

Московский Авиационный Институт  
(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики  
Кафедра вычислительной математики и программирования

**Лабораторная работа**  
**по курсу «Объектно-ориентированное программирование»**  
**III Семестр**

**Задание 1**  
**Вариант 4**  
**Простые классы**

Студент:	Синявский А.В
Группа:	М80-208Б-18
Преподаватель:	Журавлёв А.А
Оценка:	
Дата:	

# 1. Код программы на языке C++

## 1.1 main.cpp

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

int main()
{
    double tmp_l, tmp_r;

    std::cout << "Ввод левой и правой границ отрезка А:" << '\n';
    std::cin >> tmp_l;
    std::cin >> tmp_r;
    Fazzy A(tmp_l, tmp_r);

    std::cout << "Ввод левой и правой границ отрезка Б:" << '\n';
    std::cin >> tmp_l;
    std::cin >> tmp_r;
    Fazzy B(tmp_l, tmp_r);

    Fazzy res1;
    res1 = A.plus(B);
    std::cout << "A+B: ";
    res1.write(std::cout);

    Fazzy res2;
    res2 = A.minus(B);
    std::cout << "A-B: ";
    res2.write(std::cout);

    Fazzy res3;
    res3 = A.mult(B);
    std::cout << "A*B: ";
    res3.write(std::cout);

    Fazzy res4;
    res4 = A.div(B);
    std::cout << "A/B (если возможно): ";
    res4.write(std::cout);

    Fazzy res5;
    res5 = A.rev();
    std::cout << "A^-1 (если возможно): ";
    res5.write(std::cout);

    bool res6;
    res6 = A.is_more_then(B);
    std::cout << std::boolalpha;
    std::cout << "A > B (?): " << res6 << '\n';
    return 0;
}
```

## 1.2 FuzzyNumber.cpp

```
#include <iostream>
#include <cassert>
#include "FuzzyNumber.h"
```

```
Fuzzy::Fuzzy(double l, double r)
{
    if (l <= r) {
        this->l = l;
        this->r = r;
    } else {
        this->l = r;
        this->r = l;
    }
}
```

```
Fuzzy::Fuzzy()
{
    this->l = 0;
    this->r = 0;
}
```

```
Fuzzy Fuzzy::plus(const Fuzzy& A)
{
    Fuzzy res;
    res.l = this->l + A.l;
    res.r = this->r + A.r;
    return res;
}
```

```
Fuzzy Fuzzy::minus(const Fuzzy& A)
{
    Fuzzy res;
    res.l = this->l - A.r;
    res.r = this->r - A.l;
    return res;
}
```

```
Fuzzy Fuzzy::mult(const Fuzzy& A)
{
    Fuzzy res;
    double max, min, tmp[4];
    tmp[0] = this->l * A.l;
    tmp[1] = this->l * A.r;
    tmp[2] = this->r * A.l;
    tmp[3] = this->r * A.r;
    max = tmp[0];
    min = tmp[3];
    for (int i = 1; i < 4; i++)
    {
```

```

        if (tmp[i] > max) max = tmp[i];
        if (tmp[3-i] < min) min = tmp[3-i];
    }
    res.l = min;
    res.r = max;
    return res;
}

```

Fazzy Fazzy::div(const Fazzy& A)

```

{
    if (A.l * A.r > 0) {
        Fazzy res;
        double max, min, tmp[4];
        tmp[0] = this->l / A.l;
        tmp[1] = this->l / A.r;
        tmp[2] = this->r / A.l;
        tmp[3] = this->r / A.r;
        max = tmp[0];
        min = tmp[3];
        for (int i = 1; i < 4; i++)
        {
            if (tmp[i] > max) max = tmp[i];
            if (tmp[3-i] < min) min = tmp[3-i];
        }
        res.l = min;
        res.r = max;
        return res;
    } else assert(0);
}

```

Fazzy Fazzy::rev()

```

{
    if (this->l * this->r > 0)
    {
        Fazzy res;
        res.l = 1/this->r;
        res.r = 1/this->l;
        return res;
    } else assert(0);
}

```

bool Fazzy::is\_more\_then(const Fazzy& A)

```

{
    double mid1, mid2;
    mid1 = this->l + (this->r - this->l) / 2;
    mid2 = A.l + (A.r - A.l) / 2;
    return mid1 > mid2;
}

```

void Fazzy::write(std::ostream& os) const

```

{
    os << "[" << this->l << "; " << this->r << "]" << '\n';
}

```

}

## 2. Ссылка на репозиторий на GitHub

[https://github.com/Siegmeyer1/oop\\_exercise\\_01](https://github.com/Siegmeyer1/oop_exercise_01)

## 3. Набор тестов

1) 0  
2  
4  
8

2) 3  
4  
0  
9

3) 9.5  
13  
1.4  
7.8

4) 2  
1  
6  
8

## 4. Результат выполнения тестов

1)  
A+B: [4; 10]  
A-B: [-8; -2]  
A\*B: [0; 16]  
A/B (если возможно): [0; 0.5]  
A<sup>-1</sup> (если возможно): [0; 2]  
A > B (?): false

2)  
A+B: [3; 13]

A-B: [-6; 4]  
A\*B: [0; 36]  
A/B (если возможно): [3; 4]  
A<sup>-1</sup> (если возможно): [0.25; 0.333333]  
A > B (?): false

3)  
A+B: [10.9; 20.8]  
A-B: [1.7; 11.6]  
A\*B: [13.3; 101.4]  
A/B (если возможно): [1.21795; 9.28571]  
A<sup>-1</sup> (если возможно): [0.0769231; 0.105263]  
A > B (?): true

4)  
A+B: [7; 10]  
A-B: [-7; -4]  
A\*B: [6; 16]  
A/B (если возможно): [0.125; 0.333333]  
A<sup>-1</sup> (если возможно): [0.5; 1]  
A > B (?): false

## 5. Объяснение работы программы

Данная программа состоит из двух сpp файлов.

Первый - FuzzyNumber, в котором создаётся одноимённый класс для работы с отрезками. Отрезок представлен двумя полями типа double – левым и правым концами отрезка. При помощи методов класса реализованы: сложение, вычитание, умножение и деление (при возможности) отрезков, а также операция обращения, возвращающая отрезок, обратный данному и операция сравнения двух отрезков, определяющая, больше ли первый отрезок, чем второй.

Второй файл – main, запрашивает ввод границ отрезков и проделывает с ними все заданные операции. Конструктор при вызове определяет, в правильном ли порядке были введены границы отрезка, и, в случае ошибки, меняет их местами. Вывод осуществляется при помощи метода write, определённого в первом файле.

## Вывод

Проделав данную работу, я узнал, что такое классы в C++ и изучил базовые принципы их написания на практике.