

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

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

Лабораторная работа № 1

Тема: Простые классы на языке C++

Студент: Токарев Никита

Группа: 80-207

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2019

1. Постановка задачи

Реализовать класс FuzzyNumber для работы с нечеткими числами. Для нечетких чисел A и B реализовать операции: сложение, вычитание, умножение, деление, получение обратного числа, сравнение по x. Считать, что число представлено парой (x,e);

2. Описание программы

В данной работе я реализовал класс для работы с нечеткими числами. Программа работает таким образом:

1. В первую очередь вводятся числа x и e для A и B соответственно.
2. Запускается функция help() и свич меню.

Что касается самого класса, то в нем реализованы функции, необходимые по заданию и также присутствует конструктор. Пример вызова функции: A.add(B). Также в моем классе использованы модификаторы доступа public и private. Более того в Private описаны используемые переменные, а в public функции для работы с классом и функции, возвращающие значения переменных, которые объявлены в модификаторе доступа private. В функции int main() инициализируются переменные, которые являются числами нечеткого числа, затем запускается свич-меню, где операция выполняется в зависимости от того, что будет введено в консоль. Также хотелось бы отметить, что мое нечеткое число представлено тройкой чисел: (x-e, x, x+e). Из-за данного условия в модификаторе доступа private, помимо x и e, присутствует переменные l, c, r. Пример:

```
A.add(B); //вызов функция сложения
```

```
cout << "(" << A.getleft() << ", " << A.getcentre() << ", " << A.getright() << ")"
```

<< endl;

Где getleft(),getcentre(),getright() возвращает l,c,r соответственно.

3. Набор testcases

1) 2 3 4 5 a p d r s m c h e

2) 0 3 0 5 a p d r s m c e

3) 1 0 3 6 a p d r s m c e

4. Результаты выполнения тестов.

1)

A(x) = 2

A(e) = 3

B(x) = 4

B(e) = 5

A+B: press a

Print: press p

A/B: press d

A in -1: press r

A-B: press s

A*B: press m

A>?B: press c

Help: press h

Exit: press e

a

(-2,6,14)

p

A = (-1,2,5)

B = (-1,4,9)

d

(-0.111111,0.5,-5)

r

(0.2,0.5,-1)

s

(-10,-2,6)

m

(1,8,45)
c
 $B(x) > A(x)$
h
A+B: press a
Print: press p
A/B: press d
A in -1: press r
A-B: press s
A*B: press m
A>?B: press c
Help: press h
Exit: press e

e

2)
 $A(x) = 0$
 $A(e) = 3$
 $B(x) = 0$
 $B(e) = 5$
A+B: press a
Print: press p
A/B: press d
A in -1: press r
A-B: press s
A*B: press m
A>?B: press c
Help: press h
Exit: press e

a

(-8,0,8)
p
 $A = (-3,0,3)$
 $B = (-5,0,5)$
d
 $B(x) \leq 0$

r

$A(x) \leq 0$

s

$(-8, 0, 8)$

m

$(15, 0, 15)$

c

$A(x) = B(x)$

e

3)

$A(x) = 1$

$A(e) = 0$

$B(x) = 3$

$B(e) = 6$

A+B: press a

Print: press p

A/B: press d

A in -1: press r

A-B: press s

A*B: press m

A>?B: press c

Help: press h

Exit: press e

a

$(-2, 4, 10)$

p

$A = (1, 1, 1)$

$B = (-3, 3, 9)$

d

$(0.111111, 0.333333, -0.333333)$

r

$(1, 1, 1)$

s

$(-8, -2, 4)$

m

$(-3, 3, 9)$

c

$B(x) > A(x)$

e

5. Листинг программы

CmakeLists.txt

```
cmake_minimum_required(VERSION 2.8)
```

```
add_executable(fazzy Fazzynumber.cpp)
```

Fazzynumber.cpp

```
#include <iostream>
```

```
#include <cstdlib>
```

```
using namespace std;
```

```
void help() { //функция, описывающая возможные вызовы в свич-меню
```

```
printf("A+B: press a\n");
```

```
printf("Print: press p\n");
```

```
printf("A/B: press d\n");
```

```
printf("A in -1: press r\n");
```

```
printf("A-B: press s\n");
```

```
printf("A*B: press m\n");
```

```
printf("A>?B: press c\n");
```

```
printf("Help: press h\n");
```

```
printf("Exit: press e\n");
```

```
}
```

```
class fazzynumber {
```

```
private:
```

```
double a,e,l,c,r;
```

```
public:
```

```
fazzynumber(double x, double y) { //конструктор класса
```

```
a = x;
```

```
e = y;
```

```
}
```

```
void add(fazzynumber b) { //функция сложения двух нечетких чисел
```

```
double a1,e1;
```

```
a1 = a + b.a;
```

```
e1 = e + b.e;
```

```
l = a1 - e1;
```

```
c = a1;
```

```
r = a1 + e1;
```

```
}
```

```
void sub(fazzynumber b) { //функция вычитания
```

```

double a1,e1;
a1 = a - b.a;
e1 = e + b.e;
l = a1 - e1;
c = a1;
r = a1 + e1;
}
void mul(fazzynumber b) {//функция умножения
double a1,e1,a1e2,a2e1;
a1 = a * b.a;
e1 = e * b.e;
a1e2 = a * b.e;
a2e1 = b.a * e;
l = a1 - a1e2 - a2e1 + e1;
c = a1;
r = a1 + a1e2 + a2e1 + e1 ;
}
void opp() {//функция получения обратного числа
l = 1/(a + e);
c = 1/a;
r = 1/(a - e);
}
void div(fazzynumber b) {//функция деления
l = (a - e)/(b.a + b.e);
c = a/b.a;
r = (a + e)/(b.a - b.e);
}
void print() {//функция печати нечеткого числа
cout << "(" << a - e << ", " << a << ", " << a + e << ")" << endl;
}
void comp(fazzynumber b) {//функция сравнения двух чисел по x
if(a > b.a) {
cout << "A(x) > B(x)" << endl;
} else if (a < b.a) {
cout << "B(x) > A(x)" << endl;
} else {
cout << "A(x) = B(x)" << endl;
}
}
double getleft() {
return l;
}

```

```

}
double gettright() {
return r;
}
double getcentre() {
return c;
}
double getx() {
return a;
}
};

int main() {
double c1,ec;
char c;
cout << "A(x) = ";
cin >> c1;
cout << "A(e) = ";
cin >> ec;
fazzynumber A(c1,ec);
double d1,ed;
cout << "B(x) = ";
cin >> d1;
cout << "B(e) = ";
cin >> ed;
fazzynumber B(d1,ed);
help();
while (true) {
cin >> c;
switch (c) { //Свич меню.
case 'a':
A.add(B); //вызов функция сложения
cout << "(" << A.getleft() << "," << A.getcentre() << "," << A.gettright() << ")"
<< endl;
break;
case 'r':
if (A.getx() > 0) {
A.opp(); //вызов функции взятия обратного числа
cout << "(" << A.getleft() << "," << A.getcentre() << "," << A.gettright() << ")"
<< endl;
} else {

```



```

cout << "A(x) <= 0"<< endl;
}
break;
case 'p':
cout << "A = ";
A.print();
cout << "B = ";
B.print();
break;
case 'd':
if(B.getx() > 0) {
A.div(B);
cout << "(" << A.getleft() << "," << A.getcentre() << "," << A.getright() << ")"
<< endl;
} else {
cout << "B(x) <= 0"<< endl;
}
break;
case 's':
A.sub(B);
cout << "(" << A.getleft() << "," << A.getcentre() << "," << A.getright() << ")"
<< endl;
break;
case 'm':
A.mul(B);
cout << "(" << A.getleft() << "," << A.getcentre() << "," << A.getright() << ")"
<< endl;
break;
case 'c':
A.comp(B);
break;
case 'h':
help();
break;
case 'e':
return 0;
break;
}
}
return 0;
}

```

6. Выводы

В основе ООП лежат понятия **«объект»** и **«класс»**. В языке программирования объект – это переменная типа «класс». Класс описывает данные и методы (функции), которые будут использоваться объектом этого класса.

СПИСОК ЛИТЕРАТУРЫ

1. Основы сборки проектов на C/C++ при помощи Cmake URL : <https://eax.me/cmake/>
2. Классы в C++ URL : <http://cppstudio.com/post/439/>