ЛАБОРАТОРНА РОБОТА №6. СПАДКУВАННЯ

**Тема.** Класи. Спадкування.

**Мета.** Отримати знання про парадигму ООП – спадкування. Навчитися застосовувати отримані знання на практиці.

1 ВИМОГИ

**1.1 Розробник**

- Котенко Сергій Миколайович;

- Студент групи КІТ 102.8(а);

- 27-05-2019р..

**1.2 Загальне завдання**

*Модернізувати попередню лабораторну роботу шляхом:*

- додавання класу-спадкоємця, котрий буде поширювати функціонал «базового класу» у відповідності до індивідуального завдання;

- додавання ще одного класу-списку, що буде керувати лише елементами класу-спадкоємця;

- в функціях базового класу та класу-спадкоємця обов’язкове використання ключових слів final та override.

*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;

- в проекті не повинні використовуватися бібліотеки введення / виведення мови С, а також не повинні використовуватися рядки типу char\*.

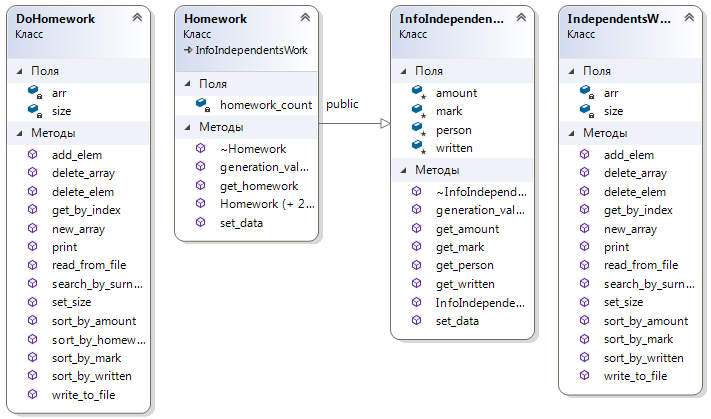
*Індивідуальне завдання:*

- В табл. 6.3 оберіть завдання для створення класу-спадкоємця у відповідності до номера у журналі групи.

10) Роботи студентів - кількість завдань для домашньої роботи

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 .

*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 .

*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**

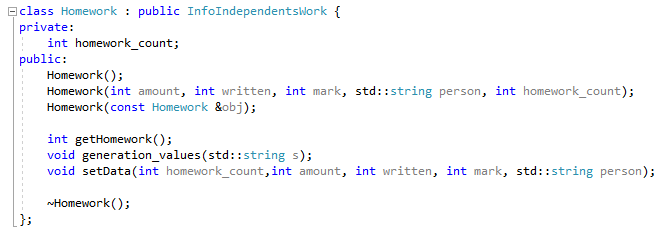


Рисунок 2.1 – Клас спадкоємець

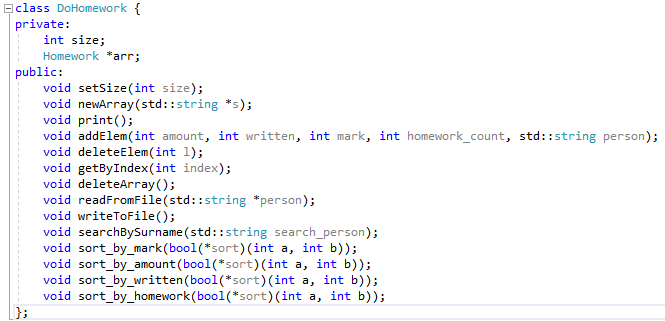


Рисунок 2.2 – Клас, що керує елементами класу спадкоємця

**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

**3.2 Ілюстрація роботи програми**

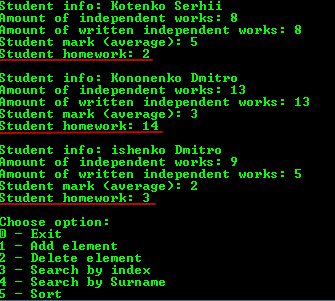


Рисунок 3.1 – Вивід даних з новою інформацією (кількість домашніх робіт)

**ВИСНОВОК**

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