

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Лабораторная работа №1 по курсу
«Объектно-ориентированное программирование»

Студент: Шаларь Игорь Павлович

Группа: М8О-208Б-20

Вариант: 23

Преподаватель: Дорохов Евгений Павлович

Оценка: _____

Дата: _____

Цель:

- Изучение системы сборки на языке C++, изучение систем контроля версии.
- Изучение основ работы с классами в C++;

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.

Необходимо настроить сборку лабораторной работы с помощью CMake. Собранная программа должна называться `oop_exercise_01` (в случае использования `Windows` `oop_exercise_01.exe`)

Необходимо зарегистрироваться на GitHub (если студент уже имеет регистрацию на GitHub то можно использовать ее) и создать репозиторий для задания лабораторной работы.

Преподавателю необходимо предъявить ссылку на публичный репозиторий на Github. Имя репозитория должно быть `https://github.com/login/oop_exercise_01`

Где `login` – логин, выбранный студентом для своего репозитория на Github. Репозиторий должен содержать файлы:

- `main.cpp` // файл с заданием работы
- `CMakeLists.txt` // файл с конфигурацией CMake
- `test_xx.txt` // файл с тестовыми данными. Где `xx` – номер тестового набора 01, 02, ... Тестовых наборов должно быть больше 1.
- `report.doc` // отчет о лабораторной работе

Задание:

Реализовать класс **FuzzyNumber** 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 $A \cdot B=(\acute{A}B-\acute{B}al-\acute{A}bl+a'l'bl, \acute{A}B, \acute{A}B+\acute{B}al+\acute{A}bl+a'l'bl)$;

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 $\langle x, e \rangle$. Реализовать операции сравнения по x .

Описание программы:

- 1)FazzyNumber.h - заголовочный файл, описывающий класс.
- 2)FazzyNumber.cpp - реализация класса.
- 3)main.cpp - основной файл, взаимодействие с пользователем.
- 4)CMakeLists.txt - указания для системы сборки CMake.

Примеры работы:

input:

1 2 1 2

output:

$A = \langle 1, 2 \rangle$

$B = \langle 1, 2 \rangle$

Comparison by x : 1

$(A == B) = 1$

$A + B = \langle 2, 6 \rangle$

$A - B = \langle 0, 4 \rangle$

$A * B = \langle 1, 9 \rangle$

$A / B = \langle 1, -3 \rangle$

Reversed $A = \langle 1, -1 \rangle$

input:

1 0.5 1 2

output:

$A = \langle 1, 0.5 \rangle$

$B = \langle 1, 2 \rangle$

Comparison by x: 1

$(A == B) = 0$

$A + B = \langle 2, 4.5 \rangle$

$A - B = \langle 0, 2.5 \rangle$

$A * B = \langle 1, 4.5 \rangle$

$A / B = \langle 1, -1.5 \rangle$

Reversed A = $\langle 1, 2 \rangle$

Дневник отладки:

Операторы изменяли значение объекта класса, а не создавали новый объект, что приводило к ошибочному изменению значения. Ошибка была исправлена.

Выводы:

Познакомился с основами работы с классами, с системой сборки CMake, с системой контроля версий Github.

Исходный код:

FazzyNumber.h:

```
#ifndef HEXAGON_H
#define HEXAGON_H
#include<iostream>
#include<stdexcept>

class FazyNumber{

public:
    FazyNumber() {this->x = 0, this->e = 0; };
    FazyNumber(double a, double b);
    friend std::istream &operator >> (std::istream &is, FazyNumber &p);
    friend std::ostream &operator << (std::ostream &os, FazyNumber &p);
    FazyNumber &operator = (const FazyNumber &right);
    bool operator == (const FazyNumber &t);
    FazyNumber operator + (const FazyNumber &t);
    FazyNumber operator - (const FazyNumber &t);
    FazyNumber operator * (const FazyNumber &t);
    FazyNumber operator / (const FazyNumber &t);
    //reverse
    FazyNumber operator-() const;
    //compare by x
    bool operator |= (const FazyNumber &t);

    double x, e;

};

#endif
```

FazyNumber.cpp:

```
#include"FazyNumber.h"
#include<iostream>

using namespace std;

FazyNumber::FazyNumber(double a, double b){
    this->x = a;
    this->e = b;
}

istream &operator >> (istream &is, FazyNumber &h){
    cin >> h.x >> h.e;
    return is;
}

ostream &operator << (ostream &os, FazyNumber &h){
    os << "<" << h.x << ", " << h.e << ">";
    return os;
}
```

```
FuzzyNumber & FuzzyNumber::operator = (const FuzzyNumber &t){
    this->x = t.x;
    this->e = t.e;
    return *this;
}
```

```
FuzzyNumber FuzzyNumber::operator + (const FuzzyNumber &t) {
    FuzzyNumber c;
    c.x = this->x + t.x;
    c.e = this->x + t.x + this->e + t.e;
    return c;
}
```

```
FuzzyNumber FuzzyNumber::operator - (const FuzzyNumber &t){
    FuzzyNumber c;
    c.x = this->x - t.x;
    c.e = this->x - t.x + this->e + t.e;
    return c;
}
```

```
FuzzyNumber FuzzyNumber::operator * (const FuzzyNumber &t) {
    FuzzyNumber 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;
}
```

```
FuzzyNumber FuzzyNumber::operator / (const FuzzyNumber &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);
    }
    FuzzyNumber c;
    c.x = this->x / t.x;
    c.e = (this->x + this->e) / (t.x - t.e);
    return c;
}
```

```
bool FuzzyNumber::operator == (const FuzzyNumber &t){
    return (this->x == t.x && this->e == t.e);
}
```

```
bool x_comp (const FuzzyNumber &a, const FuzzyNumber &b){
    return a.x == b.x;
}
```

```
FuzzyNumber FuzzyNumber::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(){
    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)

```