

Операції та вирази мови C



Лекція №6

Дисципліна «Програмування»

Пріоритети (ранги) операцій

Ранг	Операції	Асоціативність
1	<code>() [] -> .</code>	→
2	<code>! ~ + - ++ -- & * (тип) sizeof</code> (унарні)	←
3	<code>* / %</code> (мультиплікативні бінарні)	→
4	<code>+ -</code> (адитивні бінарні)	→
5	<code><< >></code> (порозрядного зсуву)	→
6	<code>< <= > =></code> (відношення)	→
7	<code>== !=</code> (відношення)	→
8	<code>&</code> (порозрядна кон'юнкція «І»)	→
9	<code>^</code> (порозрядна виключна диз'юнкція або додавання за модулем 2)	→
10	<code>!</code> (порозрядна диз'юнкція «АБО»)	→
11	<code>&&</code> (кон'юнкція «І»)	→
12	<code> </code> (диз'юнкція «АБО»)	→
13	<code>? :</code> (умовна операція)	←
14	<code>= *= /= %= += -= &= ^= = <<= >>=</code>	←
15	<code>,</code> (операція «кома»)	→



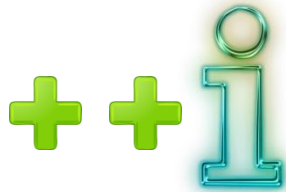
Унарні (одномісні) операції

Для зображення одномісних префіксних і постфіксних операцій використовуються такі символи:

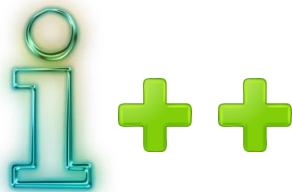
- &** – операція отримання адреси операнда;
- *** – операція звернення за адресою;
- – унарний мінус;
- +** – унарний плюс;
- ~** – порозрядне інвертування (побітове заперечення);
- !** – логічне заперечення (**НЕ**);
- ++** – збільшення на одиницю (**інкремент**);
- – зменшення на одиницю (**декремент**);
- sizeof** – операція обчислення розміру (в байтах).

Операція «Інкремент»

- 1) **префіксна операція** – збільшення значення операнда на **1** до його використання.

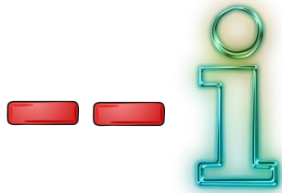


- 2) **постфіксна операція** – збільшення значення операнда на **1** після його використання.

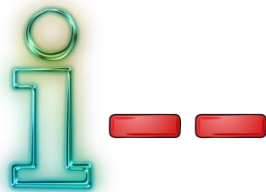


Операція «Декремент»

- 1) **префіксна операція** – зменшення значення операнда на **1** до його використання.



- 2) **постфіксна операція** – зменшення значення операнда на **1** після його використання.





Операція sizeof

sizeof – операція обчислення розміру (в байтах) для об'єкта того типу, який має операнд.

Дозволені два формати цієї операції:

sizeof вираз

sizeof (тип)



sizeof не обчислює значення виразу, а лише визначає його тип, для якого потім обчислюється розмір.

Приклади: **sizeof** z, **sizeof** (9.8 + 5)
sizeof (int), **sizeof** (char)



Бінарні (двомісні) операції

Бінарні операції поділяються на наступні групи:

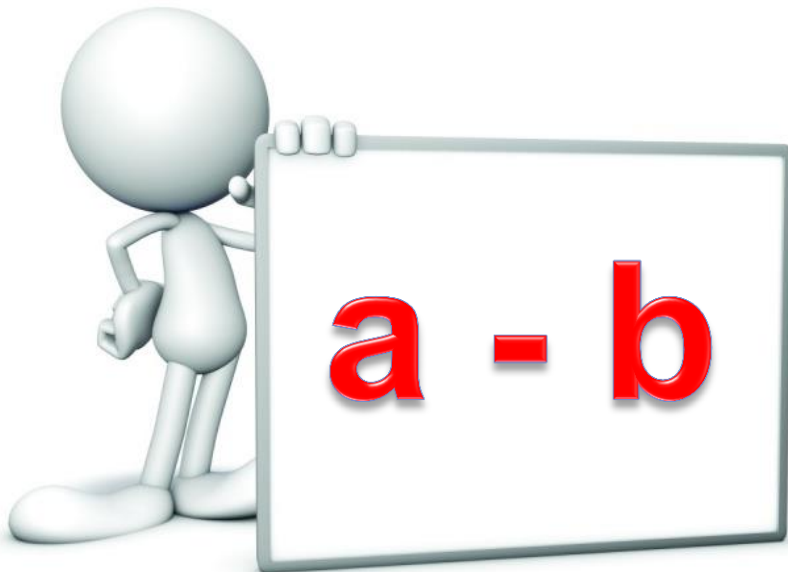
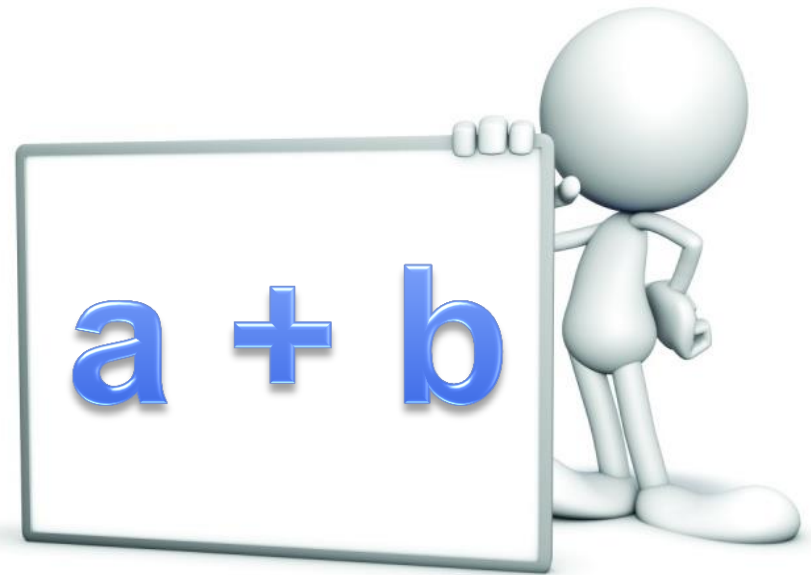
- адитивні;
- мультиплікативні;
- зсувів;
- порозрядні;
- операції відношень;
- логічні;
- присвоєння;
- вибору компонента структурованого об'єкта;
- операція «кома»;
- дужки в якості операцій.





Адитивні операції

+ – бінарний плюс – додавання арифметичних операндів або додавання вказівника до цілочислового операнда;



- – бінарний мінус – віднімання арифметичних операндів або віднімання вказівників.



Мультиплікативні операції



***** – **множення** операндів арифметичного типу;

/ – **ділення** операндів арифметичного типу.

За умови цілочислових операндів абсолютне значення результату округляється до цілого.

Наприклад, $20/3 = 6$, $-20/3 = -6$, $(-20)/3 = -6$, $20/(-3) = -6$;

% – **отримання залишку від ділення** цілочислових операндів (**ділення за модулем**).

За умови невід'ємних операндів – залишок позитивний.

В іншому випадку залишок визначається реалізацією:

$$\begin{array}{ll} 13 \% 4 = 1, & (-13) \% 4 = -1; \\ 13 \% (-4) = 1, & (-13) \% (-4) = -1. \end{array}$$

При ненульовому дільнику для цілочислових операндів завжди виконується співвідношення: **$(a / b) * b + a \% b = a$** .



Операції зсуву

Операції зсуву визначені тільки для цілочислових операндів.

Формат виразу з операцією зсуву:

операнд_лівий **операція_зсуву** операнд_правий

- <<** – **зсув вліво** бітового представлення значення лівого цілочислового операнда на кількість розрядів, що дорівнює значенню правого цілочислового операнда;
- >>** – **зсув вправо** бітового представлення значення лівого цілочислового операнда на кількість розрядів, що дорівнює значенню правого цілочислового операнда.



Порозрядні операції

До порозрядних операцій належать:

& – порозрядна кон'юнкція (І);

| – порозрядна диз'юнкція (АБО);

^ – порозрядна виключна диз'юнкція.



Результати операцій зсуву:

$$4 \ll 2 = 16; \quad 100_2 \leftarrow 10000_2 = 16_{10}$$

$$5 \gg 1 = 2; \quad 101_2 \rightarrow 10_2 = 2_{10}$$

Результати порозрядних операцій:

$$6 \& 5 = 4; \quad 6 | 5 = 7; \quad 6 \wedge 5 = 3.$$

$$\begin{array}{r} \& 110 \\ 101 \\ \hline 100 \end{array}$$

$$\begin{array}{r} | 110 \\ 101 \\ \hline 111 \end{array}$$

$$\begin{array}{r} \wedge 110 \\ 101 \\ \hline 011 \end{array}$$

Операції відношення

До операцій відношення (порівняння) належать:

- <** – менше, ніж;
- >** – більше, ніж;
- <=** – менше або дорівнює;
- >=** – більше або дорівнює;
- ==** – дорівнює;
- !=** – не дорівнює.





Логічні бінарні операції

До логічних бінарних операцій належать:

&& – **кон'юнкція** (**I**) арифметичних операндів або відношень. Цілочисловий результат **0** (хибність) або **1** (істина);

|| – **диз'юнкція** (**АБО**) арифметичних операндів або відношень. Цілочисловий результат **0** (хибність) або **1** (істина).

Результати операцій відношення та логічних операцій:

3 < 5 дорівнює 1;

3 > 5 дорівнює 0;

3 == 5 дорівнює 0;

3 != 5 дорівнює 1;

3 != 5 || 3 == 5 дорівнює 1;

3 + 4 > 5 && 3 + 5 > 4 && 4 + 5 > 3 дорівнює 1.



Операції присвоювання

= – **просте присвоювання**: присвоїти значення виразу-операнда з правої частини операнду лівої частини.

Приклад: **P = 10.3 - 2 * x;**

***=** – **присвоювання після множення**: присвоїти операнду лівої частини добуток значень обох операндів.

Приклад: **P *= 2** еквівалентно **P = P * 2;**

/= – **присвоювання після ділення**: присвоїти операнду лівої частини частку від ділення значення лівого операнда на значення правого.

Приклад: **P /= 2.2 - d** еквівалентно **P = P / (2.2 - d);**

%= – **присвоювання після ділення по модулю**: присвоїти операнду лівої частини остачу від цілочислового ділення значення лівого операнда на значення правого операнда.

Приклад: **N %= 3** еквівалентно **N = N % 3;**



Операції присвоювання

+= – **присвоювання після підсумовування**: присвоїти операнду лівої частини суму значень обох операндів.

Приклад: **A += B** еквівалентно **A = A + B**;

-= – **присвоювання після віднімання**: присвоїти операнду лівої частини різницю значень лівого і правого операндів.

Приклад: **X -= 4.3 - Z** еквівалентно **X = X - (4.3 - Z)**;

<<= – **присвоювання після зсуву розрядів вліво**: присвоїти цілочисловому операнду лівої частини значення, отримане зсувом вліво його бітового представлення на кількість розрядів, що дорівнює значенню правого цілочислового операнда.

Приклад: **a <<= 4** еквівалентно **a = a << 4**;



Операції присвоювання

>>= – присвоювання після зсуву розрядів вправо:
присвоїти цілочисловому операнду лівої частини значення, отримане зсувом вправо його бітового представлення на кількість розрядів, що дорівнює значенню правого цілочислового операнда.

Приклад: **a >>= 4** еквівалентно **a = a >> 4;**

&= – присвоювання після порозрядної кон'юнкції:
присвоїти цілочисловому операнду лівої частини значення, отримане порозрядною кон'юнкцією (І) його бітового представлення з бітовим представленням цілочислового операнда правої частини.

Приклад: **e &= 44** еквівалентно **e = e & 44;**



Операції присвоювання

$:=$ – присвоювання після порозрядної диз'юнкції: присвоїти цілочисловому операнду лівої частини значення, отримане порозрядною диз'юнкцією (АБО) його бітового представлення з бітовим представленням цілочислового операнда правої частини.

Приклад: $a := b$ еквівалентно $a = a \mid b$;

\wedge – присвоювання після порозрядної виключної диз'юнкції: присвоїти цілочисловому операнду лівої частини значення, отримане застосуванням порозрядної виключної диз'юнкції до бітових представлень значень обох операндів.

Приклад: $z \wedge = x + y$ еквівалентно $z = z \wedge (x + y)$.



Операції вибору компонентів структурованого об'єкта

До операцій вибору компонентів структурованого об'єкта належать:

- **(крапка) – прямий вибір** (виділення) **компонента структурованого об'єкта**, наприклад, об'єднання або структури.

Формат застосування операції:

ім'я_структурованого_об'єкта.ім'я_компонента

- > – **непрямий вибір** (виділення) **компонента структурованого об'єкта**, що адресується вказівником. При використанні операції потрібно, щоб з об'єктом був пов'язаний вказівник.

Формат застосування операції має вигляд:

вказівник_на_структурований_об'єкт->ім'я_компонента



Операція «кома»

Кілька виразів, розділених комами «**,**», обчислюються послідовно зліва направо.

В якості результату зберігаються тип і значення самого правого виразу.

Таким чином, операція «кома» групує обчислення зліва направо.

Тип і значення результату визначаються самим правим з розділених комами операндів (виразів).

Значення всіх лівих операндів ігноруються.

Наприклад, якщо змінна **x** має тип **int**, то значенням виразу (**x = 3 , 3 * x**) буде **9**, а змінна **x** прийме значення **3**.



Дужки в якості операцій С

Круглі () і **квадратні [] дужки** відіграють роль бінарних операцій під час виклику функцій та індексування елементів масивів.

Круглі дужки обов'язкові у зверненні до функції:

ім'я_функції(список_аргументів)

де операндами служать **ім'я_функції** і **список_аргументів**.

Результат виклику визначається (обчислюється) в тілі функції, структуру якого задає її визначення.

У виразі

ім'я_масиву[індекс]

операндами для операції **[]** служать **ім'я_масиву** та **індекс**.



Умовна операція

Умовна тернарна операція використовується з трьома операндами. У зображенні умовної операції застосовуються два символи '?' та ':' і три вирази-операнди:

вираз_1 ? вираз_2 : вираз_3

Першим обчислюється значення **виразу_1**.

Якщо він істинний, тобто не дорівнює нулю, то обчислюється значення **виразу_2**, який стає результатом.

Якщо при обчисленні **виразу_1** отримаємо **0**, то в якості результату береться значення **виразу_3**.

Класичний приклад: **$x < 0 ? -x : x$;**

Вираз повертає абсолютну величину змінної x .



Операція перетворення типу

Операція перетворення (приведення) типу:

(ім'я_типу) операнд

В якості операнда використовується унарний вираз, який в найпростішому випадку може бути змінною, константою або будь-яким виразом в круглих дужках.

Приклад: **(long)8** (4 байта) і **(char)8** (1 байт) змінюють довжину внутрішнього представлення цілих констант, не змінюючи їх значень.

Приклад: **(long double)6** або **(float)4** не тільки змінюють довжину константи, але й структуру її внутрішнього представлення. У результатах будуть виділені порядок і мантиса, значення будуть дійсними.