**­**

**Лабораторна робота №1**

З дисципліни «ООП»

«Розробка простого класу»

Виконав: Землянський Едуард

Група: КВ-22

**Постановка задачі**

Розробити клас згідно до варіанту. Не використовувати default-реалізацію

конструкторів та операторів (реалізувати самостійно). Обробка помилкових

ситуацій не є обов’язковою в рамках даної роботи. Пропонується самостійно

розробити інтерфейс класу, зокрема, виходячи з власних міркувань зручності

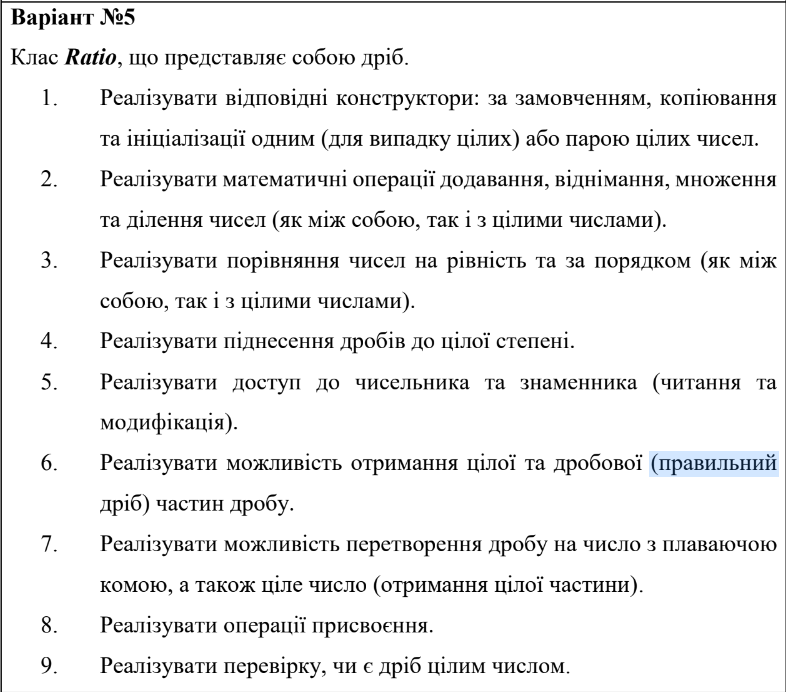
його подальшого використання (бажано зробити використання класу по

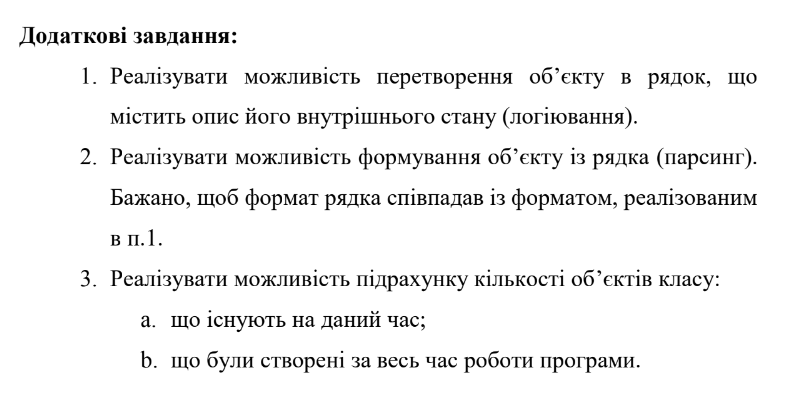
можливості інтуїтивно зрозумілим).

**Варіант завдання**

№5

**Вимоги до програми**

****

****

**Текст програми**

**Тестування програми**

1. Реалізувати відповідні конструктори: за замовченням, копіювання та ініціалізації одним (для випадку цілих) або парою цілих чисел.



1. Реалізувати математичні операції додавання, віднімання, множення та ділення чисел (як між собою, так і з цілими числами).

Ratio ratio1 = Ratio(1, 2), ratio2 = Ratio(1, 2), ratio3;

    ratio3 = ratio1 + ratio2;

    ratio1.GetRatio();

    ratio2.GetRatio();

    ratio3.GetRatio();

    cout << "---" << endl;

    ratio1 = Ratio(6, 3); ratio2 = Ratio(1, 3);

    ratio3 = ratio1 - ratio2;

    ratio1.GetRatio();

    ratio2.GetRatio();

    ratio3.GetRatio();

    cout << "---" << endl;

    ratio1 = Ratio(4, 2); ratio2 = Ratio(1, 3);

    ratio3 = ratio1 \* ratio2;

    ratio1.GetRatio();

    ratio2.GetRatio();

    ratio3.GetRatio();

    cout << "---" << endl;

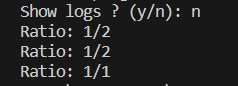
    ratio1 = Ratio(1, 2); ratio2 = Ratio(1, 3);

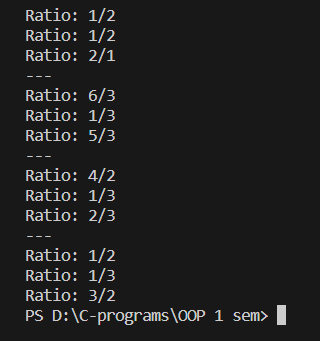
    ratio3 = ratio1 / ratio2;

    ratio1.GetRatio();

    ratio2.GetRatio();

    ratio3.GetRatio();





З цілим числом:

Ratio ratio1 = Ratio(1, 2), ratio2 = Ratio(3), ratio3;

    ratio3 = ratio1 + ratio2;

    ratio3.GetRatio();

    ratio3 = ratio1 - ratio2;

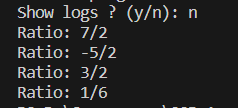
    ratio3.GetRatio();

    ratio3 = ratio1 \* ratio2;

    ratio3.GetRatio();

    ratio3 = ratio1 / ratio2;

    ratio3.GetRatio();

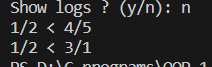


1. Реалізувати порівняння чисел на рівність та за порядком (як між собою, так і з цілими числами).

Ratio ratio1 = Ratio(1, 2), ratio2 = Ratio(3), ratio3;

    ratio1.Compare(Ratio(4, 5));

    ratio1.Compare(ratio2);

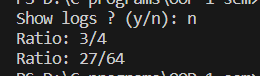


1. Реалізувати піднесення дробів до цілої степені.

Ratio ratio1 = Ratio(1, 2), ratio2 = Ratio(3, 4), ratio3;

    ratio2.GetRatio();

    ratio2.Power(3).GetRatio();



1. Реалізувати доступ до чисельника та знаменника (читання та модифікація).

Ratio ratio1 = Ratio(1, 2), ratio2 = Ratio(3, 4), ratio3;

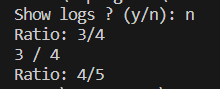
    ratio2.GetRatio();

    cout << ratio2.numerator << " / " << ratio2.denominator << endl;

    ratio2.numerator = 4;

    ratio2.denominator = 5;

    ratio2.GetRatio();

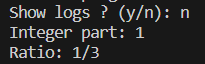


1. Реалізувати можливість отримання цілої та дробової (правильний дріб) частин дробу.

Ratio ratio1 = Ratio(1, 2), ratio2 = Ratio(4, 3), ratio3;

    cout << "Integer part: " << ratio2.GetIntegerPart() << endl;

    ratio2.GetRatioPart().GetRatio();



1. Реалізувати можливість перетворення дробу на число з плаваючою комою, а також ціле число (отримання цілої частини).

Ratio ratio1 = Ratio(1, 2), ratio2 = Ratio(4, 3), ratio3;

    cout << ratio2.GetRatioAsFloat() << endl;



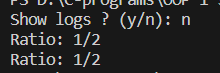
1. Реалізувати операції присвоєння.

Ratio ratio1 = Ratio(1, 2), ratio2 = Ratio(4, 3), ratio3;

    ratio3 = ratio1;

    ratio1.GetRatio();

    ratio3.GetRatio();



1. Реалізувати перевірку, чи є дріб цілим числом.

Ratio ratio1 = Ratio(12, 2), ratio2 = Ratio(4, 3), ratio3 = Ratio(3);

    ratio1.CheckIfInt();

    ratio2.CheckIfInt();

    ratio3.CheckIfInt();

