ЛАБОРАТОРНА РОБОТА №8. ПЕРЕВАНТАЖЕННЯ ОПЕРАТОРІВ

**Тема.** Перевантаження операторів. Серіалізація.

**Мета.** Отримати знання про призначення операторів, визначити їх ролі в житті об’єкта та можливість перевизначення.

1 ВИМОГИ

**1.1 Розробник**

- Котенко Сергій Миколайович;

- Студент групи КІТ 102.8(а);

- 28-05-2019р..

**1.2 Загальне завдання**

*Поширити попередню лабораторну роботу наступним чином:*

- в базовому класі, та класі/класах-спадкоємцях перевантажити:

- оператор присвоювання;

- оператор порівняння (на вибір: == , < , > , >= , <= , != );

- оператор введення/виведення;

- в класі-списку перевантажити:

- оператор індексування ( [ ] );

- оператор введення/виведення з акцентом роботи в тому числі і з файлами. При цьому продовжувати використовувати регулярні вирази для валідації введених даних.

*Додаткові умови виконання завдання:*

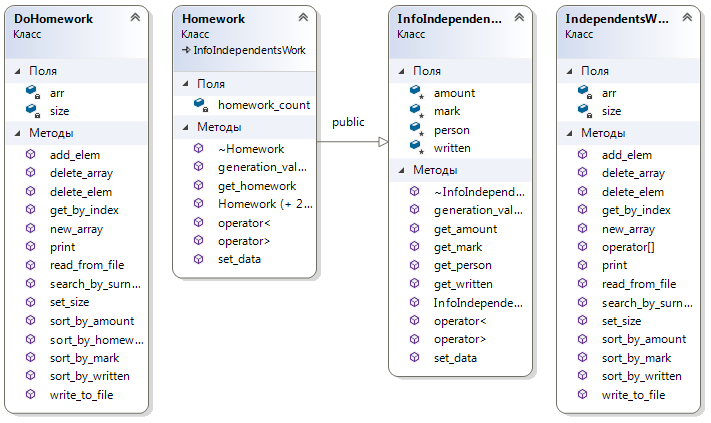
- продемонструвати відсутність витоків пам’яті;

- продемонструвати роботу розроблених методів за допомогою модульних тестів;

- не використовувати конструкцію «using namespace std;» , замість цього слід роботи «using» кожного необхідного класу:using std::string,using std::cout.

2 ОПИС ПРОГРАМИ

**2.1 Опис логічної структури**



*Діаграма класу InfoIndependentsWork*:

* ~InfoIndependentsWork - Деструктор класу;
* generation\_values – Генерація випадкових значень;
* get\_amount , get\_mark , get\_surname , get\_written - Отримання даних;
* InfoIndependentsWork - Конструктор класу;
* set\_data - Встановлення значень ;
* operator<> - Перевантаження операторів порівняння.

*Діаграма класу IndependentsWork :*

* add\_elem - Додавання нового елементу;
* delete\_array - Видалення масиву;
* delete\_elem - Видалення елементу;
* get\_by\_index - Отримання даних за індексом;
* new\_array - Створення масиву;
* print - Вивід даних на екран;
* read\_from\_file – Читання даних з файлу;
* search\_by\_surname – Пошук за прізвищем студента;
* set\_size - Отримання розміру для створення масиву;
* sort\_by\_amount, sort\_by\_mark, sort\_by\_written – Сортування даних за певним критеріем;
* write\_to\_file – Запис результату у файл.

*Діаграма класу Homework* :

* ~Homework - Деструктор класу;
* generation\_values – Генерація випадкових значень;
* get\_homework - Отримання даних;
* Homework - Конструктор класу;
* set\_data - Встановлення значень .
* operator<> - Перевантаження операторів порівняння.

*Діаграма класу DoHomework :*

* add\_elem - Додавання нового елементу;
* delete\_array - Видалення масиву;
* delete\_elem - Видалення елементу;
* get\_by\_index - Отримання даних за індексом;
* new\_array - Створення масиву;
* print - Вивід даних на екран;
* read\_from\_file – Читання даних з файлу;
* search\_by\_surname – Пошук за прізвищем студента;
* set\_size - Отримання розміру для створення масиву;
* sort\_by\_amount, sort\_by\_mark, sort\_by\_written, sort\_by\_homework – Сортування даних за певним критеріем;
* write\_to\_file – Запис результату у файл.

**2.2 Фрагменти коду**

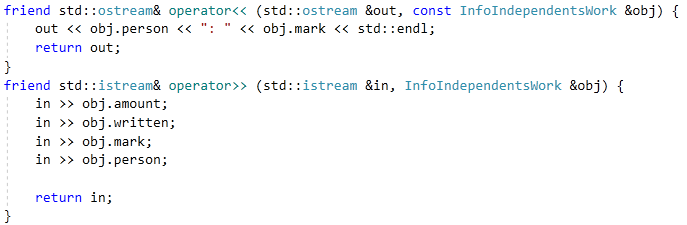


Рисунок 2.1 – Перевантаження операторів введення/виведення

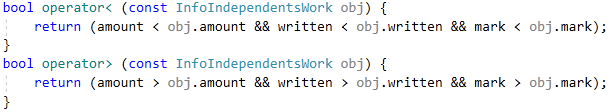


Рисунок 2.2 – Перевантаження операторів порівняння

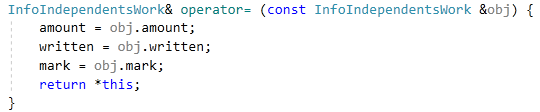


Рисунок 2.3 – Перевантаження оператора присвоювання

**3 ВАРІАНТИ ВИКОРИСТАННЯ**

**3.1 Опис поведінки програми**

Програма працює наступнім чином:

1) Введення користувачем кількості вивідних даних, створення масиву даних та виведення на екран

2) Вивід на екран можливих опцій програми , обирання користувачем опції:

2.0) Вихід з програми

2.1) Додавання нового елементу

2.2) Видалення певного елементу

2.3) Пошук за індексом

2.4) Пошук за прізвищем

2.5) Сортування за певним критерієм та у певному порядку

3) Перевірка на витоки пам’яті

**ВИСНОВОК**

В інтегрованому середовищі *Visual Studio* розроблена програма мовою С++. Виконання програми дозволяє продемонструвати коректність роботи програм для створення перевантажених операторів та їх використання.