**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.

**Завдання до лабораторної роботи**

Розширити функціональність телефонного довідника студентів групи, що був розроблений у Лабораторній роботі №1 наступними можливостями:

1) завантаження початкових данні для довідника з csv файлу. 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.

**Хід роботи**

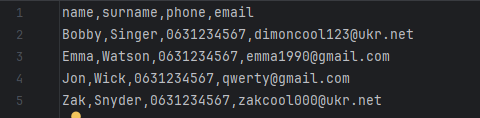
Ознайомившись з теоретичними відомостями та завданням розпочав виконувати лабораторну роботу.

Додав нові функції для опрацювання аргумента, завантаження даних із довідника та зберігання. Оптимізував існуючі функції.

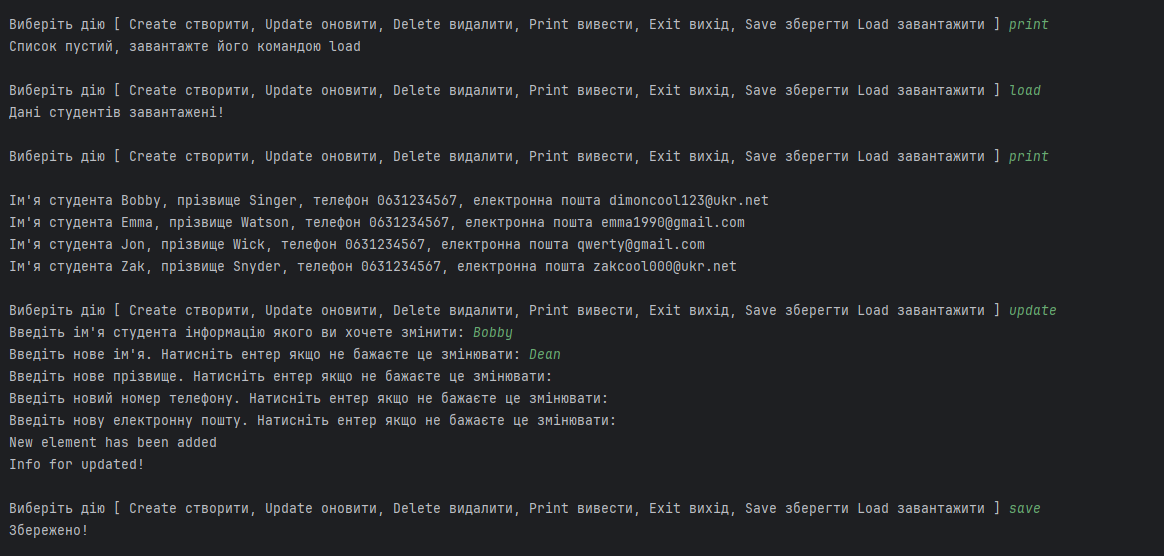
Код файла **Lab\_02.py**:

|  |
| --- |
| from sys import argv import csv  fileName = None  students = []  def printAllList():  if (len(students) == 0):  print("Список пустий, завантажте його командою load")  else:  print(" ")  for elem in students:  strForPrint = "Ім'я студента " + elem["name"] + ", прізвище " + elem["surname"] + ", телефон " + elem["phone"] + ", електронна пошта " + elem["email"]  print(strForPrint)   def askParameters():  name = input("Pease enter student name: ")  surname = input("Pease enter student surname: ")  phone = input("Please enter student phone: ")  email = input("Pease enter student email: ")  return name,surname,phone,email  def addNewElement(name,surname,phone,email):   newItem = {"name": name, "surname": surname, "phone": phone, "email": email}  # find insert position  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):   deletePosition = -1  for item in students:  if name == item["name"]:  deletePosition = students.index(item)  break  if deletePosition == -1:  print("Студента не знайдено")  else:  print("Видаляю позицію" + str(deletePosition))  del students[deletePosition]  print("Студент був видалений")  return   def updateElement():  name = input("Введіть ім'я студента інформацію якого ви хочете змінити: ")   position = -1  for item in students:  if name == item["name"]:  position = students.index(item)  break  if position == -1:  print("Студента не знайдено")  else:   name = input("Введіть нове ім'я. Натисніть ентер якщо не бажаєте це змінювати: ")  if not name:  name = students[position]["name"]   surname = input("Введіть нове прізвище. Натисніть ентер якщо не бажаєте це змінювати: ")  if not surname:  surname = students[position]["surname"]   phone = input("Введіть новий номер телефону. Натисніть ентер якщо не бажаєте це змінювати: ")  if not phone:  phone = students[position]["phone"]   email = input("Введіть нову електронну пошту. Натисніть ентер якщо не бажаєте це змінювати: ")  if not email:  email = students[position]["email"]   del students[position]   addNewElement(name,surname,phone,email)   print("Info for updated!")  def loadArgs():  if (len(argv) == 2 and argv[1].endswith(".csv")):  global fileName  fileName = argv[1]  else:  print("Можливий тільки 1 аргумент та він бути типа: file.csv")  exit()  def loadStudents():  with open(fileName) as file:  reader = csv.DictReader(file)  for row in reader:  students.append({"name": row["name"], "surname": row["surname"], "phone": row["phone"], "email": row["email"]})  print("Дані студентів завантажені!")   def saveStudents():  with open(fileName, "w", newline='') as file:  fieldnames = ["name", "surname", "phone", "email"]  writer = csv.DictWriter(file, fieldnames=fieldnames)  writer.writeheader()   writer.writerows(students)  print("Збережено!")  def main():  loadArgs()   while True:  choice = input("\nВиберіть дію [ Create створити, Update оновити, Delete видалити, Print вивести, Exit вихід, Save зберегти Load завантажити ] ")  match choice.lower().strip():  case "create":  addNewElement(\*askParameters())  printAllList()  case "update":  updateElement()  case "delete":  name = input("Введіть ім'я студента на видалення: ")  deleteElement(name)  case "print":  printAllList()  case "exit":  break  case "save":  saveStudents()  case "load":  loadStudents()  case \_:  print("Wrong choice")  if \_\_name\_\_ == "\_\_main\_\_":  main() |

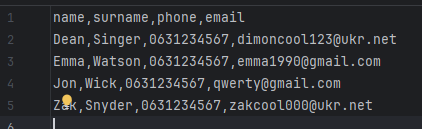
Вміст файлу **students.csv**:



Результат роботи програми:



Змінений файл studends.csv

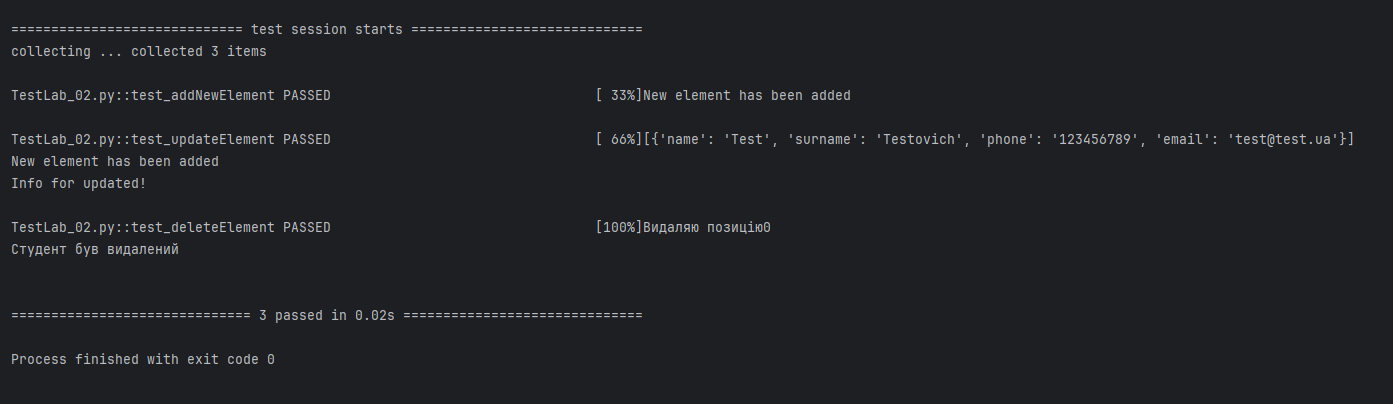


Наступним завданням було покрити основний функціонал юніт тестами.

Створив новий файл TestLab\_02.py та написав наступний код:

from lab\_02.Lab\_02 import students, addNewElement, deleteElement  
  
def test\_addNewElement():  
 addNewElement("Test", "Testovich", "123456789", "test@test.ua")  
 assert len(students) == 1  
 assert students[0]["name"] == "Test"  
  
def test\_updateElement():  
 print(students)  
 name = "Test"  
  
 position = -1  
 for item in students:  
 if name == item["name"]:  
 position = students.index(item)  
 break  
 assert position != 1  
  
 surname = "NewSurname"  
 phone = students[position]["phone"]  
 email = students[position]["email"]  
  
 assert name is not None  
 assert surname is not None  
 assert phone is not None  
 assert email is not None  
  
 del students[position]  
  
 addNewElement(name, surname, phone, email)  
  
 assert students[position]["surname"] == "NewSurname"  
  
 print("Info for updated!")  
  
def test\_deleteElement():  
 deleteElement("Test")  
 assert len(students) == 0

Результат виконання був успішним, значить все працює правильно:



**Висновок:** На цій лабораторній роботі я дороблював існуючий код, вивчав параметри командного рядка та покрив деякий функціонал свого коду юніт тестами.