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

Технології програмування

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 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:

students = [

    {"name": "Bob", "phone": "0631234567", "email": "bob@gmail.com", "group": "KB-242"},

    {"name": "Emma", "phone": "0631234567", "email": "emma@gmail.com", "group": "KB-242"},

    {"name": "Jon",  "phone": "0631234567", "email": "jon@gmail.com",  "group": "KB-242"},

    {"name": "Zak",  "phone": "0631234567", "email": "zak@gmail.com",  "group": "KB-242"}

]

def printAllList():

    for elem in students:

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

    return

def addNewElement():

    name = input("Please enter student name: ")

    phone = input("Please enter student phone: ")

    email = input("Please enter student email: ")

    group = input("Please enter student group: ")

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

    # знаходимо позицію для вставки (щоб залишалось відсортованим)

    insertPosition = 0

    for item in students:

        if name > item["name"]:

            insertPosition += 1

        else:

            break

    students.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 students:

        if name == item["name"]:

            deletePosition = students.index(item)

            break

    if deletePosition == -1:

        print("⚠️ Element was not found")

    else:

        del students[deletePosition]

        print("🗑️ Element deleted successfully")

    return

def updateElement():

    name = input("Please enter name to be updated: ")

    updatePosition = -1

    for item in students:

        if name == item["name"]:

            updatePosition = students.index(item)

            break

    if updatePosition == -1:

        print("⚠️ Student not found")

        return

    print("Leave field empty if you don’t want to change it.")

    # вводимо нові значення

    new\_name = input(f"Enter new name ({students[updatePosition]['name']}): ") or students[updatePosition]['name']

    new\_phone = input(f"Enter new phone ({students[updatePosition]['phone']}): ") or students[updatePosition]['phone']

    new\_email = input(f"Enter new email ({students[updatePosition]['email']}): ") or students[updatePosition]['email']

    new\_group = input(f"Enter new group ({students[updatePosition]['group']}): ") or students[updatePosition]['group']

    # оновлюємо дані

    students[updatePosition] = {

        "name": new\_name,

        "phone": new\_phone,

        "email": new\_email,

        "group": new\_group

    }

    # щоб залишилось відсортовано — видалимо і вставимо знову

    updated\_student = students.pop(updatePosition)

    insertPosition = 0

    for item in students:

        if updated\_student["name"] > item["name"]:

            insertPosition += 1

        else:

            break

    students.insert(insertPosition, updated\_student)

    print("✏️ Student info updated successfully")

    return

def main():

    while True:

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

        match choice:

            case "C" | "c":

                print(" Adding new element:")

                addNewElement()

                printAllList()

            case "U" | "u":

                print(" Updating element:")

                updateElement()

                printAllList()

            case "D" | "d":

                print(" Deleting element:")

                deleteElement()

                printAllList()

            case "P" | "p":

                print(" Printing list:")

                printAllList()

            case "X" | "x":

                print(" Exit()")

                break

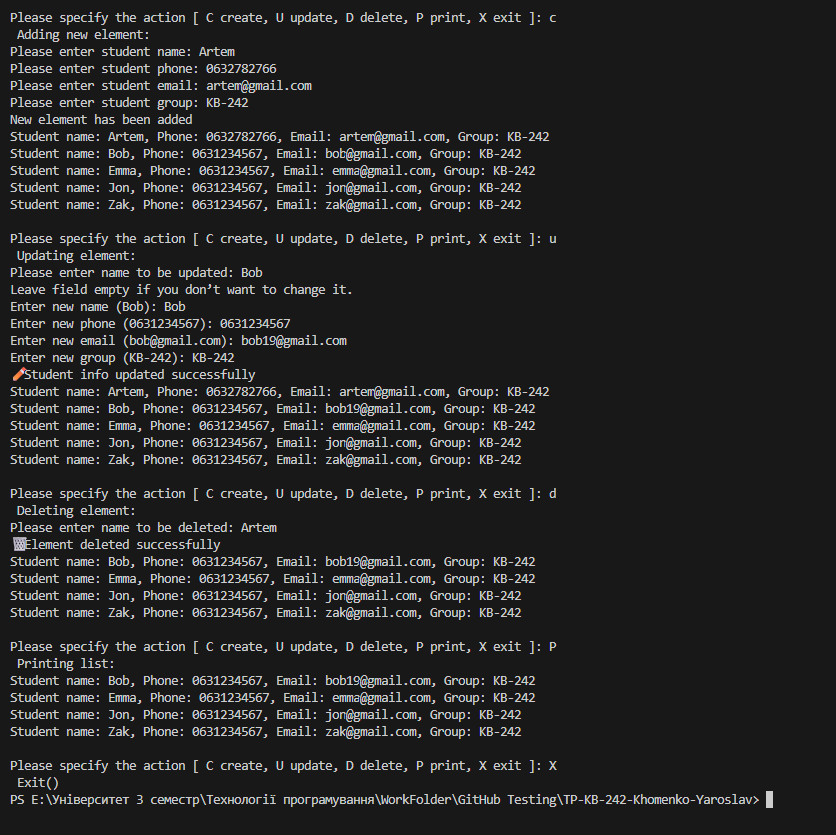
            case \_:

                print(" Wrong choice, try again.")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

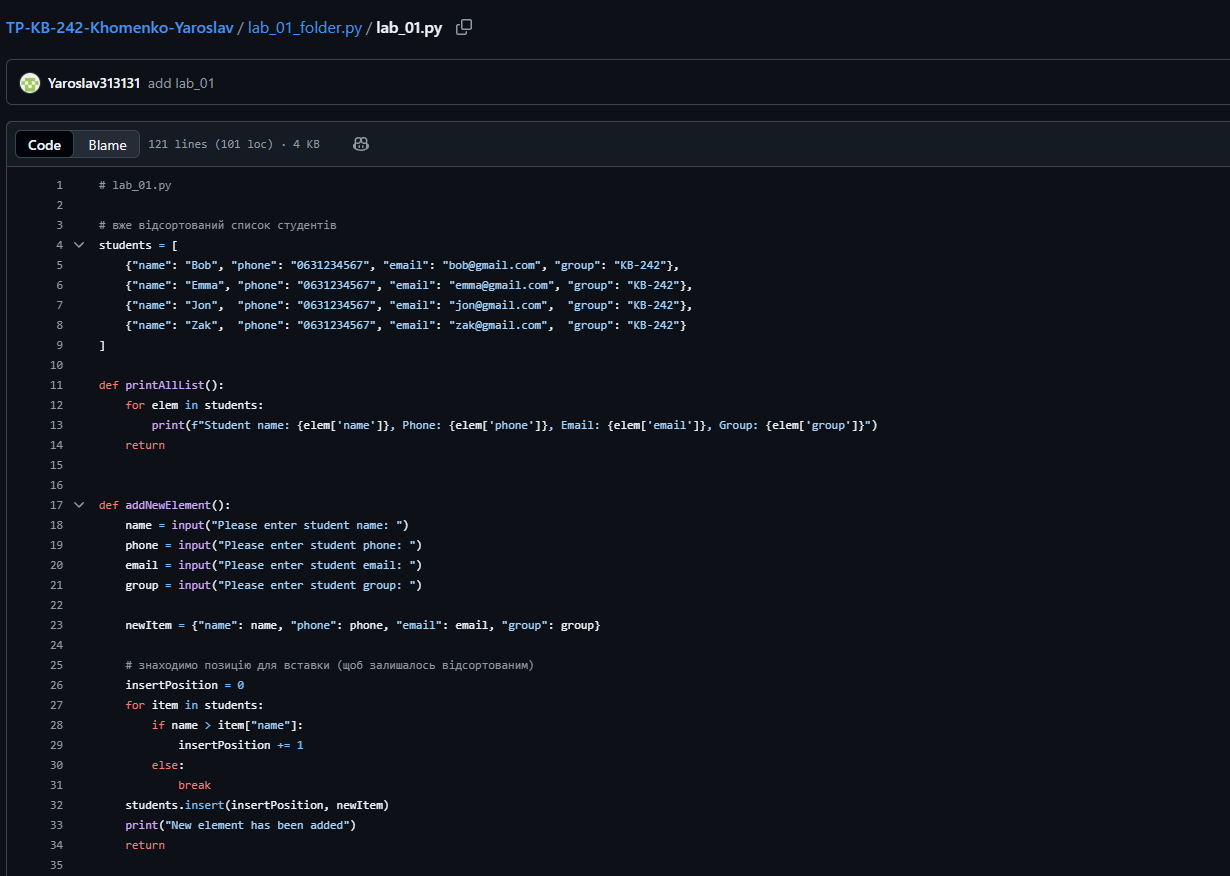
Результат виконання програми:



Посилання на github:

<https://github.com/Yaroslav313131/TP-KB-242-Khomenko-Yaroslav/blob/e4bff150b21e1c9af014f585f7b6af2cd58badb7/lab_01_folder.py/lab_01.py>

Знімок екрану з посилання на github:



Висновок : у ході лабораторної роботи було створено програму на Python для ведення відсортованого телефонного довідника студентів. Структуру запису розширено до чотирьох полів (ім’я, телефон, email, група). Реалізовано операції додавання, оновлення, видалення та виведення списку з автоматичним збереженням сортування. Мета роботи досягнута, отримано практичні навички роботи зі структурами даних та функціями мови Python.