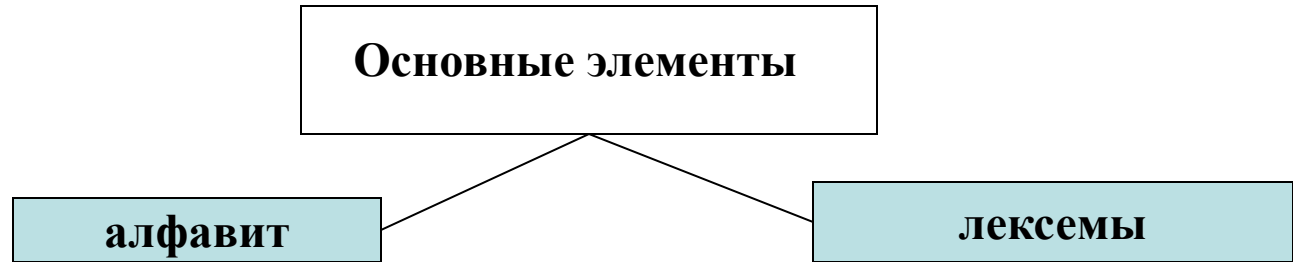


Тема 3 Элементы языка Си

Элементы языка СИ – базовые конструкции, используемые при написании Си-программы.



1. Алфавит - множество символов используемых в языке СИ (5 групп)

- 1 группа - прописные и строчные буквы латинского алфавита и символ подчеркивания

A...Z a...z _

- 2 группа – арабские цифры **0...9**
- 3 группа – знаки нумерации и специальные символы

Символ	Наименование	Символ	Наименование
,	запятая)	круглая скобка правая
.	точка	(круглая скобка левая
;	точка с запятой	}	фигурная скобка правая
:	двоеточие	{	фигурная скобка левая
?	вопросительный знак	<	меньше
'	апостроф	>	больше
!	восклицательный знак	[квадратная скобка
	вертикальная черта]	квадратная скобка
/	дробная черта(прямой слеш)	#	номер
\	обратная черта(обратный слеш)	%	процент
~	тильда	&	амперсанд
*	звездочка	^	логическое не
+	плюс	=	равно
-	минус	"	кавычки

•4 группа - **управляющие и разделительные символы.**

- пробел,
- символы табуляции,
- перевод строки,
- возврат каретки,
- новая страница и
- новая строка.

•5 группа - **управляющие последовательности**, т.е.

специальные символьные комбинации, используемые в функциях ввода и вывода информации.

Управляющая последовательность строится на основе использования

- **обратной дробной черты (\) (обязательный первый символ) и**
- **комбинацией латинских букв и цифр**

Управляющая последовательность	Наименование	Шестнадцатеричная замена
<code>\a</code>	Звонок	007
<code>\b</code>	Возврат на шаг	008
<code>\t</code>	Горизонтальная табуляция	009
<code>\n</code>	Переход на новую строку	00A
<code>\v</code>	Вертикальная табуляция	00B
<code>\r</code>	Возврат каретки	00C
<code>\f</code>	Перевод формата	00D
<code>\"</code>	Кавычки	022
<code>\'</code>	Апостроф	027
<code>\0</code>	Ноль-символ	000
<code>\\</code>	Обратная дробная черта	05C
<code>\ddd</code>	Символ набора кодов ЭВМ в восьмеричном представлении	
<code>\xdd</code>	Символ набора кодов ЭВМ в шестнадцатеричном представлении	

2. Лексемы – единицы текста программы, которые имеют определенный смысл для компилятора и которые не могут быть разбиты в дальнейшем.

Есть несколько классов лексем:

1. Знаки пунктуации (, . ; : все скобки ‘ “)
2. Идентификаторы (ИМЕНА),
3. комментарии
4. константы (целые, с плавающей точкой, символ, строки)
5. знаки операций,
6. ключевые слова
7. операторы

2. Идентификаторы (ИМЕНА) — это последовательность латинских букв, цифр и символа `_`, начинающаяся с буквы или символа `_`.

Большие и маленькие латинские буквы считаются различными!

Например, `y` и `Y` — это разные имена.

Имена присваиваются

- Переменным
- Составным типам
- Меткам
- Функциям

Рекомендуется в именах переменных использовать только строчные буквы.

Примеры правильных идентификаторов:

- `s` `c` `h` `e` `get_line` `al2` `Param1_ab`

Примеры неправильных идентификаторов:

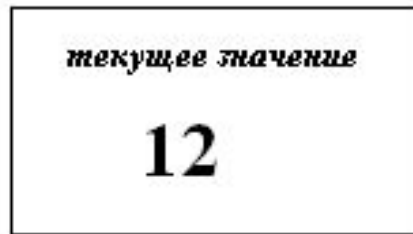
- `%ab` `12abc` `-x` `вася`

• ПЕРЕМЕННЫЕ

- Переменная - это такой "ящичек" с именем, в котором может храниться некое ЗНАЧЕНИЕ. **Имя у переменной постоянно и неизменно, значение же может меняться.**
- Например, переменная с именем "X".

переменная

X



Переменную можно изменять при помощи **операции ПРИСВАИВАНИЯ.**

В языке Си она обозначается знаком равенства.

x = 12 ;

оператор

x = x + 3;

Это не уравнение.

- 1) "взять значение переменной ИКС"
- 2) "прибавить к нему 3"
- 3) "положить новое значение в переменную ИКС", стерев в ней прежнее значение.

3. Комментарии

Символы `/*` задают начало комментария, заканчивающегося символами `*/`.

Комментарии не могут быть вложенными.

Символы `//` начинают комментарий, который заканчивается в конце строки, на которой они появились.

Примеры

```
/* Данная программа на C++ выводит сообщение “Hello  
World”*/
```

```
return 0; //необязательная строка
```


4. Константы

- целые
- с плавающей точкой
- СИМВОЛ
- строки

4.1. Целые константы

Целая константа - последовательность арабских цифр 01234 ... 9

Целая константа восьмеричная - 0 (цифра ноль) последовательность арабских цифр

Целая константа шестнадцатеричная -

0x (цифра ноль и буква x) последовательность арабских цифр

0X (цифра ноль и буква X) последовательность арабских цифр и букв от a или A до f или F, имеющие значения от 10 до 15.

Например, 1234 01234 0x1234

4.3. Символьная константа - символ заключенный в апострофы

Например, 'x' 'ф' 'Г' '3'

Значением символьной константы является численное значение символа в машинном наборе символов (алфавите).

Символьные константы считаются данными типа `int`.

Но в программе `char c = 'x';`

Неграфические символы

- '\n' - символ новой строки ,
- '\\' - обратная дробная черта ,
- '\v' - вертикальная табуляция
- '\ ' - апостроф и т.д.

4.4. Строка = строковая константа = литерал -
последовательность символов (включая строковые и
прописные буквы русского и латинского а также
цифры) заключенные в кавычки (").

Например:

"Школа N 35", "город Тамбов", "YZPT КОД "

"Школа \n N 35"

Символы строкового литерала сохраняются в области оперативной
памяти.

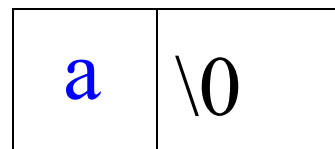
В конец каждого строкового литерала компилятором добавляется нулевой
символ, представляемый управляющей последовательностью \0.

Строковый литерал имеет тип char[]
char s[]="язык";

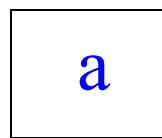
я	з	ы	к	\0
---	---	---	---	----

РАЗЛИЧИЕ

`char letter[2]="a";`



`char symbol='a';`



2.4. Знаки операций

- Унарные
- Бинарные
- Тернарная

Унарные операции



ИЛИ

—
~
!
&
*
++
--

Например

++ a

a++

- x

! x

Бинарные операции



Знаки

$+$ $-$ $*$ $/$ $\%$ - арифметические

$>$ $<$ $==$ $!=$

$>=$ $<=$ - отношения

$>>$ $<<$ - сдвиги

$\&\&$ $\|\$ \wedge - логические

$\&$ $|$ - логические поразрядные

,

Например $a+b$ $a\%b$

$(x>0)\&\&(x\leq 1)$ это означает $0<x\leq 1$

Тернарная операция (условная)



Float Y, X;

.....

Y = (X != 0.) ? (sin(X)/X) : (1.0);

2.6. Ключевые слова – заранее зарезервированные имена

Следующие идентификаторы зарезервированы для использования в качестве ключевых слов и не могут использоваться иным образом:

auto	break	case	char	class	const	continue
default	delete	do	double	else	enum	extern
float	for	goto	if	int	long	new
public	register	return	short	sizeof	static	struct
switch	typedef	union	unsigned	void	while	

2.7

ОПЕРАТОР – формальная запись предписания для выполнения действия или последовательности действий, заданных алгоритмом