

Министерство образования и науки РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

Тульский государственный университет

КАФЕДРА АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ

**КЛАССЫ. КОНСТРУКТОРЫ. ДЕСТРУКТОРЫ.**

Лабораторная работа № 1  
по курсу «Объектно-ориентированное программирование»

Вариант № 11

Выполнил:	студент группы 220601	_____	Белым А.А.
		(подпись)	
Проверил:	к. ф.-м. н., доцент каф. АТМ	_____	Середин О.С.
		(подпись)	

Тула 2013

## Цель работы

Ознакомиться с понятиями класс и экземпляр класса (объект), конструктор класса, деструктор. Спроектировать и написать программу на языке C++, использующую эти понятия.

## Задание

Задан класс:

```
class CMPLX
{
    public:
        CMPLX(); /*1 Инициализация как (0,0) */
        CMPLX (float real, float imag); /*2 Инициализация как (real,imag) */
        void Setcompl(float, float); /*3 Изменение числа */
        void Inc(); /*4 Инкремент: |z|=|z|+1 */
        void Dec(); /*5 Декремент: |z|=|z|-1 */
        CMPLX Add(CMPLX what); /*6 Сложение */
        CMPLX Sub(CMPLX what); /*7 Вычитание */
        CMPLX Mul(CMPLX what); /*8 Умножение */
        CMPLX Cmul(float what); /*9 Умножение на действительное число */
        CMPLX Div(CMPLX what); /*10 Деление */
        void Print(); /*11 Вывод в виде "("re"," im")" */
    private:
        float re;
        float im;
}
```

Реализовать следующие методы класса и написать демонстрационную программу, инициализирующую несколько переменных посредством конструктора №2, выводящую их, производящую над ними некоторые действия с помощью методов №№4,7,11; и выводящую результаты.

[illegible]

[illegible]

## Реализация класса

Ниже представлено определение класса, файл CMPLX.h:

```
#ifndef CMPLX_H
#define CMPLX_H
/*!Класс комплексных чисел.*/
class CMPLX {
public:
    /*!Конструктор по умолчанию.*/
    CMPLX();
    /*!Конструктор, задающий вещественную и мнимую части.
     * \param real вещественная часть
     * \param imag мнимая часть
     */
    CMPLX(float real, float imag);
    /*!Установка значения.
     * \param real вещественная часть
     * \param imag мнимая часть
     */
    void Setcompl(float real, float imag);
    /*!Инкремент - увеличение модуля на единицу.*/
    void Inc();
    /*!Декремент - уменьшение модуля на единицу.*/
    void Dec();
    /*!Операция сложения комплексных чисел.
     * \param what второе слагаемое
     * \return сумму комплексных чисел
     */
    CMPLX Add(CMPLX what);
    /*!Операция вычитания комплексных чисел.
     * \param what вычитаемое
     * \return разность комплексных чисел
     */
    CMPLX Sub(CMPLX what);
    /*!Операция умножения комплексных чисел.
     * \param what множитель
     * \return результат умножения
     */
    CMPLX Mul(CMPLX what);
    /*!Операция умножения на вещественное число.
     * \param what множитель
     * \return результат умножения
     */
    CMPLX Cmul(float what);
    /*!Операция деления комплексных чисел.
     * \param what делитель
     * \return результат деления
     */
    CMPLX Div(CMPLX what);
    /*!Вывод числа на экран в формате (real,imag).*/
    void Print();
private:
    /*!Вещественная часть
     float re;
     /*!Мнимая часть
     float im;
};
#endif //CMPLX_H
```

Ниже представлена реализация класса, файл CMPLX.cpp:

```
#include <iostream>
#include <cmath>
#include "CMPLX.h"
CMPLX::CMPLX() {
```

```

    re=0;im=0;
}

CMPLX::CMPLX(float real,float imag){
    re=real;im=imag;
}

void CMPLX::Setcompl(float r,float i){
    re=r; im=i;
}

void CMPLX::Inc(){
    float r=sqrt(re*re+im*im)+1,phi=atan2(im,re);
    re=r*cos(phi); im=r*sin(phi);
}

void CMPLX::Dec(){
    float r=sqrt(re*re+im*im)-1,phi=atan2(im,re);
    re=r*cos(phi); im=r*sin(phi);
}

CMPLX CMPLX::Add(CMPLX what){
    CMPLX temp;
    temp.re=re+what.re;
    temp.im=im+what.im;
    return temp;
}

CMPLX CMPLX::Sub(CMPLX what){
    CMPLX temp;
    temp.re=re-what.re;
    temp.im=im-what.im;
    return temp;
}

CMPLX CMPLX::Mul(CMPLX what){
    CMPLX temp;
    temp.re=re*what.re-im*what.im;
    temp.im=im*what.re+re*what.im;
    return temp;
}

CMPLX CMPLX::Cmul(float what){
    CMPLX temp;
    temp.re=re*what;
    temp.im=im*what;
    return temp;
}

CMPLX CMPLX::Div(CMPLX what){
    CMPLX temp;
    float denom=what.re*what.re+what.im*what.im;
    temp.re=(re*what.re+im*what.im)/denom;
    temp.im=(im*what.re-re*what.im)/denom;
    return temp;
}

void CMPLX::Print(){
    std::cout<<" ("<<re<<","<<im<<)" "<<std::endl;
}

```

## Демонстрационная программа

Далее приводится демонстрационная программа, файл lab1.cpp:

```

#include "CMPLX.h"
#include <iostream>
int main() {
    float r=0,i=0;
    std::cin>>r>>i;
    CMPLX a;

    a.Setcompl(r,i);

    std::cin>>r>>i;

    CMPLX b(r,i);

    a.Inc();b.Inc();
    CMPLX c=a.Sub(b);

    return 0;
}

```

## Инструкция программисту

Далее приводится описание функций, методов, типов данных и классов.

### ***Класс CMPLX***

Класс комплексных чисел.

```
#include <CMPLX.h>
```

#### **Открытые члены**

- **CMPLX ()**
- **CMPLX (float real, float imag)**
- **void Setcompl (float real, float imag)**
- **void Inc ()**
- **void Dec ()**
- **CMPLX Add (CMPLX what)**
- **CMPLX Sub (CMPLX what)**
- **CMPLX Mul (CMPLX what)**
- **CMPLX Cmul (float what)**
- **CMPLX Div (CMPLX what)**
- **void Print ()**

#### **Закрытые данные**

- **float re**  
*Вещественная часть*
- **float im**  
*Мнимая часть*

## Конструкторы

### CMPLX::CMPLX ()

Конструктор по умолчанию.

### CMPLX::CMPLX (float *real*, float *imag*)

Конструктор, задающий вещественную и мнимую части.

Аргументы конструктора представлены в таблице 1.

#### Аргументы:

Таблица 1 – Аргументы конструктора

<i>real</i>	вещественная часть
<i>imag</i>	мнимая часть

## Методы

### CMPLX CMPLX::Add (CMPLX *what*)

Операция сложения комплексных чисел.

Аргументы метода представлены в таблице 2.

#### Аргументы:

Таблица 2 – Аргументы метода сложения

<i>what</i>	второе слагаемое
-------------	------------------

#### Возвращает:

сумму комплексных чисел.

### CMPLX CMPLX::Cmul (float *what*)

Операция умножения на вещественное число.

Аргументы метода представлены в таблице 3.

#### Аргументы:

Таблица 3 – Аргументы метода умножения на вещественное

<i>what</i>	множитель
-------------	-----------

#### Возвращает:

результат умножения.

### void CMPLX::Dec ()

Декремент - уменьшение модуля на единицу.

### CMPLX CMPLX::Div (CMPLX *what*)

Операция деления комплексных чисел.

Аргументы метода представлены в таблице 4.



**Аргументы:**

Таблица 4 – Аргументы метода деления

<i>what</i>	делитель
-------------	----------

**Возвращает:**

результат деления.

**void CMPLX::Inc ()**

Инкремент - увеличение модуля на единицу.

**CMPLX CMPLX::Mul (CMPLX *what*)**

Операция умножения комплексных чисел.

Аргументы метода представлены в таблице 5.

**Аргументы:**

Таблица 5 – Аргументы метода умножения

<i>what</i>	множитель
-------------	-----------

**Возвращает:**

результат умножения.

**void CMPLX::Print ()**

Вывод числа на экран в формате (real,imag).

**void CMPLX::Setcompl (float *real*, float *imag*)**

Установка значения.

Аргументы метода представлены в таблице 6.

**Аргументы:**

Таблица 6 – Аргументы метода установки значения

<i>real</i>	вещественная часть
<i>imag</i>	мнимая часть

**CMPLX CMPLX::Sub (CMPLX *what*)**

Операция вычитания комплексных чисел.

Аргументы метода представлены в таблице 7.

**Аргументы:**

Таблица 7 – Аргументы метода вычитания

<i>what</i>	вычитаемое
-------------	------------

**Инструкция пользователю**

Данная программа производит инкремент и вычитание комплексных чисел.

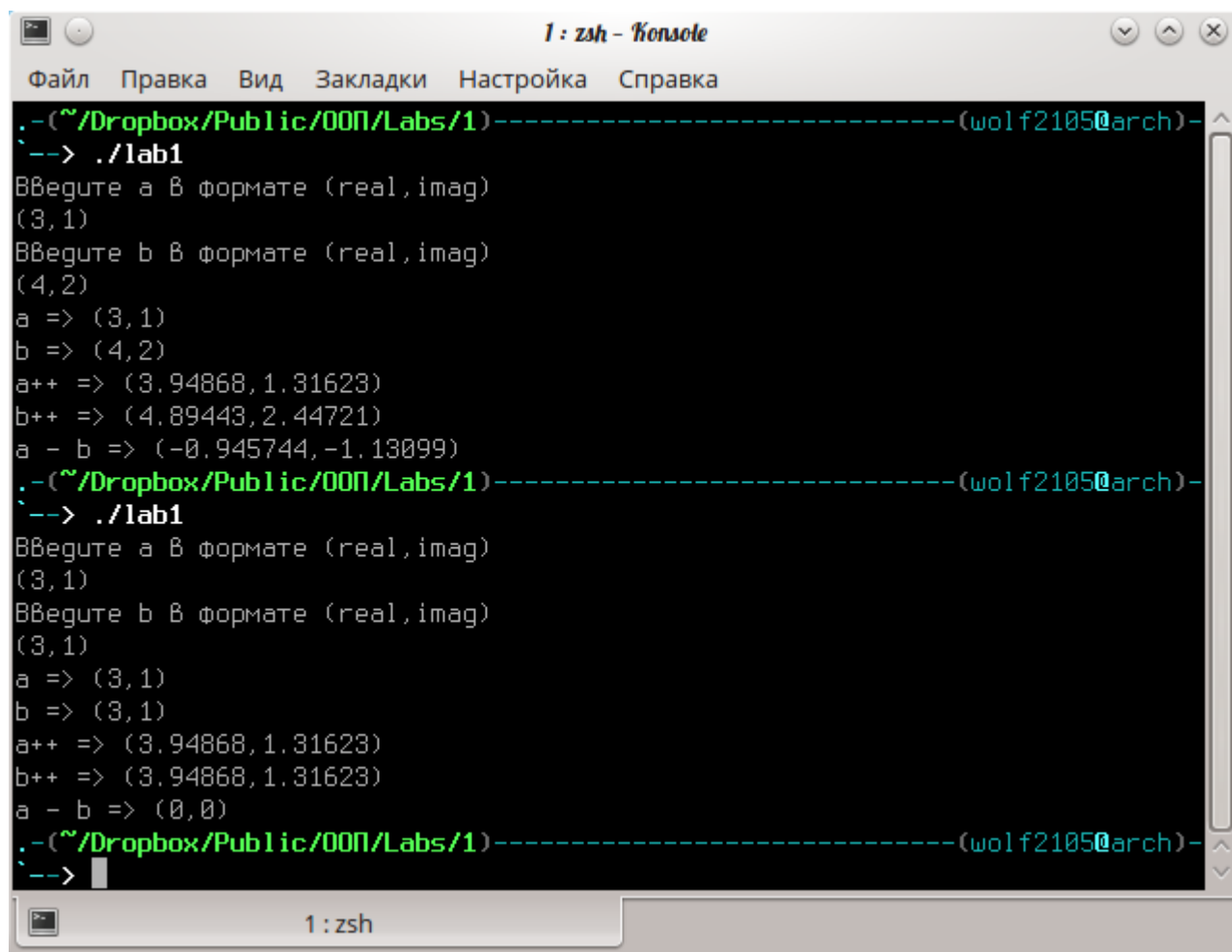
Для работы введите два комплексных числа в формате

(вещественная часть, мнимая часть).

После этого программа выведет значения этих чисел, результат их инкремента и разности.

### Контрольный пример

Ниже на рисунке 1 представлен пример работы программы, использующей класс комплексных чисел.



```
1: zsh - Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
~/.Dropbox/Public/00П/Labs/1)----- (wolf2105@arch)-
--> ./lab1
Введите a в формате (real,imag)
(3,1)
Введите b в формате (real,imag)
(4,2)
a => (3,1)
b => (4,2)
a++ => (3.94868,1.31623)
b++ => (4.89443,2.44721)
a - b => (-0.945744,-1.13099)
~/.Dropbox/Public/00П/Labs/1)----- (wolf2105@arch)-
--> ./lab1
Введите a в формате (real,imag)
(3,1)
Введите b в формате (real,imag)
(3,1)
a => (3,1)
b => (3,1)
a++ => (3.94868,1.31623)
b++ => (3.94868,1.31623)
a - b => (0,0)
~/.Dropbox/Public/00П/Labs/1)----- (wolf2105@arch)-
--> 
```

Рисунок 1— Пример работы программы, оперирующей комплексными числами

### Вывод

В данной лабораторной работе я познакомился с классами в языке программирования C++. Была написана программа, использующая класс комплексных чисел, который реализует конструкторы, деструктор и различные методы.