



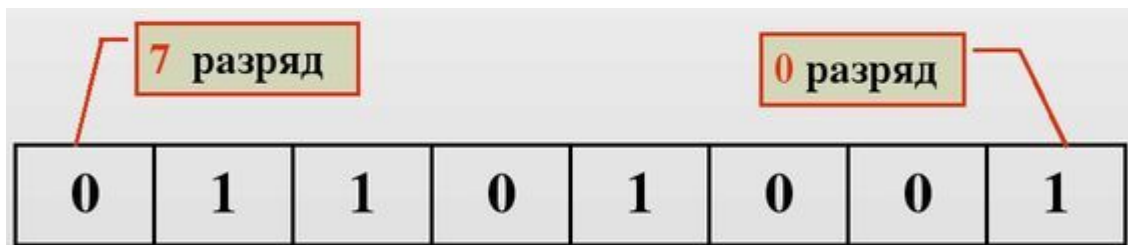
**Старший преподаватель департамента информационных технологий и автоматики**  
**Шеклеин Алексей Александрович**

# ВАРИАНТЫ ПРЕДСТАВЛЕНИЯ ИНФОРМАЦИИ В ЭВМ

- Символы
- Целые числа
- Вещественные числа

## Основные понятия

- Используется двоичный способ кодирования.
- Элементарная ячейка памяти **1 байт = 8 бит**.
- Каждый байт имеет свой номер (**адрес**).
- **Машинное слово** – наибольшая последовательность бит, которую ЭВМ может обрабатывать как единое целое.



- Длина машинного слова зависит от разрядности процессора и может быть равной 16, 32, 64 битам и т.д.

# КОДИРОВАНИЕ СИМВОЛОВ

- Для **кодирования символов** достаточно одного байта.
- С помощью 1 байта можно представить 256 символов (с десятичными кодами от 0 до 255).
- Набор символов ПК (IBM PC) – расширение **кода ASCII** (American Standard Code for Information Interchange – стандартный американский код для обмена информацией).

- Стандартный набор символов использует только 7 битов для каждого символа.
- Добавление 8-го разряда позволяет увеличить количество кодов таблицы ASCII до 255.
- Коды от 128 до 255 – это **расширение таблицы ASCII**.
- Эти коды используются для кодирования символов национальных алфавитов, а также символов псевдографики и т.п.



# КОД ASCII

32		00100000	56	8	00111000	80	P	01010000	104	h	01101000
33	!	00100001	57	9	00111001	81	Q	01010001	105	i	01101001
34	"	00100010	58	:	00111010	82	R	01010010	106	j	01101010
35	#	00100011	59	;	00111011	83	S	01010011	107	k	01101011
36	\$	00100100	60	<	00111100	84	T	01010100	108	l	01101100
37	%	00100101	61	=	00111101	85	U	01010101	109	m	01101101
38	&	00100110	62	>	00111110	86	V	01010110	110	n	01101110
39	'	00100111	63	?	00111111	87	W	01010111	111	o	01101111
40	(	00101000	64	@	01000000	88	X	01011000	112	p	01110000
41	)	00101001	65	A	01000001	89	Y	01011001	113	q	01110001
42	*	00101010	66	B	01000010	90	Z	01011010	114	r	01110010
43	+	00101011	67	C	01000011	91	[	01011011	115	s	01110011
44	,	00101100	68	D	01000100	92	\	01011100	116	t	01110100
45	-	00101101	69	E	01000101	93	]	01011101	117	u	01110101
46	.	00101110	70	F	01000110	94	^	01011110	118	v	01110110
47	/	00101111	71	G	01000111	95	_	01011111	119	w	01110111
48	0	00110000	72	H	01001000	96	`	01100000	120	x	01111000
49	1	00110001	73	I	01001001	97	a	01100001	121	y	01111001
50	2	00110010	74	J	01001010	98	b	01100010	122	z	01111010
51	3	00110011	75	K	01001011	99	c	01100011	123	{	01111011
52	4	00110100	76	L	01001100	100	d	01100100	124		01111100
53	5	00110101	77	M	01001101	101	e	01100101	125	}	01111101
54	6	00110110	78	N	01001110	102	f	01100110	126	~	01111110
55	7	00110111	79	O	01001111	103	g	01100111	127	□	01111111

Таблицы кодов ASCII содержат:

- Коды управляющих символов (0–31)
- Символы с кодами 32–127
- Символы с кодами 128–255 (для MS DOS, для MS Windows)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
0	null	►	Space	0	@	P	'	p	A	P	a	▤	L	Ш	p	Ё
1	☺	◄	!	1	A	Q	a	q	Б	С	б	▥	└	Т	с	ё
2	☺	↕	"	2	B	R	b	r	В	Т	в	▧	┘	П	т	Є
3	♥	!!	#	3	C	S	c	s	Г	У	г		┘	Ц	у	є
4	♦	¶	\$	4	D	T	d	t	Д	Ф	д	┘	—	Е	ф	İ
5	♣	§	%	5	E	U	e	u	Е	Х	е	┘	+	Ф	х	ı
6	♠	—	&	6	F	V	f	v	Ж	Ц	ж	┘	┘	П	ц	ÿ
7	●	↕	`	7	G	W	g	w	З	Ч	з	П	┘	┘	ч	ÿ
8	Bsp	↑	(	8	H	X	h	x	И	Ш	и	┘	┘	≠	ш	◦
9	Tab	↓	)	9	I	Y	i	y	Й	Щ	й	┘	┘	┘	щ	•
A	☒	→	*	:	J	Z	j	z	К	Ъ	к	┘	┘	┘	ъ	•
B	♂	Esk	+	;	K	[	k	{	Л	Ы	л	┘	┘	▀	ы	√
C	♀	L	,	<	L	\	/		М	Ь	м	┘	┘	▀	ь	No
D	♪	↔	-	=	M	J	m	}	Н	Э	н	┘	=	▀	э	я
E	♪	▲	.	>	N	^	n	~	О	Ю	о	┘	┘	▀	ю	■
F	☼	▼	/	?	O	_	o	△	П	Я	п	┘	┘	▀	я	blank



# КОДИРОВКА UNICODE

- В настоящее время чаще используется **двухбайтная кодировка Unicode**.
- Коды символов могут иметь значение от 0 до 65535 (  $2^{16} = 65536$  ).
- В этой кодировке имеются коды для практически всех применяемых символов (букв алфавитов разных языков, математических, декоративных символов и т.д.).

# ПРЕДСТАВЛЕНИЕ ЧИСЕЛ

- Последовательность нескольких битов или байтов называют **полем данных**.
- Биты в числе (в слове, в поле и т.п.) нумеруются справа налево, начиная с 0-го разряда.
- Например, нумерация бит в двухбайтовом машинном слове:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

В ПК могут обрабатываться поля постоянной и переменной длины.

### **Поля постоянной длины:**

- слово – 2 байта;
- полуслово – 1 байт;
- двойное слово – 4 байта;
- расширенное слово – 8 байт.

**Поля переменной длины** могут иметь любой размер от 0 до 256 байт, но обязательно равный целому числу байтов.

# КОДИРОВАНИЕ ЦЕЛЫХ ЧИСЕЛ

Диапазон значений величин чисел зависит от количества бит памяти, отведенных для их хранения.

Например, для **целых чисел** диапазоны могут быть:

Название типа данных	Диапазон	Количество бит для хранения
integer	от -32768 ( $-2^{15}$ ) до 32767 ( $2^{15} - 1$ )	2 байта (16 бит) слово
longint	от $-2^{31}$ до $2^{31} - 1$	4 байта (32 бита) двойное слово
word	от 0 до 65535 ( $2^{16} - 1$ )	2 байта (16 бит) слово

Запись числа  $-193_{10} = -11000001_2$  в  
разрядной сетке ПК.

Число с фиксированной запятой формата  
слово со знаком:

	Знак	Абсолютная величина числа														
№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Число	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1



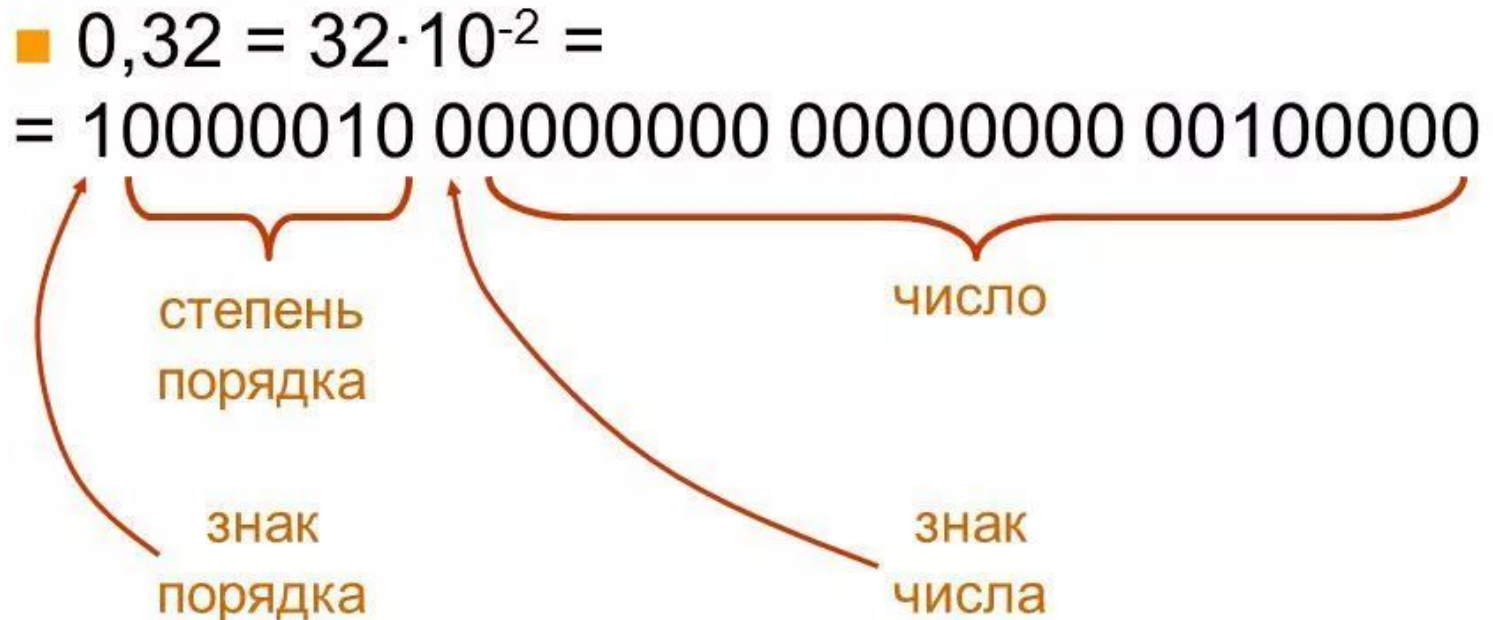
# КОДИРОВАНИЕ ВЕЩЕСТВЕННЫХ ЧИСЕЛ

Действительные типы и диапазон их значений

Тип	Диапазон	Мантисса	Байты
Real	$2,9 \times 10^{-39} \dots 1,7 \times 10^{38}$	11-12	6
Single	$1,5 \times 10^{-45} \dots 3,4 \times 10^{38}$	7-8	4
Double	$5,0 \times 10^{-324} \dots 1,7 \times 10^{308}$	15-16	8
Extended	$3,4 \times 10^{-4932} \dots 1,1 \times 10^{4932}$	19-20	10

Рассмотрим представление чисел с плавающей запятой (в нормальной форме).

Для хранения часто выделяется 8 байт (64 бита, расширенное слово).



Место, отводимое для числа с плавающей точкой, делится на два поля: мантисса и порядок.

Представление величины типа Double в памяти ЭВМ .

**Число с плавающей запятой формата расширенное слово со знаком.**

S	Смещенный порядок	Мантисса
63	62..52	51..0

Мантисса занимает младшие 52 бита. Черта указывает здесь на положение двоичной запятой. Перед запятой должен стоять бит целой части мантиссы, но поскольку она всегда равна 1, здесь данный бит не требуется и соответствующий разряд отсутствует в памяти (но он подразумевается).

Значение порядка хранится здесь не как целое число, представленное в дополнительном коде. Для упрощения вычислений и сравнения действительных чисел значение порядка в ЭВМ хранится в виде **смещенного числа**, т.е. к настоящему значению порядка перед записью его в память прибавляется смещение.

Смещение выбирается так, чтобы минимальному значению порядка соответствовал нуль. Например, для типа Double порядок занимает 11 бит и имеет диапазон от  $2^{-1023}$  до  $2^{1023}$ , поэтому смещение равно  $1023_{(10)} = 1111111111_{(2)}$ .

Наконец, бит с номером 63 указывает на знак числа.

**Мантисса** представляется в нормализованном виде, т.е. ее целая часть равна 1.

Мантисса занимает младшие 52 бита.

Черта указывает здесь на положение двоичной запятой. Перед запятой должен стоять бит целой части мантиссы, но поскольку она всегда равна 1, здесь данный бит не требуется и соответствующий разряд отсутствует в памяти (но он подразумевается).



**Смещенный порядок.** Смещение выбирается так, чтобы минимальному значению порядка соответствовал нуль.

Например, для типа Double порядок занимает 11 бит и имеет диапазон от  $2^{-1023}$  до  $2^{1023}$ , поэтому смещение равно  $1023_{(10)} = 1111111111_{(2)}$ .

**Поле S.** Бит с номером 63 указывает на **знак числа.**

## Алгоритм для получения представления действительного числа в памяти ЭВМ:

1. переведем модуль данного числа в двоичную систему счисления;
2. нормализуем двоичное число, т.е. запишем в виде  $M \times 2^p$ , где  $M$  – мантисса (ее целая часть равна  $1_{(2)}$ ) и  $p$  – порядок, записанный в десятичной системе счисления;
3. прибавим к порядку смещение и переведем смещенный порядок в двоичную систему счисления;
4. учитывая знак заданного числа (0 – положительное; 1 – отрицательное), выпишем его представление в памяти ЭВМ.

Представим код числа **-312,3125**.

1. Двоичная запись модуля этого числа имеет вид 100111000,0101.
2. Имеем  $100111000,0101 = 1,001110000101 \times 2^8$ .
3. Получаем смещенный порядок  $8 + 1023 = 1031$ .  
Далее имеем  $1031_{(10)} = 10000000111_{(2)}$ .
4. Окончательно

[illegible]

Более компактно полученный код следует записать в 16-ой СС:

**C073850000000000**<sub>(16)</sub>.

[illegible]

- 22

# УПРАЖНЕНИЯ

1. Оцените число символов алфавита, кодируемого с помощью двоичных последовательностей длиной
  - a) 4 знака
  - b) 8 знаков
  - c) 12 знаков
  - d) 16 знаков



2. С помощью кодовой таблицы ASCII декодируйте следующее сообщение

01010100 01001111 00100000 01000010 01000101  
00100000 01001111 01010010 00100000 01001110  
01001111 01010100 00100000 01010100 01001111  
00100000 01000010 01000101.

3. С помощью кодовой таблицы ASCII закодируйте в последовательность шестнадцатеричных чисел слово COMPUTER.

4. Запишите код действительного числа, интерпретируя его как величину типа Double.

a)  $-578,375$

b)  $786,375$

5. Дан код величины типа Double.  
Преобразуйте его в число.

a) 408E1300000000000

b) C07788000000000000

Спасибо за внимание!