## Лабораторна роботи №11

***Створення нових типів даних на основі структур С++. Перевантаження операцій. Елементи ООП з використанням структур.***

***Мета роботи:*** *ознайомлення з основами об'єктно-орієнтованого програмування на основі комбінованих типів.*

***Завдання***

Описати структуру даних (struct) згідно варіанта. Реалізувати функції для перевантаження операторів постановки екземпляра структури в потік (виводу на консоль, знак "<<") та отримання з потоку (вводу з консолі, знак ">>"). Реалізувати функції-члени структури, вказані у варіанті завдання. Скласти програму, яка в діалоговому режимі пропонує користувачеві ввести дані для екземплярів, та вибрати операцію, яку потрібно виконати

***Варіанти завдань***

1. Структура ***вектор в тривимірному просторі***. Визначити операції: додавання двох векторів (оператор "+"), скорочений оператор присвоєння з додаванням іншого вектора (оператор "+="), скалярний добуток двох векторів (оператор "*"), множення вектора на число (оператор "*"), отримання протилежного вектора (оператор "!").

2. Структура ***раціональне число***. Визначити операції: віднімання двох раціональних чисел (оператор "-"), ділення двох раціональних чисел (оператор "/"), піднесення раціонального числа до цілого степеня (оператор "^"), оператор постфіксного декременту (оператор "--"), скорочений оператор присвоєння з додаванням (оператор "+=").

3. Структура ***квадратна матриця 3x3***. Визначити операції: додавання двох матиць (оператор "+"), віднімання двох матриць (оператор "-"), добуток двох матриць (оператор "*"), множення матриці на число (оператор "*"), отримання оберненої (оператор "!").

4. Структура ***комплексне число***. Визначити операції: множення двох комплексних чисел (оператор "*"), додавання двох комплексних чисел (оператор "+"), порівняння модуля комплексного числа з дійсним числом на відношеннями «менше» «більше» (оператори "<", ">"), множення комплексного числа на дійсне число (оператор "*").

5. Структура ***раціональне число***. Визначити операції: множення двох раціональних чисел (оператор "*"), ділення двох раціональних чисел (оператор "/"), додавання до раціонального числа цілого числа (оператор "+"), оператор «менше» для порівняння двох раціональних

чисел (оператор “<”), скорочений оператор присвоєння з відніманням (оператор “-=”).

6. Структура ***комплексне число***. Визначити операції: віднімання двох комплексних чисел (оператор “-”), додавання двох комплексних чисел (оператор “+”), множення комплексного на дійсне (оператор “*”), , скорочені оператори присвоєння з множенням та діленням комплексних чисел (оператори “*=” і “/=”).

7. Структура ***вектор в тривимірному просторі***. Визначити операції: додавання двох векторів (оператор “+”), скорочений оператор присвоєння з відніманням іншого вектора (оператор “-=”), векторний добуток двох векторів (оператор “*”), ділення вектора на число (оператор “/”), збільшення усіх координат вектора на 1 (оператор інкременту “++”).

8. Структура ***раціональне число***. Визначити операції: віднімання двох раціональних чисел (оператор “-”), додавання двох раціональних чисел (оператор “+”), множення раціонального на ціле (оператор “*”), оператор «більше» для порівняння двох раціональних чисел (оператор “>”), скорочений оператор присвоєння з діленням (оператор “/=”).

9. Структура ***квадратна матриця 4x4***. Визначити операції: додавання двох матиць (оператор “+”), віднімання двох матриць (оператор “-”), поелементного множення двох матриць (оператор “*”), поелементного ділення матриць (оператор “/”), отримання транспонованої матриці (оператор “!”).

10. Структура ***множина цілих чисел***. Визначити операції: перетину двох множин (оператор “*”), об’єднання двох множин (оператор “+”), різниці двох множин (оператор “-”), включення цілого числа в множину (оператор “+=”), виключення елемента з множини (оператор “-=”).

11. Структура ***раціональне число***. Визначити операції: множення двох раціональних чисел (оператор “*”), додавання двох раціональних чисел (оператор “+”), порівняння раціонального числа з цілим числом на відношення «менше» (оператор “<”), оператор постфіксного інкременту (оператор “++”), оператор «менше» для порівняння двох раціональних чисел (оператор “<”).

12. Структура ***прямокутний паралелепіпед*** (зберігає розміри: ширина, висота довжина). Визначити оператори: інкременту (оператор “++”) та декременту (оператор “--”) шляхом додавання/віднімання одиниці від усіх, розмірів паралелепіпеда та усі оператори порівняння (“>”, “<”, “<=”, “>=”, “= =”, “!=”) за об’ємом, тобто більшим вважається той паралелепіпед, об’єм якого більший.

13. Структура ***комплексне число***. Визначити операції: віднімання двох комплексних чисел (оператор “-”), ділення двох комплексних чисел (оператор “/”), піднесення комплексного числа до цілого степеня (оператор “^”), скорочений оператор присвоєння з множенням (оператор “*=”), скорочений оператор присвоєння з додаванням (оператор “+=”).

14. Структура ***вектор в тривимірному просторі***. Визначити операції: віднімання двох векторів (оператор “-”), скорочений оператор присвоєння з множенням вектора на число (оператор “*=”), векторний добуток двох векторів (оператор “*”), множення вектора на число (оператор “*”), отримання протилежного вектора (оператор “унарний міну -”).

15. Структура ***раціональне число***. Визначити операції: порівняння двох раціональних чисел «більше», «менше», «рівне» (оператори “>”, “<”, “= =”), скорочені оператори присвоєння для чотирьох арифметичних операцій (оператори “+=”, “-=”, “*=”, “/=”).

***Приклад.***

16. Структура звичайний дріб. Визначити операції: додавання двох дробів (оператор “+”), множення двох дробів (оператор “*”), додавання до дробу цілого числа (оператор “+”), збільшення дробу на 1 (оператор “++”), отримання оберненого дробу (оператор “!”).

```
#include <iostream>

#include "Windows.h"

using namespace std;


struct rational

{

    int a;//чисельник дробу

    int b;//знаменнник дробу

    //оператори члени структури

    rational operator+(int);

    rational operator++(int);

    rational operator+(rational);

    rational operator!();

};
```

```
// допоміжні функції пошуку спільного знаменника
int _nsk(int a, int b);
int euklid_nsd(int a, int b);
//оператори - зовнішні функції
rational operator*(rational, rational);
istream& operator>>(istream&, rational&);
ostream& operator<<(ostream&, rational&);

void main()
{
    SetConsoleOutputCP(1251);
    SetConsoleCP(1251);
    rational a, b;
    while (true)//цикл діалогу з користувачем
    {
        system("cls");
        int choise = 0;
        cout << "Оберіть потрібну дію:\n"<<
        "1 - ввести дроби\n2 - додати дроби\n" <<
        "3 - обернений до a\n4 - знайти b + 1\n" <<
        "5 - додати до b вказане число\n6 - добуток\n"
        <<"7 - завершити роботу"<<endl;
        cin >> choise;
        rational c;
        switch (choise)
        {
        case 1: cout << "Введіть два дроби a та b";
            cout << " через пробіл, кожен з \"\/\"\n";
            cin >> a >> b;
            break;
```

```
        case 2: c = a + b;
        cout << a << " + " << b << " = " << c<< endl;
            break;
        case 3: c = !a;
        cout << "Обернений до " << a<<" = "<<c<< endl;
            break;
        case 4: c = b++;
            cout << b << "++ = " << c << endl;
            break;
        case 5: int n;
            cout << "Введіть ціле число";
            cin >> n;
            rational c = a + n;
        cout << a << " + " << n << " = " << c << endl;
            break;
        case 6: c = a * b;
        cout << a << " * " << b << " = " << c << endl;
            break;
        case 7: return;
        }
        system("pause");
    }
}
rational rational::operator+(int number)
{
    rational result;
    result.b = b;
    result.a = a + b * number;
    return result;
}
```

```
rational rational::operator++(int k)
{//постфіксний інкремент містить фіктивний
     //параметр k, щоби його можна було
     //записувати після змінної-дробу
     rational result;
     result.b = b;
     result.a = a + b;
     return result;
}
rational rational::operator+(rational other)
{//оператор + член структури додає
     //дробу a/b до якого він застосований
     //дріб other (other.a/other.b)
     rational result;
     result.b = _nsk(b, other.b);
     int dop1 = result.b / b;
     int dop2 = result.b / other.b;
     result.a = a * dop1 + other.a * dop2;
     return result;
}
rational rational::operator!()
{
     rational result;
     result.a = b;
     result.b = a;
     return result;
}
int _nsk(int a, int b)
{
     return a*b/euklid_nsd(a, b);
```

```
}
int euklid_nsd(int a, int b)
{
    while (b > 0)
    {   int nsd = b;
        b = a % b;
        a = nsd;      }
    return a;
}
rational operator*(rational p, rational q)
{
    rational result;
    int num = p.a * q.a;
    int denum = p.b * q.b;
    //скорочення дробу
    result.a = num/euklid_nsd(num,denum);
    result.b = denum/ euklid_nsd(num, denum);
    return result;
}
istream& operator>>(istream& is, rational& r)
{
    char d;//знімаємо риску друбу при вводі
    is >> r.a >> d >> r.b;
    return is;
}
ostream& operator<<(ostream& os, rational& r)
{
    os << r.a << "/" << r.b;
    return os;
}
```