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»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

по дисциплине

«Программирование в системах информационной безопасности»

на тему

«Динамическое выделение памяти»

Вариант 26

Выполнил:

студент группы БСУ1801

Чигрина А.С.

Проверил:

старший преподаватель кафедры ИБ

Барков В.В.

Москва, 2021

# Цель работы

Научиться создавать простые классы, создавать конструкторы, перегружать операции, создавать объекты классов и передавать их в функции

# 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

**Общее Задание**

**Вариант 26.**

В соответствии с Вашим вариантом разработать класс. Название класса, приватные данные-члены и необходимые публичные функции-члены указаны в таблице ниже.

Для всех вариантов необходимо перегрузить операции + - \* / и операцию приведения типа к типу double.

Составить пользовательскую функцию y, указанную в индивидуальном задании и необходимые вспомогательные функции

|  |  |  |  |
| --- | --- | --- | --- |
| Варианты | 1-5 | 6-10 | 11-15 |
| Название класса | Complex | Complex | Rational |
| Приватные данные-члены | double re; (действительная часть)  double im;  (мнимая часть) | double r;  (модуль)  double phi;  (аргумент) | int nominator;  (числитель)  int denominator;  (знаменатель)  Функция сокращения дроби |
| Публичные функции-члены | double Re() const;  double Im() const;  double R() const;  double Phi() const; | | int GetNominator() const;  int GetDenominator() const; |
| Пользовательские функции | Complex y(const Complex &z); | Complex y(const Complex &z); | Rational y1(const Rational &x);  double y2(double x); |
| Вспомогательные функции | Complex sin(const Complex &z);  Complex cos(const Complex &z);  Complex ch(const Complex &z);  Complex sh(const Complex &z);  Complex exp(const Complex &z);  Complex pow(const Complex &z1, const Complex &z2); | Complex sin(const Complex &z);  Complex tg(const Complex &z);  Complex th(const Complex &z);  Complex sh(const Complex &z);  Complex pow(const Complex &z1, const Complex &z2); |  |

**Индивидуальное задание**

# Выполнение

Листинг 1 – программный код файла Rational.cpp

#include "Header.h"

Rational::Rational() :

m\_nominator{ 0 }, m\_denominator{ 1 }{}

Rational::Rational(int x) :

m\_nominator{ x }, m\_denominator{ 1 }{}

Rational::Rational(int x, int y) :

m\_nominator{ x }, m\_denominator{ y }{Normalize();}

int Rational::Nominator() const

{

return m\_nominator;

}

int Rational::Denominator() const

{

return m\_denominator;

}

void Rational::Normalize()

{

int devider = FindGreatestCommonDivisior(m\_nominator, m\_denominator);

if (m\_denominator < 0) { devider = -devider; };

m\_nominator /= devider;

m\_denominator /= devider;

}

Rational& Rational::operator+=(const Rational& right)

{

int lcm = FindLeastCommonMultiple(m\_denominator, right.m\_denominator);

m\_nominator = m\_nominator \* (lcm / m\_denominator) + right.m\_nominator \* (lcm / m\_denominator);

m\_denominator = lcm;

Normalize();

return \*this;

}

Rational& Rational::operator-=(const Rational& right)

{

int lcm = FindLeastCommonMultiple(m\_denominator, right.m\_denominator);

m\_nominator = m\_nominator \* (lcm / m\_denominator) - right.m\_nominator \* (lcm / m\_denominator);

m\_denominator = lcm;

Normalize();

return \*this;

}

Rational& Rational::operator\*=(const Rational& right)

{

m\_nominator \*= right.m\_nominator;

m\_denominator \*= right.m\_denominator;

Normalize();

return \*this;

}

Rational& Rational::operator/=(const Rational& right)

{

m\_nominator \*= right.m\_denominator;

m\_denominator \*= right.m\_nominator;

Normalize();

return \*this;

}

Rational Rational::operator+(const Rational& right)

{

Rational res = \*this;

return res += right;;

}

Rational Rational::operator-(const Rational& right)

{

Rational res = \*this;

return res -= right;;

}

Rational Rational::operator\*(const Rational& right)

{

Rational res = \*this;

return res \*= right;;

}

Rational Rational::operator/(const Rational& right)

{

Rational res = \*this;

return res /= right;;

}

ostream& operator<<(ostream& out, const Rational& Rational) //операция вывода

{

out << Rational.m\_nominator << "/" << Rational.m\_denominator;

return out;

}

istream& operator>>(istream& in, Rational& Rational) //операция ввода

{

in >> Rational.m\_nominator >> Rational.m\_denominator;

return in;

}

Листинг 2 – программный код файла Headher.h

#include "math.h"

#include<iostream>

using namespace std;

class Rational

{

public:

Rational();

Rational(int);

Rational(int, int);

int Nominator() const;

int Denominator() const;

explicit operator double()const

{

return m\_nominator /static\_cast<double>(m\_denominator);

}

Rational operator-() const;

Rational& operator+=(const Rational&);

Rational& operator-=(const Rational&);

Rational& operator\*=(const Rational&);

Rational& operator/=(const Rational&);

Rational operator+(const Rational&);

Rational operator-(const Rational&);

Rational operator\*(const Rational&);

Rational operator/(const Rational&);

friend std::ostream& operator<<(std::ostream& out, const Rational& Rational);

friend std::istream& operator>>(std::istream& in, Rational& Rational);

private:

int m\_nominator;

int m\_denominator;

void Normalize();

};

int FindLeastCommonMultiple(int x, int y);

int FindGreatestCommonDivisior(int x, int y);

Листинг 3 – программный код файла main.cpp

#include<iostream>

#include "math.h"

#include "Header.h"

using namespace std;

Rational y1(Rational& x)

{

Rational res = (x \* 2) + (Rational(1.3) / x);

return res;

}

double y2(double x)

{

return (x \* 2) + (1.3 / x);

}

int main()

{

setlocale(LC\_ALL, "RUS");

Rational a, res;

double x, resx;

cout << "Введите рациональное число" << endl;

cin >> a;

res = y1(a);

cout << res << endl;

cout << "Введите x" << endl;

cin >> x;

resx = y2(x);

cout << resx;

}

Листинг 4 – программный код файла mth.cpp

#include<iostream>

using namespace std;

int FindGreatestCommonDivisior(int x, int y)

{

int g = std::abs(y);

int a = max(std::abs(x), std::abs(y));

int b = min(std::abs(x), std::abs(y));

while (b !=0)

{

int r = a % b;

a = b;

b = r;

}

return a;

}

int FindLeastCommonMultiple(int x, int y)

{

return abs(x \* y) / FindGreatestCommonDivisior(x, y);

}

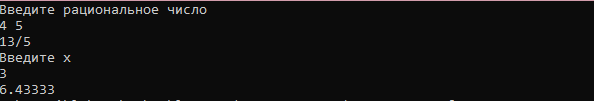


Рисунок 1 – Результат работы с программой.