**Лабораторна робота №2**

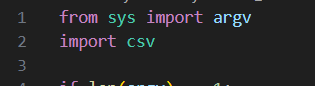
**Тема:** Робота з файлами, юніт тести

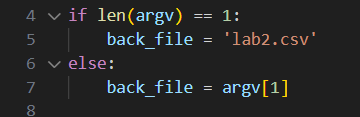
**Мета:** Використовуючи теоретичне підґрунтя про роботу з файлами та тестування коду у мові Python розширити програму телефонного довідника студентів додавши функціонал, що буде вказано в завданні до лабораторної роботи.

Хід роботи

1. Розширити функції програми з lab\_01. А саме, додати функції для роботи з csv файлами

Імпортуємо необхідні бібліотеки





Цей фрагмент коду встановлює значення змінної `back\_file`, яка вказує на ім'я файлу, з яким програма буде працювати. Це ім'я файлу отримується з командного рядка, коли програма запускається.

Давайте проаналізуємо цей фрагмент:

1. `len(argv)` - це довжина списку аргументів командного рядка, який передається програмі під час її запуску.

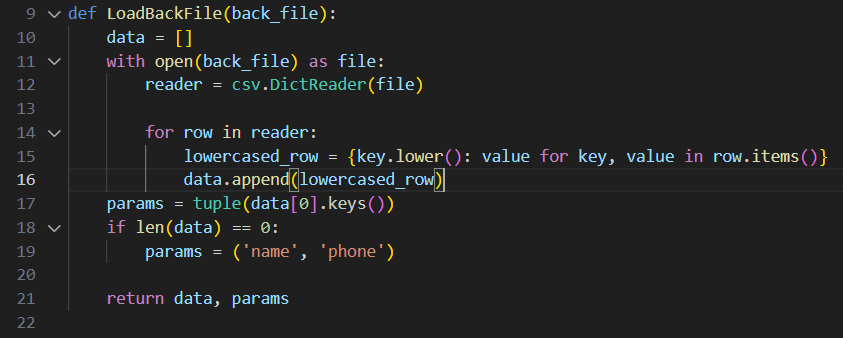
2. Умова `if len(argv) == 1:` перевіряє, чи вказано аргументи командного рядка. Якщо `len(argv)` дорівнює 1, це означає, що жодного аргумента (окрім назви самого скрипта) не було передано.

3. У разі, якщо умова виконується (`len(argv) == 1`), програма встановлює значення `back\_file` на рядок 'lab2.csv'. Це означає, що якщо користувач не вказав конкретний файл для обробки, програма автоматично використовує файл з назвою 'lab2.csv'.

4. Якщо умова не виконується (в іншому випадку, коли передано аргументи командного рядка), то значення `back\_file` встановлюється на значення першого аргумента командного рядка (тобто `argv[1]`), що може бути ім'ям іншого CSV-файлу, який користувач хоче використовувати.

Отже, цей блок коду надає гнучкість при запуску програми, дозволяючи користувачеві вказати ім'я CSV-файлу або використовувати значення за замовчуванням ('lab2.csv'), якщо жоден файл не вказано.

Є функція LoadBackFile, яка завантажує дані з CSV-файлу у вигляді списку словників. Кожен словник представляє інформацію про одного студента. Ключі словників - це назви параметрів (колонок), а значення - відповідні дані.



Ця функція `GetNewParams` використовується для отримання нових параметрів для студента від користувача через введення з клавіатури. Давайте розглянемо її детальніше:

1. \*\*Параметри функції:\*\*

- `param\_tuple`: Це кортеж, який містить імена параметрів, наприклад, `('name', 'phone')`. Ці імена використовуються для створення нового словника, де ключі - це імена параметрів, а значення вводяться користувачем.

2. \*\*Створення порожнього словника:\*\*

- `new\_params = {}`: Створюється порожній словник, в який будуть додаватися нові параметри.

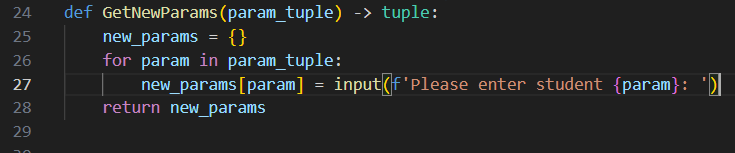
3. \*\*Цикл введення користувача:\*\*

- `for param in param\_tuple:`: Функція проходить через кожен параметр, вказаний у `param\_tuple`.

- `new\_params[param] = input(f'Please enter student {param}: ')`: Користувачеві пропонується ввести значення для кожного параметра. Введені дані зберігаються у відповідному полі словника `new\_params`.

4. \*\*Повернення словника:\*\*

- `return new\_params`: Після того як користувач ввів всі необхідні дані, функція повертає словник, де ключі - це імена параметрів, а значення - введені користувачем дані.



Ця функція `printAllList` відповідає за виведення всього списку студентів на екран у вигляді таблиці. Давайте розглянемо її детальніше:

1. \*\*Параметри функції:\*\*

- `param\_tuple`: Це кортеж, який містить імена параметрів (наприклад, `('name', 'phone')`). Використовується для отримання доступу до значень конкретних параметрів у кожному елементі списку студентів.

- `students\_list`: Це список студентів, де кожен студент представлений словником з інформацією про його параметри.

2. \*\*Виведення заголовка таблиці:\*\*

- `print("#" \* 10 + ' All list ' + '#' \* 10)`: Друкує рядок, що служить виглядовим елементом для виділення загального списку студентів.

3. \*\*Цикл для кожного студента:\*\*

- `for elem in students\_list:`: Проходить через кожен елемент (студента) у списку.

4. \*\*Виведення інформації про кожного студента:\*\*

- `print('Student : ', end='')`: Виводить початковий рядок для інформації про студента.

- `for param in param\_tuple:`: Проходить через кожен параметр, вказаний у `param\_tuple`.

- `str\_for\_print = f"{param} is " + f"{elem[param]}"`: Формує рядок для виведення інформації про кожний параметр студента.

- `print(str\_for\_print, end=', ')`: Виводить цей рядок без переведення рядка, додаючи кому для вирізнення параметрів.

- `print('\n')`: Після виведення інформації про всі параметри переходить на новий рядок.

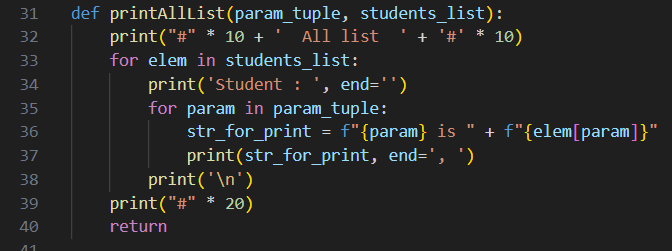
5. \*\*Виведення завершального рядка:\*\*

- `print("#" \* 20)`: Виводить рядок, що служить виглядовим елементом для виділення кінця списку студентів.

6. \*\*Повернення:\*\*

- `return`: Функція не повертає конкретного значення, оскільки працює в основному на виведенні інформації на екран.

Ця функція служить для зручного відображення інформації про студентів у вигляді таблиці при виведенні всього списку студентів.



Ця функція `addNewElement` призначена для додавання нового елементу (нового студента) до списку студентів. Розглянемо її крок за кроком:

1. \*\*Отримання нових параметрів для нового елементу:\*\*

- `new\_item = GetNewParams(param\_tuple)`: Викликає функцію `GetNewParams` для отримання нових значень параметрів для створення нового елементу (студента).

2. \*\*Визначення позиції для вставки:\*\*

- `insert\_position = 0`: Визначаємо початкову позицію для вставки нового елементу в список.

- `for item in students\_list:`: Проходимо через кожен елемент (студента) у списку.

3. \*\*Визначення правильної позиції для вставки за алфавітним порядком імен:\*\*

- `if new\_item['name'] > item["name"]:`: Порівнюємо ім'я нового елементу з ім'ям поточного елементу у списку.

- `insert\_position += 1`: Якщо ім'я нового елементу більше за ім'я поточного елементу, збільшуємо `insert\_position`.

- `else: break`: Якщо ім'я нового елементу не більше за ім'я поточного елементу, завершуємо цикл.

4. \*\*Вставка нового елементу на знайдену позицію:\*\*

- `students\_list.insert(insert\_position, new\_item)`: Вставляємо новий елемент на знайдену позицію у списку.

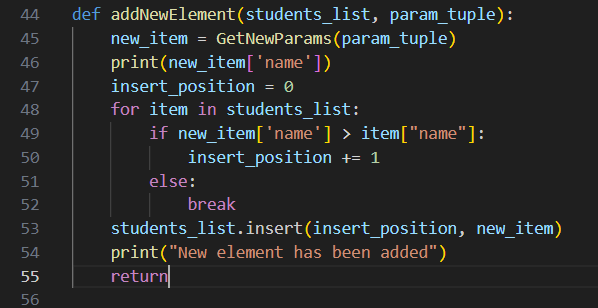
5. \*\*Виведення повідомлення про успішне додавання:\*\*

- `print("New element has been added")`: Виводить повідомлення про те, що новий елемент (студент) був успішно доданий до списку.

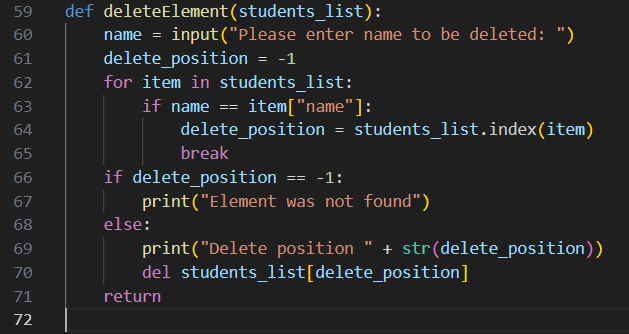
6. \*\*Повернення:\*\*

- `return`: Функція завершується без повернення конкретного значення.

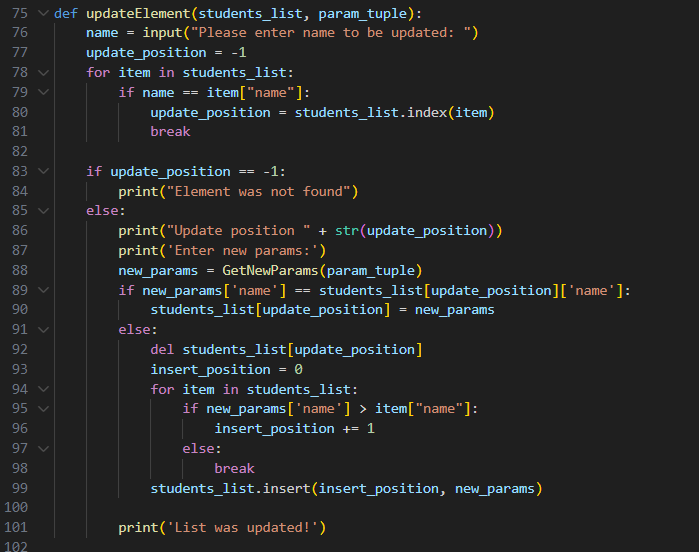
Ця функція забезпечує введення нових даних для студента та вставку його відсортованої за ім'ям позиції у список студентів.



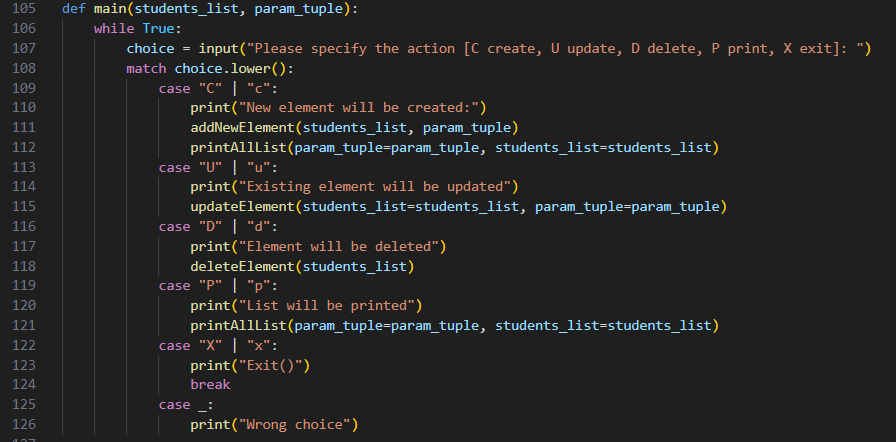
Функція видалення:



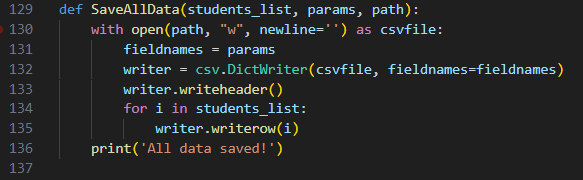
Функція оновлення:



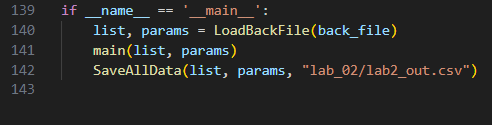
Загальна функція змін не отримала:



Функція зберігання:



Основний виклик функції:



Тестові файли:

