

Звіт про виконання лабораторної роботи №1

Тема:Списки. Словники. Кортежі

Мета: 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

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 роботи

Я додав ще два поля для списку студентів це прізвище і електронна пошта(Рис.1).

```
list = [  
    {"name": "Bob", "surname": "Jobs", "phone": "0631234567", "email": "student1@gmail.com"},  
    {"name": "Emma", "surname": "Parker", "phone": "0631234567", "email": "student2@gmail.com"},  
    {"name": "Jon", "surname": "Watson", "phone": "0631234567", "email": "student3@gmail.com"},  
    {"name": "Zak", "surname": "Zaker", "phone": "0631234567", "email": "student4@gmail.com"}  
]
```

Рис.1 Оновлений список

Потім я доробив функцію додавання студента і додав ще два поля для вводу(Рис.2)

```
def addNewElement():
    name = input("Pease enter student name: ")
    surname = input("Pease enter student surname: ")
    phone = input("Please enter student phone: ")
    email = input("Please enter student email: ")
    newItem = {"name": name, "surname": surname, "phone": phone, "email":email}

    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
```

Рис.2 функція додавання студентів

Після чого реалізував функцію зміни інформації, а саме зробив цикл, який шукає студента з таким ім'ям, якщо не знаходить, то виводить помилку, якщо є то з'являються поля для вводу нової інформації, де написана стара назва, після введення, відбувається сортування, для коректного відображення списку студентів(Рис.3).

```
def updateElement():
    name = input("Please enter name to be updated: ")
    found = None
    for item in list:
        if item["name"] == name:
            found = item
            break

    if not found:
        print("Student not found")
        return

    print(f"Updating student: {found['name']} {found['surname']}")

    new_name = input(f"Enter new name [{found['name']}]: ") or found['name']
    new_surname = input(f"Enter new surname [{found['surname']}]: ") or found['surname']
    new_phone = input(f"Enter new phone [{found['phone']}]: ") or found['phone']
    new_email = input(f"Enter new email [{found['email']}]: ") or found['email']

    list.remove(found)

    updatedItem = {"name": new_name, "surname": new_surname, "phone": new_phone, "email": new_email}

    insertPosition = 0
    for item in list:
        if new_name > item["name"]:
            insertPosition += 1
        else:
            break
    list.insert(insertPosition, updatedItem)

    print("Student has been updated")
    return
```

Рис.3 Функція оновлення інформації

```
Updating student: Bob Jobs
Enter new name [Bob]: Jake
Enter new surname [Jobs]:
Enter new phone [0631234567]:
Enter new email [student1@gmail.com]:
Student has been updated
```

Рис.4 Приклад роботи функції оновлення

Висновок: В цій лабораторній роботі, я навчився розробляти цикли для сортування списків і детальніше навчився працювати зі списками.