Міністерство освіти та науки України

Інститут «Телекомунікації, радіоелектроніки та радіотехніки»

Изображение выглядит как текст, круг, Шрифт, эмблема

Автоматически созданное описание

Звіт

З лабораторної роботи №10

З дисципліни «Об`єктно-орієнтоване програмування»

На тему: «Конструювання класів з перезавантаженням операторів»

Виконав:

Студент групи АП-22

Іщак Д.А.

Прийняв:

Алтунін С. І

Львів 2024

**Мета:** створити, відлагодити та протестувати програму, у якій створити класи та реалізувати перевантаження операторів, які підтримують роботу та взаємодію об'єктів класу для обробки даних.

**Хід роботи:**

1. Вибрати завдання згідно свого варіанту у ДОДАТКУ 5.

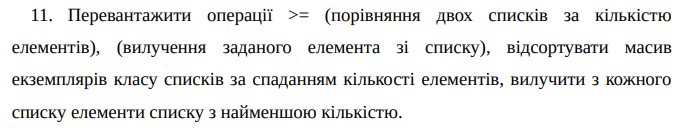
2. Ознайомитися із можливостями мови С++ для перевантаження операторів, навчитись використовувати їх при роботі з класами та об'єктами. Обдумати способи їх використання для створення програм.

3. Застосувати перевантаження операторів для досягнення цілей програми.

4. Розробити блок-схему алгоритму програми.

5. Реалізувати програму для вводу і виводу даних полів (членів) структур, вкладених у клас.

**Варіант завдання:**



**Код програми:**

**#include <iostream>**

**#include <vector>**

**#include <algorithm>**

**class List {**

**public:**

**std::vector<int> elements;**

**// Конструктор**

**List(std::vector<int> elems) : elements(elems) {}**

**// Перевантаження оператора >= для порівняння за кількістю елементів**

**bool operator>=(const List& other) const {**

**return elements.size() >= other.elements.size();**

**}**

**// Перевантаження оператора вилучення елемента**

**void operator-(int value) {**

**elements.erase(std::remove(elements.begin(), elements.end(), value), elements.end());**

**}**

**// Метод для сортування списку за кількістю елементів**

**static void sortBySize(std::vector<List>& lists) {**

**std::sort(lists.begin(), lists.end(), [](const List& a, const List& b) {**

**return a.elements.size() > b.elements.size();**

**});**

**}**

**// Метод для вилучення списку з найменшою кількістю елементів**

**static void removeSmallest(std::vector<List>& lists) {**

**auto minList = std::min\_element(lists.begin(), lists.end(), [](const List& a, const List& b) {**

**return a.elements.size() < b.elements.size();**

**});**

**if (minList != lists.end()) {**

**lists.erase(minList);**

**}**

**}**

**// Метод для виведення елементів списку**

**void print() const {**

**for (int el : elements) {**

**std::cout << el << " ";**

**}**

**std::cout << std::endl;**

**}**

**};**

**int main() {**

**List list1({1, 2, 3});**

**List list2({4, 5, 6, 7});**

**List list3({8, 9});**

**std::vector<List> lists = {list1, list2, list3};**

**// Перевірка оператора порівняння**

**if (list2 >= list1) {**

**std::cout << "List2 has more or equal elements than List1." << std::endl;**

**}**

**// Видалення елемента**

**list1 - 2;**

**list1.print(); // Виведе: 1 3**

**// Сортування за кількістю елементів**

**List::sortBySize(lists);**

**std::cout << "Sorted by size:" << std::endl;**

**for (const List& l : lists) {**

**l.print();**

**}**

**// Видалення списку з найменшою кількістю елементів**

**List::removeSmallest(lists);**

**std::cout << "After removing the smallest list:" << std::endl;**

**for (const List& l : lists) {**

**l.print();**

**}**

**return 0;**

**}**

**Результати виконання роботи:**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Блок-схема програми:**

Изображение выглядит как текст, чек, снимок экрана, Шрифт

Автоматически созданное описание

Висновок:

У цій лабораторній роботі реалізовано клас "Вектор" з конструкторами за замовчуванням і з параметрами, а також методами для введення даних через клавіатуру та перевірки їх коректності. Програма дозволяє створювати об'єкти класу, виконувати операції додавання та віднімання векторів, а також забезпечує виведення результатів. Завдяки перевірці валідності введених даних, програма гарантує коректність операцій та уникнення помилок, пов'язаних з некоректними значеннями координат.