**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 1**

Тема: Простые классы на языке С++

Студент: Шахниязов Ботир

Группа: 80-201

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2019

1. Постановка задачи

Рациональная (несократимая) дробь представляется парой целых чисел (a, b), где a — числитель, b — знаменатель. Создать класс Rational для работы с рациональными дробями. Обязательно должны быть

реализованы операции:

сложения add, (a, b) + (c, d) = (ad + bc, bd);

вычитания sub, (a, b) – (c, d) = (ad – bc, bd);

умножения mul, (a, b) (c, d) = (ac, bd);

деления div, (a, b) / (c, d) = (ad, bc);

операции сравнения.

Должна быть реализована функция сокращения дроби reduce(), которая обязательно вызывается при

выполнении арифметических операций.

1. Описание программы

Ссылка на GitHub.com: <https://github.com/BShakhniyazov/oop-exercise-01>

Вариант №3.

Создал класс Rational для выполнения операций над дробями.

Сокращение дробей (функция reduce) реализовал с помощью алгоритма Евклида (пока у числителя и знаменателя есть делитель отличный от единицы выполняем алгоритм Евклида).

Для работы с дробями в программе сделал меню.

Операторы сделал с помощью перегрузки операторов.

1. Набор testcases

|  |  |  |
| --- | --- | --- |
| Набор 1.  Простой тест, для случая когда одна из дробей равна нулю.  2  0  10  3  6  12  4  5  6  7  8  9  10 | Набор 2.  Обе дроби отличны от нуля.  2  2  5  3  3  5  4  5  6  7  8  9  10 | Набор 3.  Попытка деления на ноль.  2  1  0 |

1. Результаты выполнения тестов.

|  |
| --- |
| **1**  Программа для работы с рациональными числами.  Числа a и b вводит пользователь, число c - результат применения оператора над a и b.  Menu  0. quit  1. print menu  2. input frac a  3. input frac b  4. add  5. sub  6. mul  7. div  8. equal  9. more  10. less  a=0/1 b=0/1 c=0/1  -->2  a.numerator = 0  a.denominator = 10  10  a=0/1 b=0/1 c=0/1  -->3  b.numerator = 6  b.denominator = 12  6  a=0/1 b=1/2 c=0/1  -->4  a=0/1 b=1/2 c=1/2  -->5  a=0/1 b=1/2 c=-1/2  -->6  a=0/1 b=1/2 c=0/1  -->7  a=0/1 b=1/2 c=0/1  -->8  a=0/1 b=1/2 c=0/1  -->9  a=0/1 b=1/2 c=0/1  -->10  a=0/1 b=1/2 c=1/1  -->0 |
| **2**  Программа для работы с рациональными числами.  Числа a и b вводит пользователь, число c - результат применения оператора над a и b.  Menu  0. quit  1. print menu  2. input frac a  3. input frac b  4. add  5. sub  6. mul  7. div  8. equal  9. more  10. less  a=0/1 b=0/1 c=0/1  -->2  a.numerator = 2  a.denominator = 5  a=2/5 b=0/1 c=0/1  -->3  b.numerator = 3  b.denominator = 5  a=2/5 b=3/5 c=0/1  -->4  a=2/5 b=3/5 c=1/1  -->5  a=2/5 b=3/5 c=-1/5  -->6  a=2/5 b=3/5 c=6/25  -->7  a=2/5 b=3/5 c=2/3  -->8  a=2/5 b=3/5 c=0/1  -->9  a=2/5 b=3/5 c=0/1  -->10  a=2/5 b=3/5 c=1/1  -->0 |
| **3**  Программа для работы с рациональными числами.  Числа a и b вводит пользователь, число c - результат применения оператора над a и b.  Menu  0. quit  1. print menu  2. input frac a  3. input frac b  4. add  5. sub  6. mul  7. div  8. equal  9. more  10. less  a=0/1 b=0/1 c=0/1  -->2  a.numerator = 1  a.denominator = 0  Введенный знаменатель равен нулю.  a=0/1 b=0/1 c=0/1  -->0 |

1. Листинг программы

#include <iostream>

/\*

Вариант 3. Шахниязов Ботир, 201 группа.

Рациональная (несократимая) дробь представляется парой целых чисел (a, b), где a — числитель, b —

знаменатель. Создать класс Rational для работы с рациональными дробями. Обязательно должны быть

реализованы операции:

сложения add, (a, b) + (c, d) = (ad + bc, bd);

вычитания sub, (a, b) – (c, d) = (ad – bc, bd);

умножения mul, (a, b) \* (c, d) = (ac, bd);

деления div, (a, b) / (c, d) = (ad, bc);

операции сравнения.

Должна быть реализована функция сокращения дроби reduce(), которая обязательно вызывается при

выполнении арифметических операций.

\*/

class Rational

{

private:

int numerator; //числитель

int denominator; //знаменатель

public:

Rational()

{

numerator = 0;

denominator = 1;

}

Rational(int \_numerator, int \_denominator)

{

set(\_numerator, \_denominator);

}

Rational operator+(Rational b)

{

Rational c(this->numerator \* b.denominator + this->denominator \* b.numerator, this->denominator \* b.denominator);

return c;

}

Rational operator-(Rational b)

{

Rational c(this->numerator \* b.denominator - this->denominator \* b.numerator, this->denominator \* b.denominator);

return c;

}

Rational operator\*(Rational b)

{

Rational c(this->numerator \* b.numerator, this->denominator \* b.denominator);

return c;

}

Rational operator/(Rational b)

{

if(b.numerator)

{

Rational c(this->numerator \* b.denominator, this->denominator \* b.numerator);

return c;

} else{

std::cout << "Деление на ноль.\n";

return Rational(0,1);

}

}

int operator==(Rational b)

{

return (this->numerator == b.numerator && this->denominator == b.numerator) ? 1 : 0;

}

int operator>(Rational b)

{

return (this->numerator \* b.denominator - this->denominator \* b.numerator > 0) ? 1 : 0;

}

int operator<(Rational b)

{

return (this->numerator \* b.denominator - this->denominator \* b.numerator > 0) ? 0 : 1;

}

int get\_num()

{

return this->numerator;

}

int get\_den()

{

return this->denominator;

}

void set(int \_numerator, int \_denominator)

{

numerator = \_numerator;

if( \_denominator)

denominator = \_denominator;

else

{

std::cout << "Введённый знаменатель равен нулю.\n";

numerator = 0;

denominator = 1;

}

}

int EuclideanAlgorithm(int a, int b) //алгоритм Евклида (поиск наибольшего общего кратного)

{

if (b == 0)

return a;

return EuclideanAlgorithm(b, a % b);

}

void reduce() //сокращение дроби

{

if(denominator == 1) return;

int NOK = EuclideanAlgorithm(numerator, denominator);

while(true)

{

if(NOK == 1) break; //если общих делителей больше нет

numerator /= NOK;

denominator /= NOK;

NOK = EuclideanAlgorithm(numerator, denominator);

}

if(denominator < 0) //если знаменатель отрицателен, то "отправляем" минус в числитель

{

numerator \*= -1;

denominator \*= -1;

}

}

};

void cout\_menu()

{

std::cout << "Menu\n";

std::cout << "0. quit\n";

std::cout << "1. print menu\n";

std::cout << "2. input frac a\n";

std::cout << "3. input frac b\n";

std::cout << "4. add\n";

std::cout << "5. sub\n";

std::cout << "6. mul\n";

std::cout << "7. div\n";

std::cout << "8. equal\n";

std::cout << "9. more\n";

std::cout << "10. less\n";

}

int main()

{

std::cout << "Программа для работы с рациональными числами.\n" <<

"Числа a и b вводит пользователь, число c - результат применения оператора над a и b." << std::endl;

Rational a, b, c;

int action=1, num, den;

while(action != 0)

{

switch (action)

{

case 1: cout\_menu(); break;

case 2: std::cout << "a.numerator = "; std::cin >> num;

std::cout << "a.denominator = "; std::cin >> den;

a.set(num, den);

break;

case 3:

std::cout << "b.numerator = "; std::cin >> num;

std::cout << "b.denominator = "; std::cin >> den;

b.set(num, den);

break;

case 4: c = a + b; break;

case 5: c = a - b; break;

case 6: c = a \* b; break;

case 7: c = a / b; break;

case 8: c.set(a == b, 1); break;

case 9: c.set(a > b, 1); break;

case 10: c.set(a < b, 1); break;

default: break;

}

a.reduce(); b.reduce(); c.reduce();

std::cout << "a=" << a.get\_num() << "/" << a.get\_den() << " ";

std::cout << "b=" << b.get\_num() << "/" << b.get\_den() << " ";

std::cout << "c=" << c.get\_num() << "/" << c.get\_den() << " \n-->";

std::cin >> action;

}

return 0;

}

Выводы

Я научился создавать примитивные классы на языке C++ а также решать простые задачи с использованием классов.

СПИСОК ЛИТЕРАТУРЫ

1. ООП с примерами (часть 1) [электронный ресурс]. URL: <https://habr.com/ru/post/87119/>
2. Объектно-ориентированное программирование. Определение классов. [электронный ресурс]. URL: <https://metanit.com/cpp/tutorial/5.1.php>
3. Классы в C++. [электронный ресурс]. URL: <http://cppstudio.com/post/439/>