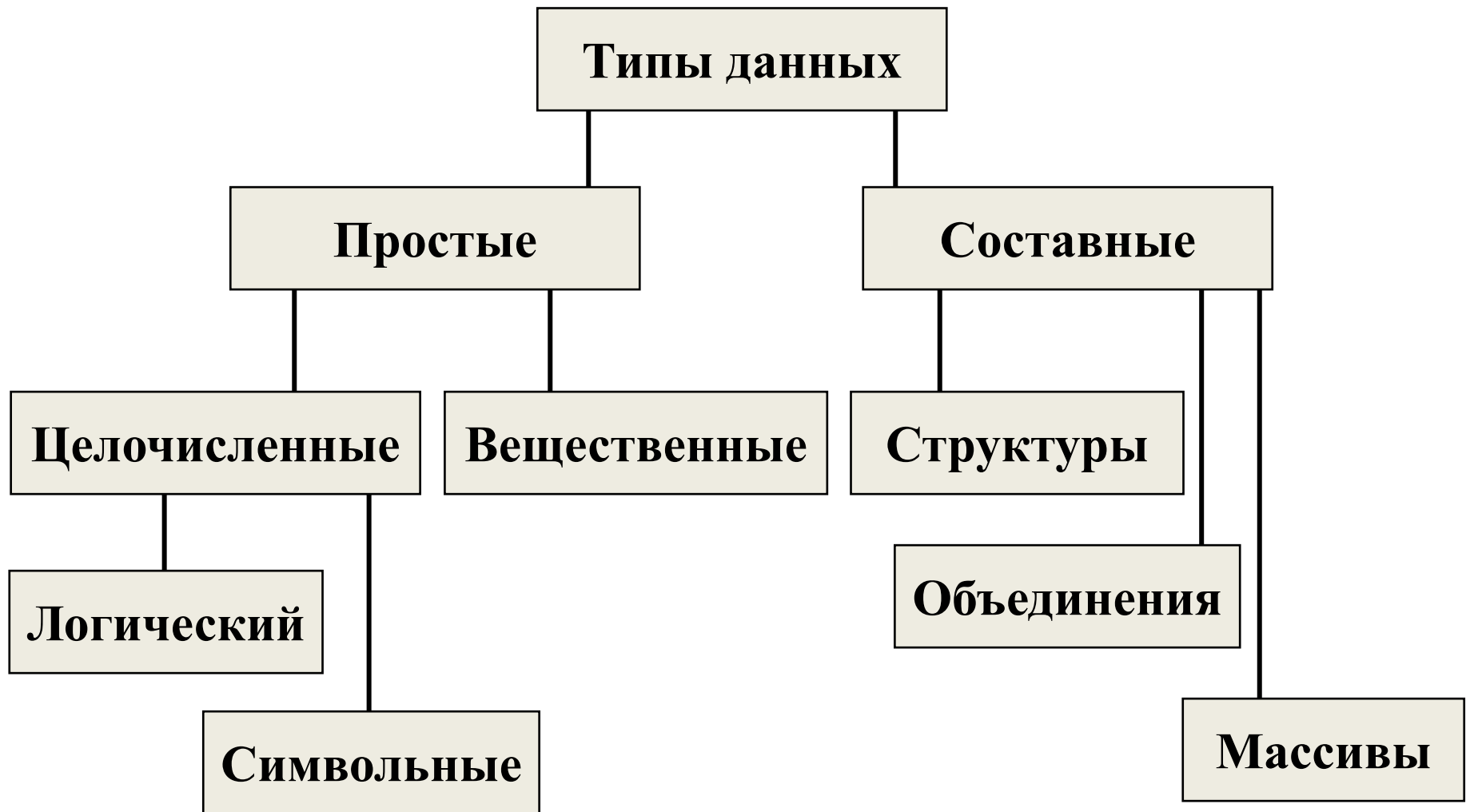


Тема 2.

Типы данных

Классификация типов данных C²



Основные (встроенные) типы данных

Переменная — это ячейка в памяти компьютера, которая имеет имя (**идентификатор**) и хранит некоторое значение определенного типа.

- Значение переменной может меняться во время выполнения программы.
- При записи в ячейку нового значения старое стирается.

bool - логические значения

int - целые числа

char - символы

**float, double - вещественные числа
(числа с плавающей запятой)**

Основы синтаксиса языка C

Идентификаторы

Имена, использующиеся для переменных, функций, меток и других определяемых пользователем объектов, называются **идентификаторами**. В идентификаторе могут использоваться буквы латинского алфавита, цифры и знак подчеркивания "_", но он не может начинаться с цифры. Прописные и строчные буквы различаются. Идентификаторы не должны совпадать с ключевыми словами языка программирования.

Примеры идентификаторов

`_code`

`TEXT`

`Text`

`icon16_16`

`iCountPersonOfBase`

`screen_width`

Правильные и неправильные идентификаторы

5

Name	правильно
Имя	неправильно
12Z	неправильно
_12Z	правильно
dValue	правильно
B34_X	правильно
VAL_ПЕРЕМ	неправильно
__M2__	правильно

Логический тип данных

Тип данных `bool` (включен в C только начиная с C99)

Переменным типа **`bool`** могут быть присвоены только значения **`false`** или **`true`**, которые являются зарезервированными словами. Если переменной логического типа присвоено целое значение, то 0 интерпретируется как **`false`**, а значение, не равное нулю, как **`true`**. В памяти **`bool`** занимает 1 байт.

Пример кода

```
bool var1 = false;  
  
bool var2 = true;
```

Целочисленные типы данных

7

Тип данных **char**

Переменным типа **char** могут быть присвоены целые значения. В памяти **char** занимает 1 байт.

Тип данных **int**

Переменным типа **int** могут быть присвоены целые значения. Размер занимаемой памяти зависит от платформы.

Модификаторы

signed

unsigned

short

long

signed означает целое число со знаком, **unsigned** - целое без знака.

short явно задает размер 2 байта, **long** - 4 байта.

Тип данных char

7 unsigned char 0

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

7 char 0

0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

↑
Знаковый разряд
(0 = +, 1 = -)

11111111 = 255

11111110 = 254

...

00000001 = 1

00000000 = 0

01111111 = 127

00000001 = 1

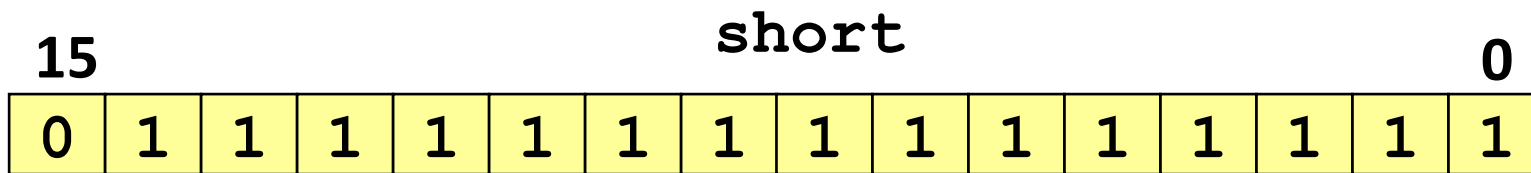
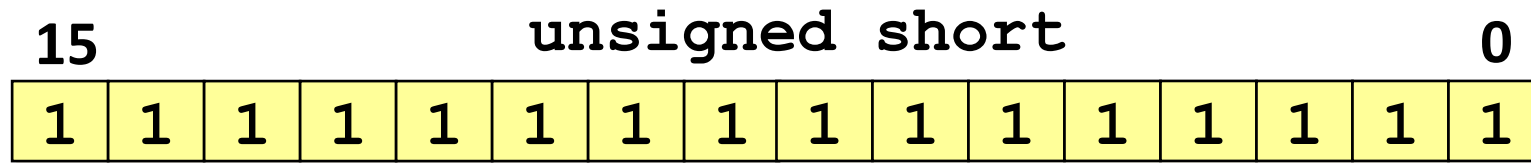
00000000 = 0

11111111 = -1

11111110 = -2

10000000 = -128

Тип данных short



↑
Знаковый разряд
(0 = +, 1 = -)

011111111111111111 = 32767
 000000000000000001 = 1
 000000000000000000 = 0
 111111111111111111 = -1
 100000000000000000 = -32768

111111111111111111 = 65535
 111111111111111110 = 65534
 000000000000000010 = 2
 000000000000000001 = 1
 000000000000000000 = 0

[illegible]

Сводная таблица

целочисленных типов данных

Тип	Размер	Диапазон
<code>bool</code>	1	<code>false</code> , <code>true</code>
<code>signed char</code>	1	-128 ... 127
<code>unsigned char</code>	1	0 ... 255
<code>signed short int</code>	2	-32768 ... 32767
<code>unsigned short int</code>	2	0 ... 65535
<code>signed long int</code>	4	-2^{31} ... $2^{31}-1$
<code>unsigned long int</code>	4	0 ... $2^{32}-1$

`short = short int`

`long = long int`

`signed` - по умолчанию

Целочисленные константы

Целочисленные константы могут записываться в десятичном, восьмеричном и шестнадцатеричном видах. Восьмеричные константы начинаются с **0**, шестнадцатеричные - с **0x** или **0X**.

Пример кода

```
int a = 15;  
int b = 015; //b = 13  
int c = 0x15; //b = 21  
int d = 0X15; //b = 21
```

Вещественные типы данных

Тип данных **float**

Переменным типа **float** могут быть присвоены вещественные значения в формате с плавающей точкой. В памяти **float** занимает 4 байта.

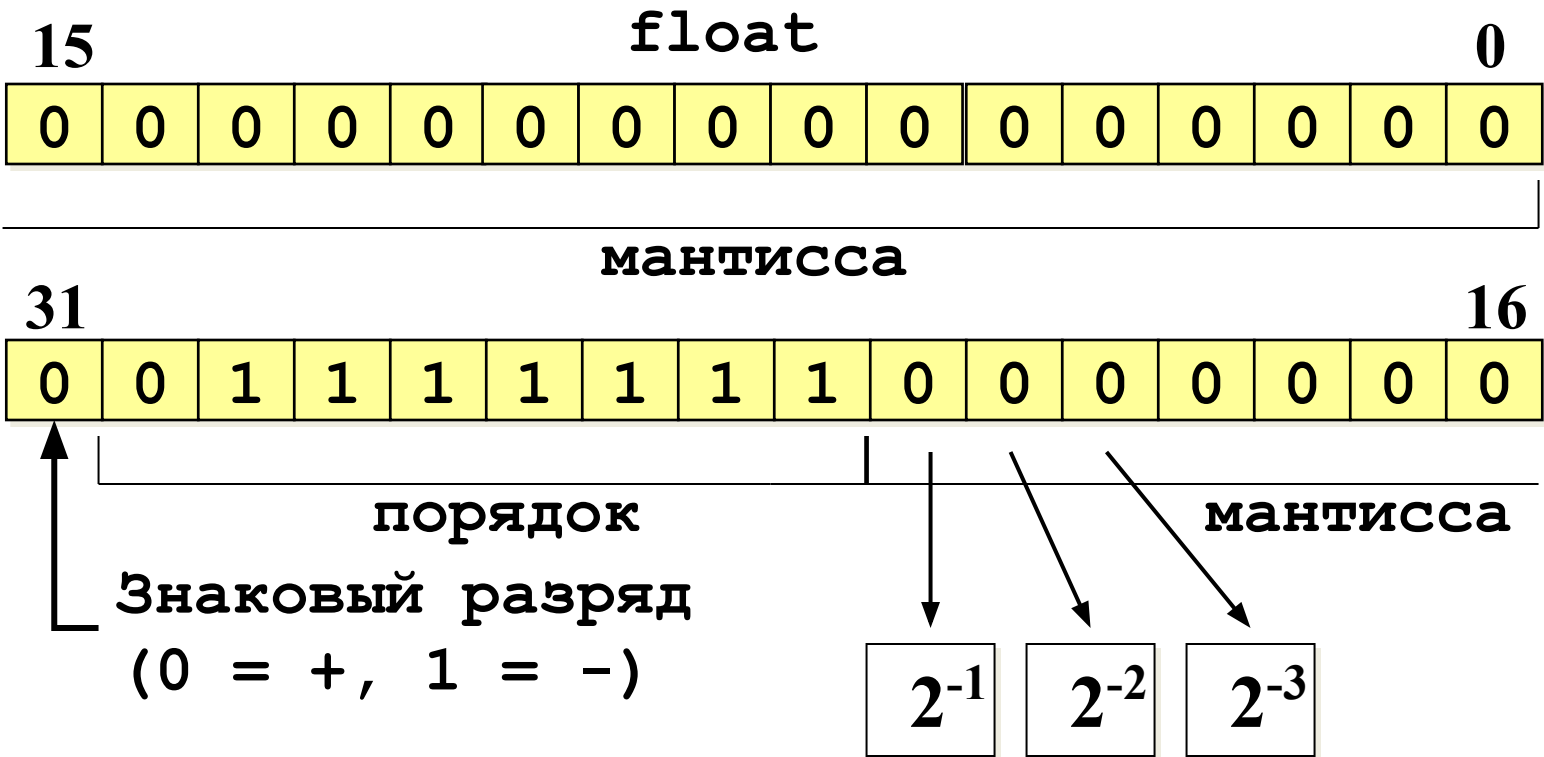
Тип данных **double**

Переменным типа **double** могут быть присвоены вещественные значения в формате с плавающей точкой. В памяти **double** занимает 8 байтов.

Тип данных **long double**

Переменным типа **long double** могут быть присвоены вещественные значения в формате с плавающей точкой. В памяти **long double** занимает 10 байтов.

Представление вещественных¹⁴ типов данных



$$\text{Число} = (1 + \text{мантисса}) * 2^{(\text{порядок} - 127)}$$

Сводная таблица вещественных типов данных

Тип	Размер	Диапазон	Точность
<code>float</code>	4	$3.4 \cdot 10^{-38} \dots 3.4 \cdot 10^{38}$	7
<code>double</code>	8	$1.7 \cdot 10^{-308} \dots 1.7 \cdot 10^{308}$	15
<code>long double</code>	10	$3.4 \cdot 10^{-4932} \dots 3.4 \cdot 10^{4932}$	19

Точность чисел с плавающей точкой

123000 - 3 значащих цифры

0.045 - 2 значащих цифры

123000.045 - 9 значащих цифр

Вещественные константы

Вещественные константы включают десятичную точку, либо могут использовать экспоненциальный формат (**e** или **E**). Тип константы по умолчанию – **double**, с помощью модификаторов **F** или **f** можно задать константу типа **float**, а с помощью **L** или **l** – типа **long double**.

Пример кода

```
float a = 5.67F;  
float b = .25f;  
float c = 5.67e4F;  
float d = 2E-2F;  
double e = 5.67;  
double f = .25;  
double g = 5.67e4;  
long double h = 2.1234E-2L;
```


СИМВОЛОВ

Тип данных char

Для представления символов используется тип данных **char**.
В памяти **char** занимает 1 байт.

'0' = 48	'A' = 65	'a' = 97
'1' = 49	'B' = 66	'b' = 98
...

Код ASCII

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

СИМВОЛЫ

Кодовая таблица Windows (CP-1251)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		kod-win1	©	€	§	€	•		°							
1		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2		!	"	#	\$	%	&		()	^	+				/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	□
8	Ђ	Ѓ	Ѕ	Ї	Љ	Њ	Ћ	Ќ		‰	Љ	«	Њ	Ћ	Ќ	Љ
9	ђ	ѓ	ѕ	ї	љ	њ	ћ	ќ		™	љ	»	њ	ќ	ћ	џ
A		у	ѳ	Ј	Ѳ	Ѓ	Ѕ	Ї	Љ	Њ	Ћ	«		-	®	Ї
B	°	±	І	і	Ѓ	μ		•	ё	№	€	»	ј	Ѕ	ѕ	ї
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

СИМВОЛЬНЫЕ КОНСТАНТЫ

Символьные константы состоят из одного символа, заключенного в апострофы. Тип константы – **char**. Для представления непечатаемых символов используется управляющие последовательности, начинающаяся с символа \ (обратный слэш).

Пример кода

```
char a = 'a' ;  
char b = '1' ;  
char c = '\t' ;    // табуляция  
char d = '\r' ;    // возврат каретки  
char e = '\n' ;    // перевод строки  
char f = '\062' ;  // символ с кодом 062  
char g = '\"' ;    // символ "  
char h = '\\ ' ;   // символ \
```

(литералы)

Строковые константы состоят из нескольких символов, заключенных в апострофы. Тип константы — **char***. Для представления непечатаемых символов используется комбинация, начинающаяся с символа \ (обратный слэш). В конце каждого строкового литерала компилятор добавляет нуль-терминатором '\0'. Поэтому длина строки всегда на 1 байт больше количества символов в ее записи.

Пример кода

```
char* a = "abcdef" ;  
char* b = "a" ;  
char* c = "" ;      // пустая строка  
char* d = "Вывод на терминал\r\n" ;  
char* e = "Оценка \"отлично\"";
```

Определение размера типов данных

21

Оператор sizeof()

Оператор **sizeof()** позволяет определить размер в байтах, занимаемый в памяти типом данных или переменной.

Пример кода

```
int i = sizeof(char) ;  
int j = sizeof(i)
```

Выбор типов данных

