

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное образовательное учреждение высшего образования

Санкт-Петербургский национальный исследовательский университет информационных
технологий, механики и оптики

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лабораторные работы №1-3

По дисциплине «Программирование»

Выполнил студент группы
№МЗ106

Казаков Никита Андреевич

Проверил

Повышев Владислав Вячеславович

Лабораторная работа №2

Классы.

Согласно варианту описать указанные классы (варианты распределяются преподавателем лично). Написать программу, использующую описанные классы: инициализация переменных (ввод пользователя), выполнение действий с экземплярами класса (в зависимости от дальнейшего ввода пользователя).

Описания и реализация должны находиться в разных файлах. Доступ к полям класса – только через методы. Внешние функции для работы с данными класса не допускаются. Перегрузка стандартных арифметических операций для класса (только в виде методов класса) – в зависимости от задания и здравого смысла (уместна консультация с преподавателем практики).

Дополнительно оценивается, если меню программы будет реализовано с помощью класса. Также возможно использование такого класса

«меню» другого студента с указанием его авторства (использование чужого «меню», естественно, дополнительно не оценивается).

Всюду под «выводом на экран» подразумевается вывод необходимой информации об объекте в текстовом виде.

main.cpp

```
#include <iostream>
#include "functions.h"
using namespace std;
int main() {
    complex first(1, 2);
    complex second(2, 3);
    complex third = second;
    complex fourth;
    double real = 1.55;
    cout << "Complex_1: " << first << '\n';
    cout << "Complex_2: " << second << '\n';
    cout << "Complex_3: " << third << '\n';
    cout << "Complex_4: " << fourth << '\n'; /home/just/ITMO_lab/lab_03
    cout << "Real_1: " << real << '\n' << '\n';
    cout << "Complex_1 = Complex_2" << '\n';
    first = second;
    cout << "Complex_1: " << first << '\n';
    first = third + second;
    cout << "Complex_1 = Complex_2 + Complex_3" << '\n' << "Complex_1: " <<
first << '\n';
    first = first * second;
    cout << "Complex_1 = Complex_1 * Complex_2" << '\n' << "Complex_1: " <<
first << '\n';
    cout << "Complex_1 length: " << first.length() << '\n';
    first = first * real;
    cout << "Complex_1 = Complex_1 * Real_1" << '\n' << "Complex_1: " << first
<< '\n';
    return 0;
}
```

```
}
```

functions.cpp

```
#include <iostream>
#include <cmath>
#include "functions.h"
complex::complex () : a(0), b(0) {}
complex::complex (double i, double i_1) : a(i), b(i_1) {}
complex::complex (const complex &value) : a(value.a), b(value.b) {}
double complex::length () {
    return sqrt(this->a * this->a + this->b * this->b);
}
```

functions.h

```
#pragma once
class complex {
public:
    explicit complex ();
    explicit complex (double, double);
    complex (const complex &);
    friend complex operator+ (const complex &left, const complex &right) {
        return complex(left.a + right.a, left.b + right.b);
    }
    friend complex &operator+= (complex &left, const complex &right) {
        left.a += right.a;
        left.b += right.b;
        return left;
    }
    friend complex operator* (const complex &left, const complex &right) {
        return complex(left.a * right.a - left.b * right.b, left.b * right.a +
left.a * right.b);
    }
    friend complex operator* (const complex &left, const double &right) {
        return complex(left.a * right, left.b);
    }
    friend complex operator* (const double &right, const complex &left) {
        return complex(left.a * right, left.b);
    }
    friend std::ostream &operator<< (std::ostream &out, const complex &value) {
        out << value.a << " + " << value.b << 'i';
        return out;
    }
    double length();
private:
    double a = 0;
    double b = 0;
};
```