**Виконав: Проценко Євгеній**

**КБ-241**

**Звіт до лабораторної роботи №1**

**Тема:** 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 для виконання:**

Реалізувати **відсортований** телефонний довідник студентів групи:

1. Розширити відомості про студента до 4х полів.
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.

**Хід роботи:**

Перед виконанням лабораторної роботи я ознайомився з теоретичними відомостями та функціоналом наданого коду для модифікації у Visual Studio Code.

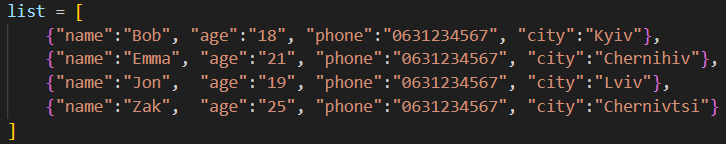
1. Додаю у список дві нові відомості про студента – вік та місто (Рис. 1).

Рис. 1 – Список відомостей про студентів

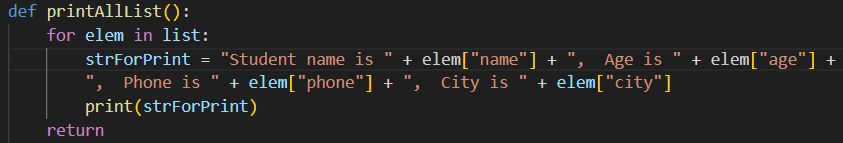
1. Перероблюю функції для виведення списку відомостей студентів та додавання нового студента, враховуючи розширення відомостей про студентів (Рис. 2-3)

Рис. 2 – Оновлена функція для виведення списку відомостей студентів

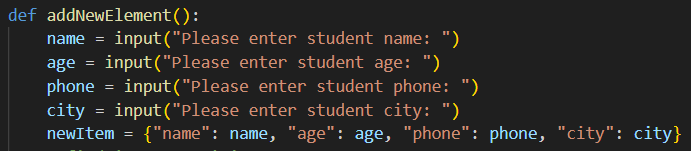


Рис. 3 – Оновлена частина функції для додавання нових студентів

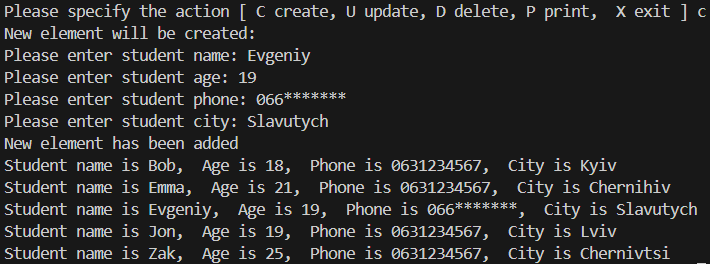
Перевіряю роботу функцій додавання відомостей та виведення списку (Рис. 4).

Рис. 4 – Тестування оновлених функцій шляхом введення своїх даних

1. Створюю функцію для оновлення інформації про студента, залишаючи список відсортованим (Рис. 5-6).

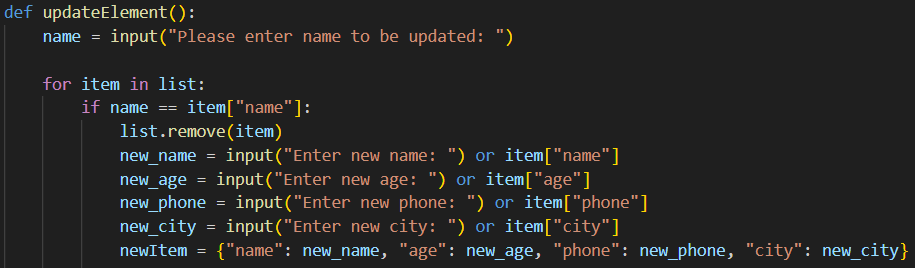


Рис. 5 – Частина функції з пошуком елемента для зміни, введенням нових значень та можливістю залишити будь-які дані студента незмінними

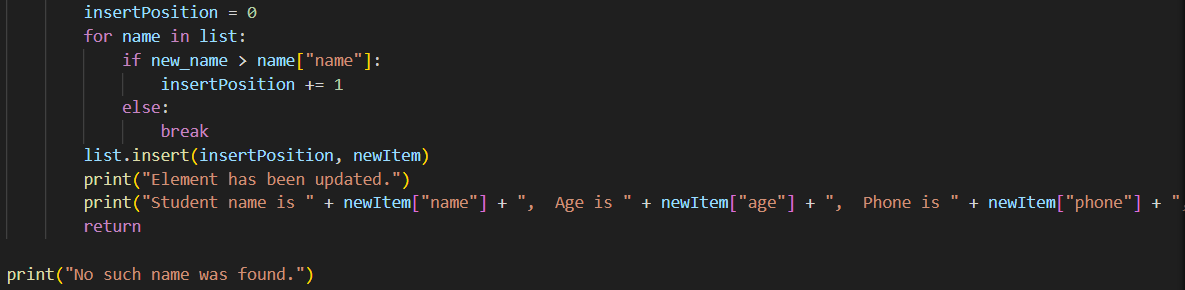


Рис. 6 – Частина функції з пошуком позиції для підтримання відсортованості списку, виведенням оновленого рядка та повідомленням на випадок відсутності імені студента у списку.

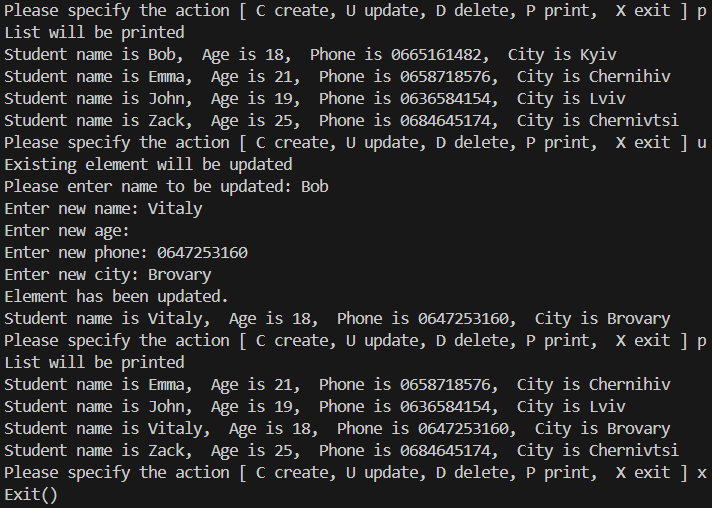
Перевіряю роботу функції для оновлення даних студента (Рис. 7).

Рис. 7 – Тестування функції за допомогою введення нового імені, номера телефону та місця проживання студента

Повний код програми доступний нижче або за посиланням на Git Hub:

<https://github.com/jekaproc/TP-KB-241-Protsenko-Evgeniy/tree/main/lab_01/lab_01.py>

|  |
| --- |
| list = [      {"name":"Bob", "age":"18", "phone":"0631234567", "city":"Kyiv"},      {"name":"Emma", "age":"21", "phone":"0631234567", "city":"Chernihiv"},      {"name":"Jon",  "age":"19", "phone":"0631234567", "city":"Lviv"},      {"name":"Zak",  "age":"25", "phone":"0631234567", "city":"Chernivtsi"}  ]  def printAllList():      for elem in list:          strForPrint = "Student name is " + elem["name"] + ",  Age is " + elem["age"] + ",  Phone is " + elem["phone"] + ",  City is " + elem["city"]          print(strForPrint)      return  def addNewElement():      name = input("Please enter student name: ")      age = input("Please enter student age: ")      phone = input("Please enter student phone: ")      city = input("Please enter student city: ")      newItem = {"name": name, "age": age, "phone": phone, "city": city}      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("Delete position " + str(deletePosition))          del list[deletePosition]      return  def updateElement():      name = input("Please enter name to be updated: ")      for item in list:          if name == item["name"]:              list.remove(item)              new\_name = input("Enter new name: ") or item["name"]              new\_age = input("Enter new age: ") or item["age"]              new\_phone = input("Enter new phone: ") or item["phone"]              new\_city = input("Enter new city: ") or item["city"]              newItem = {"name": new\_name, "age": new\_age, "phone": new\_phone, "city": new\_city}              insertPosition = 0              for name in list:                  if new\_name > name["name"]:                      insertPosition += 1                  else:                      break              list.insert(insertPosition, newItem)              print("Element has been updated.")              print("Student name is " + newItem["name"] + ",  Age is " + newItem["age"] + ",  Phone is " + newItem["phone"] + ",  City is " + newItem["city"])              return      print("No such name was found.")  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()              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 choice")  main() |

**Висновок:** У ході виконання лабораторної роботи я закріпив теоретичні знання про списки, словники та кортежі у Python і застосував їх на практиці. Модифікував наданий код телефонного довідника студентів, додавши нові відомості про вік та місто, переробив функції для виведення списку та додавання нового студента з урахуванням розширених відомостей. Створив функцію для оновлення даних студента, яка дозволяє змінювати інформацію про нього, залишаючи список відсортованим. Після виконаних змін провів тестування програми, та переконався, що всі функції працюють правильно, а список студентів відображається повним та відсортованим.