Технології програмування

Лабораторна робота №2

**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**

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

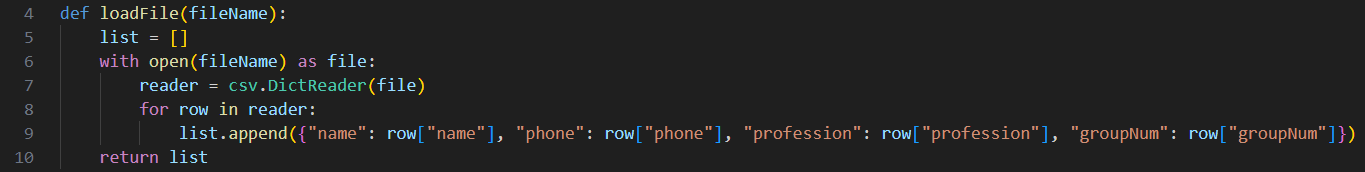
1. Завантаження початкових данні для довідника з csv файлу. Приклад csv файлу (lab2.csv), та оригінальній файл формату xlsx розміщені поряд з вказівками. Ім’я файлу для завантаження даних повинно передаватись в якості параметрів командного рядку.
2. Зберігання довідника студентів у форматі CSV перед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

**Хід роботи**

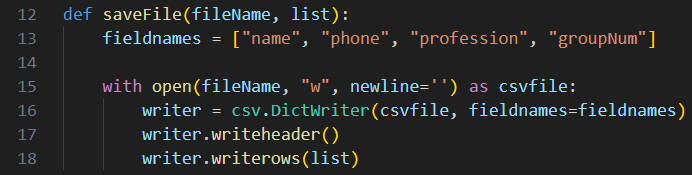
1. Імпортував два модулі csv та sys.



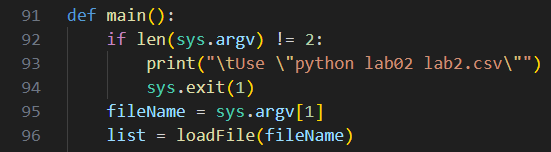
1. Створив функцію яка підвантажує дані з файлу csv в код.



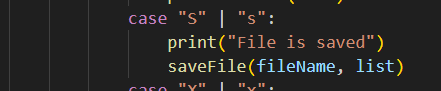
1. Створив функцію яка зберігає дані з коду в файл csv.



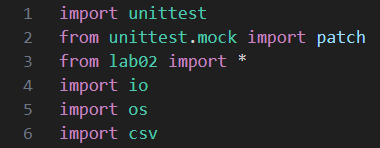
1. Модернізував функцію main для прийому аргементу (необхідно вибрати файл зі списком) при запуску коду.



1. Дописав у варіантах виклику функцій можливість виконати зберігання даних у файл csv.



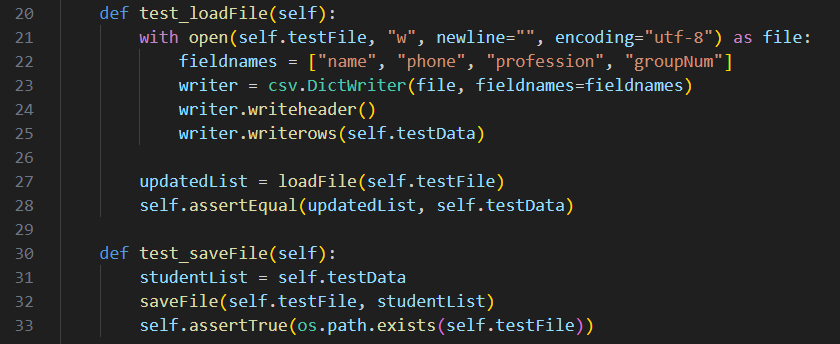
1. Створив новий файл для юніт тестів. Імпортував туди такі модулі:



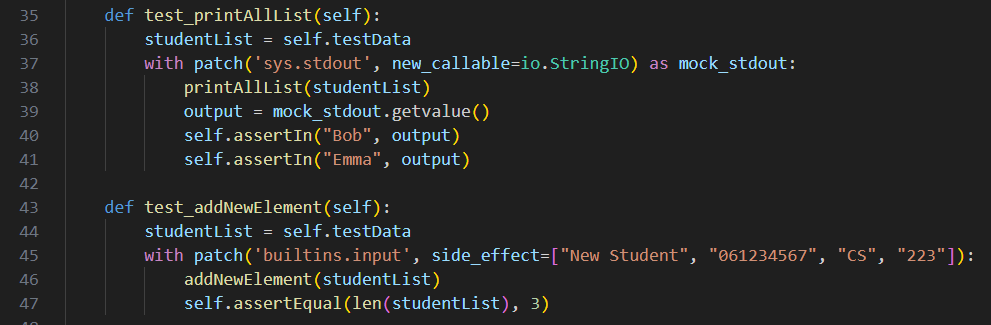
1. Написав функції для підготовки тестів та завершення тестування.



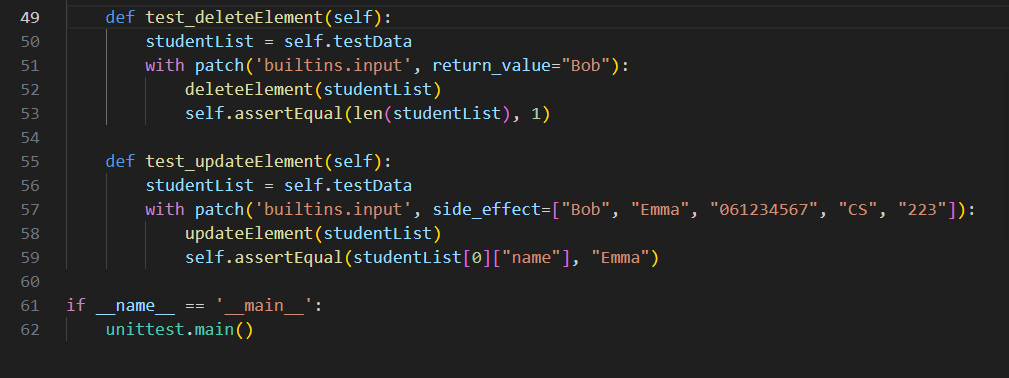
1. Написав тести для перевірки завантаження та зберігання списку.



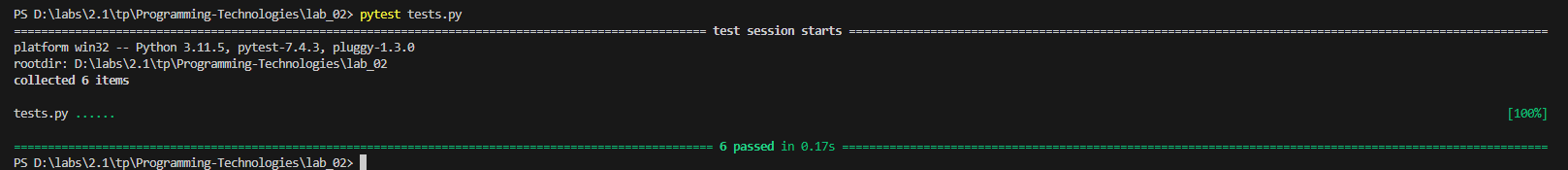
1. Для друку списка та додавання нового елементу.



1. Для видалення та оновлення елементів списка.



1. Перевірив роботу тестів з допомогою pytest. Всі тести пройшли успішно.



Код lab02.py:

import csv

import sys

def loadFile(fileName):

    list = []

    with open(fileName) as file:

        reader = csv.DictReader(file)

        for row in reader:

            list.append({"name": row["name"], "phone": row["phone"], "profession": row["profession"], "groupNum": row["groupNum"]})

    return list

def saveFile(fileName, list):

    fieldnames = ["name", "phone", "profession", "groupNum"]

    with open(fileName, "w", newline='') as csvfile:

        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

        writer.writeheader()

        writer.writerows(list)

def printAllList(list):

    for elem in list:

        strForPrint = "Student name is " + elem["name"] + ",  Phone is " + elem["phone"] + ", Profession is " + elem["profession"] + ", Group number is " + elem["groupNum"]

        print(strForPrint)

    return

def addNewElement(list):

    name = input("Please enter student name: ")

    phone = input("Please enter student phone: ")

    profession = input("Please enter student profession: ")

    groupNum = input("Please enter student group number: ")

    newItem = {"name": name, "phone": phone, "profession": profession, "groupNum": groupNum}

    # find insert position

    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

def deleteElement(list):

    name = input("Please enter name to be delated: ")

    deletePosition = -1

    for item in list:

        if name == item["name"]:

            deletePosition = list.index(item)

            break

    if deletePosition == -1:

        print("Element was not found")

    else:

        print("Delete position " + str(deletePosition))

        # list.pop(deletePosition)

        del list[deletePosition]

    return

def updateElement(list):

    name = input("Please enter name to be updated: ")

    updatePosition = -1

    trueForDel = False

    for item in list:

        if name == item["name"]:

            updatePosition = list.index(item)

            break

    if updatePosition == -1:

        print("Element was not found")

    else:

        print("Update position " + str(updatePosition))

        trueForDel = True

    if trueForDel == True:

        del list[updatePosition]

    name = input("Please enter student name: ")

    phone = input("Please enter student phone: ")

    profession = input("Please enter student profession: ")

    groupNum = input("Please enter student group number: ")

    newItem = {"name": name, "phone": phone, "profession": profession, "groupNum": groupNum}

    insertPosition = 0

    for item in list:

        if name > item["name"]:

            insertPosition += 1

        else:

            break

    list.insert(insertPosition, newItem)

    print("The element has been updated.")

    return

def main():

    if len(sys.argv) != 2:

        print("\tUse \"python lab02 lab2.csv\"")

        sys.exit(1)

    fileName = sys.argv[1]

    list = loadFile(fileName)

    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(list)

                printAllList(list)

            case "U" | "u":

                print("Existing element will be updated")

                updateElement(list)

                printAllList(list)

            case "D" | "d":

                print("Element will be deleted")

                deleteElement(list)

            case "P" | "p":

                print("List will be printed")

                printAllList(list)

            case "S" | "s":

                print("File is saved")

                saveFile(fileName, list)

            case "X" | "x":

                print("Exit")

                break

            case \_:

                print("Wrong choice")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Код tests.py:

import unittest

from unittest.mock import patch

from lab02 import \*

import io

import os

import csv

class Lab2Test(unittest.TestCase):

    def setUp(self):

        self.testFile = "testData.csv"

        self.testData = [

            {"name": "Bob", "phone": "061234567", "profession": "CS", "groupNum": "222"},

            {"name": "Emma", "phone": "061234568", "profession": "PE", "groupNum": "221"}

        ]

    def tearDown(self):

        if os.path.exists(self.testFile):

            os.remove(self.testFile)

    def test\_loadFile(self):

        with open(self.testFile, "w", newline="", encoding="utf-8") as file:

            fieldnames = ["name", "phone", "profession", "groupNum"]

            writer = csv.DictWriter(file, fieldnames=fieldnames)

            writer.writeheader()

            writer.writerows(self.testData)

        updatedList = loadFile(self.testFile)

        self.assertEqual(updatedList, self.testData)

    def test\_saveFile(self):

        studentList = self.testData

        saveFile(self.testFile, studentList)

        self.assertTrue(os.path.exists(self.testFile))

    def test\_printAllList(self):

        studentList = self.testData

        with patch('sys.stdout', new\_callable=io.StringIO) as mock\_stdout:

            printAllList(studentList)

            output = mock\_stdout.getvalue()

            self.assertIn("Bob", output)

            self.assertIn("Emma", output)

    def test\_addNewElement(self):

        studentList = self.testData

        with patch('builtins.input', side\_effect=["New Student", "061234567", "CS", "223"]):

            addNewElement(studentList)

            self.assertEqual(len(studentList), 3)

    def test\_deleteElement(self):

        studentList = self.testData

        with patch('builtins.input', return\_value="Bob"):

            deleteElement(studentList)

            self.assertEqual(len(studentList), 1)

    def test\_updateElement(self):

        studentList = self.testData

        with patch('builtins.input', side\_effect=["Bob", "Emma", "061234567", "CS", "223"]):

            updateElement(studentList)

            self.assertEqual(studentList[0]["name"], "Emma")

if \_\_name\_\_ == '\_\_main\_\_':

    unittest.main()

**Висновок:**

Під час виконання цієї лабораторної використовуючи теоретичне підґрунтя про роботу з файлами та тестування коду у мові Python розширив програму телефонного довідника студентів додавши функціонал, який вказано в завданні до лабораторної роботи. Закріпив знання роботи з файлами та ознайомився з функціоналом написання юніт тестів.