

Тема

Перевантаження функцій і операторів, дружні функції, статичні члени класу.

Мета

Навчитися використовувати механізм перевантаження функцій та операторів. Навчитися створювати та використовувати дружні функції. Ознайомитися зі статичними полями і методами та навчитися їх використовувати.

Теоретичні відомості

Перевизначення операторів

C++ підтримує спеціальні засоби, які дозволяють перевизначити вже існуючі оператори. Наприклад, для оператора + можна ввести своє власне визначення, яке реалізує операцію додавання для об'єктів певного класу. Фактично перевизначення для операцій існує і в мові С. Так, операція + може використовувати як об'єкти типу int, так і об'єкти типу float. C++ розширює цю ідею.

Для визначення оператора використовується функція, що вводиться користувачем. Тип функції визначається іменем класу, далі записується ключове слово operator, за яким слідує назва оператора, в круглих дужках дається перелік параметрів, серед яких хоча б один типу клас.

Наприклад, визначимо оператор + для класу complex комплексних чисел.

```
class complex
{
    double real, imag; // дійсна та уявна частини
public:
    // конструктор з ініціалізацією
    complex(double r, double i): real(r), imag(i) {}

    // перевизначена функція додавання викликається автоматично,
    // коли додаються два об'єкти типу complex
    complex operator+(const complex & right)
    {
        return complex(real + right.real, imag + right.imag);
    }
};

int main()
{
    complex c1(0, 1), c2(1, 0); // об'єкти типу клас complex
    complex c3 = c1 + c2;       // скорочений запис

    c3 = c1.operator+(c2);      // явний виклик функції. Результат однаковий
    return 0;
}
```

Функція operator+ повертає результат типу complex та має параметр типу complex. Функції-операції мають бути нестатичними функціями-членами класу або функціями поза межами класу та мати мінімум один аргумент типу класу, посилання на клас, перелічення (enumeration) чи посилання на перелічення.

За виключенням операції присвоєння всі перевизначені оператори наслідуються.

Дружні функції

Дружною функцією класу називається функція, яка сама не є членом класу, але має повні права на доступ до закритих та захищених елементів класу. Оскільки така функція не є членом класу, то вона не може бути вибрана з допомогою операторів (.) та (->). Дружня функція класу викликається

звичайним способом. Для опису дружньої функції використовується ключове слово friend. Наприклад,

```
class X
{
    int i{0};

    // оголошення дружньої функції
    friend void friend_func(X *, int);
public:
    void member_func(int a)
    {
        i = a;
    }
};

// доступ до закритого елемента через вказівник об'єкту
void friend_func(X * px, int a)
{
    px->i = a;
}

int main()
{
    X mm;
    mm.member_func(6); // виклик методу класу

    friend_func(&mm, 6); //виклик дружньої функції
}
```

Статичні змінні класу

До тепер вважалось, що дані в кожному об'єкті є власністю саме цього об'єкту та не використовуються іншими об'єктами цього класу. Та іноді приходится слідкувати за накопиченням даних. Наприклад, необхідно з'ясувати скільки об'єктів даного класу було створено на даний момент та скільки з них існує. Статичні змінні-члени досяжні для всіх екземплярів класу. Це є компроміс між глобальними даними, які досяжні всім елементам програми, та даними, що досяжні тільки об'єктам даного класу.

Статична змінні створюється в одному екземплярі для всіх об'єктів даного класу. Розглянемо приклад. Об'єкт класу Cat включає статичну змінну-член HowManyCats (скільки котів). Ця змінна нараховує кількість об'єктів класу Cat, що створені під час виконання програми. Для цього статична змінна HowManyCats збільшується на 1 при кожному виклику конструктора класу Cat , а при виклику деструктора зменшується на 1.

```
class Cat
{
public:
    explicit Cat(unsigned int age): itsAge(age) { ++HowManyCats; } // конструктор
    virtual ~Cat() { --HowManyCats; } // деструктор

    ...

    // оголошення статичної змінної
    static unsigned int HowManyCats;
private:
    unsigned int itsAge;
};

// визначення статичної змінної
unsigned int Cat::HowManyCats = 0;

int main()
```

```

{
    const unsigned int MaxCats = 5;
    Cat * CatHouse[MaxCats];
    for (unsigned int i = 0; i < MaxCats; ++i)
    {
        CatHouse[i] = new Cat(i); // виклик конструктора
    }
    ...
    return 0;
}

```

Статичні змінні необхідно обов'язково визначити поза межами оголошення класу. Ініціалізувати статичні змінні в межах класу можна лише, якщо вони є константні та інтегрального типу (bool, char, short, int, long, long long, etc).

Якщо необхідно обмежити доступ до статичних змінних, то оголошуйте їх закритими або захищеними.

Завдання

На основі класу з попередньої лабораторної:

- Перевантажити як мінімум три функції-члени з попереднього завдання.
- Перевантажити оператори згідно з варіантом.
- Створити дружні функції згідно з варіантом.
- Створити статичні поля та статичні методи згідно з варіантом.
- Створити статичне поле, в якому б містилась інформація про кількість створених об'єктів, а також статичні функції для роботи з цим полем.
- Продемонструвати можливості класу написавши для нього модульні тести для Google Test.
- Оформити звіт до лабораторної роботи. Звіт повинен містити:
 - Тему, мету та теоретичні відомості
 - Індивідуальне завдання
 - Код класу
 - Код тестів
 - Результат виконання тестів

Варіанти завдань

Варіант	Завдання
1	<p>Клас CFraction– звичайний дріб.</p> <p>Перевантажити оператори, як функції члени:</p> <ul style="list-style-type: none"> • Додавання (+) • Віднімання (-) • Множення (*) • Ділення (/) • Обертання дробу (!). <p>Перевантажити оператори, як дружні-функції:</p> <ul style="list-style-type: none"> • Введення дробу з потоку вводу (<<) • Виведення дробу в потік виводу (>>) • Більше (>) • Менше (<) • Рівне (==).
2	<p>Клас CComplex – комплексне число.</p> <p>Перевантажити оператори, як функції члени:</p> <ul style="list-style-type: none"> • Додавання (+) • Віднімання (-)

	<ul style="list-style-type: none"> • Множення (*) • Піднесення до n-го степеня (^) <p>Перевантажити оператори, як дружні-функції:</p> <ul style="list-style-type: none"> • Введення числа з потоку вводу (<<) • Виведення числа в потік виводу (>>) • Більше (>) • Менше (<) • Рівне (==).
3	<p>Клас CBigNumber – велике ціле число (розміром 128 біти – як два long long числа).</p> <p>Перевантажити оператори, як функції члени:</p> <ul style="list-style-type: none"> • Додавання (+) • Віднімання (-) <p>Перевантажити оператори, як дружні-функції:</p> <ul style="list-style-type: none"> • Введення числа з потоку вводу (<<) • Виведення числа в потік виводу (>>) • Більше (>) • Менше (<) • Рівне (==).
4	<p>Клас CFixedPointNumber – число із фіксованою крапкою з точністю до сотих (після крапки). Використати ціле число та сприймати розряди одиниць та десятків як такі, що після крапки.</p> <p>Перевантажити оператори, як функції члени:</p> <ul style="list-style-type: none"> • Додавання (+) • Віднімання (-) • Множення (*) • Ділення (/) • Приведення до типу double (operator double()) <p>Перевантажити оператори, як дружні-функції:</p> <ul style="list-style-type: none"> • Введення числа з потоку вводу (<<) • Виведення числа в потік виводу (>>) • Більше (>) • Менше (<) • Рівне (==).
5	<p>Клас CPoint – точка в просторі</p> <p>Перевантажити оператори, як функції члени:</p> <ul style="list-style-type: none"> • Зсув у просторі на задані значення координат (+=) • Збільшення всіх координат в певну кількість разів (*) • Відстань до іншої точки (-) <p>Перевантажити оператори, як дружні-функції:</p> <ul style="list-style-type: none"> • Введення числа з потоку вводу (<<) • Виведення числа в потік виводу (>>) • Рівне (==).
6	<p>Клас CVector – вектор в просторі.</p> <p>Перевантажити операції, як функції члени:</p> <ul style="list-style-type: none"> • Додавання (+) • Віднімання (-) • Скалярний добуток (*) • Векторний добуток (%) <p>Перевантажити операції, як дружні-функції:</p> <ul style="list-style-type: none"> • Добуток вектора на скаляр • Введення дробу з потоку вводу (<<) • Виведення дробу в потік виводу (>>) • Більше (>)

	<ul style="list-style-type: none"> • Менше (<) • Рівне (==).
7	<p>Клас CSphere – куля у просторі</p> <p>Перевантажити оператори, як функції члени:</p> <ul style="list-style-type: none"> • Перенесення центру кулі на по всіх координатах на задане число (+) • Збільшення радіусу кулі у задану кількість разів (*) <p>Перевантажити оператори, як дружні-функції:</p> <ul style="list-style-type: none"> • Введення кулі з потоку вводу (<<) • Виведення кулі в потік виводу (>>) • Рівне (==) • Нерівне (!=)
8	<p>Клас CTriangle – трикутник на площині (задаються довжини трьох сторін).</p> <p>Перевантажити оператори, як функції члени:</p> <ul style="list-style-type: none"> • Збільшення одразу всіх трьох сторін трикутника на константу (+) • Збільшення одразу всіх трьох сторін трикутника у певну кількість разів (*) • Доступ до i-ї сторони трикутника ([i]) • Приведення трикутника до дійсного типу (повертати його площу). <p>Перевантажити операції, як дружні-функції:</p> <ul style="list-style-type: none"> • Введення трикутника з потоку вводу (<<) • Виведення трикутника в потік виводу (>>) • Більше (>) • Менше (<) • Рівне (==)
9	<p>Клас CRectangle – прямокутник на площині (задаються довжини сторін)</p> <p>Перевантажити оператори, як функції члени:</p> <ul style="list-style-type: none"> • Збільшення одразу всіх трьох сторін трикутника на константу (+) • Збільшення одразу всіх трьох сторін трикутника у певну кількість разів (*) • Приведення прямокутника до дійсного типу (повертати його площу). • Обертання прямокутника – зміна сторін (!) <p>Перевантажити оператори, як дружні-функції:</p> <ul style="list-style-type: none"> • Введення прямокутника з потоку вводу (<<) • Виведення прямокутника в потік виводу (>>) • Більше (>) • Менше (<) • Рівне (==)
10	<p>Клас CPolynom2 – лінійний двочлен вигляду (ax+c).</p> <p>Перевантажити оператори, як функції члени:</p> <ul style="list-style-type: none"> • Додавання (+) • Віднімання (-) • Знаходження значення виразу для заданого x (()) • Заміна всіх коефіцієнтів полінома на протилежні за знаком (!) <p>Перевантажити оператори, як дружні-функції:</p> <ul style="list-style-type: none"> • Добуток полінома на скаляр • Введення полінома з потоку вводу (<<) • Виведення полінома в потік виводу (>>) • Доступ до i-го коефіцієнта полінома ([i]) • Рівне (==) (при порівнянні порівнювати значення всіх коефіцієнтів).
11	<p>Клас CPolynom3 – квадратичний тричлен (ax²+bx+c).</p> <p>Перевантажити оператори, як функції члени:</p> <ul style="list-style-type: none"> • Додавання (+) • Віднімання (-) • Знаходження значення виразу для заданого x (()) • Заміна всіх коефіцієнтів полінома на протилежні за знаком (!)

	<p>Перевантажити оператори, як дружні-функції:</p> <ul style="list-style-type: none"> • Добуток полінома на скаляр • Введення полінома з потоку вводу (<<) • Виведення полінома в потік виводу (>>) • Доступ до i-го коефіцієнта полінома ([]) • Рівне (==) (при порівнянні порівнювати значення всіх коефіцієнтів).
12	<p>Клас CPolynom4 – кубічний чотиричлен (ax^3+bx^2+cx+d).</p> <p>Перевантажити оператори, як функції члени:</p> <ul style="list-style-type: none"> • Додавання (+) • Віднімання (-) • Знаходження значення виразу для заданого x (()) • Заміна всіх коефіцієнтів полінома на протилежні за знаком (!) <p>Перевантажити оператори, як дружні-функції:</p> <ul style="list-style-type: none"> • Добуток полінома на скаляр • Введення полінома з потоку вводу (<<) • Виведення полінома в потік виводу (>>) • Доступ до i-го коефіцієнта полінома ([]) • Рівне (==) (при порівнянні порівнювати значення всіх коефіцієнтів).
13	<p>Клас CBitField – бітове поле (для 32 бітів).</p> <p>Перевантажити оператори, як функції члени:</p> <ul style="list-style-type: none"> • Побітове І (&) • Побітове АБО () • Побітове Виключне АБО (^) <p>Перевантажити оператори, як дружні-функції:</p> <ul style="list-style-type: none"> • Введення бітового поля з потоку вводу (<<) • Виведення бітового поля в потік виводу (>>) • Доступ до i-го біта ([]) • Рівне (==) (при порівнянні порівнювати значення всіх бітів)
14	<p>Клас CTime – час. Реалізовує роботу з часом – години, хвилини, секунди. Години та хвилини мають бути цілочисельні, а секунди – дійсне число.</p> <p>Перевантажити оператори, як функції члени:</p> <ul style="list-style-type: none"> • Збільшення часу на задану кількість секунд (дійсними числами) (+) • Збільшення часу у задану кількість разів (дійсними числами) (*) • Додавання двох об'єктів типу CTime. Результат – час, який дорівнює сумі. (+) • Різниця в секундах з іншим часом заданим через параметр (-) • Приведення часу до секунд (operator double()) <p>Перевантажити оператори, як дружні-функції:</p> <ul style="list-style-type: none"> • Введення часу з потоку вводу (<<) • Виведення часу в потік виводу (>>) • Менше (<) • Більше (>) • Рівне (==)
15	<p>Клас CDate – дата. Реалізовує роботу із датою – рік, місяць, день.</p> <p>Перевантажити оператори, як функції члени:</p> <ul style="list-style-type: none"> • Різницю в днях між двома датами (-) • Збільшення дати на задану кількість днів (+=) • Зменшення дати на задану кількість днів (-=) <p>Перевантажити оператори, як дружні-функції:</p> <ul style="list-style-type: none"> • Введення дати з потоку вводу (<<) • Виведення дати в потік виводу (>>) • Менше (<) • Більше (>) • Рівне (==)

