## Практическое задание для урока 10 модуля 3

1.

Создай класс Fraction (обыкновенная дробь). Добавь конструктор, который будет принимать два целых числа и добавлять объекту атрибуты - числитель и знаменатель. Добавь метод print который будет выводить текстовое представление объекта в формате: «Дробь {числитель}/{знаменатель}», например «Дробь 1/2». Переопредели методы арифметических операций сложение, вычитание, умножение, деление (/). Методы должны возвращать объект класса Fraction - новую дробь равную результату выполнения арифметической операции с двумя исходными дробями.

Считай с клавиатуры четыре целых чисел - числитель и знаменатель первой дроби и числитель и знаменатель второй дроби. Создай экземпляры класса Fraction с параметрами считанными с клавиатуры. Выведи результаты арифметических операций сложение, вычитание, умножение, деление (/) этих дробей.

Входные данные:

Вводится четыре строки, каждая строка содержит целое число - числитель и знаменатель первой дроби и числитель и знаменатель второй дроби.

Выходные данные:

Выводится 4 строки в формате «Дробь {числитель}/{знаменатель}» без кавычек.

**Пример работы программы:**

**Входные данные:**

2

4

1

4

**Выходные данные:**

Дробь 3/4

Дробь 1/4

Дробь 1/8

Дробь 2/1

## Решения.

1.

#include <iostream>

using namespace std;

class Fraction

{

public:

int num, den;

Fraction(int n, int d)

{

num=n/NOD(n, d);

den = d/NOD(n, d);

}

int NOD(int a, int b)

{

while(a > 0 && b > 0)

if(a > b)

a %= b;

else

b %= a;

return a + b;

}

int get\_common\_denominator(int den1,int den2)

{

int common\_den = den1 \* den2 / NOD(den1, den2);

return common\_den;

}

void print()

{

cout<<"Дробь "<< num<<"/"<<den<<endl;

}

Fraction operator + (Fraction c2)

{

int common\_den = get\_common\_denominator(this->den, c2.den);

int num = common\_den / this->den \* this->num + common\_den / c2.den \* c2.num;

num = num/NOD(num, common\_den);

int den = common\_den/NOD(num, common\_den);

return Fraction(num, den);

}

Fraction operator - (Fraction c2)

{

int common\_den = get\_common\_denominator(this->den, c2.den);

int num = common\_den / this->den \* this->num - common\_den / c2.den \* c2.num;

num = num/NOD(num, common\_den);

int den = common\_den/NOD(num, common\_den);

return Fraction(num, den);

}

Fraction operator \* (Fraction c2)

{

int num = this->num \* c2.num;

int den = this->den \* c2.den;

num = num/this->NOD(num, den);

den = den/this->NOD(num, den);

return Fraction(num, den);

}

Fraction operator / (Fraction c2)

{

int num = this->num \* c2.den;

int den = this->den \* c2.num;

num = num/this->NOD(num, den);

den = den/this->NOD(num, den);

return Fraction(num, den);

}

};

int main()

{

int num, den;

cin>>num;

cin>>den;

Fraction fraction1(num, den);

cin>>num;

cin>>den;

Fraction fraction2(num, den);

Fraction fraction3=fraction1 + fraction2;

fraction3.print();

fraction3=fraction1 - fraction2;

fraction3.print();

fraction3=fraction1 \* fraction2;

fraction3.print();

fraction3=fraction1 / fraction2;

fraction3.print();

return 0;

}