**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 2**

Тема: Перегрузка операторов в С++

Студент: Мариничев Иван Александрович

Группа: 80-208

Преподаватель: Чернышов Л.Н.

Дата: 12.10.20

Оценка:

Москва, 2020

1. Постановка задачи. Вариант 4.

**Реализовать класс FazzyNumber** для работы с нечеткими числами, которые представляются тройками чисел (*x* – *e*l, *x*, *x* + *e*r). Для чисел *A* = (*A* – *a*l, *A*, *A* + *ar*) и *B* = (*B* – *b*l, *B*, *B* + *br*) арифметические операции выполняются по следующим формулам:

* сложение *A* + *B* = (*A* + *B* – *a*l – *b*l, *A* + *B*, *A* + *B* + *ar* + *br*);
* вычитание *A* – *B* = (*A* – *B* – *a*l – *b*l, *A* – *B*, *A* – *B* + *ar* + *br*);
* умножение *A* · *B* = (*A* · *B* – *B* · *a*l – *A* · *b*l + *a*l · *b*l, *A* · *B*, *A* · *B* + *B* · *a*l + *A* · *b*l + *a*l · *b*l);
* обратное число *A* = (1 / (*A* + *ar*), 1 / *A*, 1 / (*A* – *a*l)), *A* > 0;
* деление *A* / *B* = ((*A* – *a*l) / (*B* + *br*), *A* / *B*, (*A* + *ar*) / (*B* – *b*l)), *B* > 0;

Считать *e*l = *e*r, то есть число представлено парой <*x*, *e*>.

Реализовать операции сравнения по х.

Операции сложения, вычитания, умножения, деления, сравнения (на равенство, больше и меньше) должны быть выполнены в виде перегрузки операторов.

Необходимо реализовать пользовательский литерал для работы с константами типа **FazzyNumber**.

1. Описание программы

Программа предназначена для работы с нечеткими числами. Для работы с данными числами был реализован класс FazzyNumber, принимающий **x** и **e**.

В данной программе реализованы через перегрузку операторов операции сложения, вычитания, умножения и деления нечетких чисел.

Также данная программа может проверять равенство нечетких чисел, а также сравнивать их по x.

Для взаимодействия с программой реализованы два метода:

1. ввод из тестового файла,
2. ввод из консоли
3. Руководство по использованию программы.

Таблица 1 – Функции файла main.cpp

|  |  |  |
| --- | --- | --- |
| Название | Аргументы | Описание |
| int main() | string test\_name;  x\_a, e\_a;  x\_b, e\_b. | Основная функция с помощью которой происходит демонстрация всех возможностей программы |

Таблица 2 – Методы, функции и структуры файла FazzyNumber.h

|  |  |  |
| --- | --- | --- |
| Название | Аргументы | Описание |
| class FazzyNumber | double el;  double x;  double er; | Класс нечетких чисел |
| double getEl() |  | Метод для возвращения el |
| double getX() |  | Метод для возвращения x |
| double getEr() |  | Метод для возвращения er |

Таблица 3 – FazzyNumber.cpp

|  |  |  |
| --- | --- | --- |
| Название | Аргументы | Описание |
| FazzyNumber(); |  | Конструктор |
| FazzyNumber(); | double el, double x, double er | Конструктор по умолчанию |
| FazzyNumber(); | const FazzyNumber &other | Конструктор копирования |
| const FazzyNumber &operator=(); | const FazzyNumber &other | Оператор присваивания |
| FazzyNumber operator+(); | const FazzyNumber &fn1, const FazzyNumber &fn2 | Оператор сравнения + нечетких чисел |
| FazzyNumber operator-(); | const FazzyNumber &fn1, const FazzyNumber &fn2 | Оператор сравнения - нечетких чисел |
| FazzyNumber operator\*(); | const FazzyNumber &fn1, const FazzyNumber &fn2 | Оператор сравнения \* нечетких чисел |
| FazzyNumber operator/(); | const FazzyNumber &fn1, const FazzyNumber &fn2 | Оператор сравнения / нечетких чисел |
| bool operator==(); | const FazzyNumber &fn1, const FazzyNumber &fn2 | Оператор сравнения == нечетких чисел |
| bool operator!=(); | const FazzyNumber &fn1, const FazzyNumber &fn2 | Оператор сравнения != нечетких чисел |
| bool operator<(); | const FazzyNumber &fn1, const FazzyNumber &fn2 | Оператор сравнения < нечетких чисел |
| bool operator<=(); | const FazzyNumber &fn1, const FazzyNumber &fn2 | Оператор сравнения <= нечетких чисел |
| bool operator>(); | const FazzyNumber &fn1, const FazzyNumber &fn2 | Оператор сравнения > нечетких чисел |
| bool operator>=(); | const FazzyNumber &fn1, const FazzyNumber &fn2 | Оператор сравнения >= нечетких чисел |
| FazzyNumber operator"" \_fn(); | long double num | Пользовательский литерал для нечетких чисел |
| void Inverse(); |  | Метод вывода обратного нечеткого числа |
| void Display(); |  | Метод вывода нечетких чисел |

**Результаты выполнения тестов.**

|  |  |  |
| --- | --- | --- |
| Название тестового файла | Входные данные | Результат |
| test\_00.txt | -230.123 230.125  780.546 780.23567 | Creating 2 FazzyNumber class objects...  Type test file name (e.g. test\_01.txt) or type 'console' to enter them by yourself: test\_00.txt  FazzyNumber 1: (-460.248, -230.123, 0.002)  FazzyNumber 2: (0.31033, 780.546, 1560.78)  Sum of Fazzy Numbers: (-459.938, 550.423, 1560.78)  Difference of Fazzy Numbers: (-2021.03, -1010.67, -0.30833)  Product of Fazzy Numbers: (-142.829, -179622, 3.12156)  Quotient of Fazzy Numbers: (-0.294763, -0.294823, 0.00644475)  Let's find inversed Fazzy Numbers.  Inversed FazzyNumber1: There is no inverse number.  Inversed FazzyNumber2: (0.000640705, 0.00128115, 3.22238)  Comparing 2 Fazzy Numbers by x...  FazzyNumber1 is not equal to FazzyNumber2  FazzyNumber1 is lower than FazzyNumber2  And also we can use Fazzy Number literals. So let's create FazzyNumber3 using '3.14\_fn' literal.  FazzyNumber 3: (0, 3.14, 6.28) |
| test\_01.txt | 0 0  0 0 | Creating 2 FazzyNumber class objects...  Type test file name (e.g. test\_01.txt) or type 'console' to enter them by yourself: test\_01.txt  FazzyNumber 1: (0, 0, 0)  FazzyNumber 2: (0, 0, 0)  Sum of Fazzy Numbers: (0, 0, 0)  Difference of Fazzy Numbers: (0, 0, 0)  Product of Fazzy Numbers: (0, 0, 0)  Can't divide those Fazzy Numbers.  Let's find inversed Fazzy Numbers.  Inversed FazzyNumber1: There is no inverse number.  Inversed FazzyNumber2: There is no inverse number.  Comparing 2 Fazzy Numbers by x...  FazzyNumber1 is equal to FazzyNumber2  And also we can use Fazzy Number literals. So let's create FazzyNumber3 using '3.14\_fn' literal.  FazzyNumber 3: (0, 3.14, 6.28) |
| test\_02.txt | 330.125 1200  0.1242312321 3453543.657765 | Creating 2 FazzyNumber class objects...  Type test file name (e.g. test\_01.txt) or type 'console' to enter them by yourself: test\_02.txt  FazzyNumber 1: (-869.875, 330.125, 1530.12)  FazzyNumber 2: (-3.45354e+06, 0.124231, 3.45354e+06)  Sum of Fazzy Numbers: (-3.45441e+06, 330.249, 3.45507e+06)  Difference of Fazzy Numbers: (-3.45441e+06, 330.001, 3.45507e+06)  Product of Fazzy Numbers: (3.00415e+09, 41.0118, 5.28435e+09)  Quotient of Fazzy Numbers: (5314.69, 2657.34, -0.000443059)  Let's find inversed Fazzy Numbers.  Inversed FazzyNumber1: (0.000653541, 0.00302916, -0.00114959)  Inversed FazzyNumber2: (2.89558e-07, 8.04951, -2.89558e-07)  Comparing 2 Fazzy Numbers by x...  FazzyNumber1 is not equal to FazzyNumber2  FazzyNumber1 is greater than FazzyNumber2  And also we can use Fazzy Number literals. So let's create FazzyNumber3 using '3.14\_fn' literal.  FazzyNumber 3: (0, 3.14, 6.28) |
| test\_03.txt | -2 -3  -2 -3 | Creating 2 FazzyNumber class objects...  Type test file name (e.g. test\_01.txt) or type 'console' to enter them by yourself: test\_03.txt  FazzyNumber 1: (1, -2, -5)  FazzyNumber 2: (1, -2, -5)  Sum of Fazzy Numbers: (2, -4, -10)  Difference of Fazzy Numbers: (6, 0, -6)  Product of Fazzy Numbers: (1, 4, 25)  Can't divide those Fazzy Numbers.  Let's find inversed Fazzy Numbers.  Inversed FazzyNumber1: There is no inverse number.  Inversed FazzyNumber2: There is no inverse number.  Comparing 2 Fazzy Numbers by x...  FazzyNumber1 is equal to FazzyNumber2  And also we can use Fazzy Number literals. So let's create FazzyNumber3 using '3.14\_fn' literal.  FazzyNumber 3: (0, 3.14, 6.28) |
| test\_04.txt | Несуществующий файл | File is not exists |
| console |  | Creating 2 FazzyNumber class objects...  Type test file name (e.g. test\_01.txt) or type 'console' to enter them by yourself: console  Enter x and e parts of FazzyNumber 1: 30.34563 -5.10346  Enter x and e parts of FazzyNumber 2: 30.34563 -5.10346  FazzyNumber 1: (35.4491, 30.3456, 25.2422)  FazzyNumber 2: (35.4491, 30.3456, 25.2422)  Sum of Fazzy Numbers: (70.8982, 60.6913, 50.4843)  Difference of Fazzy Numbers: (10.2069, 0, -10.2069)  Product of Fazzy Numbers: (1256.64, 920.857, 637.167)  Quotient of Fazzy Numbers: (0.59564, 1, 0.712068)  Let's find inversed Fazzy Numbers.  Inversed FazzyNumber1: (0.0396162, 0.0329537, 0.0282095)  Inversed FazzyNumber2: (0.0396162, 0.0329537, 0.0282095)  Comparing 2 Fazzy Numbers by x...  FazzyNumber1 is equal to FazzyNumber2  Creating 2 FazzyNumber class objects...  Type test file name (e.g. test\_01.txt) or type 'console' to enter them by yourself: console  Enter x and e parts of FazzyNumber 1: 30.34563 -5.10346  Enter x and e parts of FazzyNumber 2: 30.34563 -5.10346  FazzyNumber 1: (35.4491, 30.3456, 25.2422)  FazzyNumber 2: (35.4491, 30.3456, 25.2422)  Sum of Fazzy Numbers: (70.8982, 60.6913, 50.4843)  Difference of Fazzy Numbers: (10.2069, 0, -10.2069)  Product of Fazzy Numbers: (1256.64, 920.857, 637.167)  Quotient of Fazzy Numbers: (0.59564, 1, 0.712068)  Let's find inversed Fazzy Numbers.  Inversed FazzyNumber1: (0.0396162, 0.0329537, 0.0282095)  Inversed FazzyNumber2: (0.0396162, 0.0329537, 0.0282095)  Comparing 2 Fazzy Numbers by x...  FazzyNumber1 is equal to FazzyNumber2  And also we can use Fazzy Number literals. So let's create FazzyNumber3 using '3.14\_fn' literal.  FazzyNumber 3: (0, 3.14, 6.28) |

1. Листинг программы

**main.cpp**

/\* Мариничев Иван М8О-208Б-19

\*

\* github: IvaMarin

\*

\* Вариант 4:

\*Реализовать класс FazzyNumber для работы с нечеткими числами, которые представляются тройками чисел (x – el, x, x + er). Для чисел A = (A – al, A, A + ar) и B = (B – bl, B, B + br) арифметические операции выполняются по следующим формулам:

\* -сложение A + B = (A + B – al – bl, A + B, A + B + ar + br);

\* -вычитание A – B = (A – B – al – bl, A – B, A – B + ar + br);

\* -умножение A · B = (A ´ B – B ´ al – A ´ bl + al ´ bl, A ´ B, A ´ B + B ´ al + A ´ bl + al ´ bl);

\* -обратное число A = (1 / (A + ar), 1 / A, 1 / (A – al)), A > 0;

\* -деление A / B = ((A – al) / (B + br), A / B, (A + ar) / (B – bl)), B > 0; Считать el = er, то есть число представлено парой <x, e>.

\*Реализовать операции сравнения по х.

\*Операции сложения, вычитания, умножения, деления, сравнения (на равенство, больше и меньше) должны быть выполнены в виде перегрузки операторов.

\*Необходимо реализовать пользовательский литерал для работы с константами типа FazzyNumber.

\*/

#include <iostream>

#include <fstream>

#include "FazzyNumber.h"

using namespace std;

int main () {

cout << "Creating 2 FazzyNumber class objects..." << endl;

string test\_name;

cout << "Type test file name (e.g. test\_01.txt) ";

cout << "or type 'console' to enter them by yourself: ";

cin >> test\_name;

double x\_a, e\_a, x\_b, e\_b;

if (test\_name != "console") {

ifstream input;

input.open(test\_name);

if (!input.is\_open()) {

cout << "File is not exists\n"; // если не открылся

return -1;

}

input >> x\_a >> e\_a >> x\_b >> e\_b;

}else {

cout << "Enter x and e parts of FazzyNumber 1: ";

cin >> x\_a >> e\_a;

cout << "Enter x and e parts of FazzyNumber 2: ";

cin >> x\_b >> e\_b;

}

cout << endl;

FazzyNumber fn1(e\_a, x\_a, e\_a);

FazzyNumber fn2(e\_b, x\_b, e\_b);

cout << "FazzyNumber 1: ";

fn1.Display();

cout << "FazzyNumber 2: ";

fn2.Display();

cout << endl;

FazzyNumber sum = fn1 + fn2;

FazzyNumber dif = fn1 - fn2;

FazzyNumber mul = fn1 \* fn2;

FazzyNumber div = fn1 / fn2 ;

cout << "Sum of Fazzy Numbers: ";

sum.Display();

cout << "Difference of Fazzy Numbers: ";

dif.Display();

cout << "Product of Fazzy Numbers: ";

mul.Display();

if (fn2.getX() > 0) {

cout << "Quotient of Fazzy Numbers: ";

div.Display();

} else {

cout << "Can't divide those Fazzy Numbers." << endl;

}

cout << endl;

cout << "Let's find inversed Fazzy Numbers." << endl;

cout << "Inversed FazzyNumber1: ";

fn1.Inverse();

cout << "Inversed FazzyNumber2: ";

fn2.Inverse();

cout << endl;

cout << "Comparing 2 Fazzy Numbers by x..." << endl;

if (fn1 != fn2) {

cout << "FazzyNumber1 is not equal to FazzyNumber2" << endl;

if (fn1 > fn2) {

cout << "FazzyNumber1 is greater than FazzyNumber2" << endl;

}

else {

cout << "FazzyNumber1 is lower than FazzyNumber2" << endl;

}

}

else {

cout << "FazzyNumber1 is equal to FazzyNumber2" << endl;

}

cout << endl;

cout << "And also we can use Fazzy Number literals. So let's create FazzyNumber3 using '3.14\_fn' literal." <<endl;

FazzyNumber fn3 = 3.14\_fn;

cout << "FazzyNumber 3: ";

fn3.Display();

}

**FazzyNumber.cpp**

#include "FazzyNumber.h"

FazzyNumber operator+(const FazzyNumber &fn1, const FazzyNumber &fn2) { //оператор сложения нечетких чисел

return FazzyNumber(fn1.getEl() + fn2.getEl(), fn1.getX() + fn2.getX(), fn1.getEr() + fn2.getEr());

}

FazzyNumber operator-(const FazzyNumber &fn1, const FazzyNumber &fn2) { //оператор вычитания нечетких чисел

return FazzyNumber(fn1.getEl() + fn2.getEl(), fn1.getX() - fn2.getX(), fn1.getEr() + fn2.getEr());

}

FazzyNumber operator\*(const FazzyNumber &fn1, const FazzyNumber &fn2) { //оператор умножения нечетких чисел

return FazzyNumber((fn2.getX() \* fn1.getEl() + fn1.getX() \* fn2.getEl() - fn1.getEl() \* fn2.getEl()), fn1.getX() \* fn2.getX(), (fn2.getX() \* fn1.getEl() + fn1.getX() \* fn2.getEl() + fn1.getEl() \* fn2.getEl()));

}

FazzyNumber operator/(const FazzyNumber &fn1, const FazzyNumber &fn2) { //оператор деления нечетких чисел

return FazzyNumber(((fn1.getX() - fn1.getEl()) / (fn2.getX() + fn2.getEr())) - (fn1.getX() / fn2.getX()), fn1.getX() / fn2.getX(),((fn1.getX() + fn1.getEr()) / (fn2.getX() - fn2.getEl())) - (fn1.getX() / fn2.getX()));

}

bool operator==(const FazzyNumber &fn1, const FazzyNumber &fn2) { //оператор сравнения == нечетких чисел

return (fn1.getX() == fn2.getX());

}

bool operator!=(const FazzyNumber &fn1, const FazzyNumber &fn2) { //оператор сравнения != нечетких чисел

return !(fn1 == fn2);

}

bool operator<(const FazzyNumber &fn1, const FazzyNumber &fn2) { //оператор сравнения < нечетких чисел

return (fn1.getX() < fn2.getX());

}

bool operator<=(const FazzyNumber &fn1, const FazzyNumber &fn2) { //оператор сравнения <= нечетких чисел

return (fn1 < fn2) || (fn1 == fn2);

}

bool operator>(const FazzyNumber &fn1, const FazzyNumber &fn2) { //оператор сравнения > нечетких чисел

return fn2 < fn1;

}

bool operator>=(const FazzyNumber &fn1, const FazzyNumber &fn2) { //оператор сравнения >= нечетких чисел

return fn2 <= fn1;

}

void FazzyNumber::Inverse() { //метод вывода обратного нечеткого числа

if (x <= 0) {

cout << "There is no inverse number." << endl;

}

else {

double LeftInverse = 1 / (x + er);

double MiddleInverse = 1 / x;

double RightInverse = 1 / (x - el);

cout << "(" << LeftInverse << ", " << MiddleInverse << ", " << RightInverse << ")" << endl;

}

}

void FazzyNumber::Display() { //метод вывода нечетких чисел

double Left = x - el;

double Middle = x;

double Right = x + er;

cout << "(" << Left << ", " << Middle << ", " << Right << ")" << endl;

}

FazzyNumber::FazzyNumber(): el(0), x(0), er(0) {} //Конструктор по умолчанию

FazzyNumber::FazzyNumber(double el, double x, double er): el(el), x(x), er(er) {} //Конструктор

FazzyNumber::FazzyNumber(const FazzyNumber &other) {

el = other.el;

x = other.x;

er = other.er;

} //конструктор копирования

const FazzyNumber &FazzyNumber::operator=(const FazzyNumber &other) {

el = other.el;

x = other.x;

er = other.er;

return \*this;

} //оператор присваивания

FazzyNumber operator"" \_fn(long double num) {

double x = num;

double e = num;

FazzyNumber fn(e, x, e);

return fn;

} //пользовательсктй литерал для нечетких чисел

**FazzyNumber.h**

#ifndef FAZZYNUMBER\_H\_

#define FAZZYNUMBER\_H\_

#include <iostream>

using namespace std;

class FazzyNumber //класс нечетких чисел

{

private:

double el;

double x;

double er;

public:

FazzyNumber(); //конструктор

FazzyNumber(double el, double x, double er); //конструктор по умолчанию

FazzyNumber(const FazzyNumber &other); //конструктор копирования

const FazzyNumber &operator=(const FazzyNumber &other); //оператор присваивания

double getEl() const {

return el;

}

double getX() const {

return x;

}

double getEr() const {

return er;

}

void Inverse();

void Display();

};

FazzyNumber operator+(const FazzyNumber &fn1, const FazzyNumber &fn2);

FazzyNumber operator-(const FazzyNumber &fn1, const FazzyNumber &fn2);

FazzyNumber operator\*(const FazzyNumber &fn1, const FazzyNumber &fn2);

FazzyNumber operator/(const FazzyNumber &fn1, const FazzyNumber &fn2);

bool operator==(const FazzyNumber &fn1, const FazzyNumber &fn2);

bool operator!=(const FazzyNumber &fn1, const FazzyNumber &fn2);

bool operator<(const FazzyNumber &fn1, const FazzyNumber &fn2);

bool operator<=(const FazzyNumber &fn1, const FazzyNumber &fn2);

bool operator>(const FazzyNumber &fn1, const FazzyNumber &fn2);

bool operator>=(const FazzyNumber &fn1, const FazzyNumber &fn2);

FazzyNumber operator"" \_fn(long double num);

#endif /\* FAZZYNUMBER\_H\_ \*/

**CmakeLists.txt**

cmake\_minimum\_required(VERSION 3.17)

project(oop\_lab2)

set(CMAKE\_CXX\_STANDARD 17)

file(COPY ${CMAKE\_CURRENT\_SOURCE\_DIR}/test\_00.txt

DESTINATION ${CMAKE\_CURRENT\_BINARY\_DIR})

file(COPY ${CMAKE\_CURRENT\_SOURCE\_DIR}/test\_01.txt

DESTINATION ${CMAKE\_CURRENT\_BINARY\_DIR})

file(COPY ${CMAKE\_CURRENT\_SOURCE\_DIR}/test\_02.txt

DESTINATION ${CMAKE\_CURRENT\_BINARY\_DIR})

file(COPY ${CMAKE\_CURRENT\_SOURCE\_DIR}/test\_03.txt

DESTINATION ${CMAKE\_CURRENT\_BINARY\_DIR})

file(COPY ${CMAKE\_CURRENT\_SOURCE\_DIR}/test\_04.txt

DESTINATION ${CMAKE\_CURRENT\_BINARY\_DIR})

add\_executable(oop\_exercise\_02 main.cpp FazzyNumber.cpp FazzyNumber.h)

1. Вывод

В ходе лабораторной работы я изучил механизмы перегрузки операторов, ознакомился механизмами работы с пользовательскими литералами. Кроме того, я закрепил знания работы с классами в С++.

1. Список литературы.
2. Стефан К. Дьюхэрст Скользкие места С++. Как избежать проблем при проектировании и компиляции ваших программ — ISBN: 5-94074-083-9 — 266 с.
3. Пользовательские литералы в C++11 [Электронный ресурс]. URL: https://habr.com/ru/post/140357/ (дата обращения: 11.10.2020).
4. Работа с файлами в С++ [Электронный ресурс]. URL: http://cppstudio.com/post/446/ (дата обращения: 11.10.2020).
5. Нечеткие числа и операции над ними [Электронный ресурс]. URL: https://intuit.ru/studies/courses/87/87/lecture/20511 (дата обращения: 11.10.2020).