


Лекция 3

Типы данных. Данные в языке Си

Тип данных

Тип данных – это основная характеристика данных, которая определяет

- формат представления данных (способ кодирования),
 - объем выделяемой под единицу данных памяти,
 - множество допустимых операций над данными,
 - множество отношений между данным типом и другими.
- 
- множество допустимых значений

Представление данных в памяти ЭВМ

Целые числа – дополнительный код

5 – 00000101
-5 – 11111011 } однобайтовое представление

знаковый
разряд

10000000	-128
01111111	127
...	
00000001	1
00000000	0
11111111	-1
...	
10000000	-128
01111111	127

При многобайтовом представлении обычно в память записывается сначала младший байт, затем средний и старший (обратный порядок следования).

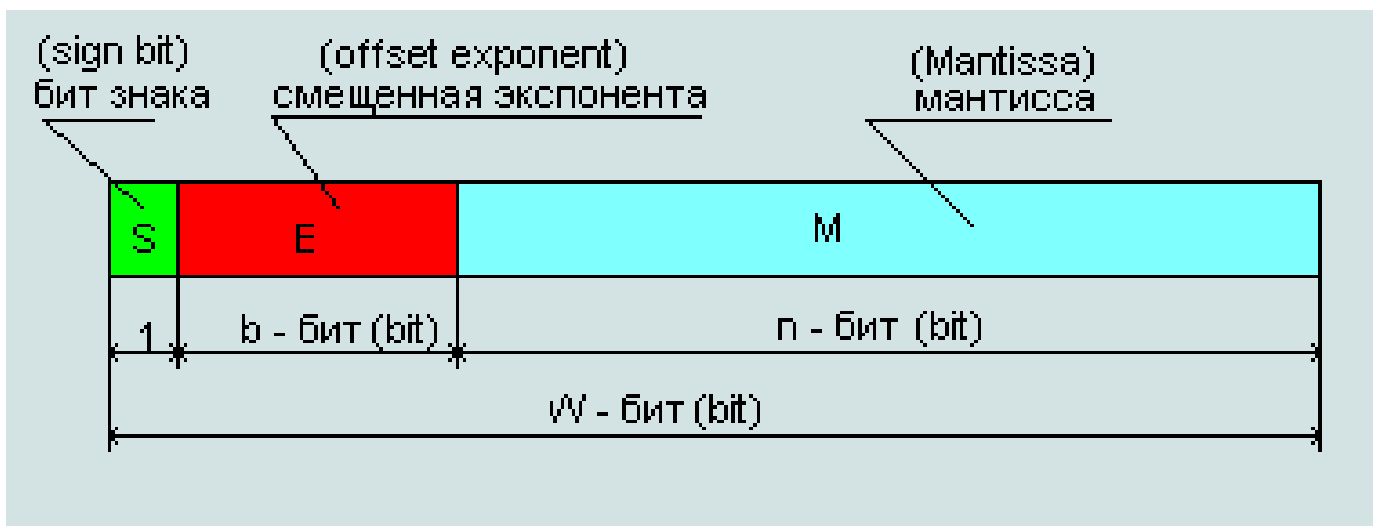
Действительные числа представляются в соответствии со стандартом **IEEE 754-2019** (вместо IEEE 754-2008 и IEEE 754-1985)

Стандарт IEEE 754-2008

Общий вид числа с плавающей точкой

$$m * f^p,$$

где m – мантисса, f – основание системы счисления, p – порядок.



binary32 – число одинарной точности, 4 байта,

E 8 бит, M 23 бита (7 значащих десятичных цифр)

binary64 – число двойной точности, 8 байт,

E 11 бит, M 52 бита (15 значащих десятичных цифр)

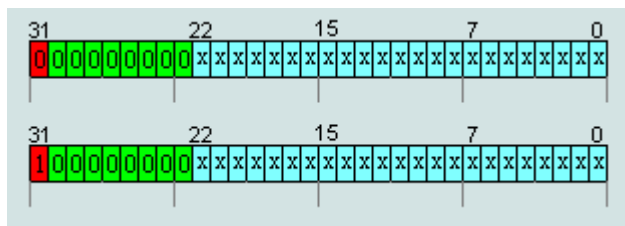
Стандарт IEEE 754-2008

- нормализованные числа с плавающей точкой

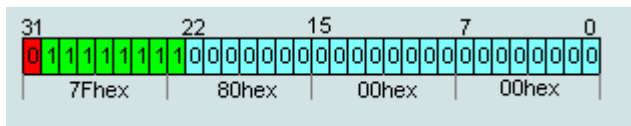
$$1 \leq m < f$$

- денормализованные числа с плавающей точкой

$$\frac{1}{f} \leq m < 1$$



- бесконечность (Infinity)

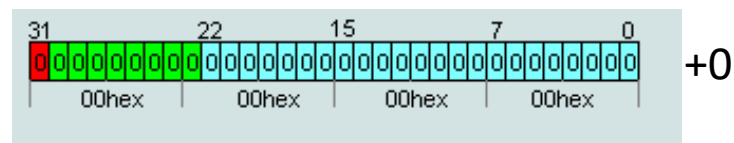


+∞

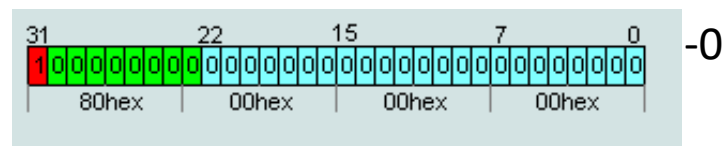


-∞

- нуль

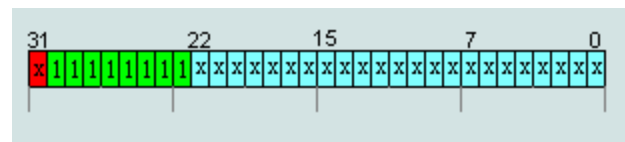


+0



-0

- "Не число" (NaN)



Базовые типы данных языка Си

char – целое число, 1 байт

int – целое число со знаком, машинное слово

double – вещественное число двойной точности,
8 байт

float – вещественное число одинарной точности,
4 байта



В языке Си все типы числовые.

void – отсутствие типа (пустой тип)

Слова *char*, *int*, *double*, *float*, *void* – **спецификаторы**
ТИПОВ.

Модифицированные типы

Модификаторы типов

- влияют на знак (**signed**, **unsigned**);
- влияют на объем памяти (**short**, **long**).

signed char – однобайтовое целое со знаком;

unsigned char – однобайтовое беззнаковое целое;

short int, short, signed short int, signed short – короткое целое со знаком, 2 байта;

short unsigned int, unsigned short – короткое целое без знака, 2 байта;

long int, long, signed long int, signed long – длинное целое со знаком, ≥ 4 байта;

long unsigned int, unsigned long – длинное целое без знака, ≥ 4 байта;

long long – очень длинное целое, 8 байт (C99);

long double – вещественное повышенной точности, 10 байт.

Производные типы данных языка Си

Производные типы формируются из уже существующих и описывают следующие конструкции:

- **указатели на объекты** заданного типа;
- **массивы** – последовательности объектов заданного типа;
- **функции**, возвращающие объекты заданного типа;
- **структуры** – последовательности объектов, возможно, различных заданных типов;
- **объединения**, каждое из которых может содержать любой из нескольких объектов различных заданных типов.

Данные в языке Си

Константа – неизменяемое данное, значение которого известно на этапе компиляции.

Константы бывают:

- явные;
- именованные.



Адрес константы получить нельзя.

Переменная – объект в оперативной памяти (область памяти), который имеет адрес, имя, тип и значение.



Если значение переменной в программе не изменяется, то является ли она константой?

Явные константы: целочисленные

Целочисленная константа – последовательность цифр одной из систем счисления:

- начинается с 0 – восьмеричная константа

034 01 0777 ~~089~~

- начинается с 0x или 0X – шестнадцатеричная

0x6A 0xFF 0x34 0X34 0XA98

- начинается не с 0 – десятичная константа

165u 34 10000L



Совпадают значения 034, 0x34 и 34?



По умолчанию все целочисленные константы типа *int*.

Явные константы: вещественные

Формы записи вещественных констант:

25.184

Целая и дробная
части отделяются
точкой

391.0 391.

Если целая или
дробная часть
отсутствует, ноль
можно не писать

0.15 .15

2e-5 6E23 2.14e+12

«Е» и «е» означают
«умножить на 10 в
степени»



Все вещественные
константы типа ***double***.

Символьные и строковые константы

Символьная константа – это последовательность из одного или нескольких символов, заключенная в одиночные кавычки (например `'x'`).



Значением символьной константы является число – код этого символа.

Для записи неотображаемых символов используются **escape-последовательности**. Примеры:

переход на новую строку `'\n'`

горизонтальная табуляция `'\t'`

возврат на шаг `'\b'`

обратная наклонная черта `'\\'`

Строковый литерал (строковая константа) – это последовательность символов, заключенная в двойные кавычки, например `"Hello!"`, `""` или `"2"`.

Что есть что?

5 5. '5' "5"

a 'a' "a"

-3 3U "3U"

0xE E 1E E1 1E1

Именованные константы

Создание именованных констант:

- директива препроцессора

```
#define N 100
```

- перечисления

```
enum bool {FALSE, TRUE};  
enum numbers {ONE=1, TWO, TEN=10};
```



После препроцессорной обработки все константы становятся явными.

Переменные

Переменная – объект в оперативной памяти (область памяти), который имеет адрес, имя, тип и значение.



В каждый момент времени переменная имеет только одно значение. При записи в переменную нового значения старое стирается.

Каждая переменная должна быть объявлена и определена.

Объявить переменную значит указать ее имя и тип.

Определить переменную значит выделить ей место в памяти.

Для большинства переменных объявление и определение совпадают.

Объявление переменных

C90: Переменные объявляются в начале блока.

Объявление переменной в общем виде:

класс квалификатор модификатор спецификатор идентификатор = инициализатор;
памяти типа типа

Обязательно!

```
int sum;  
short s;  
long int a=10, b;  
double x=3.2;  
static const char endl = '\n';
```

Классы памяти: *register*, *auto*, *static*, *extern*. По умолчанию **auto**.

Квалификаторы: *volatile*, *const*. По умолчанию **volatile**.



В неинициализированной переменной будет «мусор».