Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №4 з дисципліни

«Основи програмування»

Варіант 14

Виконав студент Кашка Максим Сергійович

( прізвище, ім'я, по батькові)

Перевірив викладач Вітковська Ірина Іванівна

( прізвище, ім'я, по батькові)

Київ 2022

**Лабораторна робота №4**

**Перевантаження операторів**

**Мета –** визначити механізми створення класів та об’єктів зі створення перевантажених операторів.

**Індивідуальне завдання:**

Код задачі:

**Код на С++ :**

**class.h :**

**#pragma once**

**#include <iostream>**

**#include <set>**

**#include <iterator>**

**using namespace std;**

**class Set**

**{**

**int element;**

**set<int> set\_;**

**bool find\_num(int);**

**public:**

**Set();**

**Set(int);**

**//Plural(const Plural& other);**

**set<int> GetSet();**

**//void SetPlural(set<int>);**

**void outputSet();**

**Set operator + (const Set& other);**

**Set operator \* (const Set& other);**

**Set operator - (const Set& other);**

**};**

**Class.cpp:**

#include "class.h"

Set::Set()

{

element = 0;

set\_ = {};

}

Set::Set(int count)

{

cout << "Enter the number of elements of the set: ";

cin >> count;

cout << "Enter " << count << " elements of the set: " << endl;

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

{

cin >> element;

if (!find\_num(element))

{

set\_.insert(element);

}

else

{

cout << "The number is already in the plural. Try again: ";

cin >> element;

set\_.insert(element);

}

}

}

set<int> Set::GetSet()

{

return set\_;

}

void Set::outputSet()

{

copy(set\_.begin(), set\_.end(), ostream\_iterator<int>(cout, " "));

}

Set Set:: operator +(const Set& other)

{

Set C;

set<int>::iterator it;

for (it = other.set\_.begin(); it != other.set\_.end(); it++)

{

this->set\_.insert(\*it);

}

set<int>::iterator it1;

for (it1 = this->set\_.begin(); it1 != this->set\_.end(); it1++)

{

C.set\_.insert(\*it1);

}

cout << endl << "Operator + :" << endl;

C.outputSet();

return C;

}

Set Set:: operator \*(const Set& other)

{

Set C;

set<int>::iterator it1;

set<int>::iterator it2;

for (it1 = other.set\_.begin(); it1 != other.set\_.end(); it1++)

{

for (it2 = this->set\_.begin(); it2 != this->set\_.end(); it2++)

{

if (\*it2 == \*it1)

{

C.set\_.insert(\*it1);

}

}

}

cout << endl << "Operator \* :" << endl;

C.outputSet();

return C;

}

Set Set:: operator -(const Set& other)

{

Set C;

set<int>::iterator it;

set<int>::iterator it1;

for (it1 = this->set\_.begin(); it1 != this->set\_.end(); it1++)

{

C.set\_.insert(\*it1);

}

for (it = other.set\_.begin(); it != other.set\_.end(); it++)

{

C.set\_.erase(\*it);

}

cout << endl << "Operator - :" << endl;

C.outputSet();

return C;

}

bool Set::find\_num(int element)

{

set<int>::iterator it;

for (it = set\_.begin(); it != set\_.end(); it++)

{

if (\*it == element)

{

return true;

}

}

return false;

}

**Main.cpp**

#include "class.h"

int main()

{

Set A(0);

Set B(0);

cout << endl << "Set A: ";

A.outputSet();

cout << endl << "Set B: ";

B.outputSet();

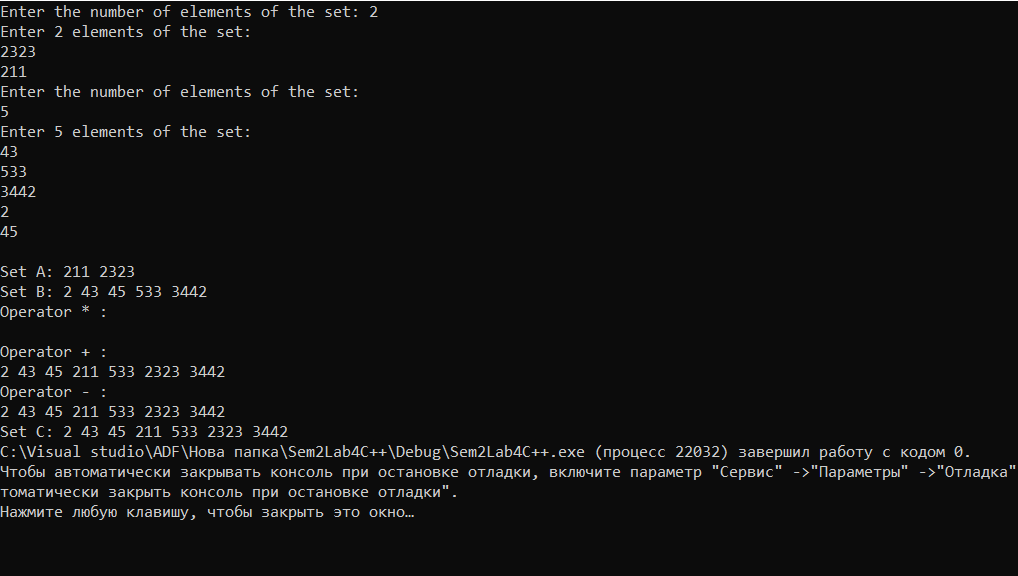
Set C = (A + B) - (A \* B);

cout << endl << "Set C: ";

C.outputSet();

}

**Скріншот результатів програми на C++ :**



**Висновок:**

Під час лабораторної роботи ми навчилися перевантажувати деякі оператори у класі (такі як +, -, \*).