

## **Практична робота №4. Проектування і реалізація програми з перевантаженням операторів**

**Мета:** засвоєння поняття статичного поліморфізму через перевантаження операторів (бінарних і унарних); набуття практичних навичок їх оголошення та використання.

### **Актуалізація опорних знань**

1. Для чого використовують перевантаження операторів?
2. Назвіть оператори мови C++, які можна перевантажувати.
3. Які оператори не можна перевантажувати?

### **Завдання до практичної роботи**

1. Створити клас для роботи з комплексними числами *Complex*.
2. Комплексне число складається з двох частин: дійсної і уявної. Нехай дано  $a + bi$  та  $c + di$ . Операції, виконувані з комплексними числами:
  - додавання:  $(a + c) + (b + d)i$ ;
  - віднімання:  $(a - c) + (b - d)i$ ;
  - множення:  $(ac - bd) + (ad + bc)i$ ;
  - множення на число  $x$ :  $(xa) + (xb)i$ ;
  - ділення:  $\frac{ac + bd}{c^2 + d^2} - \frac{bc - ad}{c^2 + d^2} i$ ;
  - перевірка чи числа рівні:  $a + bi$  та  $c + di$  рівні, якщо  $a = c$  і  $b = d$ ;
  - пошук спряженого числа:  $a + bi$  та  $c + di$  спряжені якщо їх дійсні частини рівні, а коефіцієнти при уявних частинах рівні за модулем, але протилежні за знаком (наприклад,  $4 + 3i$  та  $4 - 3i$  спряжені).
3. Клас комплексного числа має містити конструктор і деструктор, а також перевантажені операторні функції, необхідні для реалізації:
  - додавання двох комплексних чисел (оператор +);
  - віднімання двох комплексних чисел (оператор -);
  - множення комплексних чисел, у тому числі множення звичайного (цілого) числа на комплексне (оператор \*);
  - ділення комплексних чисел (оператор /);
  - порівняння двох комплексних чисел (оператор ==);
  - обчислення спряженого числа (оператор унарний -).
4. Перевантажте хоча б один оператор (на вибір) з використанням дружньої функції.
5. Програма має містити меню, яке дозволить перевірити основні функції, створені у програмі.

Зокрема, у програмі потрібно надати користувачу можливість введення комплексних чисел та забезпечити виведення у форматі  $-3 + 7i$ , а також можливість вибрати операцію, яку може виконати користувач з комплексними числами. Результат операції потрібно виводити у вигляді  $-1 + 4i + 7 - 6i = 6 - 2i$ .
5. Продемонструвати роботу створеної програми.

## Методичні рекомендації

**Перевантаження операторів** – один із способів реалізації поліморфізму, що полягає в можливості одночасного існування в одній області видимості **декількох різних варіантів застосування оператора**, що мають одне і те ж ім'я, але розрізняються типами параметрів, до яких вони застосовуються. У мові програмування C++ існує можливість перевантаження більш ніж 40 операцій.

Для перевантаження операторів використовуються **операторні функції**, які повинні мати прямий доступ до членів класу, тобто мають бути або членами класу (містять 0 або 1 операнд), або дружніми функціями (у дружню функцію не передається вказівник *this*, тому потрібно передати 1 або 2 операнди).

*Приклад операторної функції, що є членом класу:*

```
Complex operator+(Complex &c)
{
    Complex sum;
    sum.real=real+c.real;
    sum.imag=imag+c.imag;
    return sum;
}
```

*Приклад дружньої операторної функції:*

```
// прототип у класі
friend Complex operator+ (const Complex &c1, const Complex &c2);

// дружня функція поза класом
Complex operator+ (const Complex &c1, const Complex &c2)
{
    Complex sum;
    sum.real = c1.real + c2.real;
    sum.imag = c1.imag + c2.imag;
    return sum;
}
```

*Приклад додавання об'єктів класу в int main():*

```
c3=c1+c2;
c3.displaySum();
```

### Контрольні запитання:

1. Що таке дружня функція?
2. Чи можна змінити пріоритет перевантаженого оператора?
3. Чим синтаксис дружньої функції-оператора відрізняються від синтаксиста оператора-функції члена класу?