*Федоренко Іван*

*556-ІПЗ*

**23 варіант**

1. Охарактеризувати поняття явне та неявне перетворення типів.

Є 2 основних способи конвертації типів:

* *Неявна конвертація* типів, коли компілятор автоматично конвертує один фундаментальний тип даних в інший. Виконується коли ми передаємо в змінну значення не її типу (наприклад число з плаваючою комою в int = компілятор неявно заокруглить до більшого або меншого).

Неявна конвертація відбувається одним з двох способів:

числове розширення;

числова конверсія.

Коли значення з одного типу даних конвертується в інший тип даних, який є більшим (за розміром і по діапазону значень), то це називається числовим розширенням. Наприклад, тип int може бути розширений в тип long.



Натомість, числова конверсія – конвертація значення з більшого типу даних в аналогічний, але менший тип.



* *Явна конвертація* типів, коли розробник використовує один з операторів конвертації для виконання конвертації об’єкта одного типу даних в інший.

Перевагою явної конвертації є значне спрощення читання коду – замість уважно придивлятися до правильності переданих значень, розробник одразу бачить який тип даних використовується у певній операції.

|  |  |
| --- | --- |
| Наприклад результат виконання цього коду буде 3: | А цього – необхідний нам дробовий результат. |

Такий тип явного перетворення називається C-style.

1. Охарактеризувати директиви препроцесора. Навести приклади.

Препроцесор – це програма, яка опрацьовує директиви. Директиви препроцесора – це команди компілятору, котрі виконуються на початку компіляції програми. У мові С++ директиви починаються з символу # а опісля містять необхідну директиву без пробілу. Після директиви НЕ ставиться крапка з комою.

*Приклади*:

#include – означає, що до програми необхідно приєднати програмний код із зазначеного після неї файлу. Файли, які приєднують директивою #include називаються файлами заголовків ( header-файли, також відомі як модулі або бібіліотеки). У таких файлах зазвичай оголошують сталі та змінні, заголовки функцій котрі часто використовуватимуться у файлі.

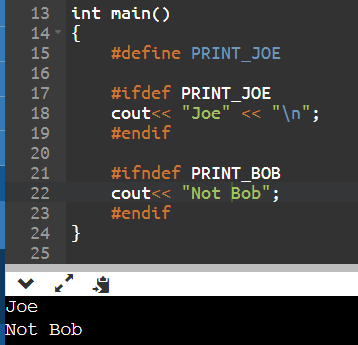
#define, #undef – записати константу і переписати на звичайну змінну – котрій при зверненні можна змінювати значення. Перевагою директиви #define є краще сприйняття коду – швидше та приємніше читання, і водночас – зрозумілі помилки, якщо значення забудуть передати.

|  |  |
| --- | --- |
|  |  |

#ifdef, #endif – перевірка, чи вже було визначене значення (If Defined). Директива #endif є кінцем блоку перевірки #ifdef.

#ifndef (If Not Defined) – якщо такого значення ще не було.

На прикладі нижче є визначена константа PRINT\_JOE – на рядку 17 директивою #ifdef компілятором перевіряється чи вже було визначено константу PRINT\_JOE. Вона вже була визначена, отож на рядку 18 я виводжу відповідне повідомлення у консоль.



Для прикладу використав директиву #ifndef – рядок 21 перевіряє чи НЕ було визначено PRINT\_BOB. Такої константи визначено не було, отже відповідне повідомлення про Боба також виводиться.

1. Скласти програму на мові С++, яка з двох введених з клавіатури цілих чисел відшукує найбільше.

|  |  |
| --- | --- |
|  |  |