Московский Авиационный Институт

(Национальный Исследовательский Университет)

**Лабораторная работа №2 по курсу**

**«Объектно-ориентированное программирование»**

Студент: Шаларь Игорь Павлович

Группа: М8О-208Б-20

Вариант: 23

Преподаватель: Дорохов Евгений Павлович

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

# Цель:

* Изучение основ работы с классами в С++;
* Перегрузка операций и создание литералов

# Требования к программе:

Разработать программу на языке C++ согласно варианту задания. Программа на C++ должна собираться с помощью системы сборки CMake. Программа должна получать данные из стандартного ввода и выводить данные в стандартный вывод.  
Реализовать над объектами реализовать в виде перегрузки операторов.  
Реализовать пользовательский литерал для работы с константами объектов созданного класса.

# Задание:

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:

сложениеA+B=(A+B–al –bl,A+B,A+B+ar +br);

вычитаниеA–B=(A–B–al –bl,A–B,A–B+ar +br);

умножениеA·B=(ÁB–B́al –Ábl +al ́bl,ÁB,ÁB+B́al +Ábl +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.

# Описание программы:

1)FazzyNumber.h -  заголовочный файл, описывающий класс.

2)FazzyNumber.cpp -  реализация класса.

3)main.cpp - основной файл, взаимодействие с пользователем, определение литералов.

4)CMakeLists.txt - указания для системы сборки CMake.

# Примеры работы:

**input:**

1 2 1 2

**output**:

A = <1, 2>

B = <1, 2>

Comparison by x: 1

(A == B) = 1

A + B = <2, 6>

A - B = <0, 4>

A \* B = <1, 9>

A / B = <1, -3>

Reversed A = <1, -1>

**input:**

1 0.5 1 2

**output:**

A = <1, 0.5>

B = <1, 2>

Comparison by x: 1

(A == B) = 0

A + B = <2, 4.5>

A - B = <0, 2.5>

A \* B = <1, 4.5>

A / B = <1, -1.5>

Reversed A = <1, 2>

# Дневник отладки:

Пытался определить литералы вне main.cpp, а такое возможно, только если определение будет внутри namespace.

# Выводы:

Освоил создание литералов.

# Исходный код:

**FazzyNumber.h**:

#ifndef HEXAGON\_H

#define HEXAGON\_H

#include<iostream>

#include<stdexcept>

class FazzyNumber{

public:

FazzyNumber() {this->x = 0, this->e = 0; };

FazzyNumber(double a, double b);

friend std::istream &operator >> (std::istream &is, FazzyNumber &p);

friend std::ostream &operator << (std::ostream &os, FazzyNumber &p);

FazzyNumber &operator = (const FazzyNumber &right);

bool operator == (const FazzyNumber &t);

FazzyNumber operator + (const FazzyNumber &t);

FazzyNumber operator - (const FazzyNumber &t);

FazzyNumber operator \* (const FazzyNumber &t);

FazzyNumber operator / (const FazzyNumber &t);

//reverse

FazzyNumber operator-() const;

//compare by x

bool operator |= (const FazzyNumber &t);

double x, e;

};

#endif

**FazzyNumber.cpp**:

#include"FazzyNumber.h"

#include<iostream>

using namespace std;

FazzyNumber::FazzyNumber(double a, double b){

this->x = a;

this->e = b;

}

istream &operator >> (istream &is, FazzyNumber &h){

cin >> h.x >> h.e;

return is;

}

ostream &operator << (ostream &os, FazzyNumber &h){

os << "<" << h.x << ", " << h.e << ">";

return os;

}

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

this->x = t.x;

this->e = t.e;

return \*this;

}

FazzyNumber FazzyNumber::operator + (const FazzyNumber &t) {

FazzyNumber c;

c.x = this->x + t.x;

c.e = this->x + t.x + this->e + t.e;

return c;

}

FazzyNumber FazzyNumber::operator - (const FazzyNumber &t){

FazzyNumber c;

c.x = this->x - t.x;

c.e = this->x - t.x + this->e + t.e;

return c;

}

FazzyNumber FazzyNumber::operator \* (const FazzyNumber &t) {

FazzyNumber c;

c.x = this->x \* t.x;

c.e = this->x \* t.x + t.x \* this->e + this->x \* t.e + this->e \* t.e;

return c;

}

FazzyNumber FazzyNumber::operator / (const FazzyNumber &t){

if (t.x <= 0){

throw invalid\_argument( "right.x must be > 0" );

exit(-1);

}

if (t.x == t.e){

throw invalid\_argument( "division by zero in /" );

exit(-1);

}

FazzyNumber c;

c.x = this->x / t.x;

c.e = (this->x + this->e) / (t.x - t.e);

return c;

}

bool FazzyNumber::operator == (const FazzyNumber &t){

return (this->x == t.x && this->e == t.e);

}

bool x\_comp (const FazzyNumber &a, const FazzyNumber &b){

return a.x == b.x;

}

FazzyNumber FazzyNumber::operator-() const {

if (this->x <= 0){

throw invalid\_argument( "x must be > 0" );

exit(-1);

}

if (this->x == this->e){

throw invalid\_argument( "division by zero in reverse" );

exit(-1);

}

FazzyNumber c;

c.x = 1 / this->x;

c.e = 1 / (this->x - this->e);

return c;

}

bool FazzyNumber::operator |= (const FazzyNumber &t){

return (this->x == t.x);

}

**main.cpp**:

#include<iostream>

#include"FazzyNumber.h"

using namespace std;

int main(){

/\*

//Literals example:

FazzyNumber q = 10\_x;

cout << q << endl;

FazzyNumber w = 10\_e;

cout << w << endl;

return 0;

\*/

FazzyNumber a, b, res;

cout << "Enter values: [x1] [e1] [x2] [e2]" << endl;

cout << "Example: 1 2 3 4" << endl;

cin >> a >> b;

cout << "A = " << a << endl;

cout << "B = " << b << endl;

cout << "Comparison by x: " << (a |= b) << endl;

cout << "(A == B) = " << (a == b) << endl;

res = a + b;

cout << "A + B = " << res << endl;

res = a - b;

cout << "A - B = " << res << endl;

res = a \* b;

cout << "A \* B = " << res << endl;

res = a / b;

cout << "A / B = " << res << endl;

a = -a;

cout << "Reversed A = " << a << endl;

return 0;

}

**CMakeLists.txt**:

cmake\_minimum\_required(VERSION 3.20)

set(CMAKE\_CXX\_STANDARD 14)

add\_executable(oop\_exercise\_01 main.cpp FazzyNumber.cpp)