# **Лабораторна робота №2**

Робота з файлами. Юніт тести.

**Мета роботи:** 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. 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.

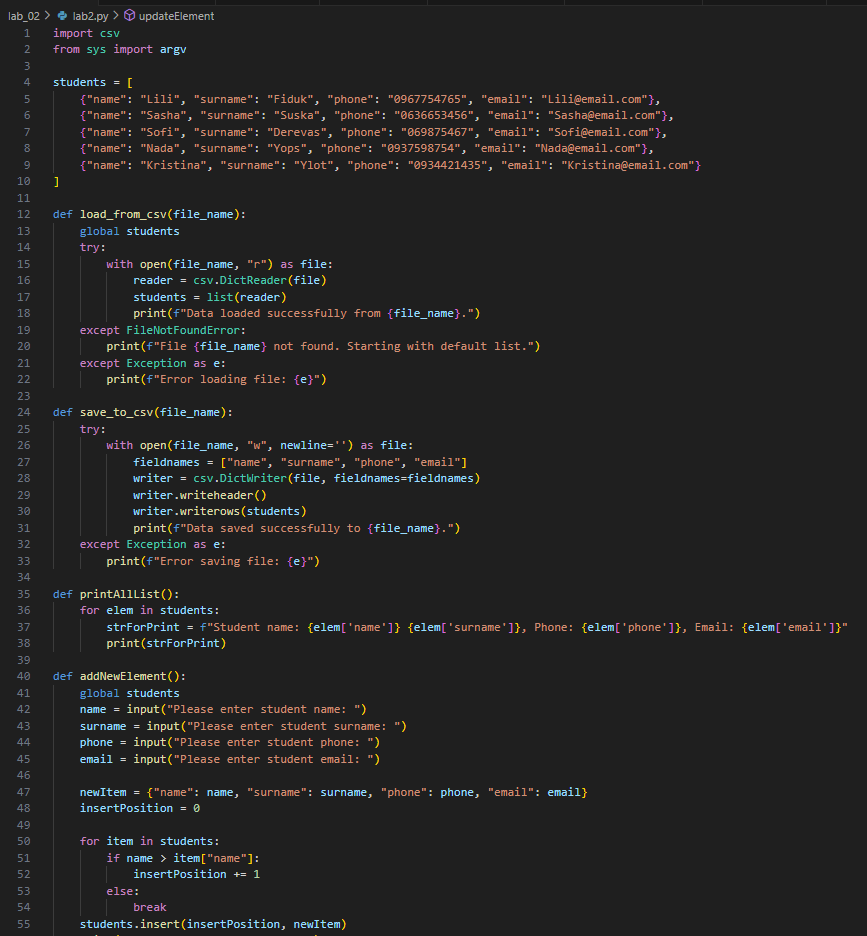
3) покриття основного функціоналу програми Юніт тестами.

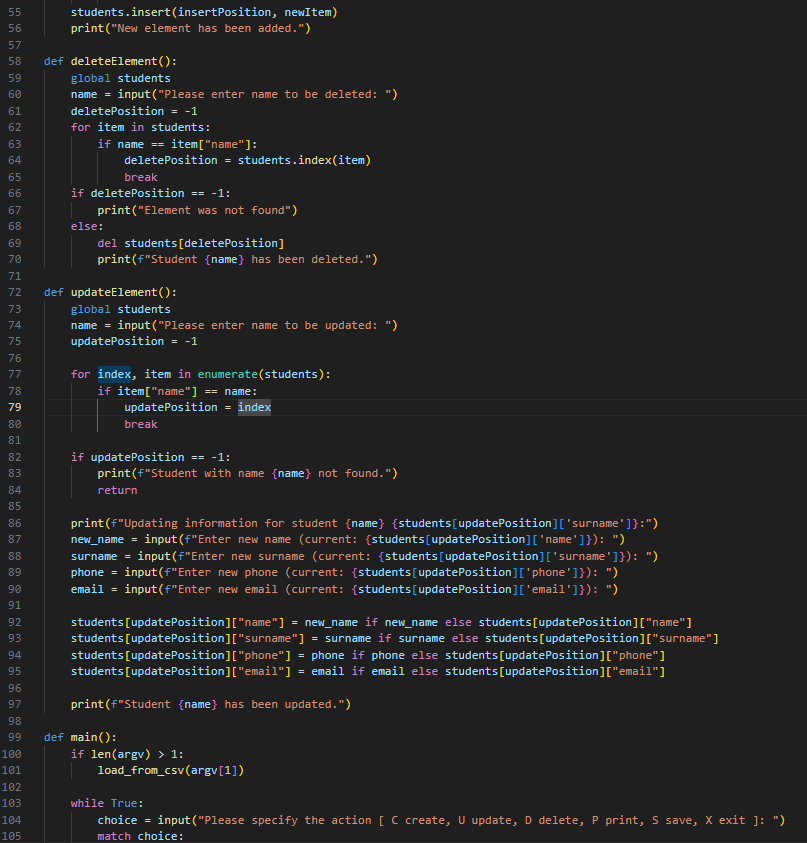
Перед виконанням роботи слід ознайомитись з прикладами.

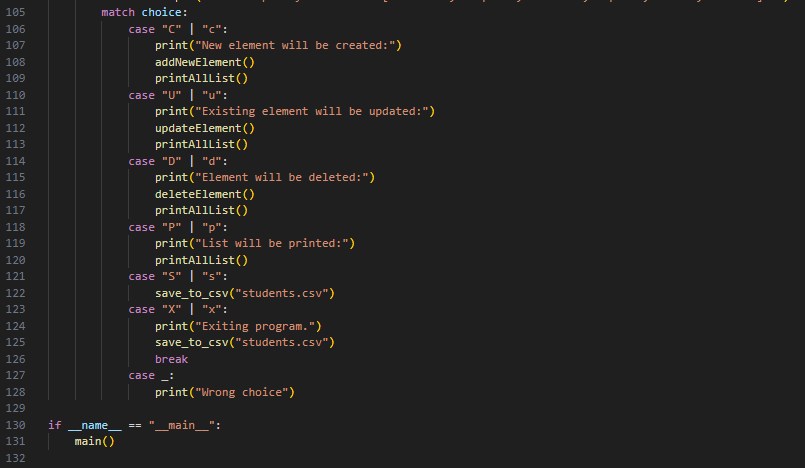
Текст програми разом зі звітом розмістити в директорії lab\_02. Директорію lab\_02 розмістити в директорії, що використовується для виконання практичних завдань по кожній лекції та має назву TP-KB-22[1 or 2]-Name-Surname.

Хід виконання завдання:

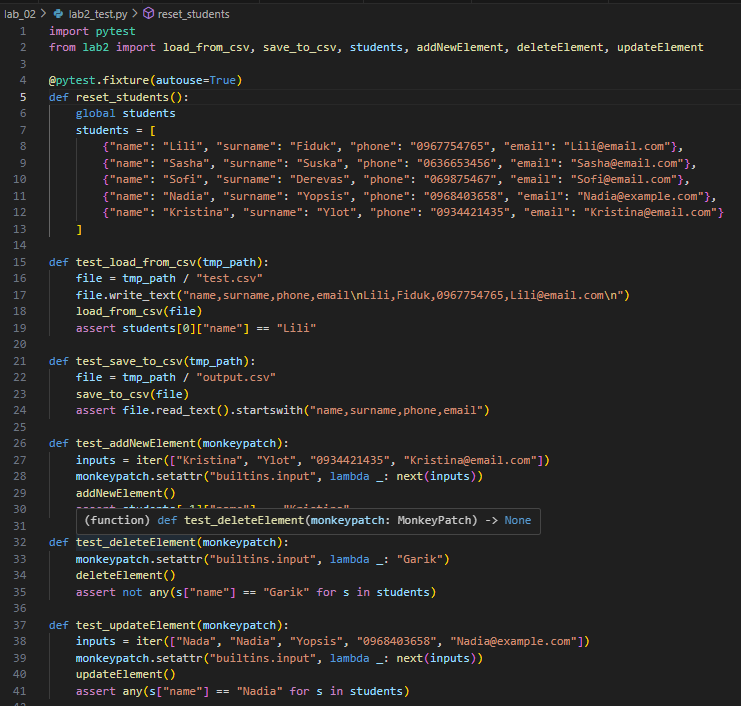
Файл lab2.py



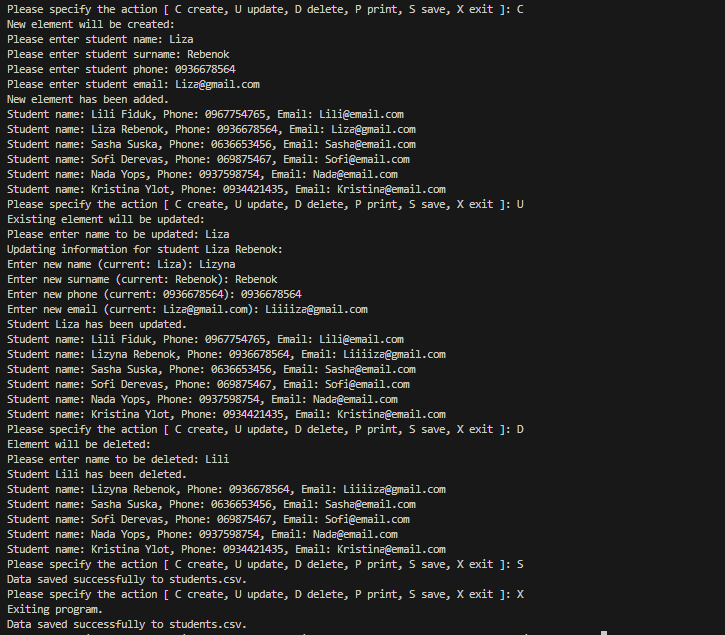




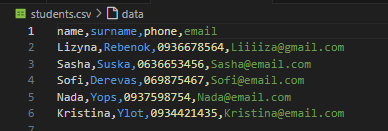
Файл lab2\_test.py



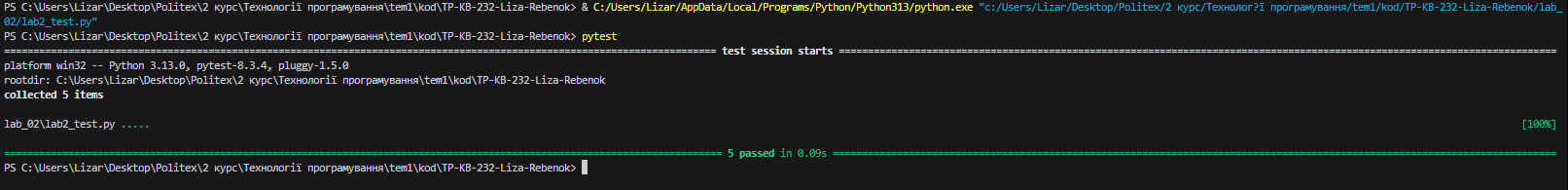
Вивід:



Кінцевий вигляд файлу students.csv



Кінцевий результат перевірки:



Текст Файлу lab2.py :

|  |
| --- |
| import csv  from sys import argv  students = [  {"name": "Lili", "surname": "Fiduk", "phone": "0967754765", "email": "Lili@email.com"},  {"name": "Sasha", "surname": "Suska", "phone": "0636653456", "email": "Sasha@email.com"},  {"name": "Sofi", "surname": "Derevas", "phone": "069875467", "email": "Sofi@email.com"},  {"name": "Nada", "surname": "Yops", "phone": "0937598754", "email": "Nada@email.com"},  {"name": "Kristina", "surname": "Ylot", "phone": "0934421435", "email": "Kristina@email.com"}  ]  def load\_from\_csv(file\_name):  global students  try:  with open(file\_name, "r") as file:  reader = csv.DictReader(file)  students = list(reader)  print(f"Data loaded successfully from {file\_name}.")  except FileNotFoundError:  print(f"File {file\_name} not found. Starting with default list.")  except Exception as e:  print(f"Error loading file: {e}")  def save\_to\_csv(file\_name):  try:  with open(file\_name, "w", newline='') as file:  fieldnames = ["name", "surname", "phone", "email"]  writer = csv.DictWriter(file, fieldnames=fieldnames)  writer.writeheader()  writer.writerows(students)  print(f"Data saved successfully to {file\_name}.")  except Exception as e:  print(f"Error saving file: {e}")  def printAllList():  for elem in students:  strForPrint = f"Student name: {elem['name']} {elem['surname']}, Phone: {elem['phone']}, Email: {elem['email']}"  print(strForPrint)  def addNewElement():  global students  name = input("Please enter student name: ")  surname = input("Please 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 students:  if name > item["name"]:  insertPosition += 1  else:  break  students.insert(insertPosition, newItem)  print("New element has been added.")  def deleteElement():  global students  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(f"Student {name} has been deleted.")  def updateElement():  global students  name = input("Please enter name to be updated: ")  updatePosition = -1  for index, item in enumerate(students):  if item["name"] == name:  updatePosition = index  break  if updatePosition == -1:  print(f"Student with name {name} not found.")  return  print(f"Updating information for student {name} {students[updatePosition]['surname']}:")  new\_name = input(f"Enter new name (current: {students[updatePosition]['name']}): ")  surname = input(f"Enter new surname (current: {students[updatePosition]['surname']}): ")  phone = input(f"Enter new phone (current: {students[updatePosition]['phone']}): ")  email = input(f"Enter new email (current: {students[updatePosition]['email']}): ")  students[updatePosition]["name"] = new\_name if new\_name else students[updatePosition]["name"]  students[updatePosition]["surname"] = surname if surname else students[updatePosition]["surname"]  students[updatePosition]["phone"] = phone if phone else students[updatePosition]["phone"]  students[updatePosition]["email"] = email if email else students[updatePosition]["email"]  print(f"Student {name} has been updated.")  def main():  if len(argv) > 1:  load\_from\_csv(argv[1])  while True:  choice = input("Please specify the action [ C create, U update, D delete, P print, S save, X exit ]: ")  match choice:  case "C" | "c":  print("New element will be created:")  addNewElement()  printAllList()  case "U" | "u":  print("Existing element will be updated:")  updateElement()  printAllList()  case "D" | "d":  print("Element will be deleted:")  deleteElement()  printAllList()  case "P" | "p":  print("List will be printed:")  printAllList()  case "S" | "s":  save\_to\_csv("students.csv")  case "X" | "x":  print("Exiting program.")  save\_to\_csv("students.csv")  break  case \_:  print("Wrong choice")  if \_\_name\_\_ == "\_\_main\_\_":  main() |

Текст Файлу lab2\_test.py:

|  |
| --- |
| import pytest  from lab2 import load\_from\_csv, save\_to\_csv, students, addNewElement, deleteElement, updateElement  @pytest.fixture(autouse=True)  def reset\_students():  global students  students = [  {"name": "Lili", "surname": "Fiduk", "phone": "0967754765", "email": "Lili@email.com"},  {"name": "Sasha", "surname": "Suska", "phone": "0636653456", "email": "Sasha@email.com"},  {"name": "Sofi", "surname": "Derevas", "phone": "069875467", "email": "Sofi@email.com"},  {"name": "Nadia", "surname": "Yopsis", "phone": "0968403658", "email": "Nadia@example.com"},  {"name": "Kristina", "surname": "Ylot", "phone": "0934421435", "email": "Kristina@email.com"}  ]  def test\_load\_from\_csv(tmp\_path):  file = tmp\_path / "test.csv"  file.write\_text("name,surname,phone,email\nLili,Fiduk,0967754765,Lili@email.com\n")  load\_from\_csv(file)  assert students[0]["name"] == "Lili"  def test\_save\_to\_csv(tmp\_path):  file = tmp\_path / "output.csv"  save\_to\_csv(file)  assert file.read\_text().startswith("name,surname,phone,email")  def test\_addNewElement(monkeypatch):  inputs = iter(["Kristina", "Ylot", "0934421435", "Kristina@email.com"])  monkeypatch.setattr("builtins.input", lambda \_: next(inputs))  addNewElement()  assert students[-1]["name"] == "Kristina"  def test\_deleteElement(monkeypatch):  monkeypatch.setattr("builtins.input", lambda \_: "Garik")  deleteElement()  assert not any(s["name"] == "Garik" for s in students)  def test\_updateElement(monkeypatch):  inputs = iter(["Nada", "Nadia", "Yopsis", "0968403658", "Nadia@example.com"])  monkeypatch.setattr("builtins.input", lambda \_: next(inputs))  updateElement()  assert any(s["name"] == "Nadia" for s in students) |

Посилання на github:

<https://github.com/Lizarebenokk>