Інформатика

Лабораторна робота № 1

Гладченко Олег

КБ-231

**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**

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 виконання роботи**

Було додано два нових поля до словника, що представляє кожного студента:

Email студента,

Адреса проживання студента.

Таким чином, кожен запис у списку тепер містить 4 поля: ім'я, номер телефону, email та адресу.

|  |
| --- |
| list = [      {"name": "Bob", "phone": "0631234567", "email": "bobisingers@example.com", "address": "123 Main St"},      {"name": "Emma", "phone": "0631234568", "email": "emma2004@example.com", "address": "456 Park Ave"},      {"name": "Jon",  "phone": "0631234569", "email": "jonyStre@example.com", "address": "789 Elm St"},      {"name": "Zak",  "phone": "0631234570", "email": "ZakKing@example.com", "address": "101 Oak St"}  ] |

Додано функцію оновлення інформації про студента. Вона дозволяє користувачу змінювати номер телефону, email або адресу існуючого студента, зберігаючи список відсортованим за іменем.

Також було оновлено процес додавання та друкування, шляхом введення нових полів для електронної пошти та адреси у список.

Реалізовано сортування після кожної операції: після додавання або оновлення даних, список студентів залишається відсортованим за ім'ям.

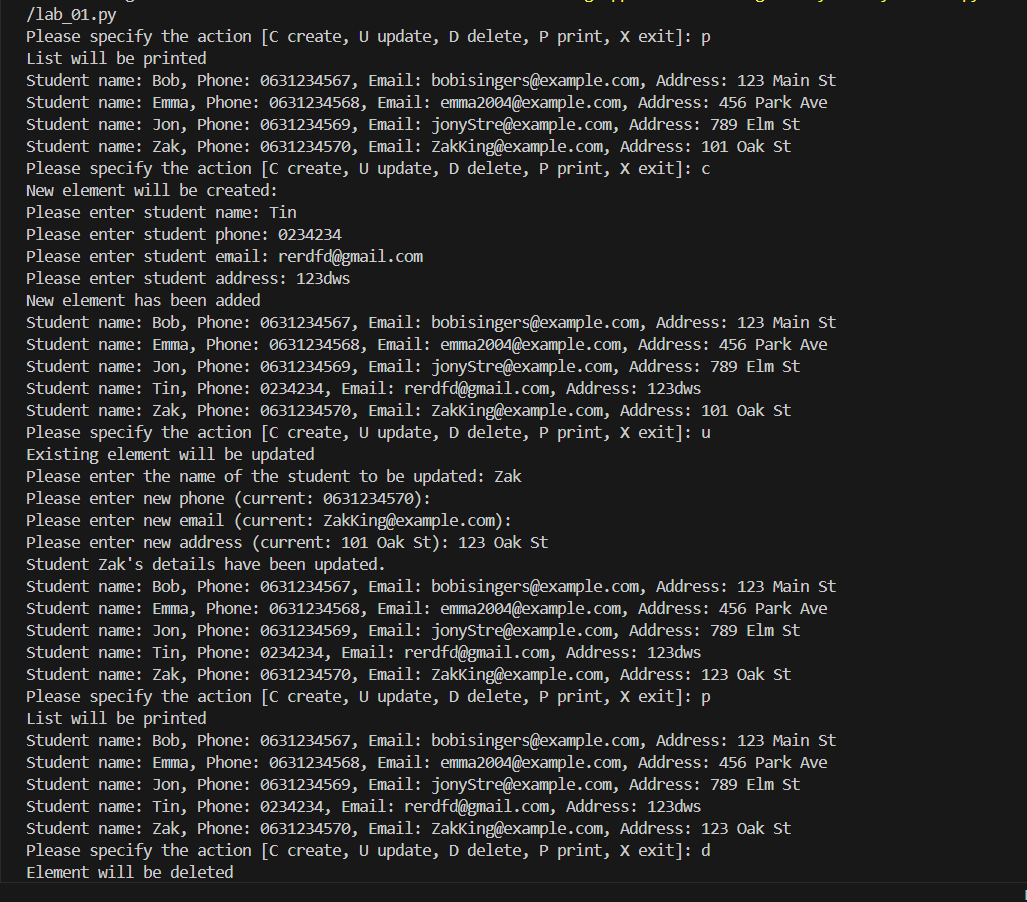
Текст програми до внесення змін:

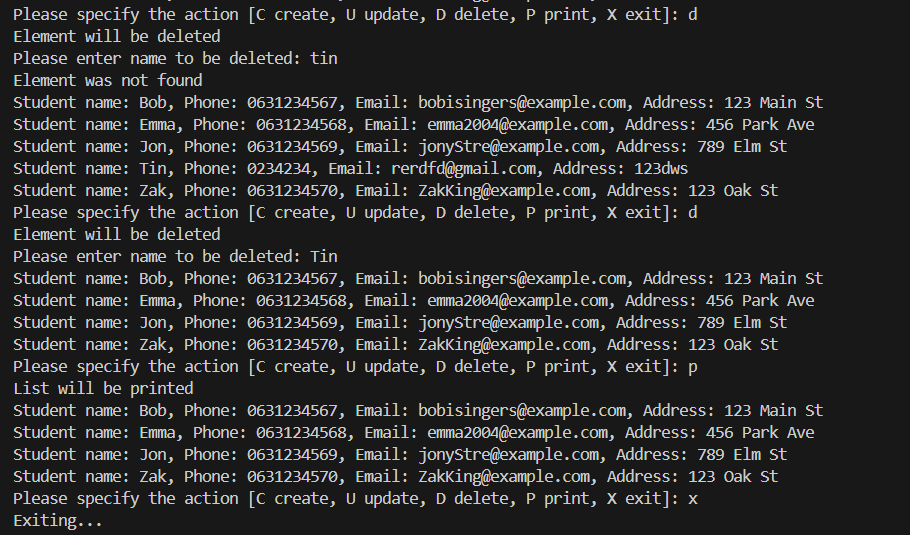
|  |
| --- |
| ## List [Item1, Item2, Item3]  ## Item {"name":"Jon", "phone":"0631234567"}  # already sorted list  list = [      {"name":"Bob", "phone":"0631234567"},      {"name":"Emma", "phone":"0631234567"},      {"name":"Jon",  "phone":"0631234567"},      {"name":"Zak",  "phone":"0631234567"}  ]  def printAllList():      for elem in list:          strForPrint = "Student name is " + elem["name"] + ",  Phone is " + elem["phone"]          print(strForPrint)      return  def addNewElement():      name = input("Pease enter student name: ")      phone = input("Please enter student phone: ")      newItem = {"name": name, "phone": phone}      # 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: ")      # 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")              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() |

Текст програми після змін:

|  |
| --- |
| # Додано нові поля: email і address для кожного студента  # Додано функцію updateElement, яка дозволяє оновлювати існуючі записи студента  # Додано можливість пропуску полів при оновленні (зберігає старі дані, якщо нові не введено)  # уже відсортований список із 4 полями: ім’я, телефон, електронна адреса, адреса  list = [      {"name": "Bob", "phone": "0631234567", "email": "bobisingers@example.com", "address": "123 Main St"},      {"name": "Emma", "phone": "0631234568", "email": "emma2004@example.com", "address": "456 Park Ave"},      {"name": "Jon",  "phone": "0631234569", "email": "jonyStre@example.com", "address": "789 Elm St"},      {"name": "Zak",  "phone": "0631234570", "email": "ZakKing@example.com", "address": "101 Oak St"}  ]  def printAllList():      for elem in list:          strForPrint = f"Student name: {elem['name']}, Phone: {elem['phone']}, Email: {elem['email']}, Address: {elem['address']}"          print(strForPrint)      return  def addNewElement():      name = input("Please enter student name: ")      phone = input("Please enter student phone: ")      email = input("Please enter student email: ")      address = input("Please enter student address: ")        newItem = {"name": name, "phone": phone, "email": email, "address": address}        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]      return  def updateElement():      name = input("Please enter the name of the student to be updated: ")      updatePosition = -1      for item in list:          if name == item["name"]:              updatePosition = list.index(item)              break        if updatePosition == -1:          print("Student not found")      else:  # Дозволяє користувачу оновлювати кожне поле або залишити його без змін          new\_phone = input(f"Please enter new phone (current: {list[updatePosition]['phone']}): ") or list[updatePosition]["phone"]          new\_email = input(f"Please enter new email (current: {list[updatePosition]['email']}): ") or list[updatePosition]["email"]          new\_address = input(f"Please enter new address (current: {list[updatePosition]['address']}): ") or list[updatePosition]["address"]    # Оновлення інформації про студента          list[updatePosition]["phone"] = new\_phone          list[updatePosition]["email"] = new\_email          list[updatePosition]["address"] = new\_address            print(f"Student {name}'s details have been updated.")  # Пересортований список          list.sort(key=lambda x: x["name"])      return  # Меню  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...")                  break              case \_:                  print("Wrong choice")  main() |

Вивід програми на консоль:





Проаналізувавши роботу зі складними структурами даних, такими як списки, словники та кортежі, можна дійти висновку, що використання цих структур дозволяє зручно зберігати, обробляти та маніпулювати великими обсягами інформації. Списки забезпечують легкий доступ до елементів, словники дозволяють зберігати дані в форматі "ключ-значення", а кортежі можуть бути корисні для незмінних наборів даних.