

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення
Дисципліна: Скриптові мови програмування (Python)

Лабораторна робота №3
Тема: «РОБОТА З ФАЙЛАМИ У PYTHON»

Виконав: ст. гр. КН-24
Куріщенко П. В.
Перевірив: ассистент
Ткаченко О.С.

Кропивницький 2024

Мета роботи - навчитися здійснювати операції читання та запису для файлів у Python.

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

Створіть файли, у яких будуть міститися рядки з іменами студентів та їх середніми балами. Кожен файл буде відповідати окремій групі.

Реалізуйте читання файлів, запис та дозапис у файли, пошук файлів у каталозі та пошук даних у файлі. Також реалізуйте сортування даних у файлі за середнім балом.

ОСНОВНІ ФУНКЦІЇ

1. `createFile(fileName)`

Призначення:

Створює новий CSV-файл із заголовками і записує в нього студентів, якщо файл ще не існує.

Принцип роботи:

- Перевірка існування файлу.
- Якщо не існує:
 - Створюється файл.
 - Записуються заголовки.
 - Дані вводяться через функцію `whatGroup`.
 - Сортування через `sortedStudents`.
 - Запис у файл.
- Якщо файл існує — виводиться повідомлення без змін.

Проектне рішення:

- Унікає перезапису існуючих файлів.
- Інкапсулює створення файлу, логіку вводу та сортування в одному місці.

Блок-схема:



Лістинг:

```
def createFile(fileName):  
    """  
    Створити файл, якщо він не існує, і заповнити його даними.  
    """  
    if not os.path.exists(fileName):  
        with open(fileName, 'w', newline='', encoding='utf-8') as file:  
            writer = csv.writer(file)  
            writer.writerow(["№", "Прізвище", "Ім'я", "Середній бал"])  
            students = whatGroup(fileName)  
            sortedStudents(students)
```

```
        writer.writerow(students)
    print(f"Файл '{fileName}' створено.")
else:
    print(f"Файл '{fileName}' вже існує.")
```

2. **showInfo(fileName)**

Призначення:

Відображає вміст CSV-файлу зі списком студентів у форматovanому вигляді.

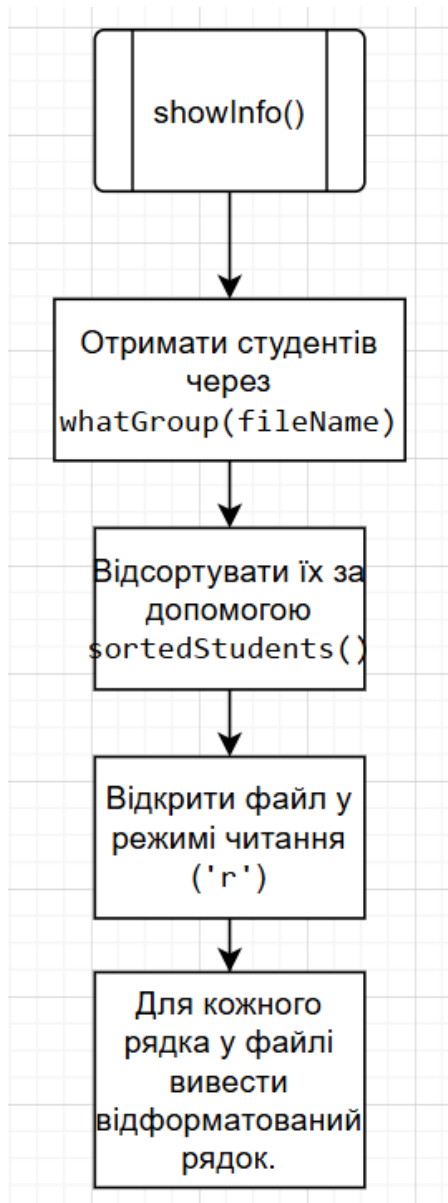
Принцип роботи:

- Зчитування та сортування студентів через зовнішні функції.
- Вивід кожного рядка файлу.

Проектне рішення:

- Відділяє логіку відображення від логіки сортування.
- Підтримує повторне використання функцій сортування.

Блок-схема:



Лістинг:

```
def showInfo(fileName):  
    """  
    Вивести інформацію про студентів.  
    """  
    sortedStudents(whatGroup(fileName))  
    with open(fileName, 'r', encoding='utf-8') as file:  
        reader = csv.reader(file)  
        for row in reader:  
            print("{:<3} {:<15} {:<15} {}".format(*row))
```

Результат роботи функції:

Меню групи \ Ввести: 1			
№	Прізвище	Ім'я	Середній бал
1	Куріщенко	Павло	99.0
2	Мельник	Дмитро	97.0
3	Мироненко	Ярослав	96.0
4	Булюкін	Володимир	94.0
5	Червонець	Артем	92.0

3. **addStudent(fileName)**

Призначення:

Додає нового студента у кінець списку, сортує його і оновлює файл.

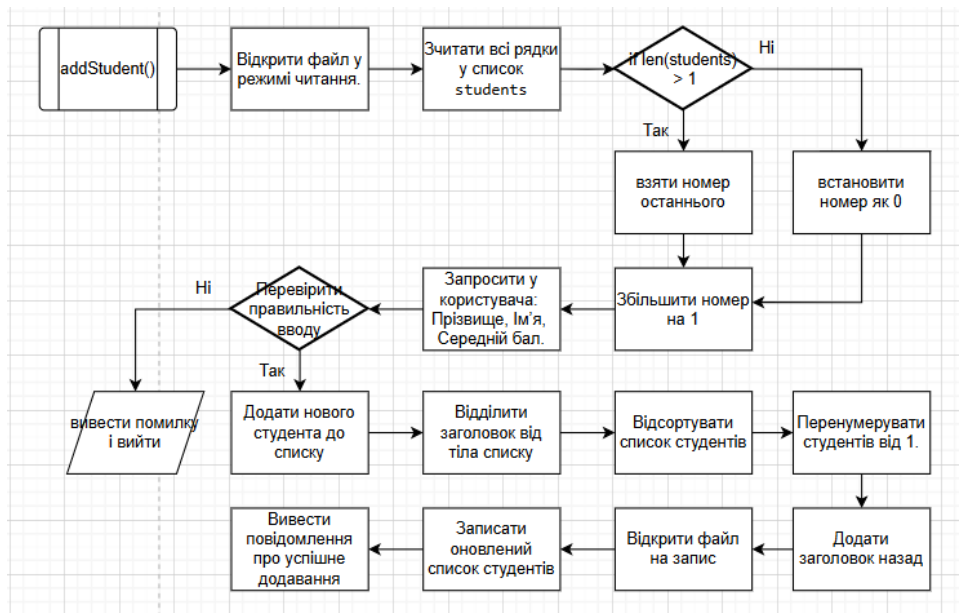
Принцип роботи:

- Зчитування існуючих даних.
- Отримання нових даних від користувача.
- Валідація вводу.
- Додавання нового студента до списку.
- Сортування та перенумерація.
- Запис оновленого списку у файл.

Проектне рішення:

- Чітке розділення етапів: зчитування → валідація → додавання → оновлення.
- Перевірка валідності даних до збереження.
- Повна заміна вмісту файлу (не дописування в кінець), що гарантує цілісність структури.

Блок-схема:



Лістинг:

```
def addStudent(fileName):
    """
    Додати студента до групи.
    """
    with open(fileName, 'r', encoding='utf-8') as file:
        reader = csv.reader(file)
        students = list(reader)

    if len(students) > 1:
        lastRow = int(students[-1][0])
    else:
        lastRow = 0

    number = lastRow + 1
    surname = input("Введіть прізвище студента: ").strip()
    name = input("Введіть ім'я студента: ").strip()
    score = float(input("Введіть середній бал студента: "))
    if not surname: return print("Прізвище не може бути порожнім.")
    if not name: return print("Ім'я не може бути порожнім.")
    if score < 0 or score > 100: return print("Середній бал повинен бути в межах від 0 до 100.")

    students.append([number, surname, name, score])

    header = students[0]
    students = students[1:]
    sortedStudents(students)

    for i, student in enumerate(students, start=1):
```

```
student[0] = i
students.insert(0, header)

with open(fileName, 'w', newline='', encoding='utf-8') as file:
    writer = csv.writer(file)
    writer.writerows(students)

print(f"Студента {surname} {name} додано до групи з оцінкою {score}.")
```

Результат роботи функції:

```
Меню групи \ Ввести: 2
Введіть прізвище студента: Ткаченко
Введіть ім'я студента: Олександр
Введіть середній бал студента: 100
Студента Ткаченко Олександр додано до групи з оцінкою 100.0.
```

4. `deleteStudent(fileName)`

Призначення:

Видаляє студента за прізвищем і оновлює список.

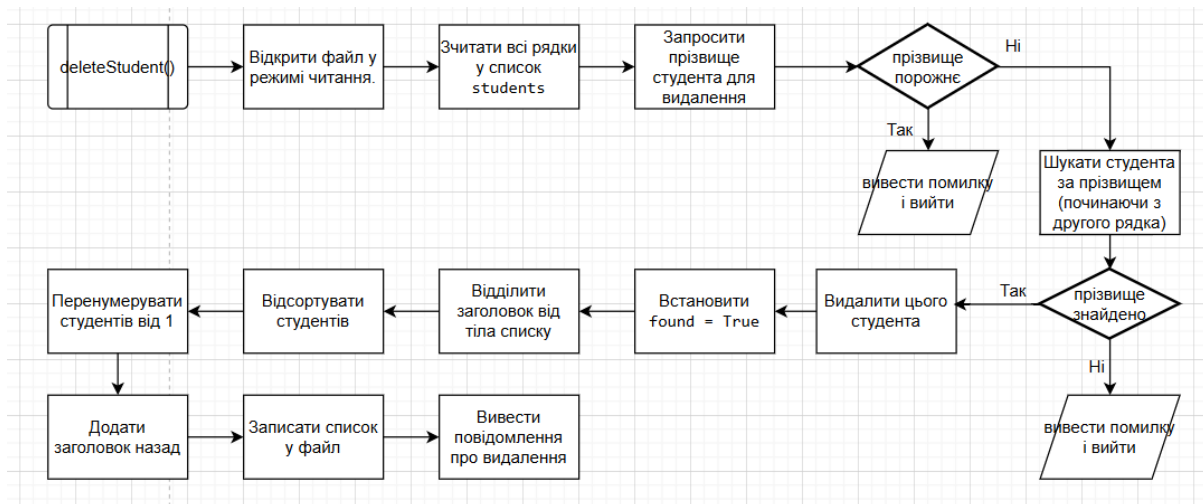
Принцип роботи:

- Зчитування даних.
- Пошук студента за прізвищем.
- Якщо знайдено — видалення.
- Сортування, перенумерація, оновлення файлу.

Проектне рішення:

- Видалення тільки першого знайденого входження (можна змінити на повне видалення всіх).
- Уніфікований підхід до збереження даних після змін (сортування + запис усіх даних).

Блок-схема:



Лістинг:

```
def deleteStudent(fileName):
    """
    Видалити студента з групи.
    """
    with open(fileName, 'r', encoding='utf-8') as file:
        reader = csv.reader(file)
        students = list(reader)

    surname = input("Введіть прізвище студента для видалення: ").strip()
    if not surname: return print("Прізвище не може бути порожнім.")

    found = False
    for i in range(1, len(students)):
        if students[i][1] == surname:
            found = True
            del students[i]
            break

    if not found: return print(f"Студента з прізвищем {surname} не знайдено.")

    header = students[0]
    students = students[1:]
    sortedStudents(sortedStudents(students))

    for i, student in enumerate(students, start=1):
        student[0] = i
    students.insert(0, header)

    with open(fileName, 'w', newline='', encoding='utf-8') as file:
        writer = csv.writer(file)
        writer.writerows(students)
```

```
print(f"Студента {surname} видалено з групи.")
```

Результат роботи функції:

```
Меню групи \ Ввести: 3  
Введіть прізвище студента для видалення: Куріщенко  
Студента Куріщенко видалено з групи.
```

5. `findStudent(fileName)`

Призначення:

Шукає і виводить студента за прізвищем.

Принцип роботи:

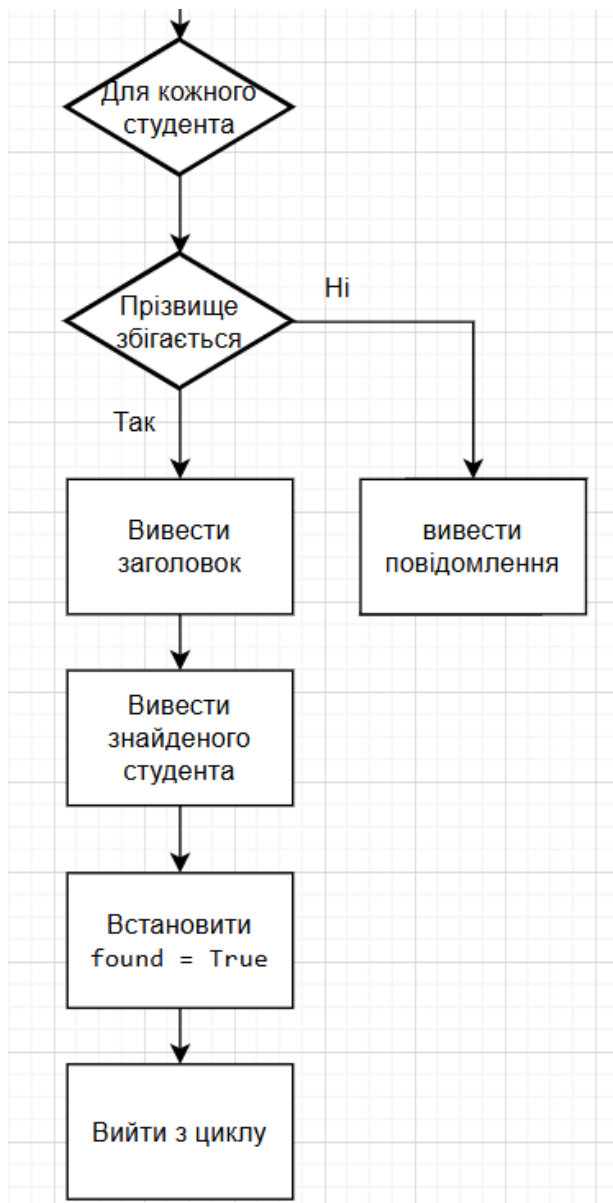
- Зчитування всіх студентів.
- Пошук у циклі за другим елементом рядка (прізвищем).
- Вивід знайденого студента або повідомлення про відсутність.

Проектне рішення:

- Мінімальна логіка — виконує тільки пошук і вивід.
- Немає зміни файлу → безпечна операція.

Блок-схема:





Лістинг:

```
def findStudent(fileName):  
    """  
    Знайти студента за його прізвищем.  
    """  
    with open(fileName, 'r', encoding='utf-8') as file:  
        reader = csv.reader(file)  
        students = list(reader)  
  
    surname = input("Введіть прізвище студента для пошуку: ").strip()  
    if not surname: return print("Прізвище не може бути порожнім.")  
  
    found = False  
    for student in students[1:]
```

```

if student[1] == surname:
    found = True
    print("{:<3} {:<15} {:<15} {}".format("№", "Прізвище", "Ім'я",
"Середній бал"))
    print("{:<3} {:<15} {:<15} {}".format(*student))
    break

if not found:
    print(f"Студента з прізвищем {surname} не знайдено.")

```

Результат роботи функції:

```

Меню групи \ Ввести: 4
Введіть прізвище студента для пошуку: Куріщенко
Студента з прізвищем Куріщенко не знайдено.
Меню групи \ Ввести: 4
Введіть прізвище студента для пошуку: Ткаченко
№   Прізвище      Ім'я      Середній бал
1   Ткаченко      Олександр 100.0

```

КОНТРОЛЬНІ ПИТАННЯ

1. Як здійснюється запис даних у файл у мові Python?

Запис у файл виконується за допомогою функції `open()` з режимом "w" (або іншим режимом запису), методу `write()` та обов'язкового закриття файлу:

```

filename = "mydata.txt"
fd = open(filename, "w") # відкриття файлу на запис
for i in range(10):
    A = i * 18
    fd.write("%i\t%.1f\n" % (i, A)) # запис у файл
fd.close() # закриття файлу

```

2. Як здійснюється читання даних з файлу у мові Python?

Читання файлу відбувається так:

```
import csv

fd = open("mydata.txt", "r") # відкриття файлу на читання
reader = csv.reader(fd, delimiter="\t") # зчитування з розділювачем
for row in reader:
    print(row) # виведення кожного рядка
fd.close() # закриття файлу
```

3. Як здійснюється копіювання файлу?

Для копіювання файлів використовується модуль `shutil` і функція `copyfile()`:

```
import shutil

shutil.copyfile("C:\\mydoc.doc", "C:\\My Documents\\mydoc_2.doc")
```

4. Який синтаксис функції `open()` у мові Python?

```
my_file = open(ім'я_файлу [, режим_доступу] [, буферизація])
```

- **Ім'я файлу** — шлях до файлу (відносний або абсолютний).
- **Режим доступу** — режим, у якому відкривається файл ("r", "w", "a" тощо).
- **Буферизація** — необов'язковий параметр для керування буферизацією.

5. Які бувають режими роботи з файлами?

Режим	Призначення
'r'	Відкриття на читання (за замовчуванням).
'w'	Відкриття на запис, файл очищується або створюється.
'x'	Ексклюзивне створення, якщо файл існує — помилка.
'a'	Дозапис у кінець файлу.

'b'	Двійковий режим.
't'	Текстовий режим (за замовчуванням).
'+'	Читання і запис одночасно.

Комбіновані приклади:

- 'r+b' — читання і запис у двійковому режимі, з початку файлу.
- 'w+b' — читання і запис у двійковому режимі, очищення файлу.
- 'a+' — додавання і читання, курсор у кінці файлу.