**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: Курносов Максим Юрьевич

Группа: 80-208

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2020

1. Постановка задачи

**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* – *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 el = er, то есть число представлено парой <*x*, *e*>.

Реализовать операции сравнения по х.

.

.

1. Описание программы

Репазиторий GitHub: https://github.com/Arhangel333/oop\_exercise\_01

Пргограмма в файле main.cpp с помощью дружественных функций и методов класса из файла FazzyNumber.cpp , где и описан весь класс

FazzyNumber ,и перегружены все заданнве операции , а так же реализованы операции сравнения, работает с входными данными из файлов test\_01.txt test\_02.txt test\_03.txt и выводит результаты в стандартный вывод.

Для изменения формата тестирование на тестирование с вводом данных через стандартный ввод(через терминал) достаточно раскоментировать в файле main.cpp строку 39 и закоментировать (“//”)38 и 40.

В результате выполнения программы выводятся значения соответствующие выполнению всех перегруженных операций , а так же сами тестовые значения перед ними.

1. Набор тестов

test\_01.txt:

88 77 8 11

test\_02.txt:

9 3 9 16

test\_03.txt:

1 0 5 2

1. Результаты выполнения тестов

1 88 77 8 11

(8, 96, 184)

(-8, 80, 168)

(-33, 704, 3135)

(0, 0, 0)

(0, 11, -55)

false

true

false

false

true

2 9 3 9 16

(-1, 18, 37)

(-19, 0, 19)

(-42, 81, 300)

(0, 0, 0)

(0, 1, -1)

false

false

true

true

true

3 1 0 5 2

(4, 6, 8)

(-6, -4, -2)

(3, 5, 7)

(1, 1, 1)

(0, 0, 0)

false

true

false

false

true

1. Листинг программы

CMakeLists.txt:

all: oop\_exercise\_01

oop\_exercise\_01: main.o FazzyNumber.o

g++ main.o FazzyNumber.o -o oop\_exercise\_01

main.o: main.cpp

g++ -c main.cpp

FazzyNumber.o: FazzyNumber.cpp

g++ -c FazzyNumber.cpp

clean:

rm -rf \*.o oop\_exercise\_01

FazzyNumber.cpp:

#include <iostream>

using namespace std;

class FazzyNumber

{

private:

int x;

int e;

int l, c, r;

public:

FazzyNumber(int a, int b){

x = a;

e = b;

l = x - e;

c = x;

r = x + e;

}

friend int valx(FazzyNumber & n){

return n.x;

}

friend int vale(FazzyNumber & n){

return n.e;

};

friend int vall(FazzyNumber & n){

int l = n.l;

return l;

}

friend int valc(FazzyNumber & n){

int c = n.c;

return c;

};

friend int valr(FazzyNumber & n){

int r = n.r;

return r;

}

friend void show(FazzyNumber & n);

friend void rshow(FazzyNumber & n);

FazzyNumber operator + (const FazzyNumber & other){

FazzyNumber s(0, 0);

s.x = this->x + other.x;

s.e = this->e + other.e;

s.l = s.x - s.e;

s.c = s.x;

s.r = s.x + s.e;

return s;

}

FazzyNumber operator - (const FazzyNumber & other){

FazzyNumber s(0, 0);

s.x = this->x - other.x;

s.e = this->e + other.e; //в задании формула из которой следует , что е складывается при вычитании объектов FazzyNumber

s.l = this->x - other.x - (this->e + other.e);

s.c = s.x;

s.r = this->x - other.x + (this->e + other.e);

return s;

}

FazzyNumber operator \* (const FazzyNumber & other){

FazzyNumber s(0, 0);

s.x = this->x \* other.x;

s.e = this->x;

s.l = this->x \* other.x + this->e \* other.e - other.x \* this->e - this->x \* other.e;

s.c = this->x \* other.x;

s.r = this->x \* other.x + this->e \* other.e + other.x \* this->e + this->x \* other.e;

return s;

}

friend FazzyNumber reverse(const FazzyNumber & other);

FazzyNumber operator / (const FazzyNumber & other){

FazzyNumber s(0, 0);

s.x = this->x / other.x;

s.e = this->x;

s.l = (this->x - this->e)/(other.x + other.e);

s.c = this->x / other.x;

s.r = (this->x + this->e)/(other.x - other.e);

return s;

}

bool operator > (const FazzyNumber & other){

return this->x > other.x;

}

bool operator < (const FazzyNumber & other){

return this->x < other.x;

}

bool operator == (const FazzyNumber & other){

return this->x == other.x;

}

bool operator >= (const FazzyNumber & other){

return this->x >= other.x;

}

bool operator <= (const FazzyNumber & other){

return this->x <= other.x;

}

}

main.cpp:

#include<iostream>

#include<cstdio>

//using namespace std;

#include"FazzyNumber.cpp"

FazzyNumber reverse(const FazzyNumber & other){

FazzyNumber s(0, 0);

s.x = 1/other.x;

s.e = other.e;

s.l = 1/(other.x + other.e);

s.c = s.x;

s.r = 1/(other.x - other.e);

//printf("\*\*\*\*\*%d %d %d %d",other.x,5,other.e,1/(other.x - other.e));

return s;

}

void show(FazzyNumber & n)

{

int l, c, r;

l = vall(n);

c = valc(n);

r = valr(n);

printf("(%d, %d, %d)\n", l, c, r);

};

void rshow(FazzyNumber & n){

int x, e;

x = valx(n);

e = vale(n);

printf("(%d, %d)\n", x, e);

};

int main(){

int a,s,d,f;

FILE \* test1 = freopen("test\_01.txt", "r", stdin);

//scanf("%d %d\n %d %d", &a, &s, &d, &f);

fscanf(test1,"%d %d\n %d %d", &a, &s, &d, &f);

std::cout << " 1 " << a << " " << s << " " << d << " " << f << endl;

FazzyNumber one31(a, s), two31(d,f);

FazzyNumber three31 = one31 + two31;

show(three31);

three31 = one31 - two31;

show(three31);

three31 = one31 \* two31;

show(three31);

three31 = reverse(one31);

show(three31);

three31 = one31 / two31;

show(three31);

if(three31 > one31){

printf("true\n");

}else{printf("false\n");

}

if(three31 < one31){

printf("true\n");

}else{printf("false\n");

}

if(three31 == one31){

printf("true\n");

}else{printf("false\n");

}

if(three31 >= one31){

printf("true\n");

}else{printf("false\n");

}

if(three31 <= one31){

printf("true\n");

}else{printf("false\n");

}

FILE \* test2 = freopen("test\_02.txt", "r", stdin);

fscanf(test2,"%d %d\n %d %d", &a, &s, &d, &f);

std::cout << " 1 " << a << " " << s << " " << d << " " << f << endl;

FazzyNumber one32(a, s), two32(d,f);

FazzyNumber three32 = one32 + two32;

show(three32);

three32 = one32 - two32;

show(three32);

three32 = one32 \* two32;

show(three32);

three32 = reverse(one32);

show(three32);

three32 = one32 / two32;

show(three32);

if(two32 > one32){

printf("true\n");

}else{printf("false\n");

}

if(two32 < one32){

printf("true\n");

}else{printf("false\n");

}

if(two32 == one32){

printf("true\n");

}else{printf("false\n");

}

if(two32 >= one32){

printf("true\n");

}else{printf("false\n");

}

if(two32 <= one32){

printf("true\n");

}else{printf("false\n");

}

FILE \* test3 = freopen("test\_03.txt", "r", stdin);

fscanf(test3,"%d %d\n %d %d", &a, &s, &d, &f);

std::cout << " 1 " << a << " " << s << " " << d << " " << f << endl;

FazzyNumber one3(a, s), two3(d,f);

FazzyNumber three3 = one3 + two3;

show(three3);

three3 = one3 - two3;

show(three3);

three3 = one3 \* two3;

show(three3);

three3 = reverse(one3);

show(three3);

three3 = one3 / two3;

show(three3);

if(three3 > one3){

printf("true\n");

}else{printf("false\n");

}

if(three3 < one3){

printf("true\n");

}else{printf("false\n");

}

if(three3 == one3){

printf("true\n");

}else{printf("false\n");

}

if(three3 >= one3){

printf("true\n");

}else{printf("false\n");

}

if(three3 <= one3){

printf("true\n");

}else{printf("false\n");

}

}