

Министерство образования и науки Украины  
Харьковский национальный университет  
радиоэлектроники



Елементи мови С

2008

**Содержание**

Введение.....	3
Теория.....	4
Алфавіт.....	4
Ідентифікатори.....	4
Константи.....	4
Коментарі.....	6
Ключові слова.....	6

## Елементи мови С

Програмування являє собою процес визначення послідовності інструкцій, які повинен виконати комп'ютер для рішення певного завдання. Для вказівки цих інструкцій використовується мова програмування, наприклад С++. За допомогою текстового редактора можна вносити програмні оператори у вихідний файл. Далі використовується спеціальна програма - компілятор - для перетворення операторів з формату, що може читати й розуміти людина, в одиниці й нулі, які розуміє комп'ютер.

Будь-яка мова (українська, російська, англійська, французька та інші) складається з декількох основних елементів - символів, слів, словосполучень і речень. В алгоритмічних мовах програмування існують аналогічні структурні елементи, тільки слова називають лексемами, словосполучення - виразами, а речення - операторами.

Лексеми в свою чергу утворюються із символів, вирази - із лексем і символів, оператори - із символів, лексем і виразів.

**Алфавіт** мови, або її символи - це основні неподільні знаки, за допомогою яких пишуться всі тексти на мові програмування.

**Лексема**, або елементарна конструкція - мінімальна одиниця мови, яка має самостійний зміст.

**Вираз** задає правило обчислення деякого значення.

**Оператор** задає кінцевий опис деякої дії.

## Теорія

### Алфавіт

Алфавіт мови Cі включає: великі та малі літери латинської абетки, арабські цифри, пробільні символи : пробіл, символи табуляції, символ переходу на наступний рядок тощо; символи , . ; : ? ' ! | / \ ~ ( ) [ ] { } < > # % ^ & - + \* =.

### Ідентифікатори

**Ідентифікатори** використовуються для іменування різних об'єктів: змінних, констант, міток, функцій тощо. При записі ідентифікаторів можуть використовуватися великі та малі літери латинської абетки, арабські цифри та символ підкреслення. Ідентифікатор не може починатися з цифри і не може містити пробілів.

Компілятор мови Cі розглядає літери верхнього та нижнього регістрів як різні символи. Тому можна створювати ідентифікатори, які співпадають орфографічно, але відрізняються регістром літер. Наприклад, кожний з наступних ідентифікаторів унікальний:

Sum, sum, sUm, SUM, sUM.

Слід також пам'ятати, що ідентифікатори не повинні співпадати з ключовими словами.

### Константи

**Константами** називають сталі величини, тобто такі, які в процесі виконання програми не змінюються. В мові Cі існує чотири типи констант: цілі, дійсні, рядкові та символні.

**Цілі константи** можуть бути десятковими, вісімковими або шістнадцятковими. Десяткова константа - послідовність десяткових цифр (від 0 до 9), яка починається не з нуля якщо це число не нуль. Приклади десяткових констант : 10, 132, 1024. Вісімкові константи починаються з символу 0, після якого розміщуються вісімкові цифри (від 0 до 7). Наприклад: 023. Запис константи вигляду 08 буде сприйматися компілятором як помилка, так як 8 не є вісімковою цифрою. Шістнадцяткові константи починаються з символів 0x або 0X, після яких розміщуються шістнадцяткові цифри (від 0 до F, можна записувати їх у верхньому чи нижньому регістрах). Наприклад : 0XF123.

**Дійсні константи** складаються з цілої частини, десяткової крапки, дробової частини, символу експоненти (e чи E) та показника степеня. Дійсні константи мають наступний формат представлення:

[ ціла\_частина ][ . дробова\_частина ][ E [-] степінь ]

У записі константи можуть бути опущені ціла чи дробова частини (але не обидві разом), десяткова крапка з дробовою частиною чи символ E (e) з показником степеня (але не разом). Приклади дійсних констант:

2.2, 220e-2, 22.E-1, .22E1.

Якщо потрібно сформувані від'ємну цілу або дійсну константу, то перед константою необхідно поставити знак унарного мінуса.

**Символьна константа** - це один або декілька символів, які заключені в апострофи. Якщо константа складається з одного символу, вона займає в пам'яті 1 байт (тип char). Двосимвольні константи займають в пам'яті відповідно 2 байти (тип int). Послідовності символів, які починаються з символу \ (обернений слеш) називаються **керуючими** або **escape-послідовностями**. Escape-послідовності представлені в таблиці 1.

Таблиця 1 - Escape-послідовності

Спеціальний символ	Шістнадцятковий код	Значення
\a	07	звуковий сигнал
\b	08	повернення на 1 символ
\f	0C	переведення сторінки
\n	0A	перехід на наступний рядок
\r	0D	повернення каретки
\t	09	горизонтальна табуляція
\v	0B	вертикальна табуляція
\\	5C	символ \
\'	27	символ '
\"	22	символ "
\?	3F	символ ?
\0	00	нульовий символ
\0ddd		вісімковий код символу
\0xddd	ddd	шістнадцятковий код символу

**Рядкові константи** записуються як послідовності символів, заключених в подвійні лапки.

"Це рядковий літерал!\n"

Для формування рядкових констант, які займають декілька рядків тексту програми використовується символ \ (обернений слеш):

"Довгі рядки можна розбивати на \ частини"

Загальна форма визначення іменованої константи має вигляд:

**const тип ім'я = значення;**

Модифікатор `const` попереджує будь-які присвоювання даному об'єкту, а також інші дії, що можуть вплинути на зміну значення. Наприклад:

```
const float pi = 3.1415926;  
const maxint = 32767;  
char *const str="Hello,P...!"; /* покажчик-константа */  
char const *str2= "Hello!"; /* покажчик на константу */
```

Використання одного лише модифікатору `const` еквівалентно `const int`.

## Коментарі

У процесі створення програми використовуються коментарі, текст яких завжди ігнорується компілятором, але дозволяє програмісту описувати призначення тої чи іншої частини програми. В мові С++ використовується два різновиду коментарів:

***/\* текст \*/*** - **багатострочний коментар**, який не може бути вкладеним, і

***// текст*** - **однострочний коментар** до кінця строки.

Коментарі здебільшого використовуються для "документування програм" та під час їх відлагодження. В програму бажано вміщувати текст, що хоч якось пояснює її роботу та призначення. Проте не слід надто зловживати коментарями, а використовувати більш розумні форми найменування змінних, констант, функцій тощо. Якщо, наприклад, функція матиме назву `add_matrix`, очевидно не зовсім раціональним буде включення у програму після її заголовної частини коментар про те, що:

```
/*функція обчислює суму матриць */
```

У цьому випадку ім'я функції пояснює її призначення. У більш сучасних версіях Сі широко застосовується так званий угорський запис імен, коли ім'я змінної містить в собі інформацію про її призначення і тип.

## Ключові слова

**Ключові слова** - це зарезервовані ідентифікатори, які мають спеціальне значення для компілятора. Їх використання суворо регламентоване. Імена змінних, констант, міток, типів тощо не можуть співпадати з ключовими словами.

Перелік ключових слів мови Сі наведено в таблиці 2.

Таблиця 2 - Ключові слова мови Сі

auto	continue	float	interrupt	short	unsigned
asm	default	for	long	signed	void
break	do	far	near	sizeof	volatile
case	double	goto	pascal	static	while
cdecl	else	huge	switch	struct	
char	enum	if	register	typedef	
const	extern	int	return	union	

