

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 2.  
Модульне програмування»

«Перевантаження операторів»

Варіант 14

Виконав студент ІП-13, Замковий Дмитро Володимирович

(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна

( прізвище, ім'я, по батькові)

## Лабораторна робота 4

### Перевантаження операторів

**Мета заняття:** вивчити механізми створення класів з використанням перевантаження операторів (операцій).

### Завдання:

Розробити клас "Множина" для представлення множини цілих чисел. Реалізувати для нього декілька конструкторів, геттери, метод визначення приналежності заданного елемента множині. Перевантажити оператори "+", "\*" та "-" для знаходження об'єднання, перетину і різниці множин відповідно. Створити дві множини (A, B), використовуючи різні конструктори. На їх основі побудувати множину  $C = (A \cup B) \setminus (A \cap B)$

### Код програми:

Програма виконана на мові програмування C++

Файл Lab4.cpp:

```
#include <iostream>
#include "Plurar.h"
using namespace std;

int main()
{
    int leng;
    cout << "Enter leng of A : ";
    cin >> leng;
    Plural A(leng), B;
    cout << "A:";
    A.output_numbers();
    cout << "B:";
    B.output_numbers();
    Plural C = (A + B) - (A * B);
    cout << "C:";
    C.output_numbers();
}
```

Файл Plurar.h:

```
#pragma once
class Plural
{
    int* numbers;
    int size;

    void del_rep();
    bool find_num(int num);
}
```

```
public:
    Plural();
    Plural(int num);
    Plural(int len, int num);
    Plural(const Plural& obj);
    ~Plural();
    void output_numbers();
    Plural operator + (const Plural obj);
    Plural operator * (const Plural obj);
    Plural operator - (const Plural obj);
};
```

Файл Plurar.cpp:

```
#include "Plurar.h"
#include <iostream>
#include <iomanip>

Plural::Plural()
{
    int num;
    srand(time(NULL));
    size = 5 + rand() % 10;
    numbers = new int[size];
    for (int i = 0; i < size; i++)
    {
        num = rand() % 10;
        if (find_num(num))
        {
            i--;
        }
        else
        {
            numbers[i] = num;
        }
    }
}

Plural::Plural(int num)
{
    size = num;
    numbers = new int[size];
    int tmp;
    std::cout << "Enters plural: ";
    for (int i = 0; i < size; i++)
    {
        std::cin >> tmp;
        numbers[i] = tmp;
    }
    del_rep();
}

Plural::Plural(int len, int num)
{
    size = len;
    numbers = new int[size];
    for (int i = 0; i < size; i++)
    {
        numbers[i] = num;
    }
}
```

```

}

Plural::Plural(const Plural& obj)
{
    size = obj.size;
    numbers = new int[size];
    for (int i = 0; i < size; i++)
    {
        numbers[i] = obj.numbers[i];
    }
}

Plural::~~Plural()
{
    delete[] numbers;
}

void Plural::output_numbers()
{
    for (int i = 0; i < size; i++)
    {
        std::cout << std::setw(5) << numbers[i];
    }
    std::cout << std::endl;
}

Plural Plural::operator+(const Plural obj)
{
    Plural tmp(size + obj.size, 0);
    for (int i = 0; i < size; i++)
    {
        tmp.numbers[i] = numbers[i];
    }
    for (int i = 0; i < obj.size; i++)
    {
        tmp.numbers[i + size] = obj.numbers[i];
    }
    tmp.del_rep();
    return tmp;
}

Plural Plural::operator*(const Plural obj)
{
    int leng,
        n = 0;
    if (size >= obj.size)
    {
        leng = size;
    }
    else
    {
        leng = obj.size;
    }
    Plural tmp(leng, 0);

    for (int i = 0; i < obj.size; i++)
    {
        if (find_num(obj.numbers[i]))
        {
            tmp.numbers[n] = obj.numbers[i];
            n++;
        }
    }
    tmp.size = n;
    tmp.del_rep();
}

```

```

        return tmp;
    }

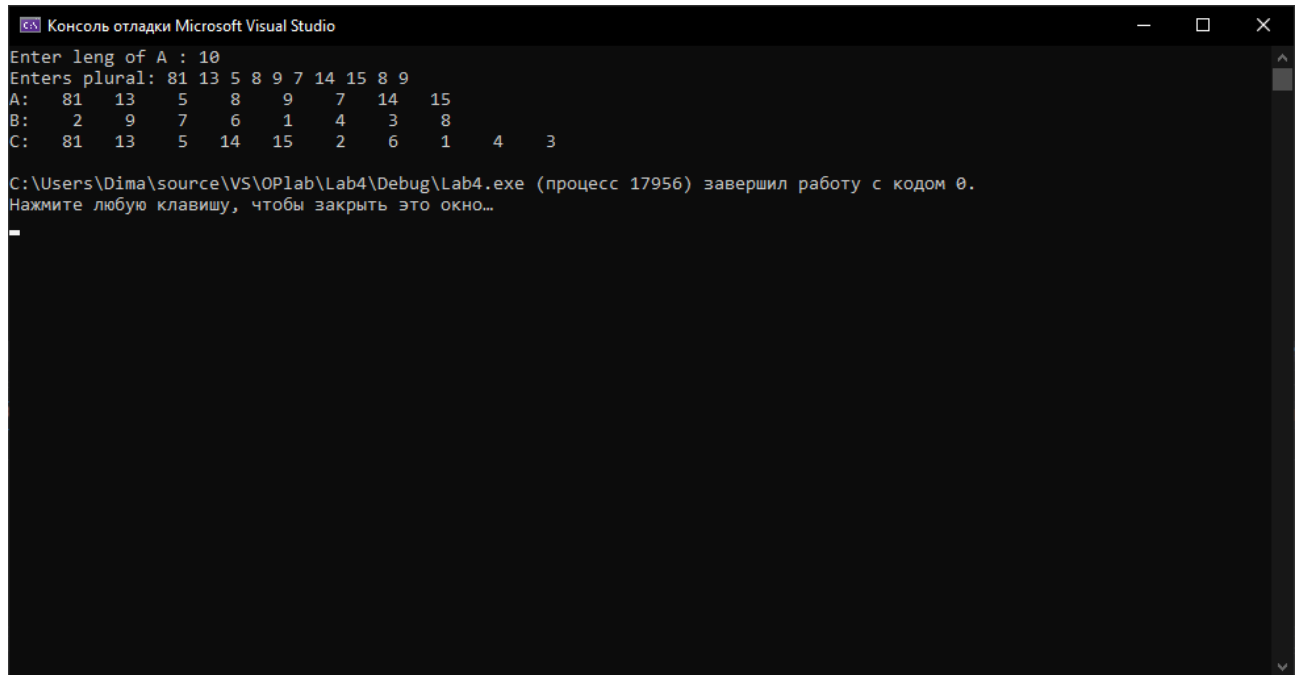
Plural Plural::operator-(const Plural obj)
{
    int n = 0;
    bool flag;
    Plural tmp(size, 0);
    for (int i = 0; i < size; i++)
    {
        flag = true;
        for (int j = 0; j < obj.size; j++)
        {
            if (obj.numbers[j] == numbers[i])
            {
                flag = false;
            }
        }
        if (flag)
        {
            tmp.numbers[n] = numbers[i];
            n++;
        }
    }
    tmp.size = n;
    return tmp;
}

void Plural::del_rep()
{
    for (int i = 0; i < size; i++)
    {
        for (int j = i + 1; j < size; j++)
        {
            if (numbers[i] == numbers[j])
            {
                for (int k = j; k < size - 1; k++)
                {
                    numbers[k] = numbers[k + 1];
                }
                size--;
            }
        }
    }
}

bool Plural::find_num(int num)
{
    for (int i = 0; i < size; i++)
    {
        if (numbers[i] == num)
        {
            return true;
        }
    }
    return false;
}

```

## Результат:



```
Консоль отладки Microsoft Visual Studio
Enter leng of A : 10
Enters plural: 81 13 5 8 9 7 14 15 8 9
A:  81  13   5   8   9   7  14  15
B:   2   9   7   6   1   4   3   8
C:  81  13   5  14  15   2   6   1   4   3

C:\Users\Dima\source\VS\OPlab\Lab4\Debug\Lab4.exe (процесс 17956) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...
```

Перевіримо правильність виконання програми.

1.  $A \cup B: \{81, 13, 5, 8, 9, 7, 14, 15, 2, 6, 1, 4, 3\}$
2.  $A \cap B: \{8, 9, 7\}$

А результатом буде різниця цих двох дій, що і відповідає тому результату, що видала програма.