

Московский Авиационный Институт
(Национальный исследовательский Университет)

Факультет: «Информационные технологии и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»

**Лабораторная работа
по курсу «ООП»**

**Тема:
Простые классы.**

Студент:	Черненко И.Д
Группа:	М80-206Б-18
Преподаватель:	Журавлев А.А.
Вариант:	4
Оценка:	
Дата:	

Москва
2019

1. Код программы на языке C++:

Fazzy_Number.hpp:

```
#ifndef LAB1_FAZZYNUMBER_HPP
#define LAB1_FAZZYNUMBER_HPP

class fn {
public:
    fn();
    fn(double a, double b);
    void print_value();
    static fn sum(const fn &a, const fn &b);
    static fn difference(const fn &a, const fn &b);
    static fn mult(const fn &a, const fn &b);
    static fn division(const fn &a, const fn &b);
    static fn inv(const fn &a);
    static int comparison(const fn &a, const fn &b);
private:
    double array[2];
};
#endif //LAB1_FAZZYNUMBER_HPP
```

Fazzy_Number.cpp:

```
#include <iostream>
#include <cassert>
#include "FazzyNumber.hpp"
#include <algorithm>
fn::fn():array{0,0} {};
fn::fn(double a, double b): array{a, b} {};
void fn::print_value() {
    std::cout<<array[0]<<" "<<array[1]<<std::endl;
};
fn fn::sum(const fn &a, const fn &b) {
    fn result;
    result.array[0] = a.array[0] + b.array[0];
    result.array[1] = a.array[1] + b.array[1];
    return result;
};
fn fn::difference(const fn &a, const fn &b) {
    fn result;
    result.array[0] = a.array[0] - b.array[1];
    result.array[1] = a.array[1] - b.array[0];
    return result;
};
fn fn::mult(const fn &a, const fn &b) {
```

```

fn result;
double t, m, p, k;
t = a.array[0] * a.array[1];
m = a.array[0] * b.array[1];
p = a.array[1] * b.array[0];
k = a.array[0] * b.array[1];
double maximum = std::max(std::max(std::max(t,m), p), k);
double minimum = std::min(std::min(std::min(t,m), p), k);
result.array[0] = minimum;
result.array[1] = maximum;
return result;
};

fn fn::division(const fn &a, const fn &b) {
    assert(b.array[0] * b.array[1]);
    /*result.array[0] = a.array[0] / b.array[1];
    result.array[1] = a.array[1] / b.array[0];*/
    fn result;
    double t, m, p, k;
    t = a.array[0] / a.array[1];
    m = a.array[0] / b.array[1];
    p = a.array[1] / b.array[0];
    k = a.array[0] / b.array[1];
    double maximum = std::max(std::max(std::max(t,m), p), k);
    double minimum = std::min(std::min(std::min(t,m), p), k);
    result.array[0] = minimum;
    result.array[1] = maximum;
    return result;
}

fn fn::inv(const fn &a) {
    fn result;
    assert(a.array[1] * a.array[0]);
    result.array[0] = 1 / a.array[1];
    result.array[1] = 1 / a.array[0];
    return result;
}

int fn::comparison(const fn &a, const fn &b) {
    if (((a.array[0]+a.array[1])/2) > ((b.array[0]+b.array[1])/2)) {
        return 1;
    } else if (((a.array[0]+a.array[1])/2) < ((b.array[0]+b.array[1])/2)) {
        return -1;
    } else {
        return 0;
    }
}

```

CmakeLists.txt:

```
project(lab1)
cmake_minimum_required(VERSION 3.2)
add_executable(lab1 lab1.cpp FuzzyNumber.cpp FuzzyNumber.hpp)
set_property(TARGET lab1 PROPERTY CXX_STANDARD 11)
```

2. Ссылка на репозиторий на GitHub

https://github.com/IlCher/oop_exercise_01

3. Набор testcases.

test_00:
-100.123 100.125
900.534 990.12333

test_01:
0.12976 -99.654
-883.3123 10.2123

test_02:
2 -33.125
0.1242312321 90000.324234

test_03:
1200 -330.125
0.1242312321 3453543.657765

test_04:

500 -12
-12 500

4. Результаты выполнения тестов.

test_00:

800.411 1090.25
-1090.25 -800.409
-90164.2 99136.1
-0.101122 0.111184
0.00998752 -0.00998772
the second operand is bigger than first

test_01:

-883.183 -89.4417
-10.0825 783.658
-114.619 -1017.7
0.0127062 0.112819
-0.0100347 7.70654
the first operand is bigger than second

test_02:

2.12423 89967.2
-89998.3 -33.2492
0.248462 -2.98126e+006
2.22221e-005 -266.64
-0.0301887 0.5
the second operand is bigger than first

test_03:

1200.12 3.45321e+006
-3.45234e+006 -330.249
149.077 -1.1401e+009
0.000347469 -2657.34
-0.00302916 0.000833333
the second operand is bigger than first

test_04

488 488
0 0
-6000 -6000
1 1
-0.0833333 0.002
operands are equivalent

5. Объяснение результатов работы программы.

- 1) При запуске программы с аргументом test_?.txt объекты a, b в основной программе получают данные из файлов test_?.txt.
- 2) Вывод данных объектов a, b в стандартный поток вывода.
- 3) Объекты a и b складываются с помощью функции-члена sum() класса fn, и результат выводится в стандартный поток вывода с помощью функции print_value() класса fn.

- 4) Из объекта `a` вычитается объект `b` с помощью функции-члена `difference()` класса `fn`, и результат выводится в стандартный поток вывода с помощью функции `print_value()` класса `fn`.
- 5) Объекты `a` и `b` умножаются с помощью функции-члена `mult()` класса `fn`, и выводится в стандартный поток вывода с помощью функции `print_value()` класса `fn`.
- 6) Объект `a` делится на `b` с помощью функции-члена `division()` класса `fn`, и выводится в стандартный поток вывода с помощью функции `print_value()` класса `fn`.
- 7) Объекты `a` и `b` проверяются на равенство (`a` также на то, что $a < b$, $a > b$) с помощью функции-члена `comparison()` класса `fn`, и результат выводится в стандартный поток вывода.
- 8) Объект `a` инвертируется с помощью функции-члена `inv()` класса `fn`, и результат выводится в стандартный поток вывода с помощью функции `print_value()` класса `fn`.

6. Вывод.

Выполняя данную лабораторную я получил опыт работы с простыми классами, с системой сборки `Cmake`, с системой контроля версий `GitHub`, а также изучил основы работы с классами в `C++`. Создал класс, соответствующий варианту моего задания, реализовал для него арифметические операции сложения, вычитания, умножения, деления, инвертирования, а также сравнения.