

**Московский авиационный институт  
(национальный исследовательский университет)**

**Факультет информационных технологий и прикладной математики**

**Кафедра вычислительной математики и программирования**

**Лабораторная работа №1 по курсу «ООП»**

Студент: Э. Л. Носов  
Преподаватель:  
Группа: М8О-308Б-21  
Дата:  
Оценка:  
Подпись:

**Москва, 2023**

# Лабораторная работа №1

## Цель:

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

**Вариант №4:** Реализовать класс `FuzzyNumber` для работы с нечеткими числами, которые представляются тройками чисел  $(x - \epsilon_1, x, x + \epsilon_2)$ .

# 1 Описание

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  $A \cdot B = (A \cdot B - B \cdot al - A \cdot bl + al \cdot bl, A \cdot B, A \cdot B + B \cdot al + A \cdot bl + al \cdot bl)$ ;
- 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  $e_l = e_r$ , то есть число представлено парой  $\langle x, e \rangle$ . Реализовать операции сравнения по  $x$ .

## 2 Исходный код

В начале необходимо определить класс FuzzyNumber. Этот класс будет содержать публичные поля `el`, `x` и `er` типа `float`, а также конструкторы по умолчанию и с параметрами. Также внутри класса определяется метод для нахождения обратного значения для объекта нашего класса. Так как аллокация памяти не производится необходимости в деструкторе нет. За пределом класса производится перегрузка необходимых операторов.

В функции `main` происходит запрос двух нечетких чисел в виде пары `<x, e>`. Ввод производится с помощью перегруженного оператора `>`. На вводе каждому объекту класса `FuzzyNumber` ожидаются два числа типа `float`, разделенные запятой. После следует демонстрация работы перегруженных операторов, демонстрируются изменения введенных данных и производится сравнение. Вывод производится при помощи перегруженного оператора `<`.

```
1  #include <iostream>
2
3  class FuzzyNumber{
4      public:
5
6          float el, x, er;
7
8          FuzzyNumber(){
9              x=0.0;
10             el=0.0;
11             er=0.0;
12         }
13
14         FuzzyNumber(float a, float b, float c){
15             el=a;
16             x=b;
17             er=c;
18         }
19
20         FuzzyNumber oposite(){
21             FuzzyNumber tmp;
22             tmp.x=1.0/x;
23             tmp.el=1/er;
24             tmp.er=1/el;
25             return tmp;
26         }
27     };
28
29     FuzzyNumber operator+(FuzzyNumber a, FuzzyNumber b){
30         FuzzyNumber c;
31         c.el=a.el+b.el;
32         c.er=a.er+b.er;
33         c.x=a.x+b.x;
34         return c;
35     }
36
37     FuzzyNumber operator-(FuzzyNumber a, FuzzyNumber b){
38         FuzzyNumber c;
39         c.el=a.el-b.el;
40         c.er=a.er-b.er;
41         c.x=a.x-b.x;
42         return c;
43     }
44
```

```

45 FuzzyNumber operator*(FuzzyNumber a, FuzzyNumber b){
46     FuzzyNumber c;
47     c.x=a.x*b.x;
48     c.el=a.el*b.el;
49     c.er=a.er*b.er;
50     return c;
51 }
52
53 FuzzyNumber operator/(FuzzyNumber a, FuzzyNumber b){
54     return a*b.opposite();
55 }
56
57 bool operator < (FuzzyNumber a, FuzzyNumber b){
58     return (a.x<b.x);
59 }
60
61 bool operator > (FuzzyNumber a, FuzzyNumber b){
62     return (a.x>b.x);
63 }
64
65 bool operator ==(FuzzyNumber a, FuzzyNumber b){
66     return (a.x==b.x);
67 }
68
69
70 std::ostream& operator <<(std::ostream& out, const FuzzyNumber &a){
71     out<<'<<a.el<<" "<<a.x<<" "<<a.er<<">";
72     return out;
73 }
74
75 std::istream& operator >>(std::istream &in, FuzzyNumber &a){
76     char c;
77     in>>a.x>>c>>a.er;
78     a.el = a.x - a.er;
79     a.er = a.x + a.er;
80     return in;
81 }
82
83 int main(){
84     FuzzyNumber a, b;
85     std::cout<<"Enter two fuzzy numbers <x, e>:\n";
86     std::cin>>a>>b;
87     std::cout<<"a = "<<a<<std::endl;
88     std::cout<<"b = "<<b<<std::endl;
89     std::cout<<"a + b="<<a+b<<std::endl;
90     std::cout<<"a - b = "<<a-b<<std::endl;
91     std::cout<<"a * b = "<<a*b<<std::endl;
92     std::cout<<"a / b = "<<a/b<<std::endl;
93     std::cout<<"1/a = "<<a.opposite()<<std::endl;
94     std::cout<<"a = "<<a<<std::endl;
95     std::cout<<"b = "<<b<<std::endl;
96     if(a>b) std::cout<<"a bigger then b" << std::endl;
97     if(a<b) std::cout<<"a lesser then b" << std::endl;
98     if(a==b) std::cout<<"a equal b" << std::endl;
99 }

```

### 3 Консоль

```
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/lab1$ ls
CMakeLists.txt  main.cpp  test_01  test_02  test_03
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/lab1$ mkdir cm; cd cm; ls
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/lab1/cm$ cmake ../
```

CMake Warning (dev) in CMakeLists.txt:

No project() command is present. The top-level CMakeLists.txt file must contain a literal, direct call to the project() command. Add a line of code such as

```
project(ProjectName)
```

near the top of the file, but after cmake\_minimum\_required().

CMake is pretending there is a "project(Project)" command on the first line.

This warning is for project developers. Use -Wno-dev to suppress it.

```
-- The C compiler identification is GNU 12.0.1
-- The CXX compiler identification is GNU 12.0.1
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
```

CMake Deprecation Warning at CMakeLists.txt:1 (cmake\_minimum\_required):

Compatibility with CMake < 2.8.12 will be removed from a future version of CMake.

Update the VERSION argument <min> value or use a ...<max> suffix to tell CMake that the project does not need compatibility with older versions.

```
-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/d/X-Files/MAI/3 sem/OOP/labs/lab1/cm
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/lab1/cm$ ls
CMakeCache.txt  CMakeFiles  Makefile  cmake_install.cmake
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/lab1/cm$ make
[ 50%] Building CXX object CMakeFiles/oop_exercise_1.dir/main.cpp.o
[100%] Linking CXX executable ../oop_exercise_1
[100%] Built target oop_exercise_1
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/lab1/cm$ ls
```

```

CMakeCache.txt  CMakeFiles  Makefile  cmake_install.cmake
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/lab1/cm$ cd ..; ls
CMakeLists.txt  cm  main.cpp  oop_exercise_1  test_01  test_02  test_03
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/lab1$ cat test_01
1, 2
2, 1
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/lab1$ ./oop_exercise_1 < test_01
Enter two fuzzy numbers <x, e>:
a = (-1, 1, 3)
b = (1, 2, 3)
a + b=(0, 3, 6)
a - b = (-2, -1, 0)
a * b = (-1, 2, 9)
a / b = (-0.333333, 0.5, 3)
1/a = (0.333333, 1, -1)
a = (-1, 1, 3)
b = (1, 2, 3)
a less than b
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/lab1$ cat test_02
2, 1
1, 2
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/lab1$ ./oop_exercise_1 < test_02
Enter two fuzzy numbers <x, e>:
a = (1, 2, 3)
b = (-1, 1, 3)
a + b=(0, 3, 6)
a - b = (2, 1, 0)
a * b = (-1, 2, 9)
a / b = (0.333333, 2, -3)
1/a = (0.333333, 0.5, 1)
a = (1, 2, 3)
b = (-1, 1, 3)
a bigger than b
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/lab1$ cat test_03
1,2
1,2
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/lab1$ ./oop_exercise_1 < test_03
Enter two fuzzy numbers <x, e>:
a = (-1, 1, 3)
b = (-1, 1, 3)
a + b=(-2, 2, 6)
a - b = (0, 0, 0)
a * b = (1, 1, 9)
a / b = (-0.333333, 1, -3)
1/a = (0.333333, 1, -1)
a = (-1, 1, 3)
b = (-1, 1, 3)
a equal b

```

## 4 Выводы

Выполнив первую лабораторную работу по курсу «ООП», я научился определять классы, их поля, методы и конструкторы, а так же использовать перегрузку операторов для пользовательских классов. Так же в ходе работы я приобрел начальные знания и навыки, необходимые для работы с системой сборки CMake.