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.

Методології програмування»

«Перевантаження операторів»

Варіант 31

Виконав студент ІП-11 Трикош Іван Володимирович

Перевірила Вітковська Ірина Іванівна

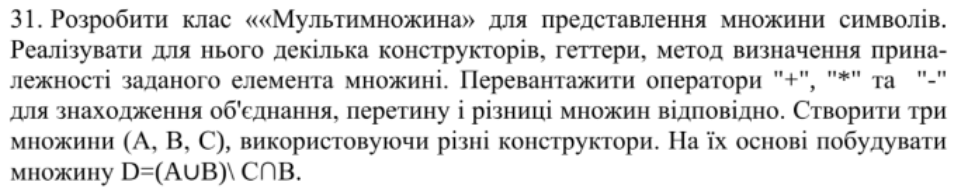
Київ 2022

**Лабораторна робота №4**

**Перевантаження операторів**

**Мета –** вивчити механізми створення класів з використанням перевантажених операторів (операцій).

Варіант 31.



**Постановка задачі.** Спочатку вводимо три множини символів у представленні рядка. Далі за заданою формулою обчислюємо четверту множину і виводимо її в консоль. Просимо у користувача ввести номер множини і символ, і перевіряємо, чи є цей символ у цій множині.

**Код та результат програми на С++:**

**Func.h**

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Multiset

{

private:

string array; // Множина символів

public:

Multiset(); // Створення пустої множини

Multiset(const char\*); // Створення множини за покажчиком на масив символів

Multiset(string&); // Створення множини за рядком

string& get\_multiset(); // Отимання множини

bool inset(char\*); // Перевірка, чи належить символ множині

void print(); // Вивід множини

void operator = (const Multiset&); // Присвоювання однієї множини іншій

Multiset operator + (Multiset&); // Об'єднання множин

Multiset operator \* (Multiset&); // Переріз множин

Multiset operator - (Multiset&); // Різниця множин

};

void input(string&, string&, string&); // Введення множин

void check(Multiset&, Multiset&, Multiset&, Multiset&); // Перевірка символу в множині

**Laba-C++.cpp**

#include "Func.h"

int main()

{

string array1; // Перша множина елементів

string array2; // Друга множина елементів

string array3; // Третя множина елементів

input(array1, array2, array3); // Вводимо множини

Multiset A(array1); // Перша множина

Multiset B = array2; // Друга множина

Multiset C(array3); // Третя множина

cout << "\nMultiset A:\n";

A.print(); // Виводимо множину

cout << "\nMultiset B:\n";

B.print(); // Виводимо множину

cout << "\nMultiset C:\n";

C.print(); // Виводимо множину

Multiset D("");

D = (A + B - C) \* B; // Остаточна множина

cout << "\nMultiset D:\n";

D.print(); // Виводимо множину

check(A, B, C, D); // Перевірка символу

}

**Func.cpp**

#include "Func.h"

void input(string& str1, string& str2, string& str3) // Вводимо множини символів як рядки

{

cout << "Input the first set (Example: akhsgjd,1n23 6): ";

getline(cin, str1);

cout << "Input the second set (Example: akhsg jd,1n236): ";

getline(cin, str2);

cout << "Input the third set (Example: ak hsgjd,1n236): ";

getline(cin, str3);

}

void check(Multiset& A, Multiset& B, Multiset& C, Multiset& D)

{

int n; // Номер множини

char m; // Символ для перевірки

cout << "\nInput the set number: ";

cin >> n;

cout << "Input the symbol: ";

cin >> m;

// Перевірка символу на наявність у множині

if (n == 1)

{

if (A.inset(&m))

{

cout << "The symbol is in the set!" << endl;

}

else

{

cout << "The symbol is not in the set!" << endl;

}

}

else if (n == 2)

{

if (B.inset(&m))

{

cout << "The symbol is in the set!" << endl;

}

else

{

cout << "The symbol is not in the set!" << endl;

}

}

else if (n == 3)

{

if (C.inset(&m))

{

cout << "The symbol is in the set!" << endl;

}

else

{

cout << "The symbol is not in the set!" << endl;

}

}

else if (n == 4)

{

if (D.inset(&m))

{

cout << "The symbol is in the set!" << endl;

}

else

{

cout << "The symbol is not in the set!" << endl;

}

}

else

{

cout << "The incorrect number!" << endl;

}

}

Multiset::Multiset()

{

array = ""; // Пуста множина

}

Multiset::Multiset(const char\* array)

{

this->array = array; // Присвоюємо рядку масив символів

}

Multiset::Multiset(string& array)

{

this->array = array; // Приствоюємо рядок рядку

}

string& Multiset::get\_multiset()

{

return array; // Повертаємо множину

}

bool Multiset::inset(char\* symbol) // Перевірка, чи належить символ множині

{

bool flag = false;

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

{

if (\*symbol == array[i])

{

flag = true;

}

}

return flag;

}

void Multiset::print() // Виведення множини символів

{

cout << "{ '";

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

{

cout << array[i];

if (i == array.size() - 1)

{

cout << "' }\n";

}

else

{

cout << "', '";

}

}

}

void Multiset::operator = (const Multiset& operand) // Приствоювання множини

{

this->array = operand.array;

}

Multiset Multiset::operator + (Multiset& operand)

{

Multiset temp; // Тимчасовий об'єкт класу

temp.array += operand.array; // Присвоюємо атрибуту тимчасового об'єкту множину

for (int i = 0; i < this->array.size(); i++)

{

if (operand.get\_multiset().find(this->array[i]) == -1) // Якщо символу немає в множині, то додаємо його в тимчасову множину

{

temp.array += this->array[i];

}

}

return temp; // Повертаємо тимчасовий об'єкт

}

Multiset Multiset::operator \* (Multiset& operand)

{

Multiset temp; // Тимчасовий об'єкт класу

for (int i = 0; i < this->array.size(); i++)

{

if (operand.get\_multiset().find(this->array[i]) != -1) // Якщо символ є у двох множинах, то присвоюємо його тимчасовій множині

{

temp.array += this->array[i];

}

}

return temp; // Повертаємо тимчасовий об'єкт

}

Multiset Multiset::operator - (Multiset& operand)

{

Multiset temp; // Тимчасовий об'єкт класу

for (int i = 0; i < this->array.size(); i++)

{

if (operand.get\_multiset().find(this->array[i]) == -1) // Якщо символ є у першій множині і відсутній у другій множині, то присвоюємо його тимчасовій множині

{

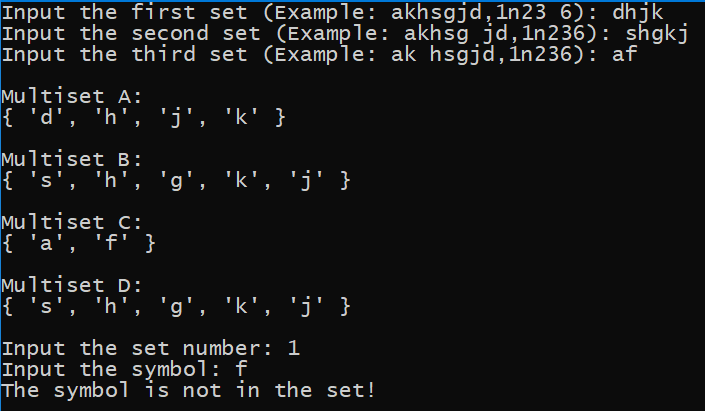
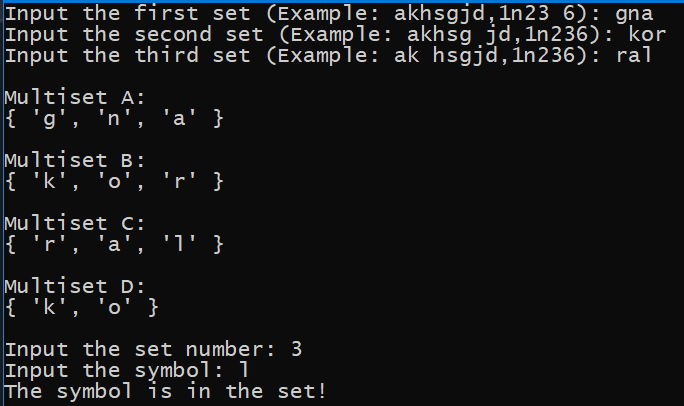
temp.array += this->array[i];

}

}

return temp; // Повертаємо тимчасовий об'єкт

}

****

**Висновок –** я вивчив механізми створення класів з використанням перевантажених операторів, покращив свої навички у роботі з класами. Я створив програму, яка здійснює операції об’єднання, перерізу, різниці множин. За допомогою цих операцій вона знаходить множину D за введеними трьома множинами.