**Министерство науки и высшего образования РФ**

Федеральное государственное бюджетное общеобразовательное учреждение высшего образования

«Чувашский Государственный Университет им. И.Н.Ульянова»

Факультет информатики и вычислительной техники

Кафедра компьютерных технологий

Отчет Лабораторной работы №4

По дисциплине: «Объектно-ориентированное программирование»

Вариант 14

Выполнил студент группы КТ-43-21

Казаков А.Ю.

Проверил ст. преподаватель:

Мытникова Е. А.

Чебоксары, 2022

#include <iostream>

#include <string>

using namespace std;

class Fraction

{

public:

double x, y;

Fraction(double in\_x, double in\_y)

{

x = in\_x;

y = in\_y;

}

~Fraction()

{

cout << "Object deleted.\n";

}

void prdouble\_fraction()

{

cout << x << "/" << y << endl;

}

Fraction operator+(const Fraction &other)

{

Fraction temp(0, 0);

temp.x = this->x + other.x;

temp.y = this->y + other.y;

return temp;

}

Fraction operator/(const Fraction &other)

{

Fraction temp(0, 0);

temp.x = this->x / other.x;

temp.y = this->y / other.y;

return temp;

}

Fraction operator\*(const Fraction &other)

{

Fraction temp(0, 0);

temp.x = this->x \* other.x;

temp.y = this->y \* other.y;

return temp;

}

Fraction operator-(const Fraction &other)

{

Fraction temp(0, 0);

temp.x = this->x - other.x;

temp.y = this->y - other.y;

return temp;

}

void reduction()

{

int divisor = 2;

int copy\_x = x;

int copy\_y = y;

while (true)

{

if (copy\_x % divisor == 0 && copy\_y % divisor == 0)

{

copy\_x /= divisor;

copy\_y /= divisor;

break;

}

else

{

divisor++;

}

}

if (copy\_x != x)

{

cout << copy\_x << "/" << copy\_y << endl;

x = copy\_x;

y = copy\_y;

}

else

{

cout << "Fraction cannot be shortened." << endl;

}

}

void print()

{

cout << "x: " << this->x << "\ny: " << this->y << endl;

}

};

int main()

{

Fraction a\_b = Fraction(1, 3);

Fraction c\_d = Fraction(1,2);

Fraction e\_f = Fraction(4,1);

Fraction g\_h = Fraction(11,2);

Fraction k\_l = Fraction(1,1);

Fraction z = ((a\_b + c\_d) / e\_f) \* (g\_h - k\_l);

z.print();

z.reduction();

z.print();

return 0;

}

