

Московский Авиационный Институт
(Национальный исследовательский Университет)

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

**Лабораторная работа
по курсу «ООП»**

**Тема:
Операторы. Литералы.**

| | |
|----------------|---------------|
| Студент: | Симонов С.Я. |
| Группа: | М8О-206Б-18 |
| Преподаватель: | Журавлев А.А. |
| Вариант: | 1 |
| Оценка: | |
| Дата: | |

Москва
2019

1. Код программы на языке C++:

complex.hpp

```
#ifndef OOP_01_COMPLEX_HPP
#define OOP_01_COMPLEX_HPP

#include <iostream>
#include <cmath>
#include <sstream>

class complex {
public:
    complex() {
        a = 0;
        b = 0;
    }
    complex(double a, double b) : a(a), b(b)
    {}

    friend std::istream& operator >> (std::istream&, complex& o);
    friend std::ostream& operator << (std::ostream&, complex& o);

    complex operator+ (const complex &o) const;
    complex operator- (const complex &o) const;
    complex operator* (const complex &o) const;
    complex operator/ (const complex &o) const;
    complex& operator*= (unsigned long long num);
    bool operator== (const complex &o) const;

    complex conj() const;
    int compare_by_abs(const complex &o) const;
private:
    double a, b;
};

complex operator""_xn(unsigned long long first);

#endif
```

complex.cpp

```
#include "complex.hpp"
```

```
std::ostream& operator << (std::ostream& os, complex& o)
{
    os << o.a << ' ' << o.b;
    return os;
}
```

```
std::istream& operator >> (std::istream& is, complex& o) {
    is >> o.a >> o.b;

    return is;
}
```

```
complex complex::operator+ (const complex &o) const {
    complex result;
    result.a = a + o.a;
    result.b = b + o.b;
    return result;
}
```

```
complex complex::operator- (const complex &o) const {
    complex result;
    result.a = a - o.a;
    result.b = b - o.b;
    return result;
}
```

```
complex complex::operator* (const complex &o) const {
    complex result;
    result.a = a * o.a - b * o.b;
    result.b = a * o.b + b * o.a;
    return result;
}
```

```
complex complex::operator/ (const complex &o) const {
    complex result;
    result.a = (a * o.a + b * o.b) / (a * a + b * b);
    result.b = (b * o.a - a * o.b) / (a * a + b * b);
    return result;
}
```

```
complex& complex::operator*=(unsigned long long num) {
```

```
    a = a * num;
    b = b * num;
    return *this;
}
```

```
bool complex::operator==(const complex &o) const {
    if ((a == o.a) && (b == o.b)) {
        return true;
    }
    return false;
}
```

```
complex operator"" _xn(unsigned long long first) {
    complex result(1, 2);
    result *= first;
    return result;
}
```

```
complex complex::conj() const {
    complex result;
    result.a = a;
    result.b = -b;
    return result;
}
```

```
int complex::compare_by_abs(const complex &o) const {
    double c, d;
    c = sqrt((a * a) + (b * b));
    d = sqrt((o.a * o.a) + (o.b * o.b));
    if (c == d) {
        return 0;
    } else if (c > d) {
        return -1;
    }
    return 1;
}
```

main.cpp:

```
#include <iostream>
#include <cmath>
#include <sstream>

#include "complex.hpp"

int main()
{
    complex f, g, c;
    std::cin >> f;
    std::cin >> g;
    std::cout << f << '\n';
    std::cout << g << '\n';
    c = f + g;
    std::cout << c << '\n';
    c = f - g;
    std::cout << c << '\n';
    c = f * g;
    std::cout << c << '\n';
    c = f / g;
    std::cout << c << '\n';
    if (f == g) {
        std::cout << "yes";
    } else {
        std::cout << "no";
    }
    std::cout << std::endl;
    c = f.conj();
    std::cout << c << '\n';
    c = g.conj();
    std::cout << c << '\n';
    if (f.compare_by_abs(g) == 0) {
        std::cout << "a = b";
    } else if (f.compare_by_abs(g) == -1) {
        std::cout << "a > b";
    } else {
        std::cout << "a < b";
    }
    std::cout << "\n";
    c = 2_xn;
    std::cout << c << "\n";

    return 0;
}
```

}

2. Ссылка на репозиторий на GitHub.

https://github.com/keoni02032/oop_exercise_02/

3. Набор testcases.

test_01.test:

2 2 2 2

test_02.test:

1 2 3 4

test_03.test:

4 3 2 1

test_04.test:

1 -2 1 -2

test_05.test:

1 -2 1 2

4. Результаты выполнения тестов.

test_01.result:

2 2

2 2

4 4

0 0

0 8

1 0

yes

2 -2

2 -2

$|a| = |b|$

test_02.result:

1 2

3 4

4 6

-2 -2

-5 -10

0.44 0.08

no equal

1 -2

3 -4

$|a| < |b|$

test_03.result:

4 3 2 1

4 3

2 1

6 4

2 2

5 10

2.2 0.4

no equal

4 -3

2 -1

$|a| > |b|$

test_04.result:

1 -2 1 -2

1 -2

1 -2

2 -4

0 0

-3 -4

1 0

yes

. 1 2

1 2

$|a| = |b|$

test_05.result:

1 -2 1 2

1 -2

1 2

2 0

0 -4

5 0

-0.6 -0.8

no equal

1 2

1 -2

$|a| = |b|$

2 4

5 Объяснение результатов работы программы.

1) При запуске скрипта с аргументом `./ 'test (1). sh' ./lab1` объекты `a`, `b` в основной программе получают данные из файлов `test_?.test`. Над этими данными выполняются следующие операции: сложение, вычитание, умножение, деление, сравнение, выводится сопряженное число, реализация операции сравнения модулей.

2) Вывод данных объектов `a`, `b` в стандартный поток вывода.

3) Объекты `a` и `b` складываются с помощью функции-члена `add()` класса `complex`, и результат выводится в стандартный поток вывода с помощью функции `write()`.

4) С помощью перегрузки операторов сложение, вычитание, умножения и деления вычисляем соответствующие значения.

5) Для объектов `a` и `b` вычисляется сопряженное число с помощью функции-члена `conj()` класса `complex`.

6) С помощью перегрузки операторов сравнения проводим сравнение заданных комплексных чисел по модулю.

7) Пользовательский литерал `_xp`, который умножает комплексное число с вещественной частью 1 и мнимой частью 2 на заданное целое число `first` типа `unsigned long long`.

6. Вывод.

Выполняя данную лабораторную я получил опыт работы с простыми классами, с системой сборки Cmake, с системой контроля версий Git, а также изучил основы работы с классами в C++. Перегрузил операторы сложения, вычитания, умножения, деления, ввода и вывода, а также создал пользовательский литерал, который умножает комплексное число с вещественной частью 1 и мнимой частью 2 на заданное целое число.