Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет)

Факультет «Робототехника и комплексная автоматизация»

Кафедра «Системы автоматизированного проектирования»

**Отчет по лабораторной работе**

По курсу «Объектно-ориентированное программирование»

Выполнил: Студент Петраков С.А.

Группа РК6-26Б

Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2020 г.

**ПЯВУ Вариант 16R**

**Задание:**

Разработать объектно-ориентированную программу для перемножения целого числа и обыкновенной дроби, которые должны передаваться ей через аргументы командной строки. Результат операции должен отображаться строкой потока стандартного вывода в формате обыкновенной дроби. В любом случае для записи обыкновенной дроби должен применяться символьный формат, где ее числитель и знаменатель разделены знаком **'/'**. Программная реализация вычислений должна быть основана на разработке класса обыкновенных дробей с приватными полями данных для целочисленных значений числителя и знаменателя, а также дружественным методом перегрузки оператора **'\*'**. Конструкторы класса должны выполнять преобразования обоих операндов в указанный числовой формат обыкновенных дробей. Для приведения исходной и результирующей обыкновенных дробей к несократимому виду следует предусмотреть компонентный метод, реализующий алгоритм Евклида, и перегрузку оператора присваивания.

**Алгоритм:**

В нашем классе хранятся приватные поля, в котором имеются целочисленные числитель и знаменатель. При считывании дроби, сокращаем её с помощью алгоритма Евклида (находим наибольший общий делитель числителя и знаменателя, далее сокращаем на это число и числитель, и знаменатель). Перегружаем оператор ‘\*’ для целого числа и дроби (умножаем числитель на целое число), опять сокращаем обыкновенную дробь. Выводим её на экран.

**Входные данные:**

2 числа - аргументы командной строки: целое число и обыкновенная дробь, записанная в формате с ‘/’.

**Выходные данные:**

Результат умножения целого числа и обыкновенной дроби.

**Текст программы:**

#include <string.h>

#include <stdlib.h>

#include <iostream>

#include <stdio.h>

using namespace std;

class Fraction

{

private:

int nom;

int den;

public:

Fraction(int n = 0, int m = 1) : nom(n), den(m) {};

Fraction(char\* s);

void reduce();

operator char\* ();

int getDen() { return den; };

Fraction& operator=(Fraction& f);

friend Fraction& operator\*(Fraction& f, int n);

};

Fraction::Fraction(char\* s)

{

char\* p = strchr(s, '/');

den = 1;

if (p != NULL)

{

\*p++ = '\0';

den = atoi(p);

}

nom = atoi(s);

}

int euclide(int n, int m)

{

int r = 1;

while (n != 0)

{

r = m % n;

m = n;

n = r;

}

return m;

}

void Fraction::reduce()

{

int gcd = euclide(abs(nom), den);

nom /= gcd;

den /= gcd;

}

Fraction::operator char\* ()

{

static char s[32];

sprintf(s, "%d/%d", nom, den);

return s;

}

Fraction& Fraction::operator=(Fraction& f)

{

nom = f.nom;

den = f.den;

return \*this;

}

Fraction& operator\*(Fraction& f, int n)

{

f.nom \*= n;

return f;

}

int main(int argc, char\* argv[])

{

if (argc != 3)

{

cout << "Wrong number of arguments." << endl;

return (-1);

}

Fraction f(argv[2]);

f.reduce();

if (f.getDen() == 0)

{

cout << "Denominator should not be 0." << endl;

return (-1);

}

int num = atoi(argv[1]);

f = (f \* num);

f.reduce();

cout << (char\*)f << endl;

return 0;

}

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Тест №** | **Входные данные** | **Выходные данные** |
| 1 | -2 1/2 | -1/1 |
| 2 | 7 2/21 | 2/3 |

**Список использованной литературы:**

* Волосатова Т.М., Родионов С.В. Лекции по курсу «Объектно-ориентированное программирование»
* bigor.bmstu.ru