

Лекция 3. Элементы языка Си. Основные понятия языка Си. Основные понятия языка Си. Структура программы в языке Си

3.1. Элементы языка Си

Множество символов, используемых в Си можно разделить следующим образом [3-5]:

1. символы, используемые для образования ключевых слов (зарегистрированные операторы и команды) и идентификаторов (переменные). Входят прописные и строчные буквы латинского алфавита, символ подчеркивания;
2. группа прописных и строчных букв русского алфавита и арабские цифры (1,2,3...);
3. знаки нумерации и специальные символы. Используются для организации вычислений, а также для передачи компилятору инструкций;
4. управляющие и разделительные символы: пробел, табуляция, перевод строки, возврат каретки и т.д. Используется для разделения объектов;
5. Управляющие последовательности. Чаще используются в функциях ввода/вывода, например, `\n` – переход на новую строку, `\t` – горизонтальная табуляция и др.

3.2. Основные понятия языка Си

Программа, написанная на языке Си, состоит из операторов. Каждый оператор вызывает выполнение некоторых действий на соответствующем шаге выполнения программы.

При написании операторов применяются латинские прописные и строчные буквы, цифры и специальные знаки. К таким знакам, например, относятся: точка (.), запятая (,), двоеточие (:), точка с запятой (;) и др. Совокупность символов, используемых в языке, называется **алфавитом языка**. В персональном компьютере символы хранятся в виде кодов. Соответствие между каждым символом и его кодом задается специальной кодовой таблицей. На нее разработан стандарт ASCII, поэтому коды символов называют ASCII-кодами.

Различают видимые и управляющие символы. Первые могут быть отображены на экране дисплея. Вторые вызывают определенные действия в машине, например, звуковой сигнал - код 710, возврат курсора на один шаг - код 810, горизонтальная табуляция - код 910, перевод курсора на новую строку - код 1010, перемещение курсора в начало строки - код 1310 и т.д. Такие управляющие символы имеют десятичные номера 0 - 31, 127.

Для представления каждого символа в персональном компьютере используется один байт, поэтому общее число символов равно $2^8 = 256$. В таблице 3.1 представлены ASCII коды основных символов (**DEC**- десятичная

система счисления, **HEX** – шестнадцатеричная система счисления, **BIN** – двоичная система счисления, **CHAR** – символьный вид) [3-5].

Таблица 3.1.

Таблица ASCII кодов

Dec	Hex	Char	Cmd	Dec	Hex	Char	Cmd	Dec	Hex	Char	Cmd	Dec	Hex	Char	Cmd
0	00		NUL	32	20		(sp)	64	40	@		96	60	`	
1	01	☺	SOH	33	21	!		65	41	A		97	61	a	
2	02	☼	STX	34	22	"		66	42	B		98	62	b	
3	03	♥	ETX	35	23	#		67	43	C		99	63	c	
4	04	♦	EOT	36	24	\$		68	44	D		100	64	d	
5	05	♣	ENQ	37	25	%		69	45	E		101	65	e	
6	06	♠	ACK	38	26	&		70	46	F		102	66	f	
7	07	•	BEL	39	27	'		71	47	G		103	67	g	
8	08	▣	BS	40	28	(72	48	H		104	68	h	
9	09	○	TAB	41	29)		73	49	I		105	69	i	
10	0A	▣	LF	42	2A	*		74	4A	J		106	6A	j	
11	0B	♂	VT	43	2B	+		75	4B	K		107	6B	k	
12	0C	♀	FF	44	2C	,		76	4C	L		108	6C	l	
13	0D	♪	CR	45	2D	-		77	4D	M		109	6D	m	
14	0E	♫	SO	46	2E	.		78	4E	N		110	6E	n	
15	0F	☼	SI	47	2F	/		79	4F	O		111	6F	o	
16	10	►	DLE	48	30	0		80	50	P		112	70	p	
17	11	◄	DC1	49	31	1		81	51	Q		113	71	q	
18	12	↕	DC2	50	32	2		82	52	R		114	72	r	
19	13	!!	DC3	51	33	3		83	53	S		115	73	s	
20	14	¶	DC4	52	34	4		84	54	T		116	74	t	
21	15	§	NAK	53	35	5		85	55	U		117	75	u	
22	16	—	SYN	54	36	6		86	56	V		118	76	v	
23	17	↕	ETB	55	37	7		87	57	W		119	77	w	
24	18	↑	CAN	56	38	8		88	58	X		120	78	x	
25	19	↓	EM	57	39	9		89	59	Y		121	79	y	
26	1A	→	SUB	58	3A	:		90	5A	Z		122	7A	z	
27	1B	←	ESC	59	3B	;		91	5B	[123	7B	{	
28	1C	└	FS	60	3C	<		92	5C	\		124	7C		
29	1D	↔	GS	61	3D	=		93	5D]		125	7D	}	
30	1E	▲	RS	62	3E	>		94	5E	^		126	7E	~	
31	1F	▼	US	63	3F	?		95	5F	_		127	7F	␣	DEL

Важным понятием языка является **идентификатор**, который используется в качестве имени объекта (функции, переменной, константы и др.). Идентификаторы должны выбираться с учетом следующих правил:

- идентификаторы должны начинаться с буквы латинского алфавита (*a,...,z*, *A,...,Z*) или с символа подчеркивания (*_*);
- в идентификаторах могут использоваться буквы латинского алфавита, символ подчеркивания и цифры (от 0 до 9). Использование других символов в идентификаторах запрещено;
- в языке Си буквы нижнего регистра (*a,...,z*), применяемые в идентификаторах, отличаются от букв верхнего регистра (*A,...,Z*). Это означает, что следующие идентификаторы считаются различными: **function**, **FUNCTION**, **Function** и т.д;
- идентификаторы объектов не должны совпадать с ключевыми словами языка и именами стандартных функций из применяемых библиотек.

Пробелы, символы табуляции и перехода на новую строку в программах на языке Си игнорируются. Это позволяет записывать различные выражения в удобном для прочтения и понимания виде. Кроме того, строки программы

можно начинать с любой позиции, что дает возможность выделять в тексте группы операторов.

3.3. Структура программы в языке Си

Программа на языке Си всегда состоит из функций и переменных. Функции содержат набор инструкций, описывающих вычисления, которые необходимо выполнить. А переменные хранят значения, которые используются в процессе вычислений [3-5].

Обычно разработчик может определять любые имена для своих функций. Однако в языке Си есть особая функция с именем "**main**". Любая программа начинается с первой инструкции функции **main**. Т.е. данная функция должна присутствовать в любой программе. Не допускается вызов функции **main** из других функций программы – она является управляющей.

Следующие за именем функции круглые скобки предназначены для указания параметров (или аргументов), которые передаются в функцию при обращении к ней, т.е. которые необходимы для выполнения инструкций внутри функции. В случае записи **main()** в функцию не передается никаких аргументов, поэтому список аргументов в круглых скобках пустой. Функцию **main()** можно записать двумя способами:

int main()
void main()

Перед именем функции указывается тип возвращаемого значения. При обращении к главной функции значение возвращается операционной системе. Последняя запись (**void main()**) не будет возвращать никакого значения. В свою очередь первая запись **int main()** сообщает компилятору о том, что после выполнения функции будет возвращено целочисленное значение, которое необходимо операционной системе и сообщает ей о том, что программа завершилась корректно. Если же это значение не возвращено, то операционная система понимает, что программа завершилась в аварийном режиме.

Для возврата целочисленного значения перед завершением функции добавляется строка:

return 0;

В зависимости от программы, достаточно часто данная строка не требуется.

Язык Си является блочно-структурированным. Каждый блок заключается в фигурные скобки {}.

Каждое действие в языке Си заканчивается символом "точка с запятой" - ;. В качестве действия может выступать вызов функции или осуществление некоторых операций.

В Си для написания комментариев используются следующие символы: /* - начало комментария, */ - конец комментария. При этом все что будет заключено между этими символами будет являться комментариями, т.е. не будет восприниматься компилятором как инструкция или набор инструкций. Такое ограничение области комментариев удобно использовать при

комментировании блока программы, состоящего из нескольких строк. При необходимости комментария одной строки удобно использовать символ //, при этом комментарием будет являться все, что расположено между символом // и концом строки (т.е. все, что расположено справа от данного символа). Примеры вариантов комментариев представлены в табл. 3.2 [3-5].

Таблица 3.2.

Примеры написания комментариев в Си

Обозначение комментария двух строк.	int a; /* целая переменная */
Обозначение комментария блока программы. При этом в ходе выполнения программы будут выполнены операции $b=a$ и $d=a-b$. А то что заключено между символами /* и */ выполняться не будет.	int a,b,c,d; float e; void main(void) { b=a; /*c=d e=a+b;*/ d=a-b; }
Комментирование одной строки. При этом, то что находится правее символа // будет являться комментарием. Т.е. в данной программе фраза «определение переменных» является комментарием, а также в ходе выполнения программы операция $e=a+b$ выполняться не будет, так как она закомментирована.	int a,b,c; //определение переменных float e; void main(void) { b=a; //e=a+b; }

Обычно **main** для выполнения своей работы пользуется услугами других функций; одни из них пишутся самим программистом, а другие берутся готовыми из имеющихся библиотек. Например, строка программы:

#include <stdio.h>

сообщает компилятору, что он должен включить стандартную библиотеку ввода-вывода. Эта строка встречается в начале многих файлов программ на языке Си. Существует большое количество различных библиотек, которые выполняют различные функции. В таблице 3.3 представлены примеры стандартных библиотек языка Си.

Таблица 3.3.

Примеры стандартных библиотек языка Си

<assert.h>	Макрос assert()
<ctype.h>	Обработка символов

<errno.h>	Сообщения об ошибках
<float.h>	Значения с плавающей точкой, зависящие от конкретной реализации компилятора
<iso646.h>	Макросы, соответствующие различным операторам, например && и ^. Добавлено в 1995 году (Поправка 1)
<limits.h>	Различные ограничения, зависящие от конкретной реализации компилятора
<locale.h>	Функция setlocale()
<math.h>	Различные определения, используемые библиотекой math
<setjmp.h>	Нелокальные переходы
<signal.h>	Значения сигналов
<stdarg.h>	Списки аргументов переменной длины
<stddef.h>	Распространенные константы
<stdio.h>	Ввод-вывод файлов
<stdlib.h>	Смешанные объявления
<string.h>	Функции обработки строк
<time.h>	Функции системного времени и даты

Описание и функции данных библиотек легко найти как в различных литературных источниках.