**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.

**Теоретичні відомості**

**Списки**

Масив – набір фіксованої кількості елементів, що розміщені в пам’яті комп’ютера безпосередньо один за одним, а доступ до них здійснюється за індексом (номер даного елементу в масиві).

В Python для реалізації масиву використовуються списки. Список – тип даних, що представляє собою послідовність певних значень, що можуть повторюватись. Але на відміну від масиву – кількість елементів у списку може бути довільною.

Списки – структура даних, що може містити елементи різних типів, що перераховані через кому та заключені в квадратні дужки.

Списки служать для того, щоб зберігати об’єкти в певному порядку, особливо якщо порядок або вміст можуть змінюватися. Можна змінювати список, додати в нього нові елементи, а також видалити або перезаписати існуючі. Можна змінити кількість елементів у списку, а також самі елементи. Одне і те ж значення може зустрічатися в списку кілька разів.

**Словники**

Словник дуже схожий на список, але порядок елементів в ньому не має значення, і вони вибираються не за допомогою зміщення. Замість цього для кожного значення вказується пов’язаний з ним унікальний ключ. Таким ключем може бути об’єкт одного з незмінних типів: рядок, булева змінна, ціле число, число з плаваючою точкою, кортеж і іншими об’єктами. Елементи словника можуть містити об’єкти довільного типу даних і мати необмежений рівень вкладеності. Елементи в словниках розташовуються в довільному порядку.

**Кортежі**

Кортежі, як і списки, є послідовностями довільних елементів. На відміну від списків кортежі незмінні.

Всі операції над списками, що не змінюють список (додавання, множення на число, функції index() і count() і деякі інші операції) можна застосовувати до кортежів. Можна також по-різному змінювати елементи місцями і так далі.

**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.

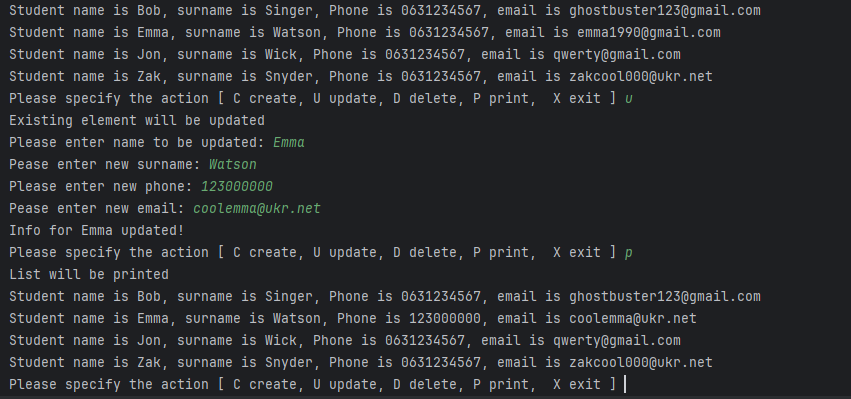
Необхідно реалізувати з нуля функціонал зміни інформації про студента враховуючи той факт, що вже існує реалізація додавання нового запису та видалення існуючого. **При зміні інформації про студента список має залишатись відсортованим**.

**Хід роботи**

Ознайомившись з теоретичними відомостями та завданням розпочав виконувати лабораторну роботу.

**Код програми:**

|  |
| --- |
| ## List [Item1, Item2, Item3] ## Item {"name":"Jon", "phone":"0631234567"}  # already sorted list list = [  {"name":"Bob", "surname":"Singer", "phone":"0631234567", "email":"ghostbuster123@gmail.com"},  {"name":"Emma", "surname":"Watson","phone":"0631234567", "email":"emma1990@gmail.com"},  {"name":"Jon", "surname":"Wick", "phone":"0631234567", "email":"qwerty@gmail.com"},  {"name":"Zak", "surname":"Snyder", "phone":"0631234567", "email":"zakcool000@ukr.net"} ]  def printAllList():  for elem in list:  strForPrint = "Student name is " + elem["name"] + ", surname is " + elem["surname"] + ", Phone is " + elem["phone"] + ", email is " + elem["email"]  print(strForPrint)  return  def addNewElement():  name = input("Pease enter student name: ")  surname = input("Pease enter student surname: ")  phone = input("Please enter student phone: ")  email = input("Pease enter student email: ")   newItem = {"name": name, "surname": surname, "phone": phone, "email": email}  # find insert position  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 delated: ")  deletePosition = -1  for item in list:  if name == item["name"]:  deletePosition = list.index(item)  break  if deletePosition == -1:  print("Element was not found")  else:  print("Dele position " + str(deletePosition))  # list.pop(deletePosition)  del list[deletePosition]  return   def updateElement():  name = input("Please enter name to be updated: ")   position = -1  for item in list:  if name == item["name"]:  position = list.index(item)  break  if position == -1:  print("Element was not found")  else:  surname = input("Pease enter new surname: ")  phone = input("Please enter new phone: ")  email = input("Pease enter new email: ")   del list[position]   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("Info for " + name + " updated!")   def main():  while True:  chouse = input("Please specify the action [ C create, U update, D delete, P print, X exit ] ")  match chouse:  case "C" | "c":  print("New element will be created:")  addNewElement()  printAllList()  case "U" | "u":  print("Existing element will be updated")  updateElement()  case "D" | "d":  print("Element will be deleted")  deleteElement()  case "P" | "p":  print("List will be printed")  printAllList()  case "X" | "x":  print("Exit()")  break  case \_:  print("Wrong chouse")   main() |

Результат роботи функції update: 

**Висновок:** На цій лабораторній роботі я вивчав списки, словники, кортежі та дописав деякий функціонал існуючому коду.