Технології програмування

Лабораторна робота №1

**Списки. Словники. Кортежі**

**Новик Максим Вікторович, КБ-231**

**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.

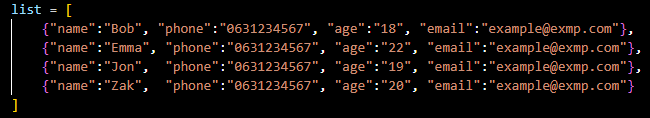
Перед виконанням роботи слід ознайомитись з існуючим функціоналом. Необхідно розширити відомості про студента до 4х полів. На даний час використовується лише два поля (name та phone). Необхідно переробити існуючий функціонал враховуючи розширення відомості про студента до 4х полів.

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**.

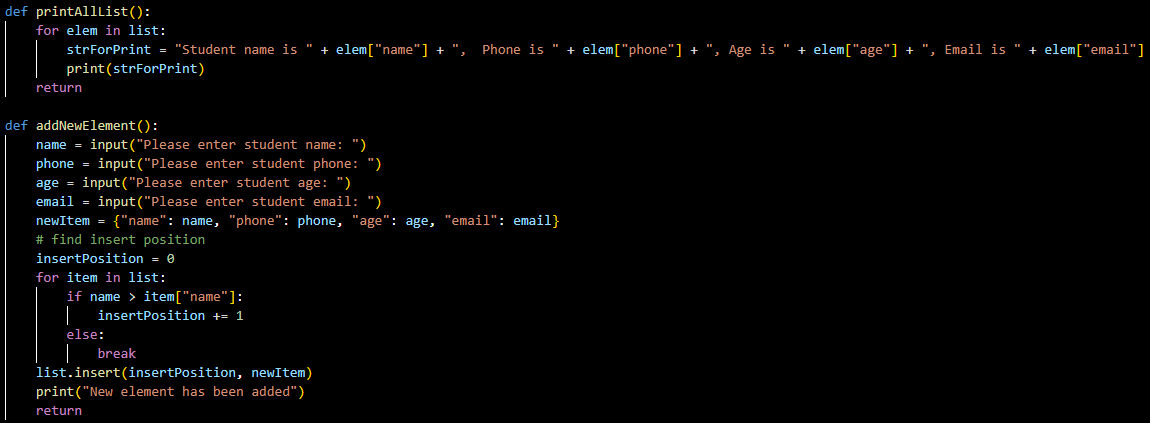
Текст програми разом зі звітом розмістити в каталозі lab\_01. Каталог lab\_01 розмістити в каталозі, що використовується для виконання практичних завдань по кожній лекції та має назву **TP-KB-23[1 or 2]-Name-Surname**.

**Хід виконання завдання:**

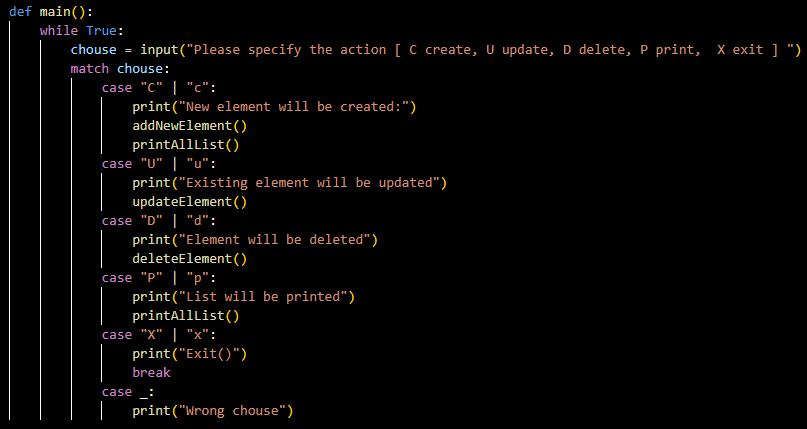
Було розширені відомості про студентів до 4х полів. Зараз там є: name, phone, age та email.



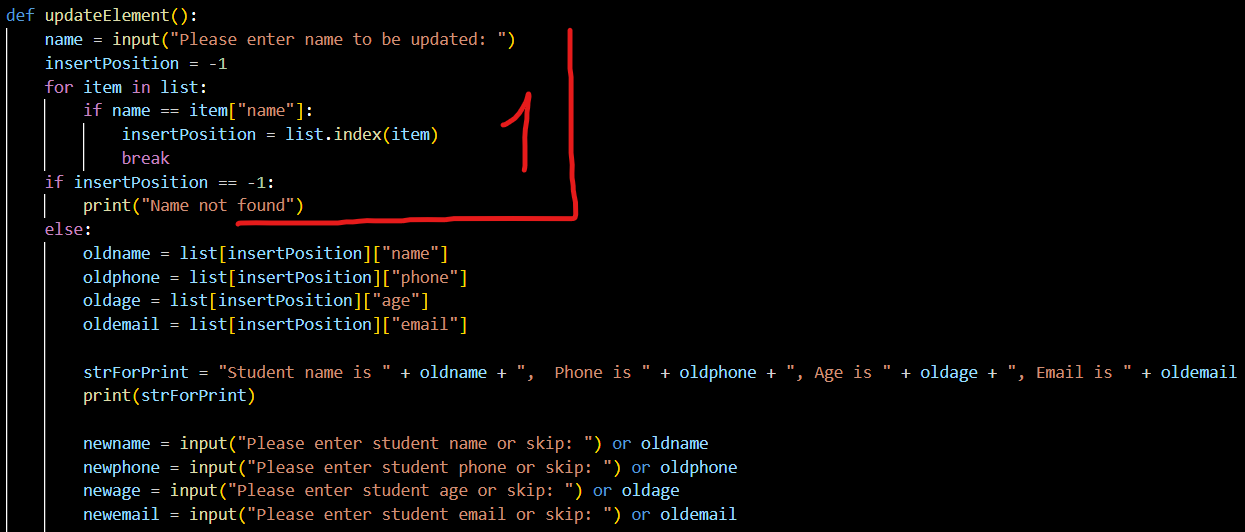
Перероблюю функції printAllList() та addNewElement() під вивід і додавання вже чотирьох полів даних студентів.



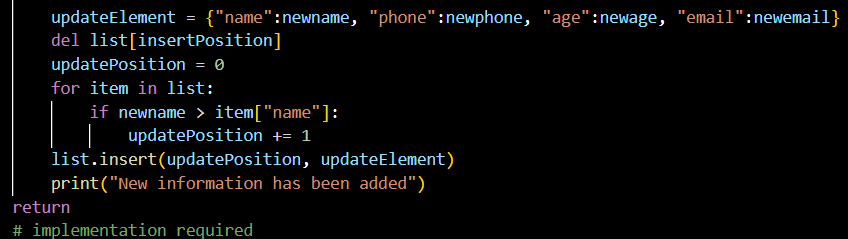
Також виправляю відсутність виклику функції updateElement():



Далі, використовуючи вже наявний в коді пошук елементів, починаю будувати функції updateElement() оновлення даних студентів. Перша частина взята з функції deleteElement(). Вона виконує пошук позиції елемента за введеним ім’ям:



В другій частині програма зберігає дані та виводить їх користувачу, після чого користувачу надається можливість ввести нові дані, або залишити старі.



В третій частині формується змінений елемент, видаляється старий і відбувається пошук позиції для вставки оновленого елементу.

Текст програми:

|  |
| --- |
| ## List [Item1, Item2, Item3]  ## Item {"name":"Jon", "phone":"0631234567"}  # already sorted list  list = [      {"name":"Bob", "phone":"0631234567", "age":"18", "email":"example@exmp.com"},      {"name":"Emma", "phone":"0631234567", "age":"22", "email":"example@exmp.com"},      {"name":"Jon",  "phone":"0631234567", "age":"19", "email":"example@exmp.com"},      {"name":"Zak",  "phone":"0631234567", "age":"20", "email":"example@exmp.com"}  ]  def printAllList():      for elem in list:          strForPrint = "Student name is " + elem["name"] + ",  Phone is " + elem["phone"] + ", Age is " + elem["age"] + ", Email is " + elem["email"]          print(strForPrint)      return  def addNewElement():      name = input("Please enter student name: ")      phone = input("Please enter student phone: ")      age = input("Please enter student age: ")      email = input("Please enter student email: ")      newItem = {"name": name, "phone": phone, "age": age, "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: ")      insertPosition = -1      for item in list:          if name == item["name"]:              insertPosition = list.index(item)              break      if insertPosition == -1:          print("Name not found")      else:          oldname = list[insertPosition]["name"]          oldphone = list[insertPosition]["phone"]          oldage = list[insertPosition]["age"]          oldemail = list[insertPosition]["email"]          strForPrint = "Student name is " + oldname + ",  Phone is " + oldphone + ", Age is " + oldage + ", Email is " + oldemail          print(strForPrint)          newname = input("Please enter student name or skip: ") or oldname          newphone = input("Please enter student phone or skip: ") or oldphone          newage = input("Please enter student age or skip: ") or oldage          newemail = input("Please enter student email or skip: ") or oldemail          updateElement = {"name":newname, "phone":newphone, "age":newage, "email":newemail}          del list[insertPosition]          updatePosition = 0          for item in list:              if newname > item["name"]:                  updatePosition += 1          list.insert(updatePosition, updateElement)          print("New information has been added")      return      # implementation required  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() |

**Висновок:**

Під час виконання лабораторної роботи я використовував теоретичне підґрунтя про складні структури даних, такі як: списки, словники та кортежі. Використовував існуючий код, щоб доробити програму додавши функціонал.

**Посилання на github:**

https://github.com/GAGGAGX/TP-KB-231-Novyk-Maksym

**Знімок екрану з посилання на github:**

