Технології програмування

Лабораторна робота №3

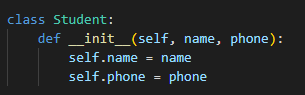
**Робота з файлами. Юніт тести.**

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

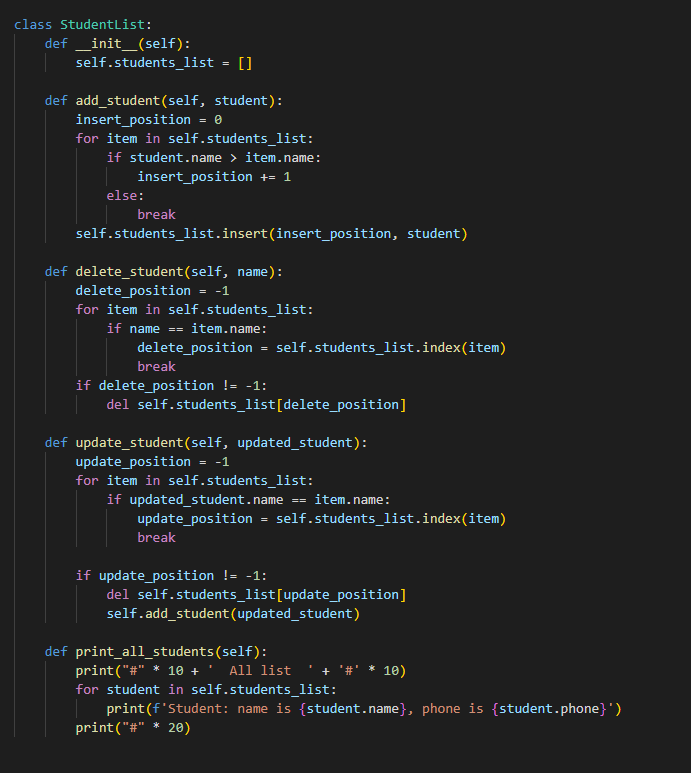
**Хід роботи**

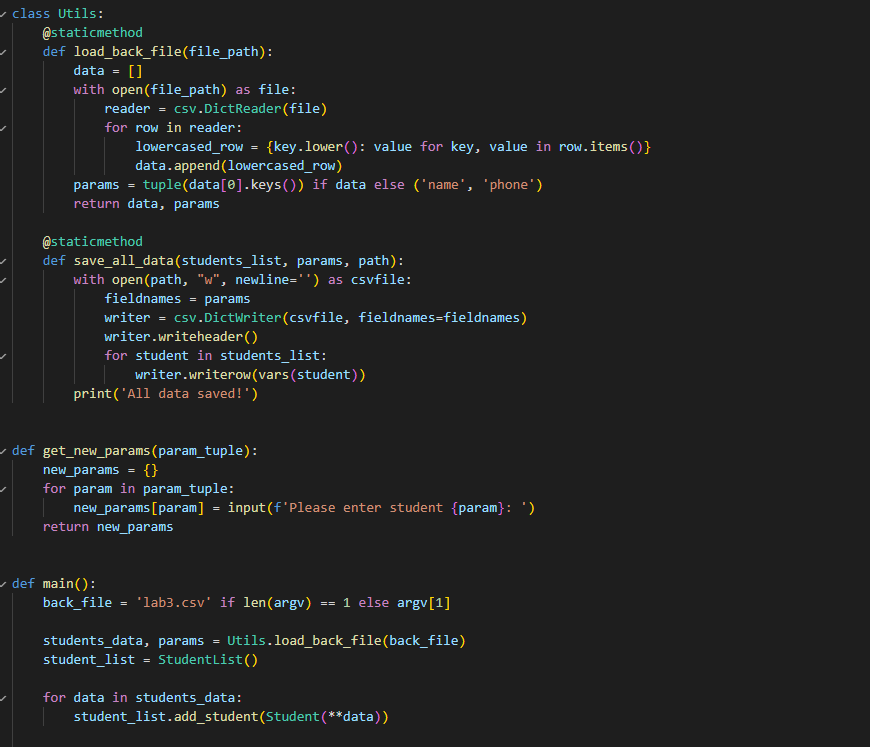
**Переробити** функціональність телефонного довідника студентів групи, що був розроблений у Лабораторній роботі №2 використовуючи принципи ООП:

1)розробити клас Студент групи з відповідними атрибутами;



2) Розроблення класу "Список групи" передбачає використання об'єктів класу "Студент групи" замість словників, як це було в лабораторній роботі №2. Додавання нового запису, видалення існуючого або зміна даних повинні виконуватися через методи цього класу. Ось перефразований варіант:  
  
Створіть клас "Список групи", який містить об'єкти класу "Студент групи" замість словників, як це було реалізовано у лабораторній роботі №2. Реалізуйте методи цього класу для додавання нового запису, видалення існуючого або зміни даних. Всі зазначені операції повинні виконуватися через відповідні 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)Cписок 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.



Висновок: Під час оптимізації програми телефонного довідника студентів були успішно використані принципи об'єктно-орієнтованого програмування (ООП) у мові Python. ООП спрямований на організацію коду у вигляді об'єктів, які об'єднують дані та методи для їх обробки.  
  
Перш за все, був створений клас Student, який відображає студента з атрибутами name та phone. Цей клас дозволяє більш ефективно та логічно моделювати інформацію про студентів, а використання екземплярів цього класу робить код більш зрозумілим та підтримуваним.  
  
Далі був створений клас StudentList, який представляє список студентів та включає методи для додавання, видалення та оновлення даних студентів. Використання цього класу дозволяє групувати функціонал, пов'язаний із списком студентів, що полегшує обслуговування та підтримання коду.