**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования**

**«МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ»**

**(МТУСИ)**

Кафедра

«Информационная безопасность»

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине:

«Объектно-ориентированное программирование систем защиты информации»

на тему:

«Классы с динамическими структурами данных. Шаблонные классы.»

Выполнила: Яковлева К.А.

Группа: БПЗ1801

Вариант №12 (27)

Проверил: Барков В.В.

Москва 2021

**ОБЩЕЕ ЗАДАНИЕ.**

Выберите для выполнения лабораторной работы свой вариант.

Для выбранного варианта определите класс, включив в него:

· конструктор по умолчанию;

· конструктор инициализации;

· конструктор преобразования базового типа к типу, определяемому разрабатываемым классом;

В разрабатываемом классе перегрузите потоковые операции для объектов класса. Для выполнения задания предложенного варианта перегрузите необходимые математические операции.

При разработке класса вашего варианта учтите индивидуальные уточнения для функций – членов класса.

Разработайте для объектов вашего класса предложенные в каждом варианте пользовательские функции.

Разработайте функцию main, организующую ввод данных и демонстрацию работы разработанных функций

**Программный код**

Листинг 1 - лаба4.cpp

#include <iostream>

#include <math.h>

using namespace std;

double NOD(int count1, int count2)

{

int del;

while (count2 != 0)

{

del = count2;

count2 = count1 % count2;

count1 = del;

}

return count1;

}

double NOK(int count1, int count2)

{

return count1 \* count2 / NOD(count1, count2);

}

class Rational

{

int nominator;

int denominator;

double x;

public:

Rational(int nominator, int denominator)

:nominator(nominator), denominator(denominator) {}

int GetNominator() const

{

return nominator;

}

int GetDenominator() const

{

return denominator;

}

void PrintCount()

{

cout << nominator << "/" << denominator;

}

friend Rational operator +(Rational a, Rational b)

{

int del = NOK(a.GetDenominator(), b.GetDenominator());

return Rational(a.GetNominator() \* (del / b.GetDenominator()) + b.GetNominator() \* (del / a.GetDenominator()), del);

}

friend Rational operator -(Rational a, Rational b)

{

int del = NOK(a.GetDenominator(), b.GetDenominator());

return Rational(a.GetNominator() \* (del / b.GetDenominator()) - b.GetNominator() \* (del / a.GetDenominator()), del);

}

friend Rational operator \*(Rational a, Rational b)

{

return Rational(a.GetNominator() \* b.GetNominator(), a.GetDenominator() \* b.GetDenominator());

}

friend Rational operator /(Rational a, Rational b)

{

return Rational(a.GetNominator() \* b.GetDenominator(), a.GetDenominator() \* b.GetNominator());

}

};

void main()

{

double x;

int nominator, denominator, del;

cout << "Enter x = ";

cin >> x;

nominator = 0.5 \* x \* x - x;

denominator = 3 \* x - (0.2 \* x + 0.5 \* x \*x) ;

del = NOD(nominator, denominator);

nominator /= del;

denominator /= del;

Rational y1(nominator, denominator);

cout << "Rational count = ";

y1.PrintCount();

cout << endl;

Rational y2(1, 40);

Rational y3(0, 0), y4(0, 0), y5(0, 0), y6(0, 0);

y3 = y1 + y2;

y3.PrintCount();

cout << endl;

y4 = y1 - y2;

y4.PrintCount();

cout << endl;

y5 = y1 \* y2;

y5.PrintCount();

cout << endl;

y6 = y1 / y2;

y6.PrintCount();

}

**Результат работы**

