

2.2. Алфавіт, лексеми, синтаксис мови

У природній мові спілкування виділяють чотири основні елементи: символ, слово, словосполучення та речення. Подібні елементи існують і в алгоритмічній мові, тільки слова мають назву лексеми, словосполучення — вирази, а речення — оператори. Лексеми створюються із символів, вирази — із лексем та символів, оператори — з символів, виразів і лексем.

Алфавіт мови C++ включає:

- великі (**A-Z**) і малі (**a—z**) літери латинського алфавіту та символ підкреслення (**_**);
- арабські цифри від **0** до **9**;
- знаки арифметичних дій **+**, **-**, *****, **/**, **%**, **++**, **--**;
- знаки побітових операцій **<<**, **>>**, **&**, **|**, **~**, **^**;
- знаки відношень **<**, **<=**, **=**, **!=**, **>**, **>=**;
- знаки логічних операцій **&&**, **||**, **!**;
- розділові знаки **,** **;** **:** **пропуск**;
- спеціальні знаки **.**, **=**, **->**, **?**, ****, **\$**, **#**, **'**, **"**;
- символи дужок **(**, **)**, **[**, **]**, **{**, **}**.

Інші символи, а також літери кирилиці не використовуються для побудови базових елементів мови або для їх розділу, але вони можуть застосовуватись у символічних константах та коментарях.

Лексеми, тобто базові елементи мови з певним самостійним значенням, складаються із символів алфавіту. До них відносять ідентифікатори, ключові слова, знаки операцій, константи, роздільники (дужки, крапка, кома, символи пропуску). Межі лексем визначаються іншими лексемами-роздільниками або знаками операцій.

Ідентифікатором, тобто ім'ям програмного об'єкта, називається будь-яка послідовність літер латинського алфавіту, цифр і символу підкреслення за умови, що першою стоїть літера або символ підкреслення, а не цифра.

Існує два різновиди ідентифікаторів:

- *стандартні*, наприклад, імена всіх вбудованих у мову функцій;
- *користувальницькі*.

Характерно, що мова C++ чутлива до регістру літер, тому компілятор розпізнає великі і малі літери латинського алфавіту як різні символи. Це дає можливість створювати ідентифікатори, що однаково читаються, але відрізняються написом одного або декількох символів. Наприклад, ідентифікатори «**Sigma**», «**sigma**» і «**sigMa**» вважаються різними.

Ідентифікатори можуть мати будь-яку довжину, але значимими є не більше 31 символу від початку ідентифікатора, а в деяких компіляторах це обмеження ще більш суворе (не більше 8 символів). Імена програмних об'єктів створюються на етапі оголошення даних, після цього їх можна використовувати в різних операторах програми.

Надамо декілька порад щодо вибору ідентифікатора:

- рекомендується не жалкувати часу на створення ідентифікаторів переважно за їх змістовим призначенням;
- ім'я програмного об'єкта повинно легко розпізнаватись і, бажано, не мати символів, які можна переплутати між собою;
- ідентифікатор не повинен збігатися з ключовими словами, а також з іменами стандартних об'єктів мови C++;

- не слід починати ідентифікатори із символу підкреслення, бо вони можуть співпадати з іменами системних функцій або змінних;
- для розділу частин імені можна використовувати символ підкреслення;
- всередині ідентифікатора не можна розміщати символи пропуску.

Ключовими (службовими) словами називають ряд зарезервованих ідентифікаторів, що вживаються для побудови конструкцій мови і мають фіксоване значення. За смисловим навантаженням службові слова поділяються на такі основні групи:

- специфікатори типів — **char, int, long, typedef, short, float, double, enum, struct, union, signed, unsigned, void;**
- кваліфікатори типів — **const i volatile;**
- класи пам'яті — **auto, extern, register, static;**
- для побудови операторів — **for, while, do, if, else, switch, case, continue, goto, break, return, default, sizeof.**

У *табл. 2.1* наведено список основних ключових слів C++.

Таблиця 2.1

Ключові слова C++

asm	delete	goto	register	throw
auto	do	if	return	try
break	double	inline	short	typedef
case	else	int	signed	typename
catch	enum	long	sizeof	union
char	explicit	new	static	unsigned
class	extern	operator	struct	virtual
const	float	private	switch	void
continue	for	protected	template	volatile
default	friend	public	this	while

Як роздільники лексем застосовуються такі символи: пропуск, табуляція, символ нового рядка, коментар. Між будь-якими двома лексемами допускається довільна кількість символів-роздільників. Крім того, деякі лексеми («*», «+», «,», « », «(», «->» тощо) самі є роздільниками і відділяти їх від інших лексем символами-роздільниками необов'язково.