**ЛАБОРАТОРНАЯ РАБОТА №5**

**ПЕРЕГРУЗКА ОПЕРАТОРОВ В ЯЗЫКЕ С++**

**Цель**:

Целью данной лабораторной работы является изучение перегрузки операторов в языке С++ и использование перегруженных операторов на практике.

**Задания:**

1. Для класса Complex перегрузить операторы

присваивания, инкремента, декремента, сравнения, ввода и вывода.

2. Для класса Vector перегрузить операторы присваивания,

сравнения, ввода и вывода.

**Задание №1**

#include "pch.h"

#include "windows.h"

#include <iostream>

#include <string>

#include<cmath>

using namespace std;

class Complex

{

public:

double Re, Im;

Complex()

{

cout << "Введите действительную часть: " << endl;

cin >> Re;

cout << "Введите мнимую часть: " << endl;

cin >> Im;

}

Complex(const Complex &other)

{

this->Re = other.Re;

this->Im = other.Im;

}

~Complex() {};

void set\_c()

{

cout << "Введите действительную часть: " << endl;

cin >> Re;

cout << "Введите мнимую часть: " << endl;

cin >> Im;

}

string get\_c()

{

return to\_string(Re) + "+" + to\_string(Im) + "i";

}

double modul()

{

return sqrt(Re\*Re + Im \* Im);

}

double arg()

{

return atan(Im / Re);

}

friend istream& operator>> (std::istream &in, Complex &point)

{

cout << "Введите действительную часть: " << endl;

in >> point.Re;

cout << "Введите мнимую часть: " << endl;

in >> point.Im;

return in;

}

friend ostream& operator<< (std::ostream &out, const Complex &point)

{

out << point.Re << "+i\*" << point.Im;

return out;

}

bool operator < (Complex &other)

{

if (this->modul() < other.modul())return true;

else return false;

}

bool operator >(Complex &other)

{

if (this->modul() > other.modul())

{

return true;

}

else

{

return false;

}

}

bool operator ==(Complex &other)

{

if (this->modul() == other.modul())return true;

else return false;

}

Complex& operator ++()

{

this->Re++;

return \*this;

}

Complex& operator --()

{

this->Re--;

return \*this;

}

};

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

system("color 0A");

Complex a;

Complex b = a;

cout << "a: " << a << endl;

cout << "b=a: " << b << endl;

cin >> a;

cout << "a-измененное: " << a << endl;

cout << "Модуль a: " << a.modul() << endl;

cout << "Аргумент a: " << a.arg() << endl;

cout << "a++: " << ++a << endl;

cout << "a--: " << --a << endl;

if (a > b)

{

cout << "a > b" << endl;

}

else if (a < b)

{

cout << "a < b" << endl;

}

else if (a == b)

{

cout << "a = b";

}

system("pause");

return 0;

}

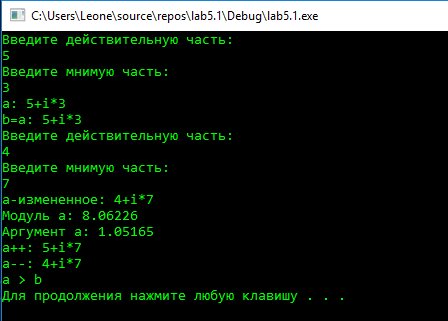


Рис.1

**Задание №2**

#include "pch.h"

#include "windows.h"

#include <iostream>

#include <string>

#include <cmath>

using namespace std;

class Vector

{

double x;

double y;

public:

Vector() : x(0), y(0) {}

Vector(double x, double y) : x(x), y(y) {};

Vector(const Vector& v);

Vector operator= (const Vector& v);

friend Vector operator+ (const Vector& v1, const Vector& v2);

friend Vector operator- (const Vector& v1, const Vector& v2);

friend const bool operator== (const Vector& v1, const Vector& v2);

friend const istream& operator>> (istream& in, Vector& v);

friend const ostream& operator<< (ostream& out, const Vector& v);

};

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

system("color 0A");

cout << "Введите X и Y:";

Vector v1;

cin >> v1;

Vector v2 = v1;

if (v1 == v2)

{

cout << "Сумма v1 и v2:"<< v1 + v2;

}

cout << endl;

system("pause");

return 0;

}

Vector::Vector(const Vector& v)

{

x = v.x;

y = v.y;

}

Vector Vector::operator= (const Vector& v)

{

return Vector(v.x, v.y);

}

Vector operator+ (const Vector& v1, const Vector& v2)

{

return Vector(v1.x + v2.x, v1.y + v2.y);

}

Vector operator- (const Vector& v1, const Vector& v2)

{

return Vector(v1.x - v2.x, v1.y - v2.y);

}

const bool operator== (const Vector& v1, const Vector& v2)

{

return (v1.x == v2.x) && (v1.y == v2.y) ;

}

const istream& operator>> (istream& in, Vector& v)

{

in >> v.x >> v.y ;

return in;

}

const ostream& operator<< (ostream& out, const Vector& v)

{

out << v.x << " " << v.y ;

return out;

}

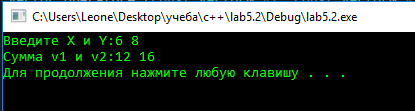
****

Рис.2

**Вывод:** В ходе выполнения лабораторной работы №5 были реализованы различные перегрузки.