|  |  |
| --- | --- |
| *voenmeh* | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение  высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**  **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
| БГТУ.СМК-Ф-4.2-К5-02 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Факультет |  | О |  | Естественнонаучный |
|  |  | шифр |  | наименование |
| Кафедра |  | О7 |  | Информационные системы и программная инженерия |
|  |  | шифр |  | наименование |
| Дисциплина |  | Информационные технологии и программирование | | |

|  |
| --- |
| Практическая работа №1 |
| Классы |
| Вариант №25 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | | | |  | И924Б |
| Фокин В. К. | | | | | |
| Фамилия И.О. | | | | | |
| **Преподаватель:** | | | | | |
| Гладевич А. А. | |  |  | | |
| Фамилия И.О. Подпись | | | | | |
| Оценка |  | | | |  |
| «\_\_\_\_\_» |  | | | | 2023г. |

САНКТ-ПЕТЕРБУРГ

2023 г.

**Цель работы**

Изучить понятие класса, механизмы работы с классами, научиться обеспечивать вызов методов и обращение к полям классов, изменять видимость компонентов в определении класса с использованием спецификаторов доступа, перегружать операции и использовать дружественные функции.

**Постановка задачи**

Описать класс в соответствии с индивидуальным вариантом задания и реализовать все его методы. Каждый класс должен содержать, помимо указанных в варианте методов, конструктор с параметрами, конструктор копирования, деструктор, методы ввода с клавиатуры, установки и получения значений полей, вывода этих значений на экран. В каждом методе класса, включая конструкторы и деструктор, предусмотреть отладочную печать сообщения, содержащего имя метода. Написать программу для тестирования всех методов класса, выбор метода должен осуществляться с помощью меню.

**Вариант 25**

**Класс** Рациональное число.

**Поля:** числитель, знаменатель.

**Методы:** возведение РЧ в целую степень.

**Операторы:** ^ как возведение вещественного числа в степень, заданную РЧ, деления / двух рациональных, сравнения != рационального числа с действительным, сложения += увеличения значения на величину другого рационального числа.

**Текст программы:**

#include <iostream>

#include <cmath>

using namespace std;

class ratio\_fraction

{

int numerator, denomirator;

double real;

public:

ratio\_fraction()

{

numerator = 0;

denomirator = 1;

real = 0;

cout << "Default constructor called" << endl;

}

ratio\_fraction(int num, int denom, double rel)

{

numerator = num;

denomirator = denom;

real = rel;

cout << "Constructor with parameters called" << endl;

}

ratio\_fraction(const ratio\_fraction& r)

{

numerator = r.numerator;

denomirator = r.denomirator;

real = r.real;

cout << "Copy constructor called" << endl;

}

~ratio\_fraction()

{

cout << "Destructor called" << endl;

}

void set\_real(double rel)

{

real = rel;

cout << "Real setter called" << endl;

}

double get\_real()

{

return real;

cout << "Real getter called" << endl;

}

void set\_num(int num)

{

numerator = num;

cout << "Numerator setter called" << endl;

}

int get\_num()

{

return numerator;

cout << "Numerator getter called" << endl;

}

void set\_denom(int denom)

{

do

{

if (denom == 0)

{

cout << "Denominator cannot be zero." << endl;

cout << "Denominator: ";

cin >> denom;

}

denomirator = denom;

}

while (denom == 0);

cout << "Denominator setter called" << endl;

}

int get\_denom()

{

return denomirator;

cout << "Denominator getter called" << endl;

}

void power()

{

int n, num1, denom1;

double rez;

cout << "Entere degree: ";

cin >> n;

num1 = pow(numerator, n);

denom1 = pow(denomirator, n);

rez = (double)num1 / denom1;

cout << "Original fraction: " << numerator << "/" << denomirator << endl;

cout << "Raised to the " << n << " degree" << endl;

cout << "Result: " << num1 << "/" << denom1 << " = " << rez << endl;

}

double to\_double()

{

if (denomirator == 0)

{

cout << "Denominator cannot be 0, so was changed to 1";

denomirator = 1;

}

return (double)numerator / denomirator;

}

friend ostream& operator << (ostream& out, const ratio\_fraction& r)

{

cout << r.numerator << "/" << r.denomirator;

return out;

}

friend ostream& operator < (ostream& out, const ratio\_fraction& r)

{

cout << r.real;

return out;

}

double operator^(const ratio\_fraction& r)

{

cout << "Operator ^" << endl;

return pow(real, r.numerator/r.denomirator);

}

double operator/(const ratio\_fraction& r)

{

int num2, denom2;

cout << "Operator /" << endl;

num2 = numerator \* r.denomirator;

denom2 = denomirator \* r.numerator;

return (double)num2 / denom2;

}

bool operator!=(const ratio\_fraction& r)

{

cout << "Operator !=" << endl;

if ((double)numerator/denomirator != r.real)

{

return false;

}

else

{

return true;

}

}

double operator+=(const ratio\_fraction& r)

{

int num3, denom3;

cout << "Operator +=" << endl;

if (denomirator == r.denomirator)

{

num3 = numerator + r.numerator;

cout << num3 << "/" << denomirator << " = ";

return num3 / denomirator;

}

else

{

denom3 = denomirator \* r.denomirator;

num3 = (numerator \* r.denomirator) + (r.numerator \* denomirator);

}

cout << num3 << "/" << denom3 << " = ";

return (double)num3 / denom3;

}

};

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

{

int x, y;

char menu;

cout << "-----------------------------------" << endl;

cout << "------ Checking constructors ------" << endl;

ratio\_fraction number1;

cout << "Number1: " << number1 << endl;

cout << "-----------------------------------" << endl;

ratio\_fraction number2(4, 9, 0);

cout << "Number2: " << number2 << endl;

cout << "-----------------------------------" << endl;

cout << "---------- from keyboard ----------" << endl;

cout << "Enter a rational fraction" << endl;

cout << "Numerator: ";

cin >> x;

do

{

cout << "Denominator: ";

cin >> y;

if (y == 0)

{

cout << "Denominator cannot be 0" << endl;

}

}

while (y == 0);

ratio\_fraction number3(x, y, 0);

cout << "Introduced number3: " << number3 << endl;

cout << "-----------------------------------" << endl;

ratio\_fraction number4(number3);

cout << "Number4: " << number4 << endl;

do

{

cout << "-----------------------------------" << endl;

cout << "-------------- MENU ---------------" << endl;

cout << "1. Raising a rational number to an integer power" << endl;

cout << "2. Raising a real number to a power given by a rational number (^)" << endl;

cout << "3. Division of two rational numbers (/)" << endl;

cout << "4. Comparison of rational and real numbers (!=)" << endl;

cout << "5. Addition of two rational numbers (+)" << endl;

cout << "0. Exit" << endl;

cout << "-----------------------------------" << endl;

cin >> menu;

switch (menu)

{

case '1':

{

system("cls");

cout << "---------- from keyboard ----------" << endl;

cout << "Entere a rational fraction: " << endl;

cout << "Numerator: ";

cin >> x;

number4.set\_num(x);

cout << "Denominator: ";

cin >> y;

number4.set\_denom(y);

cout << "Introduced number: " << number4 << " = " << number4.to\_double() << endl;

number4.power();

break;

}

case '2':

{

double rez, dbl;

system("cls");

cout << "---------- from keyboard ----------" << endl;

cout << "Enter a real number: ";

cin >> dbl;

ratio\_fraction real(0, 1, 0);

real.set\_real(dbl);

cout < real;

cout << "\nEnter a rational fraction: " << endl;

cout << "Numerator: ";

cin >> x;

number4.set\_num(x);

cout << "Denominator: ";

cin >> y;

number4.set\_denom(y);

cout << "Introduced number: " << number4 << " = " << number4.to\_double() << endl;

rez = real ^ number4;

cout << rez << endl;

break;

}

case '3':

{

system("cls");

double rez;

cout << "---------- from keyboard ----------" << endl;

cout << "Enter a first rational fraction: " << endl;

cout << "Numerator: ";

cin >> x;

number3.set\_num(x);

cout << "Denumirator: ";

cin >> y;

number3.set\_denom(y);

cout << "Introduced number: " << number3 << endl;

cout << "-----------------------------------" << endl;

cout << "Enter a second rational fraction: " << endl;

do

{

cout << "Numerator: ";

cin >> x;

if (x == 0)

{

cout << "The second fraction cannot be 0." << endl;

}

}

while (x == 0);

number4.set\_num(x);

cout << "Denumirator: ";

cin >> y;

number4.set\_denom(y);

cout << "Introduced number: " << number4 << endl;

cout << "-----------------------------------" << endl;

rez = number3 / number4;

cout << rez << endl;

break;

}

case '4':

{

double dbl;

system("cls");

cout << "---------- from keyboard ----------" << endl;

cout << "Enter a rational fraction: " << endl;

cout << "Numerator: ";

cin >> x;

number3.set\_num(x);

cout << "Denumirator: ";

cin >> y;

number3.set\_denom(y);

cout << "Introduced number: " << number3 << endl;

cout << "-----------------------------------" << endl;

cout << "Enter a real number: " << endl;

ratio\_fraction rel(0, 1, 0);

cin >> dbl;

rel.set\_real(dbl);

cout << "-----------------------------------" << endl;

cout << "First fraction == Real number?" << endl;

cout << number3.to\_double() << " == " < rel;

cout << "?" << endl;

cout << ((number3 != rel) ? "YES" : "NO") << endl;

break;

}

case '5':

{

system("cls");

double rez;

cout << "---------- from keyboard ----------" << endl;

cout << "Enter a first rational fraction: " << endl;

cout << "Numerator: ";

cin >> x;

number3.set\_num(x);

cout << "Denumirator: ";

cin >> y;

number3.set\_denom(y);

cout << "Introduced number: " << number3 << endl;

cout << "-----------------------------------" << endl;

cout << "Enter a second rational fraction: " << endl;

cout << "Numerator: ";

cin >> x;

number4.set\_num(x);

cout << "Denumirator: ";

cin >> y;

number4.set\_denom(y);

cout << "Introduced number: " << number4 << endl;

cout << "-----------------------------------" << endl;

rez = number3 += number4;

cout << rez << endl;

break;

}

case '0':

{

cout << "Exiting program..." << endl;

break;

}

default:

{

cout << "-----------------------------------" << endl;

cout << "Invalid choice. Try again." << endl;

}

}

}

while (menu != '0');

return 0;

}

**Результат работы программы:**

При включении программы запускаются стандартный конструктор и конструктор с параметрами, что представлено на рисунке 1.

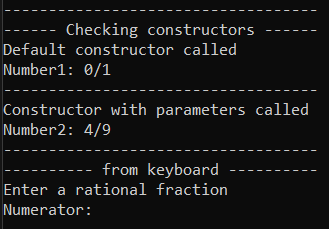


Рисунок 1 − Вызов конструкторов

Затем нужно ввести рациональную дробь, чтобы сработал конструктор копирования. После этого сразу появится меню, что предоставлено на рисунке 2.

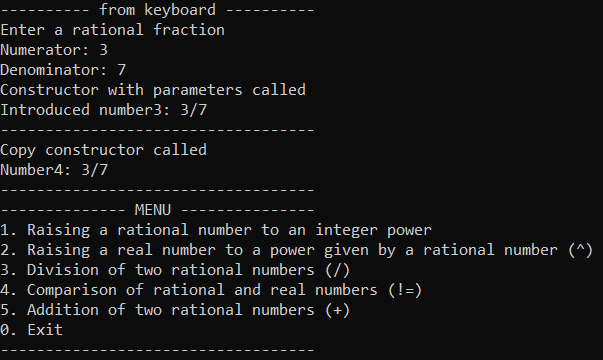


Рисунок 2 − Запуск конструктора копирования

Если в знаменатель дроби ввести «0», появится сообщение об ошибке, что предоставлено на рисунке 3.

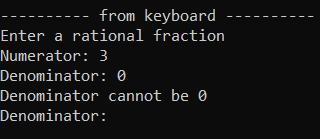


Рисунок 3 − Вывод ошибки при неподходящем значении

При нажатии на «1» можно возвести рациональную дробь в целую степень. Дробь и степень вводятся с клавиатуры, что предоставлено на рисунке 4.

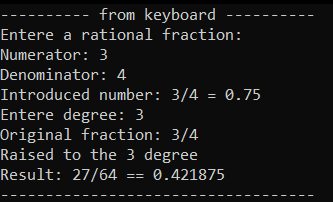


Рисунок 4 − Работа программы при выборе «1» в меню

Если в знаменатель дроби ввести «0», то появится сообщение об ошибке, что предоставлено на рисунке 5.

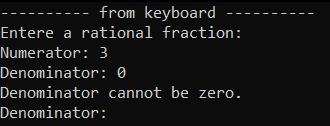


Рисунок 5 − Вывод ошибки при неподходящем значении

При нажатии на «2» можно возвести вещественное число в степень, заданную рациональной дробью. Вещественное число и рациональная дробь вводятся с клавиатуры, что предоставлено на рисунке 6.

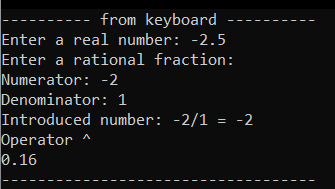


Рисунок 6 − Работа программы при выборе «2» в меню

Если в знаменатель дроби ввести «0», появится сообщение об ошибке, что предоставлено на рисунке 7.

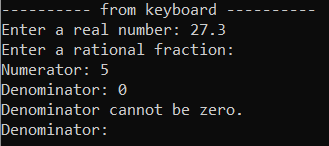


Рисунок 7 − Вывод ошибки при неподходящем значении

При нажатии на «3» разделить одну рациональную дробь на другую. Дроби вводятся с клавиатуры, что предоставлено на рисунке 8.

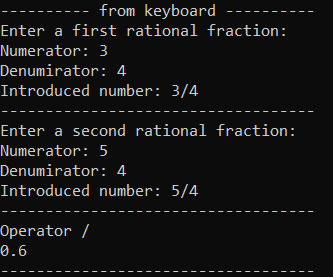


Рисунок 8 − Работа программы при выборе «3» в меню

Если в знаменатель любой дроби ввести «0», появится сообщение об ошибке, что предоставлено на рисунке 9.

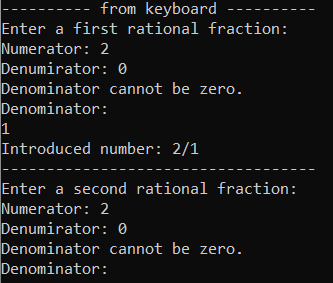


Рисунок 9 − Вывод ошибки при неподходящем значении

Если в числитель второй дроби ввести «0», появится сообщение об ошибке, что предоставлено на рисунке 10.

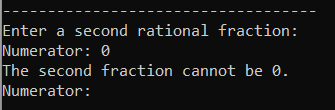


Рисунок 10 − Вывод ошибки при неподходящем значении

При нажатии на «4» можно сравнить рациональную дробь с вещественным числом. Дробь и вещественное число вводятся с клавиатуры, что предоставлено на рисунке 11.

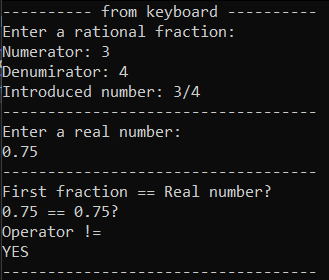


Рисунок 11 − Работа программы при выборе «4» в меню

Если в знаменатель дроби ввести «0», появится сообщение об ошибке, что предоставлено на рисунке 12.

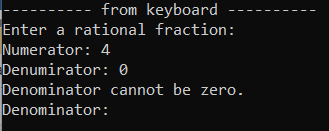


Рисунок 12 − Вывод ошибки при неподходящем значении

При нажатии на «5» можно сложить две рациональные дроби. Обе дроби вводятся с клавиатуры, что предоставлено на рисунке 13.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 13 − Работа программы при выборе «5» в меню

Если в знаменатель любой дроби ввести «0», появится сообщение об ошибке, что предоставлено на рисунке 14.

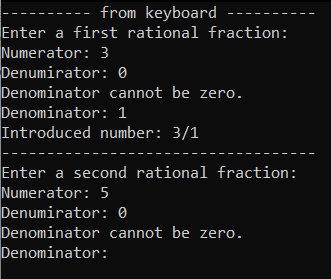


Рисунок 14 − Вывод ошибки при неподходящем значении

При нажатии на «0» программа прекращает работу и срабатывают деструкторы, что предоставлено на рисунке 15.

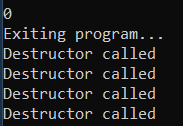


Рисунок 15 − Завершение работы программы