

Балтийский государственный технический университет
«ВОЕНМЕХ» им. Д. Ф. Устинова

Кафедра И5 «Информационные системы и программная инженерия»

Лабораторная работа №1

по дисциплине «Программирование на языке высокого уровня»

по теме «Классы: Основные понятия и определения»

Выполнил:

Студент Альков В. С.

Группа И407Б

Преподаватель:

Кимсанбаев К. А.

Санкт-Петербург

2021 г.

Задача: Описать класс в соответствии с индивидуальным вариантом задания и реализовать все его методы. Каждый класс должен содержать, помимо указанных в варианте методов, конструктор с параметрами, конструктор копирования, деструктор, методы ввода с клавиатуры, установки и получения значений полей, вывода этих значений на экран. В каждом методе класса, включая конструкторы и деструктор, предусмотреть отладочную печать сообщения, содержащего имя метода. Написать программу для тестирования всех методов класса, выбор метода должен осуществляться с помощью меню.

Класс «Комплексное число». Поля класса: действительная и мнимая части.

Методы: возведение комплексного числа в целую степень, перегрузка операций сравнения («==», «!=») двух комплексных чисел и увеличения значения «+=» на величину другого комплексного или вещественного числа.

Текст программы с классом

```
#include <iostream>
#include <math.h>
//отладочная печать
#define PrintMethodName 1
//определение класса
class ComplexNumber
{
private:
    double realCoefficient;
    double imagnateCoefficient;
public:
    //конструктор с параметрами
    ComplexNumber(double a = 0, double b = 0);
    //конструктор копирования
    ComplexNumber(ComplexNumber& number);
    void setRealCoefficient(double num);
    void setImagnateCoefficient(double num);
    void readRealCoefficient();
    void readImagnateCoefficient();
    double getRealCoefficient();
    double getImagnateCoefficient();
    void exponent(int n);
    void print(const char a[] = "");
    void printRealCoefficient();
    void printImagnateCoefficient();
    //перегрузка операторов
    int operator==(ComplexNumber &num);
    int operator!=(ComplexNumber &num);
    void operator+=(ComplexNumber &num);
    void operator+=(double num);
    //деструктор
    ~ComplexNumber();
};

ComplexNumber::ComplexNumber(double a, double b)
{
    if(PrintMethodName)
        std::cout << "constructor ComplexNumber(double, double)\n";
    realCoefficient = a;
    imagnateCoefficient = b;
}
```

```

ComplexNumber::ComplexNumber(ComplexNumber& number)
{
    if(PrintMethodName)
        std::cout << "constructor ComplexNumber(ComplexNumber&) \n";
    realCoefficient = number.realCoefficient;
    imagnateCoefficient = number.imagnateCoefficient;
}

void ComplexNumber::setRealCoefficient(double num)
{
    if(PrintMethodName)
        std::cout << "method setRealCoefficient\n";
    realCoefficient = num;
}

void ComplexNumber::setImagnateCoefficient(double num)
{
    if(PrintMethodName)
        std::cout << "method setImagnateCoefficient\n";
    imagnateCoefficient = num;
}

void ComplexNumber::readRealCoefficient()
{
    if(PrintMethodName)
        std::cout << "method readRealCoefficient\n";
    std::cin >> realCoefficient;
}

void ComplexNumber::readImagnateCoefficient()
{
    if(PrintMethodName)
        std::cout << "method readImagnateCoefficient\n";
    std::cin >> imagnateCoefficient;
}

double ComplexNumber::getRealCoefficient()
{
    if(PrintMethodName)
        std::cout << "method getRealCoefficient\n";
    return realCoefficient;
}

double ComplexNumber::getImagnateCoefficient()
{
    if(PrintMethodName)
        std::cout << "method getImagnateCoefficient\n";
    return imagnateCoefficient;
}
//метод возведения в степень комплексного числа по формуле Муавра
void ComplexNumber::exponent(int n)
{
    if(PrintMethodName)
        std::cout << "method exponent\n";
    double argument;
    double modul = sqrt(realCoefficient*realCoefficient +
    imagnateCoefficient*imagnateCoefficient);
    modul = pow(modul, n);
    if (realCoefficient > 0)
        argument = atan(imagnateCoefficient/realCoefficient);
    else
        if (imagnateCoefficient > 0)

```

```

        argument = M_PI + atan(imaginateCoefficient/realCoefficient);
    else
        argument = -M_PI + atan(imaginateCoefficient/realCoefficient);
    realCoefficient = modul*cos(n*argument);
    imagineCoefficient = modul*sin(n*argument);
}

void ComplexNumber::print(const char a[])
{
    if(PrintMethodName)
        std::cout << "method print\n";
    std::cout << a << realCoefficient << " + " << imagineCoefficient <<
    "*i\n";
}

void ComplexNumber::printRealCoefficient()
{
    if(PrintMethodName)
        std::cout << "method printRealCoefficient\n";
    std::cout << realCoefficient << "\n";
}

void ComplexNumber::printImagineCoefficient()
{
    if(PrintMethodName)
        std::cout << "method printImagineCoefficient\n";
    std::cout << imagineCoefficient << "\n";
}

int ComplexNumber::operator==(ComplexNumber &num)
{
    if(PrintMethodName)
        std::cout << "operator==\n";
    return realCoefficient == num.realCoefficient && imagineCoefficient ==
    num.imagineCoefficient;
};

int ComplexNumber::operator!=(ComplexNumber &num)
{
    if(PrintMethodName)
        std::cout << "operator!=\n";
    return realCoefficient != num.realCoefficient || imagineCoefficient !=
    num.imagineCoefficient;
};

void ComplexNumber::operator+=(ComplexNumber &num)
{
    if(PrintMethodName)
        std::cout << "operator+=(ComplexNumber) \n";
    realCoefficient += num.realCoefficient;
    imagineCoefficient += num.imagineCoefficient;
};

void ComplexNumber::operator+=(double num)
{
    if(PrintMethodName)
        std::cout << "operator+=(double) \n";
    realCoefficient += num;
};

ComplexNumber::~ComplexNumber()
{
    if(PrintMethodName)

```

```

        std::cout << "~ComplexNumber";
    }
}

```

Текст тестирующей программы

```

#include "complexnumber.cpp"
#include <iostream>
#include <stdlib.h>
#include <locale.h>
int main()
{
    setlocale(LC_ALL, "rus");
    ComplexNumber a(1,2), b(a);
    double c;
    int menu, menu2, n;
    printf("Нажмите Enter..\n");
    getchar();
    do
    {
        system ("CLS");
        puts (" 1. Ввести с клавиатуры действительную часть");
        puts (" 2. Ввести с клавиатуры мнимую часть");
        puts (" 3. Задать действительную часть из переменной c");
        puts (" 4. Задать мнимую часть из переменной c");
        puts (" 5. Записать действительную часть в переменную c");
        puts (" 6. Записать мнимую часть в переменную c");
        puts (" 7. Возвести комплексное число в степень");
        puts (" 8. Вывести комплексное число");
        puts (" 9. Вывести действительную часть комплексного числа");
        puts ("10. Вывести мнимую часть комплексного числа");
        puts ("11. Проверить на равенство комплексные числа a и b");
        puts ("12. Проверить на неравенство комплексные числа a и b");
        puts ("13. Сложить комплексное число a с b (a+=b)");
        puts ("14. Сложить комплексное число a с вещ. числом c (a+=c)");
        puts ("15. Ввести с клавиатуры c");
        puts ("16. Вывести c");
        puts ("17. Выйти");
        scanf("%d%c", &menu);
        system ("CLS");
        switch(menu)
        {
            case 1: printf("Выберите число:\n1. a\n2. b\n");
                    scanf("%d%c", &menu2);
                    system ("CLS");
                    switch(menu2)
                    {
                        case 1: a.readRealCoefficient(); break;
                        case 2: b.readRealCoefficient(); break;
                    };
                    break;
            case 2: printf("Выберите число:\n1. a\n2. b\n");
                    scanf("%d%c", &menu2);
                    system ("CLS");
                    switch(menu2)
                    {
                        case 1: a.readImaginateCoefficient(); break;
                        case 2: b.readImaginateCoefficient(); break;
                    };
                    break;
            case 3: printf("Выберите число:\n1. a\n2. b\n");
                    scanf("%d%c", &menu2);
                    system ("CLS");
                    switch(menu2)

```

```

        {
            case 1: a.setRealCoefficient(c); break;
            case 2: b.setRealCoefficient(c); break;
        };
        break;
case 4: printf("Выберите число:\n1. a\n2. b\n");
scanf("%d%c", &menu2);
system ("CLS");
switch(menu2)
{
    case 1: a.setImaginateCoefficient(c); break;
    case 2: b.setImaginateCoefficient(c); break;
};
break;
case 5: printf("Выберите число:\n1. a\n2. b\n");
scanf("%d%c", &menu2);
system ("CLS");
switch(menu2)
{
    case 1: c = a.getRealCoefficient(); break;
    case 2: c = b.getRealCoefficient(); break;
};
break;
case 6: printf("Выберите число:\n1. a\n2. b\n");
scanf("%d%c", &menu2);
system ("CLS");
switch(menu2)
{
    case 1: c = a.getImaginateCoefficient(); break;
    case 2: c = b.getImaginateCoefficient(); break;
};
break;
case 7: printf("Введите степень: ");
scanf("%d", &n);
printf("Выберите число:\n1. a\n2. b\n");
scanf("%d%c", &menu2);
system ("CLS");
switch(menu2)
{
    case 1: a.exponent(n); break;
    case 2: b.exponent(n); break;
};
break;
case 8: printf("Выберите число:\n1. a\n2. b\n");
scanf("%d%c", &menu2);
system ("CLS");
switch(menu2)
{
    case 1: a.print(); break;
    case 2: b.print(); break;
};
break;
case 9: printf("Выберите число:\n1. a\n2. b\n");
scanf("%d%c", &menu2);
system ("CLS");
switch(menu2)
{
    case 1: a.printRealCoefficient(); break;
    case 2: b.printRealCoefficient(); break;
};
break;
case 10: printf("Выберите число:\n1. a\n2. b\n");
scanf("%d%c", &menu2);

```

```

        system ("CLS");
        switch (menu2)
        {
            case 1: a.printImaginateCoefficient(); break;
            case 2: b.printImaginateCoefficient(); break;
        };
        break;
    case 11: printf("a==b = %d", a==b); break;
    case 12: printf("a!=b = %d", a!=b); break;
    case 13: a+=b; break;
    case 14: a+=c; break;
    case 15: printf("Введите число: ");
        scanf("%lf%c", &c);
        break;
    case 16: printf("%lf", c); break;
    case 17: break;
    default: printf("Неправильный ввод"); break;
};
    getchar();
} while (menu != 17);
return 0;
}

```

Результат работы программы

```

constructor ComplexNumber(double, double)
constructor ComplexNumber(ComplexNumber&)
Нажмите Enter..

```

1. Ввести с клавиатуры действительную часть
2. Ввести с клавиатуры мнимую часть
3. Задать действительную часть из переменной c
4. Задать мнимую часть из переменной c
5. Записать действительную часть в переменную c
6. Записать мнимую часть в переменную c
7. Возвести комплексное число в степень
8. Вывести комплексное число
9. Вывести действительную часть комплексного числа
10. Вывести мнимую часть комплексного числа
11. Проверить на равенство комплексные числа a и b
12. Проверить на неравенство комплексные числа a и b
13. Сложить комплексное число a с b (a+=b)
14. Сложить комплексное число a с вещ. числом c (a+=c)
15. Ввести с клавиатуры c
16. Вывести c
17. Выйти

1. Ввести с клавиатуры действительную часть

Выберите число:

1. a
 2. b
- 1

```

method readRealCoefficient
12

```

2. Ввести с клавиатуры мнимую часть

```
Выберите число:  
1. a  
2. b  
1
```

```
method readImaginateCoefficient  
3
```

8. Вывести комплексное число

```
Выберите число:  
1. a  
2. b  
1
```

```
method print  
12 + 3*i
```

9. Вывести действительную часть комплексного числа

```
Выберите число:  
1. a  
2. b  
1
```

```
method printRealCoefficient  
12
```

10. Вывести мнимую часть комплексного числа

```
Выберите число:  
1. a  
2. b  
1
```

```
method printImaginateCoefficient  
3
```

5. Записать действительную часть в переменную c

```
Выберите число:  
1. a  
2. b  
1
```

```
method getRealCoefficient
```

16. Вывести c

```
12,000000
```

6. Записать мнимую часть в переменную c


```
Выберите число:  
1. a  
2. b  
1
```

```
method getImaginateCoefficient
```

16. Вывести с

```
3,000000
```

3. Задать действительную часть из переменной c

```
Выберите число:  
1. a  
2. b  
2
```

```
method setRealCoefficient
```

4. Задать мнимую часть из переменной c

```
Выберите число:  
1. a  
2. b  
2
```

```
method setImaginateCoefficient
```

8. Вывести комплексное число

```
Выберите число:  
1. a  
2. b  
2
```

```
method print  
3 + 3*i
```

7. Возвести комплексное число в степень

```
Введите степень: 3  
Выберите число:  
1. a  
2. b  
2
```

```
method exponent
```

8. Вывести комплексное число

```
Выберите число:  
1. a  
2. b  
2
```

```
method print  
-54 + 54*i
```

8. Вывести комплексное число

```
Выберите число:  
1. a  
2. b  
1
```

```
method print  
12 + 3*i
```

11. Проверить на равенство комплексные числа a и b

```
operator==  
a==b = 0
```

12. Проверить на неравенство комплексные числа a и b

```
operator!=  
a!=b = 1
```

13. Сложить комплексное число a с b ($a+=b$)

```
operator+=(ComplexNumber)
```

8. Вывести комплексное число

```
Выберите число:  
1. a  
2. b  
1
```

```
method print  
-42 + 57*i
```

15. Ввести с клавиатуры c

```
Введите число: 10
```

14. Сложить комплексное число a с вещ. числом c ($a+=c$)

```
operator+=(double)
```

8. Вывести комплексное число

```
Выберите число:  
1. a  
2. b  
1
```

```
method print  
-32 + 57*i
```

17. Выйти

```
~ComplexNumber~ComplexNumber  
Process returned 0 (0x0)   execution time : 1695.075 s  
Press any key to continue.
```