ЛАБОРАТОРНА РОБОТА №7. ПОЛІМОРФІЗМ

**Тема.** Класи. Поліморфізм. Абстрактні класи.

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

1 ВИМОГИ

**1.1 Розробник**

- Котенко Сергій Миколайович;

- Студент групи КІТ 102.8(а);

- 02-06-2019р..

**1.2 Загальне завдання**

Модернізувати попередню лабораторну роботу шляхом:

- додавання ще одного класу-спадкоємця до базового класу. Поля обрати самостійно;

- базовий клас зробити абстрактним. Додати абстрактні поля;

- розроблені класи-списки поєднуються до одного таким чином, щоб він міг працювати як з базовим класом, так і з його спадкоємцями. При цьому, серед полів класу-списку повен бути лише один масив, що містить усі типи класів ієрархії. Оновити методи, що працюють з цим масивом.

Додаткові умови виконання завдання:

- продемонструвати відсутність витоків пам’яті;

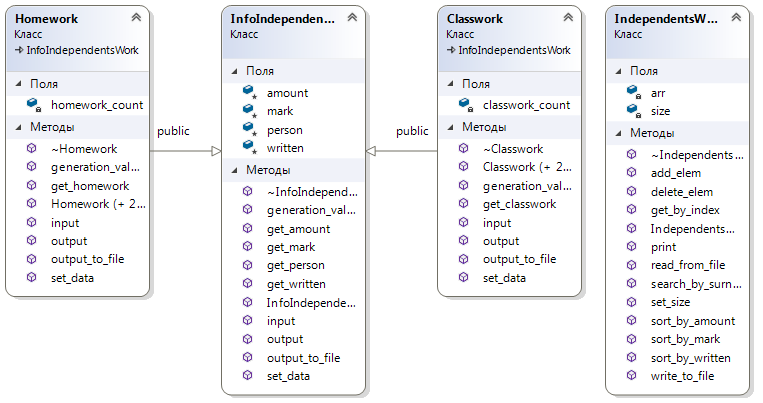
- продемонструвати роботу розроблений методів за допомогою модульних тестів;

- 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\*.

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;
* input – Введення нових даних;
* output – Вивід на екран;
* output\_to\_file – Вивід даних у файл;
* set\_data - Встановлення значень .

*Діаграма класу IndependentsWork :*

* ~IndependentsWork – Деструктор класу;
* add\_elem - Додавання нового елементу;
* delete\_elem - Видалення елементу;
* get\_by\_index - Отримання даних за індексом;
* IndependentsWork – Конструктор класу;
* 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;
* input – Введення нових даних;
* output – Вивід на екран;
* output\_to\_file – Вивід даних у файл;
* set\_data - Встановлення значень .

*Діаграма класу Classwork* :

* ~Classwork - Деструктор класу;
* generation\_values – Генерація випадкових значень;
* get\_homework - Отримання даних;
* Classwork - Конструктор класу;
* input – Введення нових даних;
* output – Вивід на екран;
* output\_to\_file – Вивід даних у файл;
* set\_data - Встановлення значень .

**2.2 Фрагменти коду**

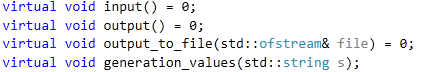


Рисунок 2.1 – Віртуальні методи основного класу

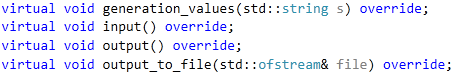


Рисунок 2.2 – Віртуальні методи в класах спадкоємцях

**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**

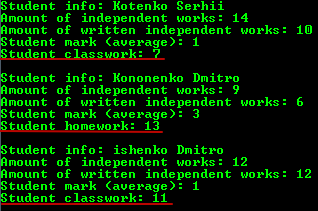


Рисунок 3.1 – Створенні дані з інформацією з класів спадкоємців

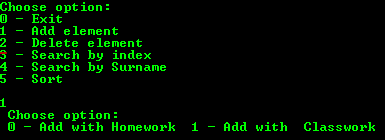


Рисунок 3.2 – Додавання елемента за певнім критерієм (певним об’єктом)

**ВИСНОВОК**

В інтегрованому середовищі *Visual Studio* розроблена програма мовою С++. Виконання програми дозволяє продемонструвати коректність роботи програм з використанням поліморфізму та коректність їх роботи.