

Программирование на языке С

Модуль 1. **ВВЕДЕНИЕ В ЯЗЫК С**

- Лексемы и пробельные символы
- Основные типы данных
- Диапазоны представляемых значений
- Декларация переменных
- Литералы



Немного истории (начало)

- **1969–1972** — на базе языка B («би») — упрощенного варианта BCPL (Basic Combined Programming Language — Мартин Ричардс, Кембриджский университет, 1966) — начинается проектирование нового языка программирования
- **1972** — сотрудник Bell Telephone Laboratories Деннис Ритчи создает язык C («си») для разработки ОС UNIX на платформе PDP-7
- **1973** — на язык C перенесен значительный фрагмент ядра Unix для PDP-11, ранее разработанный на языке ассемблера
- **1978** — в США выходит книга Брайана Кернигана и Денниса Ритчи с описанием языка C, надолго ставшая неформальным стандартом для программистов (K&R C)

Немного истории (продолжение)

- **1985** — книга Б. Кернигана и Д. Ритчи переведена на русский язык
Керниган Б., Ритчи Д., Фьюер А. Язык программирования Си. Задачи по языку Си / Пер. с англ. — М.: Финансы и статистика, 1985. — 279 с.
- **1988** — выходит в свет 2-е издание книги Б. Кернигана и Д. Ритчи (первое описание будущего стандарта ANSI C)
Kernighan, Brian W.; Ritchie, Dennis M. The C Programming Language (2nd ed.). Englewood Cliffs, NJ: Prentice Hall (1988)
- **1989** — созданный в 1983 г. комитет Американского института стандартов (ANSI) ратифицирует стандарт X3.159-1989 “Programming Language C” (ANSI C, Standard C, C89)
- **1990** — Международная организация по стандартизации (ISO) принимает стандарт ANSI C как стандарт ISO/IEC 9899:1990 (C90)

Немного истории (окончание)

- **1992** — 2-е издание книги Б. Кернигана и Д. Ритчи выходит в России
Керниган Б., Ритчи Д. Язык программирования Си / Пер. с англ.
— М.: Финансы и статистика, 1992. — 272 с.
- **1999** — опубликован стандарт ISO/IEC 9899:1999 (C99)
к числу стандартных возможностей языка добавлены встроенные (inline) функции, новые типы данных (long long int, complex и др.), массивы переменной длины и однострочные комментарии
- **2011** — опубликован стандарт ISO/IEC 9899:2011 (C11)

Алфавит и лексемы языка C

- **Алфавит**

- буквы: A, B, C, ..., Z, a, b, c, ..., z
- цифры: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- специальные символы: + - / % . ? ! " < > | \ ' _ & ~ ^
- знаки пунктуации языка: [] () { } , ; : ... * = #
- пробельные символы: _ (пробел), \t (символ табуляции), \n (символ перевода строки)
- прочие символы — только в комментариях к тексту программы

- **Лексемы** — идентификаторы, ключевые слова, константы, операции, разделители

- единицы текста программы, которые при компиляции воспринимаются как единое целое и по смыслу не могут быть разделены на более мелкие элементы [Под04]

Лексемы

Лексема - это единица текста программы, которая имеет **определенный смысл для компилятора** и которая **не может быть разбита в дальнейшем**. Компилятор разбивает текст программы на лексемы и ставит им в соответствие последовательности машинных команд.

1	Знаки пунктуации и специальные символы	<u>Используются для различных целей</u> , от организации текста программы до определения заданий, которые будут выполнены компилятором или программой.
2	Знаки операций	Один или более специальных символов, определяющих действие, которое должно быть выполнено над операндами. <u>Используются для формирования выражений.</u>
3	Ключевые слова	<u>Предопределенные идентификаторы</u> , которые <u>имеют специальное значение для компилятора</u> . Их можно использовать только так как они определены.
4	Константы	<u>Неизменяемые величины в программе.</u>
5	Идентификаторы	<u>Условные обозначения объектов программ.</u>
6	Комментарии	<u>Пояснения к тексту программы</u> . Каждый из них воспринимается <u>компилятором</u> как отдельный пробельный символ и <u>игнорируется</u> .

Границы лексем определяются:

- 1) пробельными символами
- 2) другими лексемами

Идентификаторы и ключевые слова

- **Идентификатор** — любая последовательность букв A, B, C, ..., Z, a, b, c, ..., z, цифр 0, 1, ..., 9 и символов подчеркивания `_`, не начинающаяся с цифры и имеющая длину не более 31 символа. Строчные и прописные буквы в идентификаторах различаются компиляторами
- **Ключевое слово** — одно из слов языка, входящих в следующий список:
 - спецификаторы типов: `char`, `double`, `enum`, `float`, `int`, `long`, `short`, `struct`, `signed`, `union`, `unsigned`, `void`, `typedef`
 - квалификаторы типов: `const`, `volatile`
 - квалификаторы классов памяти: `auto`, `extern`, `register`, `static`
 - операторы языка и идентификаторы специального назначения: `break`, `continue`, `do`, `for`, `goto`, `if`, `return`, `switch`, `while`; `default`, `case`, `else`, `sizeof`
 - модификаторы и псевдопеременные: конкретный набор зависит от компилятора

Константные значения (начало)

- **Константа** — неизменяемое арифметическое значение целого, вещественного, символьного или перечислимого типа, нулевой указатель либо строковый литерал:
 - целые — записываются в системах счисления по основаниям 10, 8, 16:
 - (целочисленный) ноль в любой системе счисления — 0
 - десятичные — последовательность десятичных цифр, не начинающаяся с нуля
 - восьмеричные — последовательность восьмеричных цифр, начинающаяся с нуля
 - шестнадцатеричные — последовательность шестнадцатеричных цифр, начинающаяся с 0x или 0X
 - вещественные — записываются в десятичной системе в следующих форматах:
 - $[+|-]<\text{целая часть}>.[<\text{дробная часть}>]$
 - $[+|-]<\text{целая часть}>\{e|E\}[+|-]<\text{порядок}>$
 - $.[<\text{дробная часть}>][\{e|E\}[+|-]<\text{порядок}>]$где $<\text{целая часть}>$ есть целая часть абсолютной величины десятичной мантииссы, $<\text{дробная часть}>$ — дробная часть абсолютной величины десятичной мантииссы, $<\text{порядок}>$ — абсолютная величина десятичного порядка (экспоненциальной части числа)

Константные значения (окончание)

- символные — записываются естественным образом* или посредством ESC-последовательностей**, *** согласно следующим правилам:
 - * символы, имеющие экранное представление — любой входящий или не входящий в алфавит языка единичный символ в обрамлении апострофов (');
 - ** ряд символов, лишенных экранного представления — одна из следующих управляющих последовательностей: '\n' — перевод строки; '\t' — горизонтальная табуляция; '\r' — возврат каретки; '\\ ' — обратная косая черта; '\ ' — апостроф; '\" ' — двойная кавычка; '\0' — нулевой символ; '\a' — звонок; '\b' — возврат на одну позицию; '\f' — перевод страницы; '\v' — вертикальная табуляция; '\?' — знак вопроса;
 - *** любой символ — собственный восьмеричный код в виде '\ooo', где o — цифра от 0 до 7, либо шестнадцатеричный код в виде '\xhh' или '\Xhh', где h — цифра от 0 до F;
- перечислимые — задаются в определении программистом собственного типа-перечисления;
- нулевой указатель — единственная неарифметическая константа, представимая различными компиляторами как 0, 0L или NULL (значение NULL может не совпадать с нулем (0) и (или) нулевым символом ('\0'));
- строковый литерал — заключенная в двойные кавычки (") последовательность символов, записанных по правилам для символьных констант*, **, *** без обрамляющих апострофов

Знаки и приоритет операций (начало)

Приоритет операций	Знаки операций	Порядок выполнения операций с равным приоритетом
1	() [] -> .	слева направо
2	! ~ + - ++ -- & * (<имя типа>) sizeof	справа налево
3	* / %	слева направо
4	+ -	слева направо
5	<< >>	слева направо
6	< <= >= >	слева направо
7	== !=	слева направо
8	&	слева направо
9	^	слева направо

Знаки и приоритет операций (окончание)

Приорите т операций	Знаки операций	Порядок выполнения операций с равным приоритетом
10		слева направо
11	&&	слева направо
12		слева направо
13	?:	справа налево
14	= *= /= %= += -= = &= ^= = <<= >>=	справа налево
15	,	слева направо

Комментарии - пояснения к тексту программы.

Например,

*/*Это комментарий языка C*/*

*/**

Может

быть

многострочным /, но не может быть вложенным*/*

**/*

//Это однострочный комментарий языка C++ и стандарта C99



Разделители

- **Разделитель** — парный или одиночный знак пунктуации, входящий в следующий список:

[] () { } , ; : ... * = #

Переменные и константы

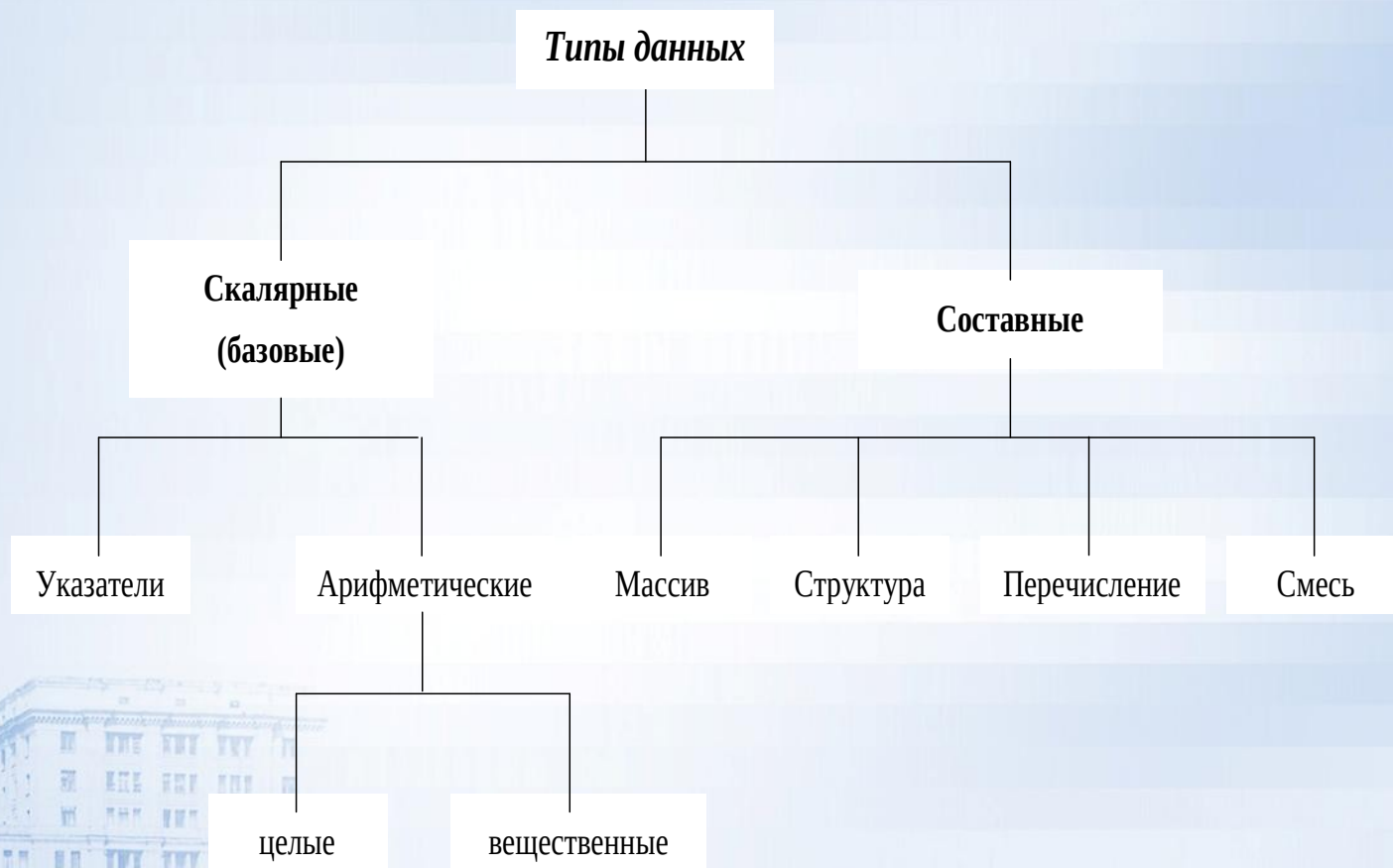
- **Описание переменных**

<имя типа> <переменная 1>[[= <значение 1>], ..., <переменная N> [= <значение N>]];

- **Описание констант**

const [<имя типа>] <имя константы> = <значение константы>;

- при опускании типа константы подразумевается int



Основные типы данных

Имя типа	Размер области памяти (бит)	Диапазон значений (для вещественных типов — по абсолютной величине)
unsigned char	8	0 ... 255, '\x00' ... '\xFF'
[signed] char	8	-128 ... 127
unsigned short [int]	16	0 ... 65535
[signed] short [int]	16	-32768 ... 32767
enum	32	-2147483648 ... 2147483647
unsigned [int]	32	0 ... 4294967295
[signed] int	32	-2147483648 ... 2147483647
unsigned long	32	0 ... 4294967295
[signed] long	32	-2147483648 ... 2147483647
float	32	3.4E-38 ... 3.4E38

Новые типы данных (C11)

- **long long int** **64 бит** см. **limits.h**
- **long double** **128 бит**
 __mingw_printf("%Lg", имя переменной)
- **stdbool.h** => **bool**
- **tgmath.h** => **complex**
 double complex **z1=2-5I**
 creal(z1)
 cimag(z1)

Список литературы

- [КР92] Керниган Б., Ритчи Д. Язык программирования Си / Пер. с англ. — М.: Финансы и статистика, 1992. — 272 с.
- [КР06] Керниган Б., Ритчи Д. Язык программирования C / Пер. с англ. — М.: Вильямс, 2006. — 304 с.
- [Под04] Подбельский В.В., Фомин С.С. Программирование на языке Си. – 2-е доп. изд. – М., Финансы и статистика, 2004. – 600 с.