**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**ИНЖЕНЕРНО-ТЕХНОЛОГИЧЕСКАЯ АКАДЕМИЯ**

**Институт компьютерных технологий и информационной безопасности**

**Кафедра математического обеспечения и применения ЭВМ**

**Отчёты по лабораторным работам**

по курсу «Объектно-ориентированный анализ и проектирование»

**Вариант №2**

Выполнили: студенты группы КТбо2-1: Шутенко Остап Русланович

Проверил: доцент кафедры МОП ЭВМ Тарасов

ТАГАНРОГ 2018

**Лабораторная работа №1**

**Классы и объекты в С++**

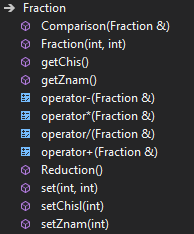
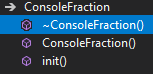
**Цель работы**

Ознакомление с технологией построения классов, разработкой методов, дружественных функций и перегрузкой операций.

**Формулировка задачи**

Определить класс Fraction, в котором производятся операции над обыкновенными дробями (сравнение, сложение, умножение, деление, сокращение).

**Диаграммы классов**



**Перечень сигнатур методов**

Метод Add (): складывает две дроби, приводя их к общему знаменателю.

Метод Div (): делит дробь, проверяя при этом, чтобы числитель второй дроби не был равен нулю.

Метод Mult (): перемножает две дроби.

Метод Print (): выводит на экран дробь.

Метод Initialization (): инициализирует дробь.

Метод Comparison (): сравнивает две дроби, приводя их к общему знаменателю и сравнивая числители.

Метод Reduction (): функция сокращает дробь

**Листинг программы lab1\_oop.cpp на языке С++**

#include <iostream>

#include "ConsoleFraction.h"

int main()

{

ConsoleFraction CF;

CF.init();

return 0;

}

**Листинг программы ConsoleFraction.h на языке С++**

#pragma once

#include "Fraction.h"

#include <iostream>

class ConsoleFraction

{

public:

ConsoleFraction();

~ConsoleFraction();

void init();

}

**Листинг программы Fraction.h на языке С++**

#pragma once

class Fraction

{

public:

// Конструктор

Fraction() {};

// Деструктор

~Fraction() {};

// Конструктор присваивания числителя и знаменателя, проверка знаменателя на равенство нулю

// В переменную а присваивается числитель, в переменную b - знаменатель, но только если он не равен нулю

Fraction(int chislit, int znamenat);

// Сложение

Fraction operator+(Fraction& s);

// разность

Fraction operator-(Fraction& s);

// Умножение

Fraction operator\*(Fraction& s);

// Деление

Fraction operator/(Fraction& s);

// Сокращение

Fraction Reduction();

// Сравнение

int Comparison(Fraction& s);

// Вывод

int getChis();

int getZnam();

// Инициализация дроби(изменение) - числителя и знаменателя

void set(int a, int b);

void setChisl(int a);

void setZnam(int b);

private:

int a = 0; //a-числитель

int b = 1; //b-знаменатель

};

};

**Листинг программы ConsoleFraction.cpp на языке С++**

#include "ConsoleFraction.h"

ConsoleFraction::ConsoleFraction() {};

ConsoleFraction::~ConsoleFraction() {};

void ConsoleFraction::init()

{

Fraction fa, fb;

while (true)

{

std::cout << "type first Fraction\n";

int a, b;

std::cin >> a >> b;

fa.set(a, b);

std::cout << "type last Fraction\n";

std::cin >> a >> b;

fb.set(a, b);

std::cin >> a;

switch (a)

{

case 0:fb = fa; fb.set(5, 6); std::cout << "b=a b=" << fb.getChis() << "/" << fb.getZnam() << "\n" << "a =" << fa.getChis() << "/" << fa.getZnam() << "\n"; break;

case 1:fb = fb + fa; std::cout << "b+a =" << fb.getChis() << "/" << fb.getZnam() << "\n"; break;

case 2:fb = fb - fa; std::cout << "b-a =" << fb.getChis() << "/" << fb.getZnam() << "\n"; break;

case 3:fb = fb \* fa; std::cout << "b\*a =" << fb.getChis() << "/" << fb.getZnam() << "\n"; break;

case 4:fb = fb / fa; std::cout << "b/a =" << fb.getChis() << "/" << fb.getZnam() << "\n"; break;

case 5:std::cout << fb.Comparison(fa) << "\n"; break;

}

if (a == -1)

break;

}

};

**Листинг программы Fraction.cpp на языке С++**

#include "Fraction.h"

#include <cstdlib>

Fraction::Fraction(int chislit, int znamenat)

{

try

{

if (znamenat == 0) // Если знаменатель равен нулю, то мы предупреждаем о неправильном вводе данных

throw 1;

else

{

this->a = chislit;

this->b = znamenat;

}

}

catch (int thr)

{

this->a = chislit;

this->b = 1;

exit(1);

}

}

// Сложение

Fraction Fraction::operator+(Fraction& s)

{

Fraction Sum;

// Приводим обе дроби к общему знаменателю

Sum.a = a \* s.b + b \* s.a;

Sum.b = b \* s.b;

try

{

if (Sum.b == 0) // Если знаменатель равен нулю, то мы предупреждаем о неправильном вводе данных

throw 1;

// Знаменатель конечной дроби равен произведению числителя второй дроби и числителя второй

else {

// Знаменатель конечной дроби равен произведению знаменателя первой и второй дроби

Sum.b = b \* s.b;

}

}

catch (int thr)

{

exit(1);

Sum.b = 1;

}

// Возвращаем новую дробь, являющейся суммой первой и второй дроби

return Sum.Reduction();

}

// разность

Fraction Fraction::operator-(Fraction& s)

{

Fraction Sum;

// Приводим обе дроби к общему знаменателю

Sum.a = a \* s.b - b \* s.a;

Sum.b = b \* s.b;

try

{

if (Sum.b == 0) // Если знаменатель равен нулю, то мы предупреждаем о неправильном вводе данных

throw 1;

// Знаменатель конечной дроби равен произведению числителя второй дроби и числителя второй

else {

// Знаменатель конечной дроби равен произведению знаменателя первой и второй дроби

Sum.b = b \* s.b;

}

}

catch (int thr)

{

exit(1);

Sum.b = 1;

}

// Возвращаем новую дробь, разностью суммой первой и второй дроби

return Sum.Reduction();

}

// Умножение

Fraction Fraction::operator\*(Fraction& s)

{

Fraction Mult;

// Числитель конечной дроби равен произведению числителя первой и второй дроби

Mult.a = a \* s.a;

Mult.b = b \* s.b;

// Возвращаем новую дробь, являющейся произведением первой и второй дроби

try

{

if (Mult.b == 0) // Если знаменатель равен нулю, то мы предупреждаем о неправильном вводе данных

throw 1;

// Знаменатель конечной дроби равен произведению числителя второй дроби и числителя второй

else {

// Знаменатель конечной дроби равен произведению знаменателя первой и второй дроби

Mult.b = b \* s.b;

}

}

catch (int thr)

{

exit(1);

Mult.b = 1;

}

return Mult.Reduction();

}

// Деление

Fraction Fraction::operator/(Fraction& s)

{

Fraction Div;

// Числитель конечной дроби равен произведению числителя первой дроби и знаменателя второй

Div.a = a \* s.b;

// Проверка на то что числитель второй дроби не равен 0

try

{

if (Div.b == 0) // Если знаменатель равен нулю, то мы предупреждаем о неправильном вводе данных

throw 1;

// Знаменатель конечной дроби равен произведению числителя второй дроби и числителя второй

else {

Div.b = b \* s.a;

}

}

catch(int thr)

{

exit(1);

Div.b = 1;

}

// Возвращаем дробь поделенную на другую

return Div.Reduction();

}

// Сокращение

Fraction Fraction::Reduction()

{

Fraction Red;

Red.a = a;

Red.b = b;

// Ищем общий делитель для числителя и знаменателя, чтобы сократить дробь

for (int i = 1; i <= a && i <= b; i++)

{

int aa = a % i;

int bb = b % i;

if (!(a % i) && !(b % i))

{

Red.a = a / i;

Red.b = b / i;

}

}

// Возвращаем сокращённую дробь

return Red;

}

// Сравнение

int Fraction::Comparison(Fraction& s)

{

Fraction d1, d2;

// Приводим обе дроби к общему знаменателю

d1.a = a \* s.b;

d1.b = b \* s.b;

d2.a = s.a \* b;

d2.b = s.b \* b;

// Сравниваем числителя после приведения к общему знаменателю

if (d1.a > d2.a)

return 1;

if (d1.a < d2.a)

return 2;

if (d1.a == d2.a)

return 0;

}

// Вывод

int Fraction::getChis()

{

return this->a;

}

int Fraction::getZnam()

{

return this->b;

}

// Инициализация дроби(изменение) - числителя и знаменателя

void Fraction::set(int a, int b)

{

this->a = a;

try

{

if (b == 0) // Если знаменатель равен нулю, то мы предупреждаем о неправильном вводе данных

throw 1;

else

{

this->b = b;

}

}

catch (int thr)

{

exit(1);

this->b = 1;

}

}

void Fraction::setChisl(int a)

{

this->a = a;

}

void Fraction::setZnam(int b)

{

try

{

if (b == 0)

throw 1;

else

{

this->b = b;

}

}

catch (int thr)

{

exit(1);

this->b = 1;

}

}

**Заключение**

В процессе выполнения лабораторной работы №1, мы ознакомились с технологией построения классов и разработкой методов.

Автор Преподаватель: Тарасов

Шутенко Остап Русланович Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись \_\_\_\_\_\_\_ Дата: \_\_.\_\_.20\_\_