/\*

Юркова Мирослава

8 группа

Аудиторные занятия 22.04

Класс OX

\*/

**#include <iostream>**

**#include "Ox.h"**

**using namespace std;**

**int main()**

**{**

**OX a, b, c = 30;**

**OX d = a + b;**

**cout << a << " " << b << " " << c << " " << d;**

**return 0;**

**}**

#include<iostream>

using namespace std;

class OX

{

int n;

public:

OX();

OX(int x);

OX &operator =(OX a);

OX& operator +=(const OX& a);

OX& operator -=(const OX\* a);

OX& operator \*=(const OX\* a);

OX& operator /=(const OX\* a);

OX operator ++();

friend OX operator +(OX a, OX b);

friend OX operator -(OX a, OX b);

friend OX operator \*(OX a, OX b);

friend OX operator /(OX a, OX b);

friend ostream& operator <<(ostream &out, OX &a);

friend istream& operator >>(istream &in, OX &a);

};

#include "Ox.h"

OX::OX(){

n = 10;

}

OX::OX(int x){

n = x;

}

OX& OX::operator =(OX a){

this->n = a.n;

return \*this;

}

OX& OX:: operator +=(const OX& a){

this->n += a.n;

return \*this;

}

OX& OX:: operator -=(const OX\* a){

this->n -= a->n;

return \*this;

}

OX& OX:: operator /=(const OX\* a){

this->n /= a->n;

return \*this;

}

OX& OX:: operator \*=(const OX\* a){

this->n \*= a->n;

return \*this;

}

OX OX::operator++(){

this->n++;

return \*this;

}

OX operator +(OX a, OX b){

OX c;

c = a.n + b.n;

return c;

}

OX operator -(OX a, OX b){

OX c;

c = a.n - b.n;

return c;

}

OX operator \*(OX a, OX b){

OX c;

c = a.n \* b.n;

return c;

}

OX operator /(OX a, OX b){

OX c;

if(b.n != 0) c = a.n / b.n;

else throw("Error!");

return c;

}

std::ostream& operator <<(std::ostream &out, OX &a){

out << a.n << endl;

return out;

}

std::istream& operator >>(std::istream &in, OX &a){

in >> a.n;

return in;

}