Звіт до лабораторної роботи №2

**Робота з файлами. Юніт тести.**

Під час виконання практичного завдання до лабораторної роботи №2 було надано варіанти рішення до наступних задач:

1. **завантаження** початкових данні для довідника з csv файлу. Приклад csv файлу (lab2.csv), та оригінальній файл формату xlsx розміщені поряд з вказівками. Ім’я файлу для завантаження даних повинно передаватись в якості параметрів командного рядку.
2. **зберігання** довідника студентів у форматі CSV перед виходом із програми.
3. покриття основного функціоналу програми **Юніт тестами**.

Виконання роботи:

**Текст програми:**

|  |
| --- |
| import operator  import sys  import csv  # Список студентів  students = [{"name": "Bob", "phone": "0641234567", "surname": "Litvinenko", "course": "4"},          {"name": "Emma", "phone": "0651234567", "surname": "Ilyenko", "course": "1"},          {"name": "Jon", "phone": "0661234567", "surname": "Kotsyuba", "course": "4"},          {"name": "Zak", "phone": "0671234567", "surname": "Starodub", "course": "3"}]  def loadFromCsv(filename):      global students      try:          with open(filename, newline='') as csvfile:              reader = csv.DictReader(csvfile)              students = sorted(reader, key=lambda x: x['name'].lower())      except FileNotFoundError:          print(f"Error: The file {filename} was not found.")      except IOError:          print(f"Error: Could not read the file {filename}.")  def saveToCsv(filename):      global students      try:          with open(filename, mode="w", newline='', encoding="utf-8") as csvfile:              fieldnames = ['name', 'phone', 'surname', 'course']              writer = csv.DictWriter(csvfile, fieldnames=fieldnames)              writer.writeheader()              for student in students:                  writer.writerow(student)      except IOError:          print(f"Error: Could not write to the file {filename}.")  def print\_all\_list():      """Функція для виведення всього списку студентів."""      for elem in students:          str\_for\_print = (              f"Student name is {elem['name']}, "              f"Phone is {elem['phone']}, "              f"surname is {elem['surname']}, "              f"Course is {elem['course']}"          )          print(str\_for\_print)  def add\_new\_element():      """Функція для додавання нового студента у відсортоване положення списку."""      name = input("Please enter student name: ")      phone = input("Please enter student phone: ")      surname = input("Please enter student surname: ")      course = input("Please enter student course: ")      new\_item = {"name": name, "phone": phone, "surname": surname, "course": course}      insert\_position = next((i for i, item in enumerate(students) if name < item["name"]), len(students))      students.insert(insert\_position, new\_item)      print("New student added.")  def delete\_element():      """Функція для видалення елементу зі списку студентів."""      name = input("Please enter name to be deleted: ")      phone = input("Please enter phone to be deleted: ")      delete\_position = -1      for i, item in enumerate(students):          if name == item["name"] and phone == item["phone"]:              delete\_position = i              break      if delete\_position == -1:          print("Element was not found.")      else:          del students[delete\_position]          print("Element has been deleted.")  def update\_element():      """Функція для оновлення інформації про студента у відсортованому списку."""      name = input("Please enter the name of the student to update: ")      phone = input("Please enter the phone of the student to update: ")      for i, student in enumerate(students):          if student["name"].lower() == name.lower() and student["phone"] == phone:              print(f"Updating information for {student['name']}")              new\_name = input(f"New name (Enter to keep '{student['name']}'): ") or student["name"]              new\_phone = input(f"New phone (Enter to keep '{student['phone']}'): ") or student["phone"]              new\_surname = input(f"New surname (Enter to keep '{student['surname']}'): ") or student["surname"]              new\_course = input(f"New course (Enter to keep '{student['course']}'): ") or student["course"]              # Видаляємо старий запис і додаємо новий у відсортоване положення              del students[i]              new\_student = {"name": new\_name, "phone": new\_phone, "surname": new\_surname, "course": new\_course}                # Знаходимо індекс для нової позиції студента              insert\_position = len(students)  #  елемент буде в кінці списку              for j, item in enumerate(students):                  if new\_name < item["name"]:                      insert\_position = j                      break              # Вставляємо нового студента в обчислену позицію              students.insert(insert\_position, new\_student)              print("Student information updated and placed in sorted position.")              return      print("Student not found.")  def main(filename):      """Головна функція, яка організовує взаємодію з користувачем."""      loadFromCsv(filename)      while True:          choice = input("Please specify the action [C create, U update, D delete, P print, X exit]: ")          match choice.lower():              case "c":                  print("New element will be created:")                  add\_new\_element()                  print\_all\_list()              case "u":                  print("Existing element will be updated")                  update\_element()                  print\_all\_list()              case "d":                  print("Element will be deleted")                  delete\_element()              case "p":                  print("List will be printed")                  print\_all\_list()              case "x":                  print("Exit...")                  saveToCsv(filename)                  break              case \_:                  print("Wrong choice")    if \_\_name\_\_ == "\_\_main\_\_":      if len(sys.argv) != 2:          print("Usage: python script.py <filename>")      else:          main(sys.argv[1]) |

**Посилання на github:**

<https://github.com/Alexxxxo/TP-KB-231-Oleksandr-Kozachok/tree/main/lab_02>

Висновок: у процесі виконання лабораторної роботи №2 було реалізовано завантаження даних із CSV-файлу, збереження довідника студентів у цьому ж форматі та створення юніт-тестів для перевірки основного функціоналу програми. Завдання виконано відповідно до вимог, результат роботи підтверджено тестуванням.