Технології програмування

Лабораторна робота №3

**ООП**

**Мета роботи**: Використовуючи теоретичне підґрунтя про ООП у мові Python переробити програму телефонного довідника студентів використовуючи принципи ООП для формування відомостей про студентів.

**Завдання до лабораторної роботи**

Переробити функціональність телефонного довідника студентів групи, що був розроблений у Лабораторній роботі №2 використовуючи принципи ООП:

1. розробити клас Студент групи з відповідними атрибутами;
2. розробити клас Список групи, має містити не словники, як виконано в лабораторній роботі №2, а об’єкти класу Студент групи; додавання нового запису, видаленні існуючого чи зміна даних має бути виконана через методи класу Список групи.
3. розробити клас для роботи з файлами для зчитування початкової інформації про список групи та збереження інформації по завершенню програми.
4. список студентів має містити не словники, як виконано в лабораторній роботі №2, а об’єкти класу Студент групи;
5. описання всіх класів мають міститися в окремих файлах, що мають відповідні імена(наприклад Studen, StudentList, Utils)
6. основний функціонал програми має бути покритий Юніт тестами.

Текст програми разом зі звітом розмістити в директорії lab\_03. Директорію lab\_03 розмістити в директорії, що використовується для виконання практичних завдань по кожній лекції та має назву **TP-KB-22[1 or 2]-Name-Surname**.

Хід роботи

1. Використання принципів ООП:

* Створено клас Student, який представляє модель студента з атрибутами name, phone, email та group.
* Методи класу дозволяють оновлювати дані студента.

1. Управління списком студентів:

* У попередній роботі телефонний довідник був представлений у вигляді списку словників. У новій версії створено клас StudentList, що оперує списком об’єктів Student.
* Додані методи:
  + add\_student – для додавання нового студента.
  + delete\_student – для видалення студента за іменем.
  + update\_student – для оновлення інформації про студента.

1. Робота з файлами через окремий клас:

* Створено клас FileManager, який забезпечує зчитування даних із CSV-файлу та збереження списку студентів у файл.

1. Організація коду:

* Код розділено на окремі модулі: student.py, student\_list.py, file\_manager.py, main.py.
* Це забезпечує структурованість і полегшує тестування та підтримку.

1. Покращена логіка сортування:

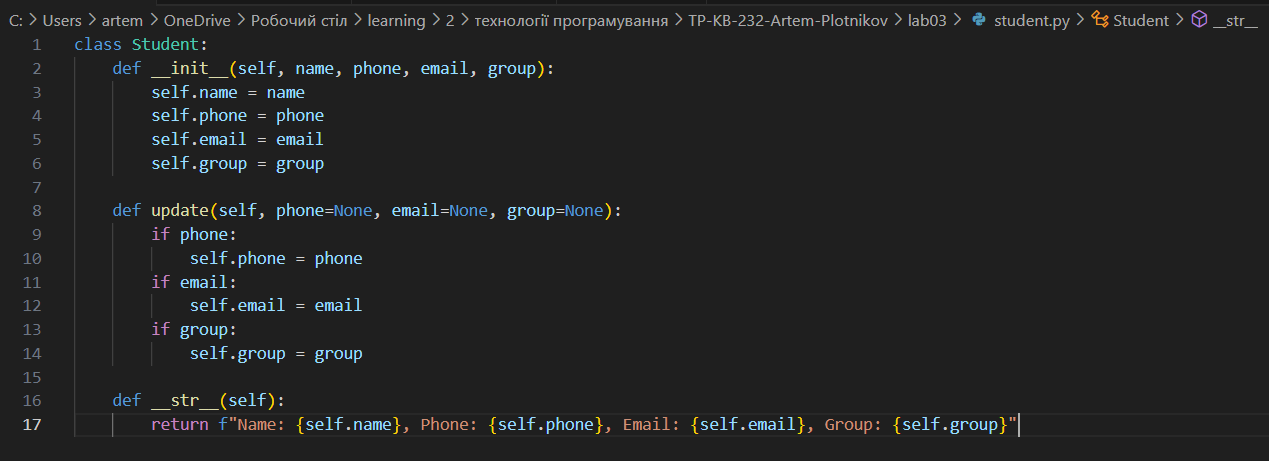
* Усі дії зі списком студентів автоматично сортують записи за іменем, що полегшує пошук і перегляд.

1. Підготовка до тестування:

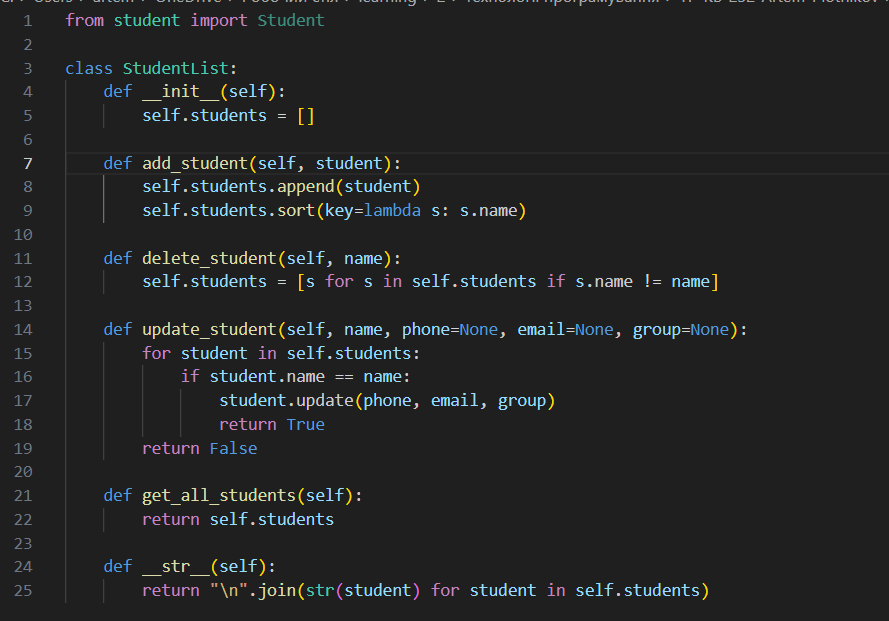
* Структура класів і методів дозволяє легко написати юніт-тести для кожного модуля.

Що було створено:

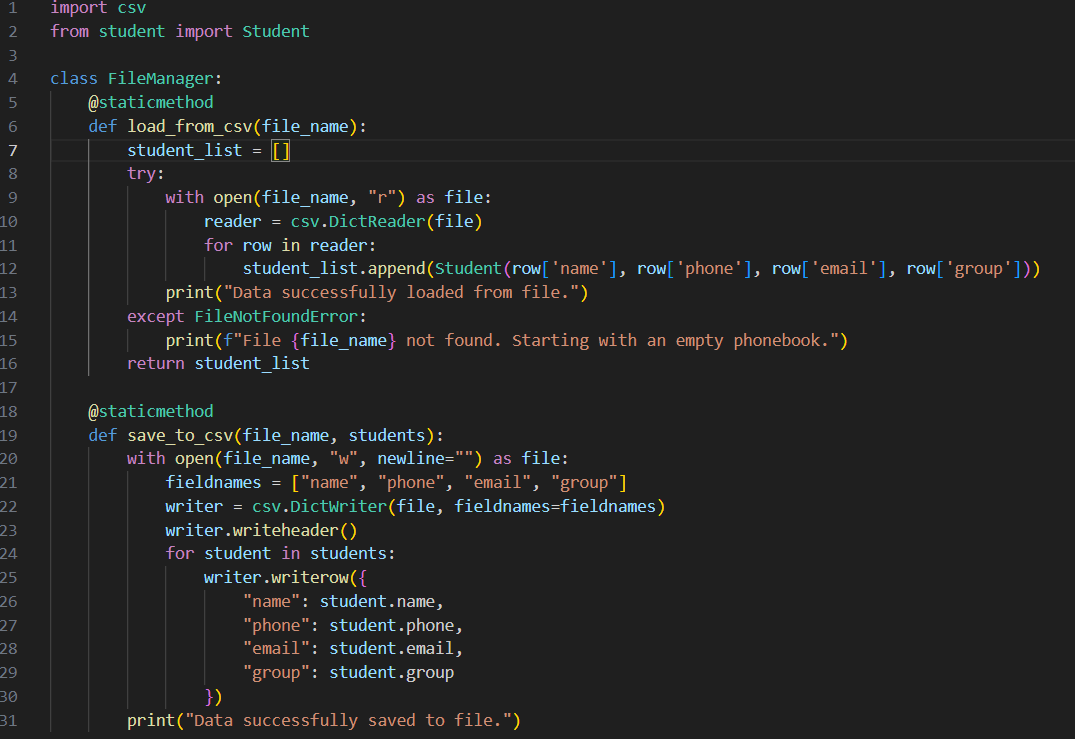
* Клас Student для опису сутності студента.



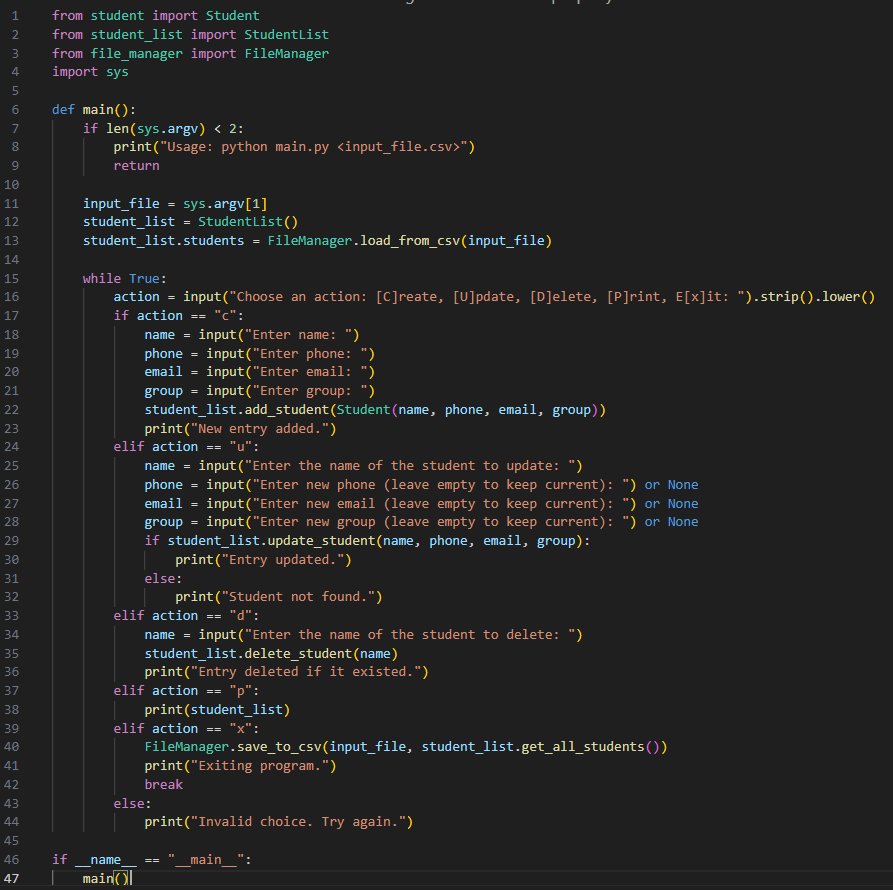
* Клас StudentList для управління списком студентів.



* Клас FileManager для роботи з файлами.



* Головний файл main.py для інтерактивної взаємодії з користувачем.



Приклад запуску коду:

