Технології програмування

Лабораторна робота №1

**Списки. Словники. Кортежі**

**Мета роботи**: Використовуючи теоретичне підґрунтя про складні структури даних Списки, Словники Кортежі, та використовуючи існуючий код, доробити програму додавши функціонал, що буде вказано в завданні до лабораторної роботи.

**Завдання до лабораторної роботи**

Реалізувати **відсортований** телефонний довідник студентів групи.

Для виконання задання надано частину готового функціоналу, яка розміщена в одній директорії з завданням до лабораторної роботи та має назву **lab\_01.py.**

Частина готового функціоналу реалізує безкінечний цикл запитів до користувача. Типи запитів: додати нового студента, змінити данні про існуючого студента, видалити запис, роздрукувати всю таблицю та вихід із програми. Реалізован функціонал додавання нового запису та видалення існуючого. Всі дії відбуваються з **відсортованим** списком студентів.

Перед виконанням роботи слід ознайомитись з існуючим функціоналом.

Необхідно розширити відомості про студента до 4х полів. На даний час використовується лише два поля (name та phone).

Необхідно переробити існуючий функціонал враховуючи розширення відомості про студента до 4х полів.

Необхідно реалізувати з нуля функціонал зміни інформації про студента враховуючи той факт, що вже існує реалізація додавання нового запису та видалення існуючого. **При зміні інформації про студента список має залишатись відсортованим**.

Текст програми разом зі звітом розмістити в каталозі lab\_01. Каталог lab\_01 розмістити в каталозі, що використовується для виконання практичних завдань по кожній лекції та має назву **TP-KB-22[1 or 2]-Name-Surname**.

#### 1.Опис програми

Програма працює з початковим списком студентів, представленим у вигляді масиву словників. Кожен студент має такі поля:

* name — ім'я студента.
* phone — номер телефону.
* group — назва групи.
* avarage mark — середній бал.

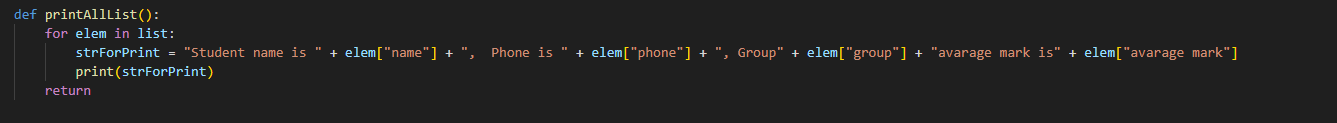
**Кодові функції:**

1. **printAllList()**

Призначення: Виводить список студентів у зручному форматі.

Алгоритм:

* + 1. Ітерація по кожному елементу списку.
    2. Формування рядка зі значеннями полів.
    3. Вивід результату в консоль.



**Приклад результату**:

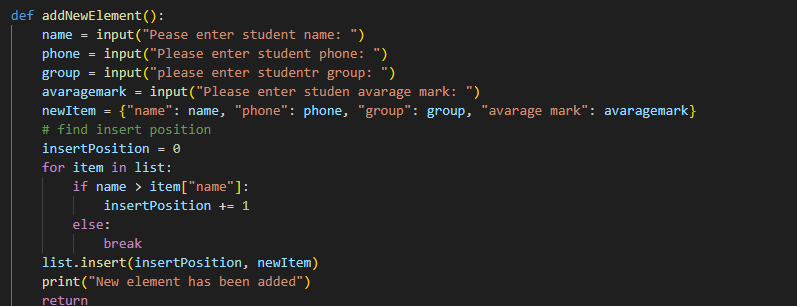
Student name is Bob, Phone is 0631234567, Group KB-231, average mark is 81

1. **addNewElement()**

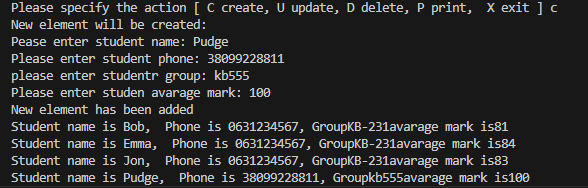
Призначення: Додає нового студента до списку.

Алгоритм:

* 1. Зчитування даних студента від користувача.
  2. Визначення позиції вставки нового елемента (щоб список залишався відсортованим за іменем).
  3. Додавання студента у відповідну позицію.
  4. Вивід повідомлення про успішне додавання.



**Приклад результату**:

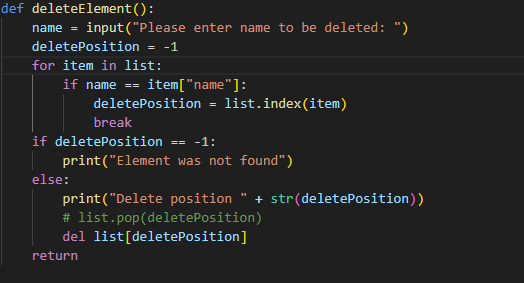


1. **deleteElement()**

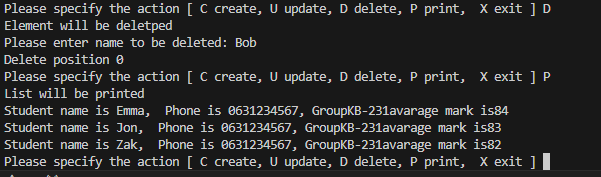
Призначення: Видаляє студента за іменем.

Алгоритм:

* 1. Отримання імені студента від користувача.
  2. Перевірка наявності студента у списку.
  3. Видалення елемента, якщо ім'я знайдено.
  4. Вивід повідомлення про успіх або помилку, якщо елемент не знайдено.



**Приклад результату**:

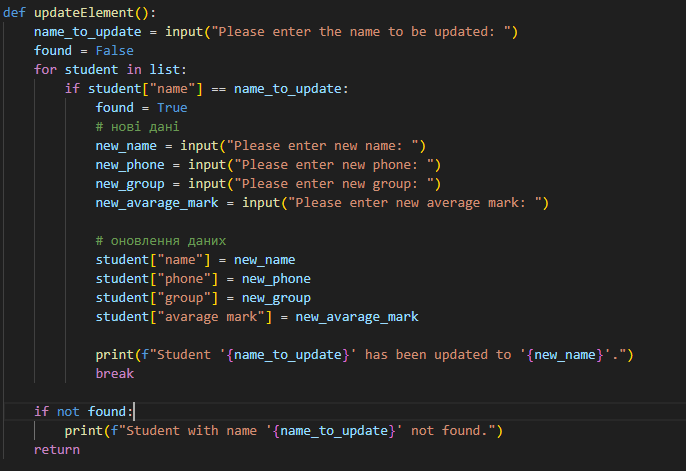


4. **updateElement()**

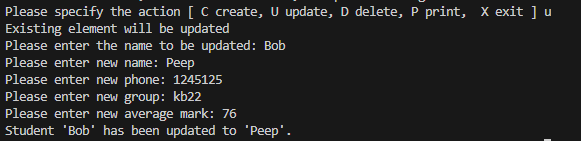
Призначення: Оновлює інформацію про студента за його ім'ям.

Алгоритм:

* 1. Зчитування імені студента для оновлення.
  2. Перевірка наявності студента.
  3. Заміна старих значень на нові, які вводить користувач.
  4. Вивід результату оновлення.



**Приклад результату**:

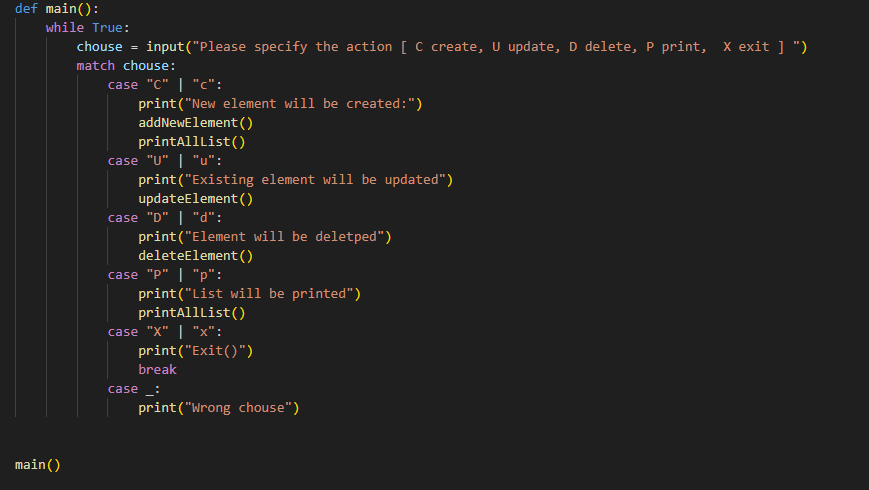


* 1. **main()**

Призначення: Меню взаємодії з користувачем.

Алгоритм:

* 1. Виведення підказки з доступними опціями: C (Create), U (Update), D (Delete), P (Print), X (Exit).
  2. Виклик відповідних функцій залежно від вибору користувача.
  3. Завершення роботи при виборі опції X.



**Приклад результату:**

**Висновок**

Програма реалізує базові операції управління списком студентів і демонструє навички роботи з масивами, функціями та інтерфейсом користувача.