**Лабораторна робота №1**

**Тема**: Списки. Словники. Кортежі.

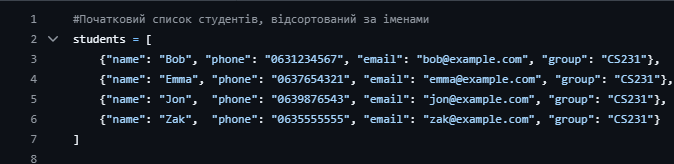
**Мета**: Використовуючи теоретичне підґрунтя про складні структури даних Списки, Словники Кортежі, та використовуючи існуючий код, доробити програму додавши функціонал, що буде вказано в завданні до лабораторної роботи.

**Завдання до лабораторної роботи**

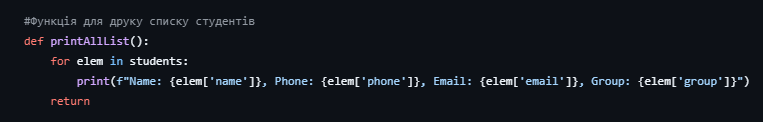
1. Розширити відомості про студента до 4х полів. На даний час використовується лише два поля (name та phone).
2. Переробити існуючий функціонал враховуючи розширення відомості про студента до 4х полів.
3. Реалізувати з нуля функціонал зміни інформації про студента враховуючи той факт, що вже існує реалізація додавання нового запису та видалення існуючого. **При зміні інформації про студента список має залишатись відсортованим**.

**Хід роботи**

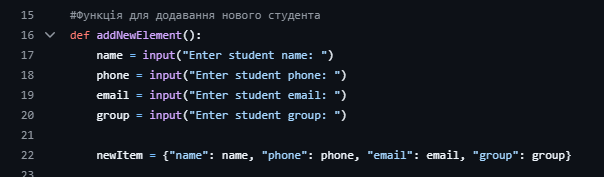
1. Спочатку створено початковий список студентів, який містить інформацію про їхні імена, номери телефонів, електронні адреси та групи. Список відсортований за іменами.



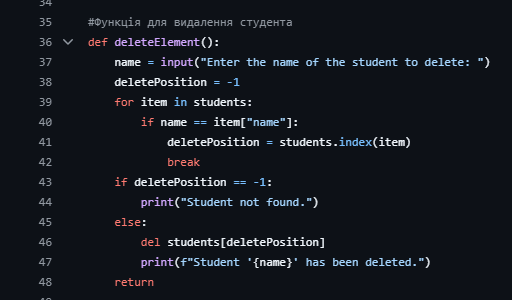
2. Для виведення списку студентів створена функція printAllList. Вона проходить по кожному елементу списку і друкує його у форматі: ім'я, телефон, email та група.



3. Додавання нових студентів реалізовано у функції addNewElement. Користувач вводить дані нового студента, а програма визначає правильне місце для його вставки, щоб список залишався відсортованим за іменами. Після додавання виводиться повідомлення про успішне завершення операції.



4. Функція deleteElement дозволяє видаляти студентів зі списку. Користувач вводить ім'я студента, якого потрібно видалити. Якщо такий студент знайдений, його запис видаляється. Якщо ж ім'я відсутнє у списку, користувач отримує відповідне повідомлення.



5. Для оновлення інформації про студента використовується функція updateElement. Користувач вводить ім'я студента, чиї дані потрібно змінити. Якщо студент знайдений, програма дозволяє змінити номер телефону, email чи групу. Якщо певне поле залишено порожнім, старе значення залишається без змін. Оновлена інформація додається у список, зберігаючи порядок сортування.



6. Головна функція main організовує роботу програми. Вона пропонує користувачеві вибрати дію: додати студента, оновити дані, видалити запис, переглянути список чи завершити роботу. Для обробки вибору використовується конструкція match-case.



7. Програма працює у безперервному циклі, доки користувач не вибере опцію виходу (X). При цьому виводиться повідомлення про завершення роботи.

Текст програми:

|  |
| --- |
| #Початковий список студентів, відсортований за іменами  students = [  {"name": "Bob", "phone": "0631234567", "email": "bob@example.com", "group": "CS231"},  {"name": "Emma", "phone": "0637654321", "email": "emma@example.com", "group": "CS231"},  {"name": "Jon", "phone": "0639876543", "email": "jon@example.com", "group": "CS231"},  {"name": "Zak", "phone": "0635555555", "email": "zak@example.com", "group": "CS231"}  ]  #Функція для друку списку студентів  def printAllList():  for elem in students:  print(f"Name: {elem['name']}, Phone: {elem['phone']}, Email: {elem['email']}, Group: {elem['group']}")  return  #Функція для додавання нового студента  def addNewElement():  name = input("Enter student name: ")  phone = input("Enter student phone: ")  email = input("Enter student email: ")  group = input("Enter student group: ")  newItem = {"name": name, "phone": phone, "email": email, "group": group}  #Пошук позиції для вставки (щоб список залишався відсортованим)  insertPosition = 0  for item in students:  if name > item["name"]:  insertPosition += 1  else:  break  students.insert(insertPosition, newItem)  print("New student has been added.")  return  #Функція для видалення студента  def deleteElement():  name = input("Enter the name of the student to delete: ")  deletePosition = -1  for item in students:  if name == item["name"]:  deletePosition = students.index(item)  break  if deletePosition == -1:  print("Student not found.")  else:  del students[deletePosition]  print(f"Student '{name}' has been deleted.")  return  #Функція для зміни інформації про студента  def updateElement():  name = input("Enter the name of the student to update: ")  updatePosition = -1  for item in students:  if name == item["name"]:  updatePosition = students.index(item)  break  if updatePosition == -1:  print("Student not found.")  else:  print("Enter new details for the student (leave empty to keep current values):")  phone = input(f"Current phone ({students[updatePosition]['phone']}): ") or students[updatePosition]["phone"]  email = input(f"Current email ({students[updatePosition]['email']}): ") or students[updatePosition]["email"]  group = input(f"Current group ({students[updatePosition]['group']}): ") or students[updatePosition]["group"]  #Оновлення даних  students.pop(updatePosition)  newItem = {"name": name, "phone": phone, "email": email, "group": group}  #Вставка оновленого елемента у відповідне місце (зберігаємо сортування)  insertPosition = 0  for item in students:  if name > item["name"]:  insertPosition += 1  else:  break  students.insert(insertPosition, newItem)  print(f"Student '{name}' has been updated.")  return  #Основна програма  def main():  while True:  choice = input("Specify action [C create, U update, D delete, P print, X exit]: ")  match choice.lower():  case "c":  print("Adding a new student...")  addNewElement()  case "u":  print("Updating an existing student...")  updateElement()  case "d":  print("Deleting a student...")  deleteElement()  case "p":  print("Student list:")  printAllList()  case "x":  print("Exiting the program. Goodbye!")  break  case \_:  print("Invalid choice. Please try again.")  main() |

**Висновок:** Під час виконання роботи я використовуючи теоретичне підґрунтя про складні структури даних списки, кортежі, словники та використовуючи існуючий код, додаємо вказаний функціонал.