокого уровня)

Алгоритмический язык – удобный для человека
ЯЗЫК записи алгоритмов.

- наглядность;
- набор операций не зависит от компьютера;
- расширенный набор типов данных.



Обзор лекции No.3

Компьютер: оперативная память, процессор

Ячейки оперативной памяти

Форматы хранения чисел

Операции с нормализованными числами

Система команд

Регистры процессора

Порядок исполнения команд

Программирование в кодах

Мнемокод, ассемблер

Алгоритмические языки высокого уровня

--- Конец лекции No. 3 ---