# **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.

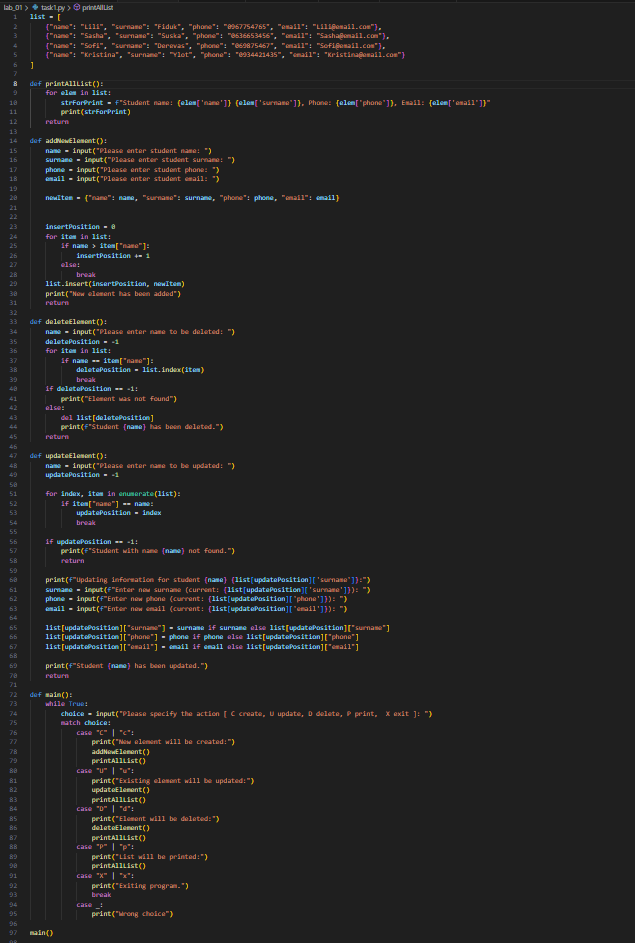
Необхідно розширити відомості про студента до 4х полів. На даний час використовується лише два поля (name та phone).

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

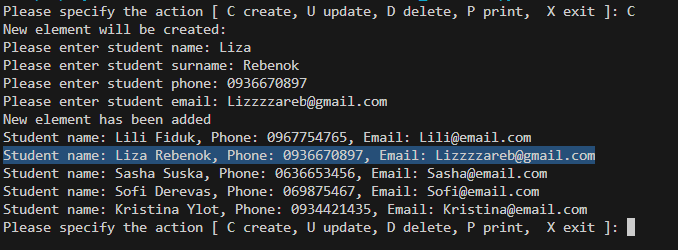
Необхідно реалізувати з нуля функціонал зміни інформації про студента враховуючи той факт, що вже існує реалізація додавання нового запису та видалення існуючого. **При зміні інформації про студента список має залишатись відсортованим**.

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-232-Liza-Rebenok**.

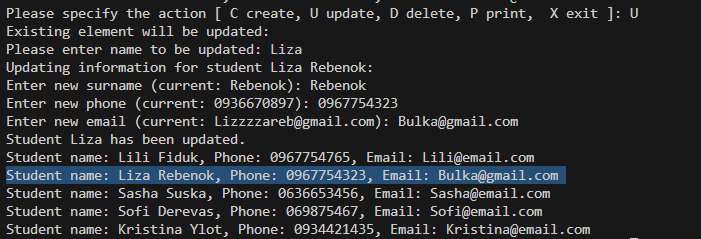
Хід виконання завдання:



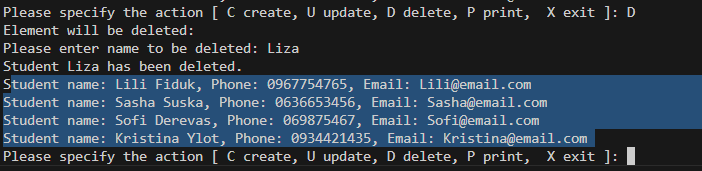
Вивід create:



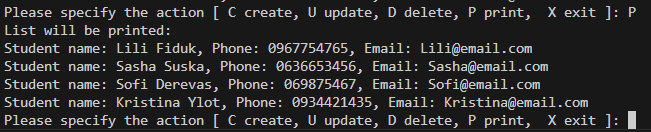
Вивід update:



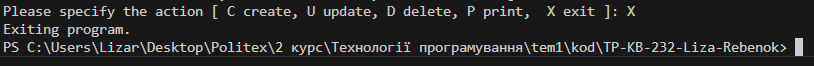
Вивід delete:



Вивід print:



Вивід exit:



Текст програми:

|  |
| --- |
| list = [  {"name": "Bob", "surname": "Smith", "phone": "0631234567", "email": "bob@email.com"},  {"name": "Emma", "surname": "Johnson", "phone": "0631234567", "email": "emma@email.com"},  {"name": "Jon", "surname": "Doe", "phone": "0631234567", "email": "jon@email.com"},  {"name": "Zak", "surname": "Taylor", "phone": "0631234567", "email": "zak@email.com"}  ]  def printAllList():  for elem in list:  strForPrint = f"Student name: {elem['name']} {elem['surname']}, Phone: {elem['phone']}, Email: {elem['email']}"  print(strForPrint)  return  def addNewElement():  name = input("Please enter student name: ")  surname = input("Please enter student surname: ")  phone = input("Please enter student phone: ")  email = input("Please enter student email: ")  newItem = {"name": name, "surname": surname, "phone": phone, "email": email}    insertPosition = 0  for item in list:  if name > item["name"]:  insertPosition += 1  else:  break  list.insert(insertPosition, newItem)  print("New element has been added")  return  def deleteElement():  name = input("Please enter name to be deleted: ")  deletePosition = -1  for item in list:  if name == item["name"]:  deletePosition = list.index(item)  break  if deletePosition == -1:  print("Element was not found")  else:  del list[deletePosition]  print(f"Student {name} has been deleted.")  return  def updateElement():  name = input("Please enter name to be updated: ")  updatePosition = -1  for index, item in enumerate(list):  if item["name"] == name:  updatePosition = index  break  if updatePosition == -1:  print(f"Student with name {name} not found.")  return  print(f"Updating information for student {name} {list[updatePosition]['surname']}:")  surname = input(f"Enter new surname (current: {list[updatePosition]['surname']}): ")  phone = input(f"Enter new phone (current: {list[updatePosition]['phone']}): ")  email = input(f"Enter new email (current: {list[updatePosition]['email']}): ")  list[updatePosition]["surname"] = surname if surname else list[updatePosition]["surname"]  list[updatePosition]["phone"] = phone if phone else list[updatePosition]["phone"]  list[updatePosition]["email"] = email if email else list[updatePosition]["email"]  print(f"Student {name} has been updated.")  return  def main():  while True:  choice = input("Please specify the action [ C create, U update, D delete, P print, X exit ]: ")  match choice:  case "C" | "c":  print("New element will be created:")  addNewElement()  printAllList()  case "U" | "u":  print("Existing element will be updated:")  updateElement()  printAllList()  case "D" | "d":  print("Element will be deleted:")  deleteElement()  printAllList()  case "P" | "p":  print("List will be printed:")  printAllList()  case "X" | "x":  print("Exiting program.")  break  case \_:  print("Wrong choice")  main() |

Посилання на github:

<https://github.com/Lizarebenokk>