**Звіт про виконання лабораторної роботи з курсу Технології програмування на мові Python**

Ткачова Вероніка КБ-242

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 виконання**

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 завдання:**

Текст програми:

# Початковий список студентів

students = [

    {"name": "Bob",  "phone": "0631111111", "group": "KB-242", "email": "bob@gmail.com"},

    {"name": "Emma", "phone": "0632222222", "group": "KB-242", "email": "emma@gmail.com"},

    {"name": "Jon",  "phone": "0633333333", "group": "KB-242", "email": "jon@gmail.com"},

    {"name": "Zak",  "phone": "0634444444", "group": "KB-242", "email": "zak@gmail.com"}

]

# Вивід усіх студентів

def printAllList():

    for elem in students:

        print(f"Name: {elem['name']}, Phone: {elem['phone']}, Group: {elem['group']}, Email: {elem['email']}")

    print()

# Додавання нового студента

def addNewElement():

    name = input("Enter student name: ")

    phone = input("Enter student phone: ")

    group = input("Enter student group: ")

    email = input("Enter student email: ")

    newItem = {"name": name, "phone": phone, "group": group, "email": email}

    # Знайти місце для вставки (щоб список залишався відсортованим)

    insertPosition = 0

    for item in students:

        if name > item["name"]:

            insertPosition += 1

        else:

            break

    students.insert(insertPosition, newItem)

    print("New student added!\n")

# Видалення студента

def deleteElement():

    name = input("Enter name to delete: ")

    deletePosition = -1

    for item in students:

        if name == item["name"]:

            deletePosition = students.index(item)

            break

    if deletePosition == -1:

        print("Student not found\n")

    else:

        del students[deletePosition]

        print("Student deleted\n")

# Оновлення інформації про студента

def updateElement():

    name = input("Enter student name to update: ")

    found = None

    for item in students:

        if name == item["name"]:

            found = item

            break

    if not found:

        print("Student not found\n")

        return

    print("Enter new data (press Enter to leave unchanged):")

    new\_name = input(f"Name [{found['name']}]: ") or found['name']

    new\_phone = input(f"Phone [{found['phone']}]: ") or found['phone']

    new\_group = input(f"Group [{found['group']}]: ") or found['group']

    new\_email = input(f"Email [{found['email']}]: ") or found['email']

    # Видаляємо старий запис і вставляємо оновлений у відсортованому порядку

    students.remove(found)

    newItem = {"name": new\_name, "phone": new\_phone, "group": new\_group, "email": new\_email}

    insertPosition = 0

    for item in students:

        if new\_name > item["name"]:

            insertPosition += 1

        else:

            break

    students.insert(insertPosition, newItem)

    print("Student information updated!\n")

# Головне меню

def main():

    while True:

        choice = input("Choose action [C create, U update, D delete, P print, X exit]: ")

        match choice:

            case "C" | "c":

                print("Adding new student:")

                addNewElement()

            case "U" | "u":

                print("Updating student info:")

                updateElement()

            case "D" | "d":

                print("Deleting student:")

                deleteElement()

            case "P" | "p":

                print("Student list:")

                printAllList()

            case "X" | "x":

                print("Exit program.")

                break

            case \_:

                print("Wrong choice, try again.\n")

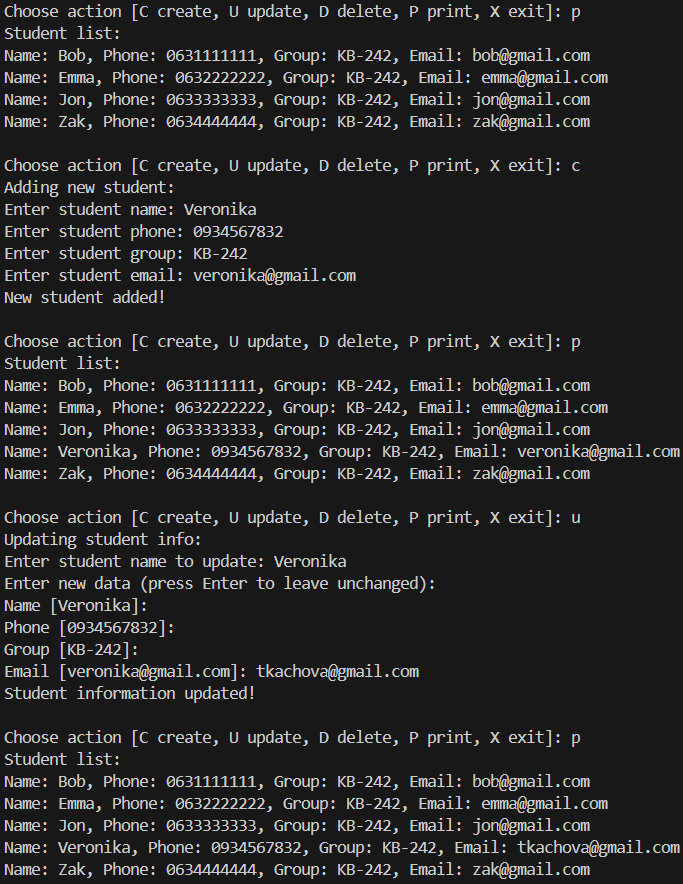
# Запуск програми

main()

Посилання на github:

Знімок екрану з посилання на github:

Виконання завдання:



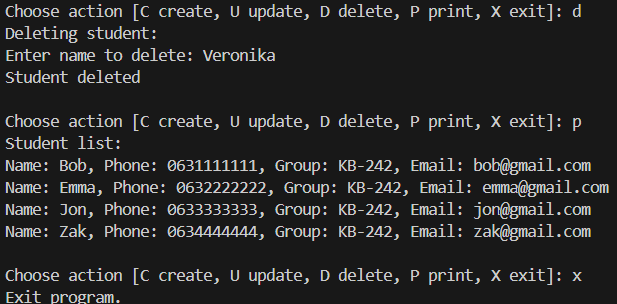


Рис. 3 – 4 – результат виконання завдання для лабораторної роботи №1.

**Висновок:** під час виконання цієї лабораторної роботи, я використовуючи існуючий код та теоретичне підґрунтя про складні структури даних Списки, Словники, Кортежі, доробила програму «Телефонний довідник студентів групи».