Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Звіт

З лабораторної роботи № 5, варіант 23

З дисципліни « ПРОГРАМУВАННЯ, ЧАСТИНА 2 (ОБ’ЄКТНООРІЄНТОВАНЕ ПРОГРАМУВАННЯ) »

На тему: «Перевантаження операторів»

Виконав: ст. гр. КІ-15

Пилипів Р. С.

Перевірив: викладач

Козак Н.Б.

Львів – 2020

**Завдання:**

23. +=, %, ==, !

**Код програми:**

//Soldier.h

#include<iostream>

#include<stdlib.h>

#include<time.h>

using namespace std;

class CSoldier

{

private:

double war;

double ser;

int\* ptr;

int size;

public:

CSoldier(double l = 1, double w = 1);

CSoldier(const CSoldier& rect);

virtual ~CSoldier();

double rank();

void name(double);

void num(double);

void en(double);

void time(double);

CSoldier(int s = 10);

CSoldier(CSoldier& arr);

friend ostream& operator<< (ostream& output, CSoldier& arr);

void Randomize(int num = 10);

CSoldier& operator= (CSoldier& arr);

CSoldier& operator+= (CSoldier& arr);

CSoldier operator% (CSoldier& arr);

CSoldier& operator== (CSoldier& arr);

CSoldier operator!();

};

//Soldier.cpp

#include "Soldier.h"

#include <math.h>

#include <iostream>

using namespace std;

CSoldier::CSoldier(double l, double w)

{

war = l;

ser = w;

}

CSoldier::CSoldier(const CSoldier& rect)

{

war = rect.war;

ser = rect.ser;

}

CSoldier::~CSoldier()

{

delete[] ptr;

cout << "Destructor" << endl;

}

double CSoldier::rank()

{

return int((ser + pow(war, 5)) / 50);

}

void CSoldier::name(double x)

{

string n;

cin >> n;

}

void CSoldier::num(double x)

{

cin >> x;

}

void CSoldier::en(double x)

{

cin >> x;

}

void CSoldier::time(double x)

{

cin >> x;

}

CSoldier::CSoldier(int s)

{

size = s;

ptr = new int[size];

for (int i = 0; i < size; i++)

{

ptr[i] = 0;

}

cout << "Constructor" << endl;

}

//----------------------------------------------------------------------

CSoldier::CSoldier(CSoldier& arr)

{

size = arr.size;

ptr = new int[size];

for (int i = 0; i < size; i++)

{

ptr[i] = arr.ptr[i];

}

cout << "Copy Constructor" << endl;

}

//----------------------------------------------------------------------

void CSoldier::Randomize(int num)

{

for (int i = 0; i < size; i++)

{

ptr[i] = rand() % num;

}

}

//----------------------------------------------------------------------

ostream& operator<< (ostream& output, CSoldier& arr)

{

for (int i = 0; i < arr.size; i++)

{

output << arr.ptr[i] << " ";

}

output << endl;

return output;

}

//----------------------------------------------------------------------

CSoldier& CSoldier::operator= (CSoldier& arr)

{

if (this != &arr)

{

delete[] ptr;

size = arr.size;

ptr = new int[size];

for (int i = 0; i < size; i++)

{

ptr[i] = arr.ptr[i];

}

}

return \*this;

}

//----------------------------------------------------------------------

CSoldier& CSoldier::operator+= (CSoldier& arr)

{

for (int i = 0; i < size; i++)

{

ptr[i] = ptr[i] + arr.ptr[i];

}

cout << "Operation += : " << endl;

return \*this;

}

//----------------------------------------------------------------------

CSoldier CSoldier::operator%(CSoldier& arr)

{

for (int i = 0; i < size; i++)

{

if (arr.ptr[i] != 0)

{

ptr[i] = ptr[i] % arr.ptr[i];

}

else ptr[i] = 0;

}

cout << "Operation % :" << endl;

return \*this;

}

//----------------------------------------------------------------------

CSoldier& CSoldier::operator==(CSoldier& arr)

{

for (int i = 0; i < size; i++)

{

ptr[i] = arr.ptr[i];

}

cout << "Operation == :" << endl;

return \*this;

}

//----------------------------------------------------------------------

CSoldier CSoldier::operator!()

{

for (int i = 0; i < size; i++)

{

ptr[i] = !ptr[i];

}

cout << "Operation ! : " << endl;

return \*this;

}

//main.cpp

#include<iostream>

#include<math.h>

#include"Soldier.h"

using namespace std;

int main(void)

{

double n = 0;

CSoldier name(n);

CSoldier num(n);

CSoldier en(n);

CSoldier time(n);

double ser, war;

cout << "Enter soldier name: " << endl;

name.name(n);

cout << "The part number in which it serves" << endl;

num.num(n);

cout << "Number of participation in wars" << endl;

cin >> war;

cout << "The number of enemies destroyed" << endl;

en.en(n);

cout << "Service life" << endl;

time.time(n);

cout << "Combat experience" << endl;

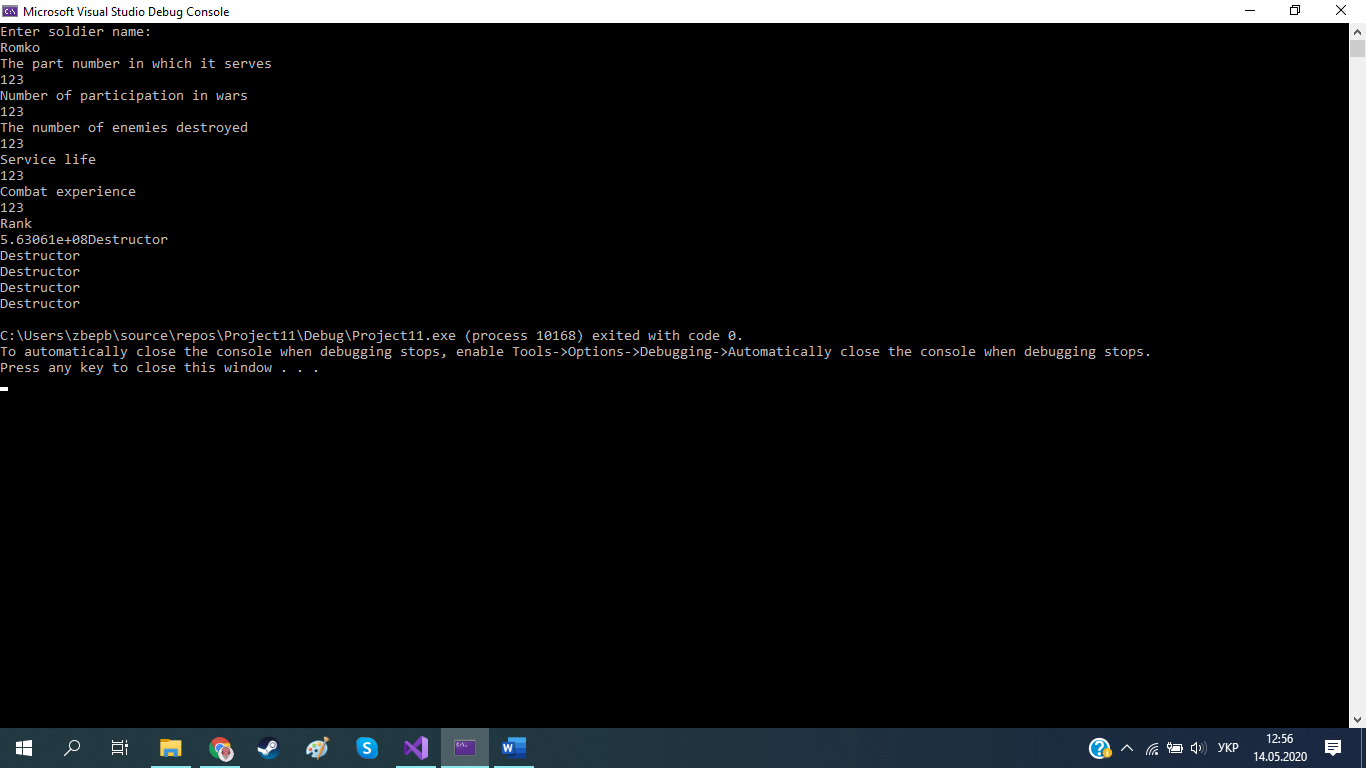
cin >> ser;

CSoldier rank(war, ser);

cout << "Rank" << endl << rank.rank();

}

**Вікно результату:**



**Висновок.** На цій лабораторній роботі я познайомився з перенавантаженням операторів.