**Лабораторна робота №1**

Завдання: Маючи вже файл lab\_01.py з початковим кодом, виконати такі завдання:

1) Розширити відомості про студента до 4х полів(зараз є лише два поля (name та phone)).

2) Переробити існуючий функціонал враховуючи розширення відомості про студента до 4х полів.

3) Реалізувати з нуля функціонал зміни інформації(функція updateElement()) про студента враховуючи той факт, що вже існує реалізація додавання нового запису та видалення існуючого, але при цьому список має залишатися сортованим.

Виконання завданння:

Щоб виконати завдання перше, спочатку вручну додамо нові і заповнимо в список ці поля(email та group), щоб не було конфліктів. Далі змінимо трохи функцію printAllList: додамо на вивід нові поля.

Тепер друге завдання, змінимо функцію addNewElement() додавши запити до користувача ще для двох нових полів. І відповідно змінив вміст змінни newItem, щоб всі поля не два, чотири додавалися до масиву словників.

Далі третє завдання, реалізуємо оновлення даних в функції updateElement(). Робимо змінну з списком list\_students глобальною, щоб посилатися на цю змінну, що поза функцією. Далі запитаємо користувача про ім’я дані, якого він хоче оновити і на яке ім’я змінити попереднє ім’я. Потім напишемо ще змінну deletePosition і присвоїмо їй значення -1, ця змінна знадобиться нам для видалення даних студента і заміною на нові дані. Далі пропишемо цикл, який перебирає студентів у списку list\_students і в цьому циклу пропишемо умови, що якщо ім’я на яке ми хочемо змінити(name\_update) попереднє ім’я(name) дорівнює поточному імені елементу(item[“name”]), якого ми перебираємо, і ім’я, яке хочемо змінити дорівнює поточному імені елементу, якого ми перебираємо, то далі запитуємо інші дані, які необхідні для оновлення даних студента, потім присвоюємо нові дані(значення) по характеристикам(ключам) студента. Виводимо повідомлення про оновлення даних і змінній deletePosition значення -2, для того, щоб позначити, що ми оновлюємо елемент. Якщо ж ця умова не виконується, то я прописав іншу далі умову, що якщо ім’я(name), якому дані ми хочемо оновити дорівнює поточному елементу(item[“name”]), який перебираємо і водночас ім’я(name\_update), на яке хочемо змінити попереднє ім’я, не дорівнює поточному елементу, то шукаємо індекс цього елемента в масиві і присвоюємо результат в змінну deletePosition і ще перевірка щодо цієї змінної, якщо deletePosition залишився такий самий(-1), то ми виводимо повідомлення про не знайдення студента в списку, але якщо позиція знайдена, то ми повідомляємо позицію по якій видаляємо і видаляємо елемент, далі запитуємо нові значення для характеристик студента(телефон, е-пошта, група). Потім всі ці отримані значення складаємо в один словник newItem і шукаємо позицію для вставки елемента роблячи перевірки по імені чи більше, чи менше від нашого нового імені студента. Рахунок позиції записуємо insertPosition. Далі після завершення перевірок, ми за допомогою insert() вставляємо новий елемент, виводимо повідомлення, що елемент встановлено. Згодом, коли цикл завершиться, ми робимо перевірку щодо змінної deletePosition, якщо вона -1, то це означає, що студента не було знайдено. Це все щодо реалізації функції updateElement(). Не забудемо викликати цю функцію в main(), де це необхідно.

Код завдання:

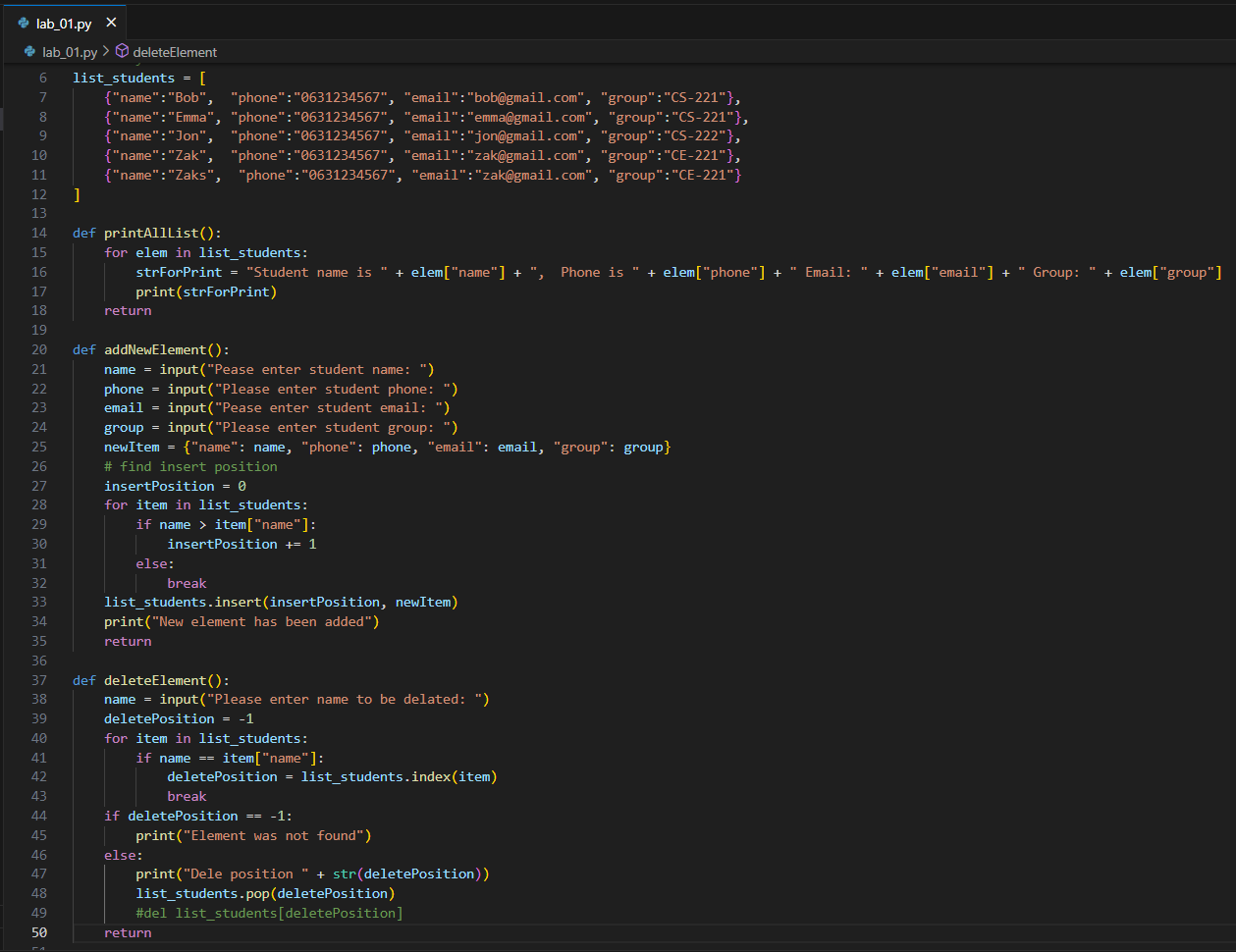


Рис. 1. Код файлу lab\_01.py.

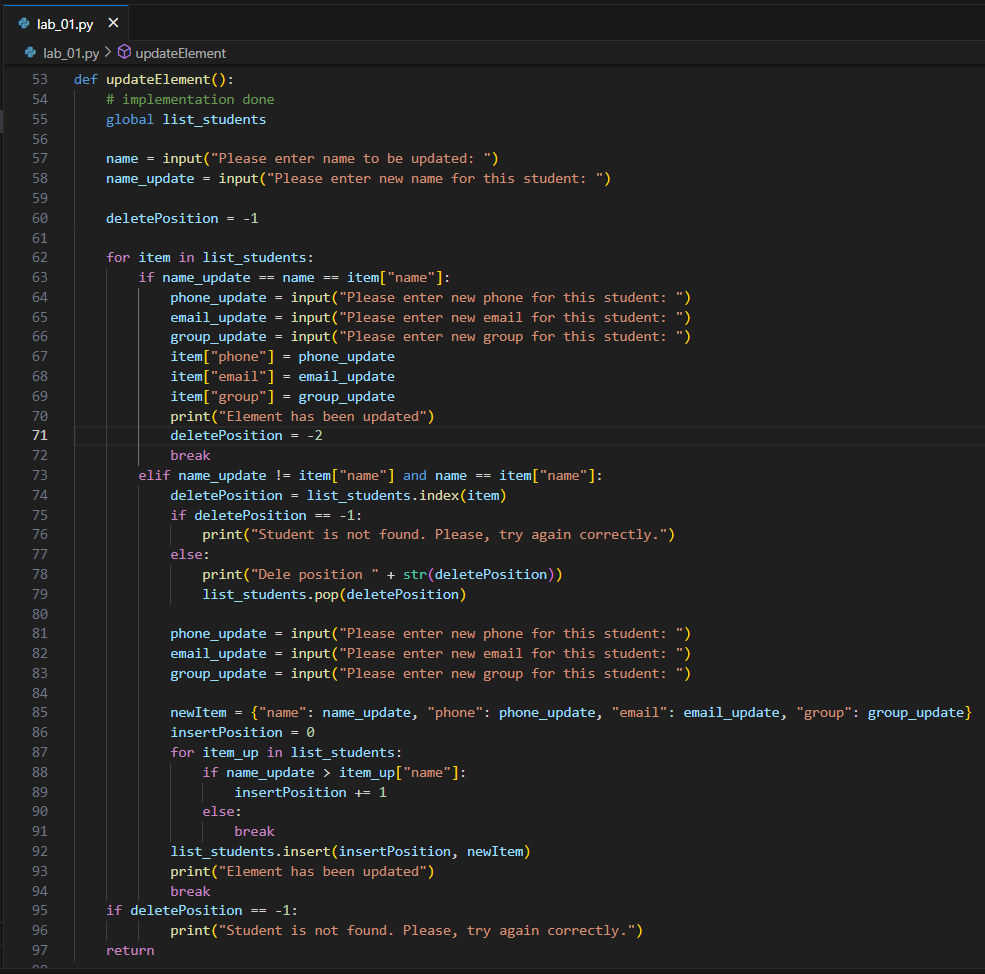


Рис. 2. Продовження коду файлу lab\_01.py.

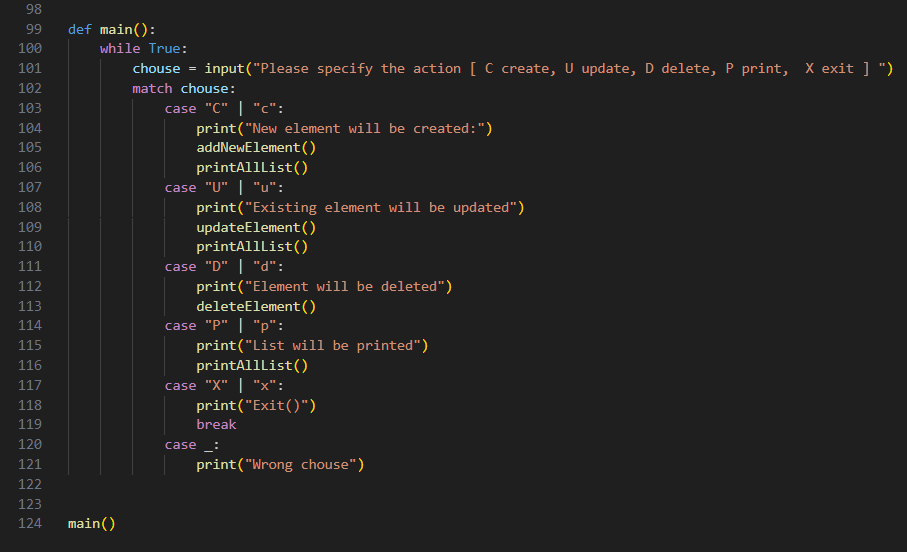


Рис. 3. Продовження коду файлу lab\_01.py.

GitHub:

Посилання на GitHub на цей файл: <https://github.com/danylosykylinda/TP-KB-221-Danylo-Sykylinda/blob/main/lab_01/lab_01.py>.

Скріншот коду з GitHub:

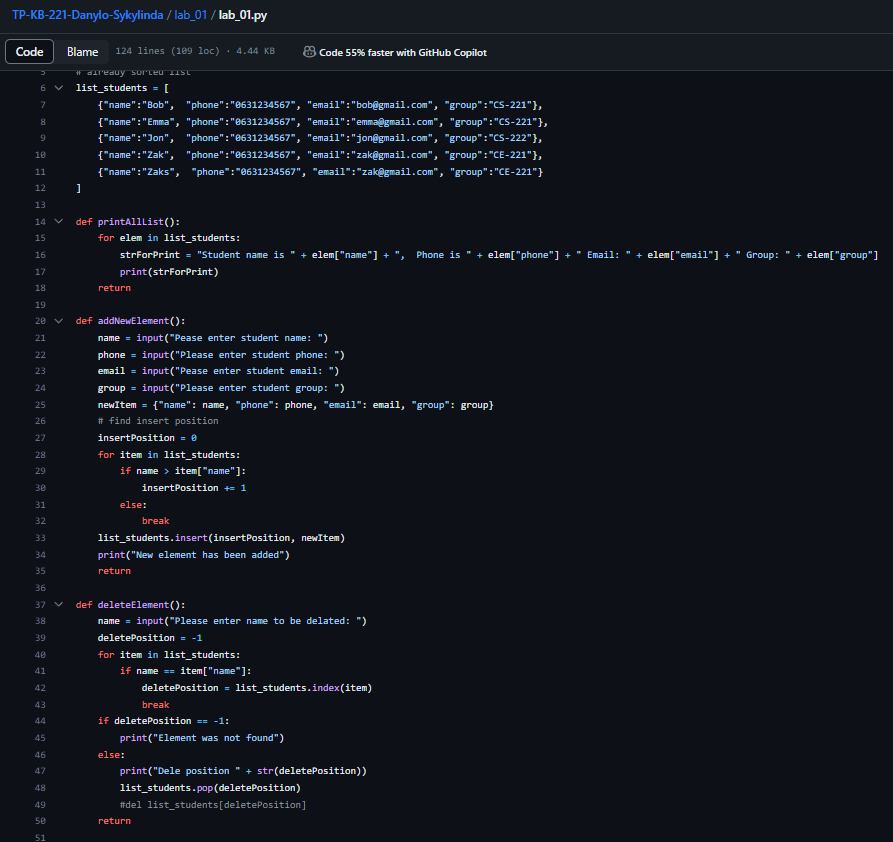


Рис. 4. Код файлу lab\_01.py.

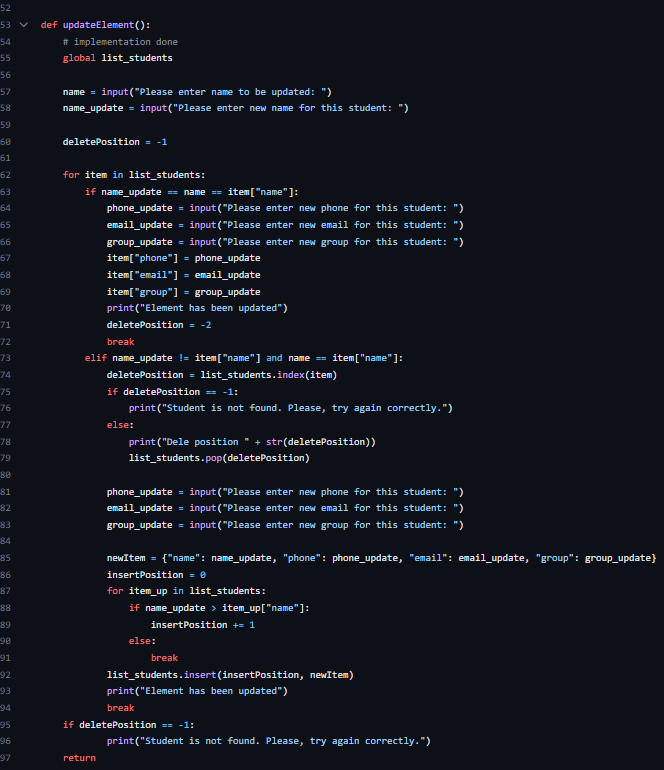


Рис. 5. Продовження коду файлу lab\_01.py.

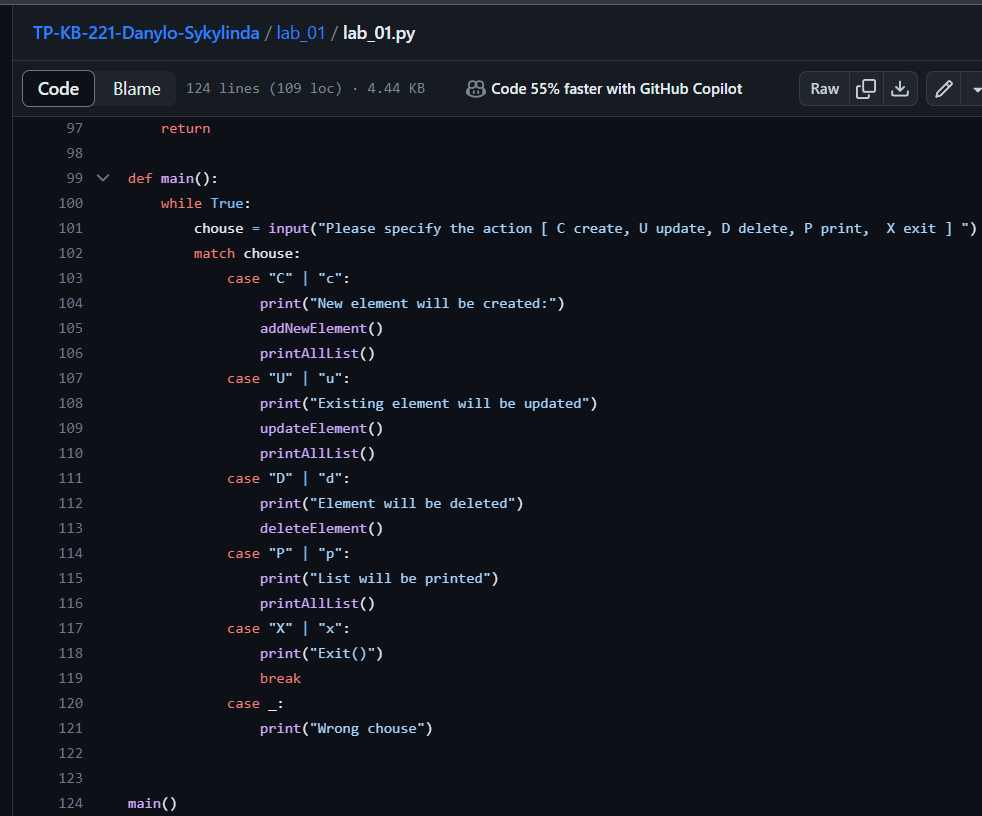


Рис. 6. Продовження коду файлу lab\_01.py.