

# ЛЕКЦИЯ 6

---

Структура программы.  
Основные понятия языка.  
Система базовых типов.  
Операторы объявлений.

# Структура программы

Программа состоит из двух частей:

1. Раздел подключений библиотек
2. Раздел главной функции программы

# Библиотеки языка С

Все библиотеки языка С разделяются на две группы:

- системные библиотеки;
- пользовательские библиотеки.

# Библиотеки языка С

Системные библиотеки могут быть разделены на следующие две группы:

- стандартные библиотеки;
- нестандартные библиотеки.

Стандартные библиотеки:

`stdio.h` `stdlib.h` `math.h` `string.h`

Нестандартные библиотеки:

`conio.h` `io.h` `malloc.h` `memory.h`

# Библиотеки языка С

Подключение библиотек осуществляется с помощью директивы препроцессора:

```
#include <имя_библиотеки>
```

Например:

```
#include <stdio.h>
```

# Пример

```
#include<stdio.h>
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    printf(“Привет мир!\n”);
```

```
    return 0;
```

```
}
```

# Содержание функции `main`

Тело функции `main` состоит из операторов, которые записываются в императивной (процедурной) форме. Каждый оператор завершается знаком ';' и состоит из одной или нескольких лексем (англ., token).

Лексемы могут разделяться любым количеством пробелов, табуляций или переводов строк. Сама лексема должна писаться слитно. Лексемами могут быть:

- ключевые (зарезервированные) слова языка C,
- идентификаторы языка C;
- константы;
- строковые литералы;
- знаки пунктуации.

# Ключевые слова языка C

<b>auto</b>	<b>enum</b>	<b>restrict</b>	<b>unsigned</b>
<b>break</b>	<b>extern</b>	<b>return</b>	<b>void</b>
<b>case</b>	<b>float</b>	<b>short</b>	<b>volatile</b>
<b>char</b>	<b>for</b>	<b>signed</b>	<b>while</b>
<b>const</b>	<b>goto</b>	<b>sizeof</b>	<b>_Bool</b>
<b>continue</b>	<b>if</b>	<b>static</b>	<b>_Complex</b>
<b>default</b>	<b>inline</b>	<b>struct</b>	<b>_Imaginary</b>
<b>do</b>	<b>switch</b>	<b>switch</b>	
<b>double</b>	<b>long</b>	<b>typedef</b>	
<b>else</b>	<b>register</b>	<b>union</b>	
<b>__asm</b>	<b>__finally</b>	<b>__try</b>	<b>_forceinline</b>
<b>__cdecl</b>	<b>__forceinline</b>	<b>__asm</b>	<b>_inline</b>
<b>__declspec</b>	<b>__inline</b>	<b>__cdecl</b>	<b>_stdcall</b>
<b>__except</b>	<b>__leave</b>	<b>__declspec</b>	
<b>__fastcall</b>	<b>__stdcall</b>	<b>__fastcall</b>	



# Идентификаторы и константы

**Идентификатор** — это сочетание букв латинского алфавита, цифр и знака подчеркивания «\_», начинающееся или с буквы, или со знака подчеркивания. Длина идентификатора не ограничена. Заглавные и строчные буквы различаются. Идентификаторы используются для обозначения имен переменных, константных переменных, функций, типов.

**Константы** — это числа, символы или строки, которые могут использоваться как значения в программе, но не могут быть модифицированы в ней.

# Строковые литералы и знаки пунктуации

**Строковые литералы** – это последовательности символов заключенных в двойные кавычки, которые могут рассматриваться как строковые константы.

**Знаки пунктуации** – символы, используемые для разделения и группировки лексем, обозначения операций. В языке C используются следующие знаки пунктуации:

[ ] ( ) { } . ->  
++ -- & \* + - ~ !  
/ % << >> < > <= >= == != ^ | && ||  
? : ; ...  
= \*= /= %= += -= <<= >>= &= ^= |=  
, # ##  
<: :> <% %> %: %:%:

# Операторы языка C

Все операторы языка C можно разделить на четыре группы:

- операторы объявлений типов и переменных;
- операторы присвоений, математических и логических операций;
- операторы вызова функций;
- операторы управления.

# Система базовых типов

**Тип** – это описание диапазона значений, которые может принимать переменная, указанного типа.

Каждый тип данных характеризуется:

- размером (количество байт занимаемых значением данного типа в памяти компьютера)
- диапазоном (множеством) принимаемых значений.

# Система базовых типов

Все типы данных можно разделить на следующие виды:

- простые (скалярные) и сложные (векторные) типы;
- базовые (системные) и пользовательские типы.

# Система базовых типов

В языке C систему базовых типов образуют четыре типа данных:

- СИМВОЛЬНЫЙ,
- ЦЕЛОЧИСЛЕННЫЙ,
- ВЕЩЕСТВЕННЫЙ ОДИНАРНОЙ ТОЧНОСТИ,
- ВЕЩЕСТВЕННЫЙ ДВОЙНОЙ ТОЧНОСТИ.

# СИМВОЛЬНЫЙ ТИП

**char**

Размер: 1 байт (8 бит)

Диапазон значений: -128 ... 127

# Целочисленный тип

**int**

Размер: 4 байта (32 бита)

Диапазон значений:

-2147483648 ... 2147483647



# Вещественный тип одинарной ТОЧНОСТИ

## **float**

Размер: 4 байта (32 бита)

23 бита – мантисса;

8 бит – порядок;

1 бит – знак.

Диапазон значений:  $\pm 3.4E\pm 38$

Точность до 7 знаков после запятой

# Вещественный тип двойной точности

## **double**

Размер: 8 байт (64 бита)

52 бита – мантисса;

11 бит – порядок;

1 бит – знак.

Диапазон значений:  $\pm 1.7E\pm 308$

Точность до 17 знаков после запятой

# Система базовых типов

В языке C предусмотрены две группы модификаторов типа:

- модификаторы знака: **signed** и **unsigned**;
- модификаторы размера: **short** и **long**.

# Модификатор `signed`

Модификатор **`signed`** осуществляет преобразование целочисленных значений к знаковым числам. Используется совместно с типами **`char`** и **`int`**.

используется по умолчанию

# Модификатор `unsigned`

Модификатор `unsigned` осуществляет преобразование целочисленных значений к незнакомым числам. Используется совместно с типами `char` и `int`.

**`unsigned char`**

Диапазон: 0 ... 255

**`unsigned int`**

Диапазон: 0 ... 4294967295

# Модификатор `short`

Модификатор **`short`** уменьшает размер целочисленного типа в два раза. Используется только с типом **`int`**.

## **`short int`**

Размер: два байта

Диапазон: -32768 ... 32767

## **`unsigned short int`**

Размер: два байта

Диапазон: 0 ... 65535

# Модификатор `long`

Модификатор `long` увеличивает размер типа данных. Используется совместно с типами `int` и `double`.

`long double` – тоже, что и `double`

Но в стандарте IEEE:

80 бит:

1 бит – знак

64 бита – мантисса

15 бит - порядок

# Модификатор long

long int – то же, что и int

## **long long int**

Размер: 8 байт (64 бита)

Диапазон значений:  $-2^{63} \dots 2^{63}-1$

## **unsigned long long int**

Размер: 8 байт (64 бита)

Диапазон значений:  $0 \dots 2^{64}-1$



# Комплексные числа

Комплексные числа введены в стандарте C99.

**float** \_Complex

**double** \_Complex

**long double** \_Complex

Библиотека **complex.h**

**float** complex

**double** complex

**long double** complex

# Система базовых типов

Минимальные и максимальные значения всех базовых типов данных языка C описаны в библиотеках:

- **limits.h** – содержит диапазоны целочисленных значений,
- **float.h** – содержит диапазоны вещественных значений.

# Логический тип

Стандарт **C89**:

Логический тип – int

0 – ложь (false);

не 0 – истина (true).

Стандарт **C99**:

Логический тип - `_Bool`

Ключевые слова: `bool true false`

Библиотека **`stdbool.h`**

# Операторы объявлений

**Переменная** – именованная область памяти вычислительной машины, предназначенная для хранения значений определенного типа, с произвольным методом доступа: чтение и запись.

**Имя переменной** – разрешенный идентификатор языка С не использовавшийся ранее для обозначения других переменных, типов, элементов перечислений или имен функций.

# Операторы объявлений

Оператор объявления переменных имеет следующий синтаксис:

`<тип> <имя1>{',' <имя2>}* ':'`

Примеры:

```
int a, b, c;
```

```
double x, y;
```

```
char ch;
```

# Операторы объявлений

Правила хорошего стиля программирования предъявляют следующие требования к форматированию операторов объявлений переменных:

- объявление переменных нового типа всегда начинается с новой строки;
- имя переменной должно быть информативным, т.е. из имени переменной должно быть понятно ее назначение;
- имя переменной не должно быть слишком длинным;
- после объявления переменной в этой же строке или на предыдущей строке в комментарии желательно дать краткое описание назначения переменной;
- необходимо разделять имена переменных пробелами (в разумных пределах).

# Операторы объявлений

Оператор объявления переменных с инициализацией имеет следующий синтаксис:

```
<тип> <имя1> [= <значение1>] {', <имя2> [= <значение2>]}* ';' ;
```

Примеры:

```
int a=26, b=032, c=0x1A;
```

```
double x=2.5e2, y=0x1.ffe-3;
```

```
char ch='Z';
```

# Константы

В языке C присутствует три вида констант:

- целочисленные,
- вещественные,
- СИМВОЛЬНЫЕ.



# Целочисленные константы

- Десятичная константа указывается десятичным числом в обычной форме.
- Восьмеричная константа указывается числом, начинающимся с цифры ноль и содержащим цифры 0...7.
- Шестнадцатеричная константа указывается целым числом с префиксом 0x или 0X, содержащим цифры 0...9 и буквы латинского алфавита a...f, A...F.

# Суффиксы целочисленных констант

Суффикс	Десятичные константы	Восьмеричные и шестнадцатеричные константы
нет	int long int long long int	int unsigned int long int unsigned long int long long int unsigned long long int
<b>U</b> или <b>u</b>	unsigned int unsigned long int unsigned long long int	unsigned int unsigned long int unsigned long long int
<b>L</b> или <b>l</b>	long int long long int	long int unsigned long int long long int unsigned long long int
Оба <b>u</b> или <b>U</b> и <b>l</b> или <b>L</b>	unsigned long int unsigned long long int	unsigned long int unsigned long long int
<b>LL</b> или <b>ll</b>	long long int	long long int unsigned long long int
Оба <b>u</b> или <b>U</b> и <b>ll</b> или <b>LL</b>	unsigned long long int	unsigned long long int

# Вещественные константы

Вещественные константы записываются в десятичной или шестнадцатеричной системе исчисления. Позиция запятой указывается точкой, экспонента указывается после латинской буквы `e` (или `E`).

Суффиксы вещественных констант:

- **f** или **F** – тип **float**,
- **l** или **L** – тип **long double**.

# Символьные константы

Символьные константы записываются в одинарных кавычках. В языке C присутствуют специальные символы:

- `'\'` – одинарная кавычка,
- `'\"'` – двойная кавычка,
- `'\\'` – обратный слеш,
- `'\?'` – знак вопроса,
- `'\a'` – звуковой сигнал,
- `'\b'` – забой символа,
- `'\f'` – прокрутка страницы,
- `'\n'` – перевод строки,
- `'\r'` – возврат каретки,
- `'\t'` – горизонтальная табуляция,
- `'\v'` – вертикальная табуляция.

# Константные переменные

В языке C можно также создавать переменные, имеющие константное значение (их значение нельзя изменить). Объявление таких «переменных» имеет следующий синтаксис:

```
'const' <тип> <имя1>'='<значение1>[',' <имя2> '=' <значение>]*;
```

Примеры

```
const unsigned int x=80, y=25;
```

```
const double pi=3.1415;
```

# Оператор создания типов

К операторам объявлений следует также отнести и оператор создания пользовательских типов данных.

Синтаксис оператора имеет следующий вид:

```
'typedef' <имя_старого_типа> <имя_нового_типа>;
```

Пример:

```
typedef unsigned int word;
```

# Операторы объявлений

Согласно стандарту **C99** операторы объявления переменных могут располагаться в любом месте программы, если это не противоречит синтаксису языка C.

Оператор объявления пользовательских типов также может располагаться в любом месте программы, но рекомендуется располагать его сразу после подключения библиотек (до описания функции **main**).

# Операторы объявлений

В языке C присутствует оператор определения размера значения определенного типа **sizeof**, который возвращает значение целого типа равное числу байт занимаемых значением в памяти.

Пример:

```
sizeof(int)      //4  
sizeof(char)    //1  
sizeof(double) //8
```