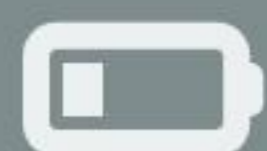


Дійсні Числа та іотаніп

Тема 1.4: Числа з плаваючою точкою та Математика

Типи даних з плаваючою точкою

У реальному житті не все можна порахувати цілими числами (вага, температура, гроші).



float

4 байти

Точність: ~6–7 знаків

Проста графіка (економія)



double

8 байт

Точність: ~15–16 знаків

Стандарт C++



long double

10–16 байт

Точність: 18+ знаків

Наукові розрахунки

Чому double? float втрачає точність вже після 7-го знаку (помилки в "пі").

Форматування виводу (iomanip)

Керування виводом

За замовчуванням C++ виводить ~6 знаків. Для грошей або фізики цього може бути мало або забагато.

- Бібліотека: `#include`
- `fixed` — закріпити кому.
- `setprecision(n)` — кількість знаків після коми.

```
#include using namespace std; int main() {  
    double pi = 3.1415926535; // 2 знаки (гроші)  
    cout << fixed << setprecision(2) << pi; // 3.14  
    // 5 знаків (фізика) cout << fixed <<  
    setprecision(5) << pi; // 3.14159 return 0; }
```

Інкремент (++) та Декремент (--)

Програмісти придумали короткі команди для додавання/віднімання одиниці.



Інкремент (Level Up)

```
i++;
```

Аналог: $i = i + 1;$

Збільшити на 1



Декремент

```
i--;
```

Аналог: $i = i - 1;$

Зменшити на 1

⚠ Пастка: Відсотки

Проблема `int / int`

Ми знаємо, що `int / int` дає ціле число.

Задача: Знайти 50% від числа 90.

$$(50 / 100) * 90 = 0$$

Помилка: `50 / 100` дасть 0.

Рішення: `Double`

Використовуйте крапку, щоб вказати дробовий тип.

```
// Неправильно: int res = (50 / 100) * 90; //  
0 // Правильно: double res = (50.0 / 100.0) *  
90; // Результат: 45.0
```

Математичні Прогресії

Арифметична

Кожне наступне число більше на крок d .

$$a_n = a_1 + d(n-1)$$

$$S_n = ((a_1 + a_n) \cdot n) / 2$$

Геометрична

Кожне наступне число множиться на q .

$$b_n = b_1 \cdot q^{n-1}$$

Код: `b1 * pow(q, n-1)`

(Використовуємо бібліотеку `cmath`)

Практика

Перевірте знання на практичній роботі 1.4!

