

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет

ЗВІТ
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 3
з навчальної дисципліни
“Скриптові мови програмування (Python)”

Робота з файлами у мові Python

ВИКОНАВ
студент академічної групи КН-24
Мироненко Я.М

ПЕРЕВІРИВ
асистент кафедри кібербезпеки та
програмного забезпечення
Ткаченко О.С

Тема: Робота з файлами у мові Python

Мета: навчитися здійснювати операції читання та запису для файлів у Python

Завдання

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.

Лістинг функції 1

```
def new_file(): # 1)Створення нового файлу
    """Функція для створення нового файлу .csv та директорії files/ якщо потрібно"""

    # Перевіряємо чи існує папка files, якщо ні – створюємо її
    folder_name = 'files'
    if not os.path.exists(folder_name): # перевірка наявності директорії files
        os.makedirs(folder_name) # створення директорії files

    name_file = input('Введіть назву файлу: ')
    file_path = os.path.join(folder_name, f'{name_file}.csv') # шлях до файлу

    with open(file_path, 'w+', encoding='utf-8', newline='') as csvfile: # створення
        файлу
        name_columns = ['Прізвище Ім\'я', 'Група', 'Середній бал'] # назви
        колонок
        writer = csv.DictWriter(csvfile, fieldnames=name_columns) # запис колонок у
        файл
        writer.writeheader() # запис заголовка
        print(f'Файл {name_file}.csv створено')
```

Принцип роботи функції

Функція new_file() створює новий текстовий файл формату .csv у директорії files/ та записує в нього заголовки колонок.

Принцип роботи:

Запит користувача на введення назви файлу

Використовується input(), щоб отримати назву файлу від користувача.

Відкриття файлу у режимі w+

Файл створюється (або перезаписується, якщо вже існує) у директорії files/.

Використовується кодування utf-8 та newline="" для коректного запису CSV-даних.

Формування заголовків колонок

Створюється список заголовків: 'Прізвище Ім'я', 'Група', 'Середній бал'.

Запис заголовків у файл

Використовується `csv.DictWriter`, щоб записати заголовки у файл.

`writeheader()` записує перший рядок у вигляді заголовка.

Виведення повідомлення про успішне створення файлу

Виводиться повідомлення Файл {name_file} створено, щоб повідомити користувача про завершення операції.

Функція дозволяє швидко створювати .csv файли для збереження списку студентів із зазначенням їхніх імен, груп і середнього балу.

Тестові запуски функції

```
Вас вітає програма для роботи з файлами .csv
В цій програмі ви можете обрати одну з наступних дій
які в свою чергу виконують обрану функцію

1: Створити новий файл .csv
2: Додати студента до списку
3: Вивести вміст файлу
4: Знайти студента за прізвищем
5: Знайти студента за номером у списку
6: Відредагувати данні про студента
7: Сортувати список студентів
8: Видалити студента зі списку
9: Вихід

Оберіть функцію: 1
Введіть назву файлу: new
Файл new створено
```

Рисунок 1

Блок-Схема



Рисунок 2

Лістинг функції 2

```

def derectory_reader(): # 2)Вибір файлу для роботи
    """функція читає директорію files/ та дає можливість обрати файл для читання
    файл обирається за номером у списку"""

    file_list = os.listdir('files') # список файлів у директорії files/
    file_list = [folder for folder in file_list if
os.path.isfile(os.path.join('files', folder)) and folder.endswith('.csv')] # вибірка
файлів з розширенням .csv

    if not file_list: # якщо файлів немає
        print("У директорії 'files/' немає файлів.")
        return None

    print("Оберіть файл для роботи:")
        "\nСписок доступних файлів:")
    for number, file in enumerate(file_list, 1): # вивід списку файлів
        print(f"{number}. {file}") # вивід файлів

    while True:
        try:
            choice = int(input("\nВведіть номер файлу для вибору: "))
            if 1 <= choice <= len(file_list): # перевірка чи введене число в
діапазоні

                selected_file = file_list[choice - 1] # вибір файлу
                selected_path = os.path.join('files', selected_file) # шлях до файлу

```

```

        print(f"Вибрано файл: {selected_file}") # вивід вибраного файлу
        return selected_path # повернення шляху до файлу
    else:
        print(f"Введіть число від 1 до {len(file_list)}")
except ValueError:
    print("Введіть правильне число")

```

Принцип роботи функції

Функція зчитує список файлів у директорії files/, відфільтровує лише файли з розширенням .csv і дозволяє користувачу вибрати один із них за номером у списку.

Лістинг функції 3

```

def add_student(): # 3) Додавання студента
    """Функція для додавання студента в файл обраний функцією derectory_reader()
    з можливістю додавання одного або декількох студентів"""

    selected_path = derectory_reader() # вибір файлу для додавання студентів

    print("\nДодавання студентів:")
    while True:
        student = input("Введіть дані студента у форматі 'Прізвище ім'я, група, середній бал': ")

        parts = [part.strip() for part in student.split(',')] # розділення введених даних

        if len(parts) < 3: # перевірка чи введені всі дані
            print("Недостатньо даних. Потрібно: Прізвище, Ім'я, Група, Середній бал")
            continue

        # Валідація ПІБ (перевірка на наявність цифр)
        if any(char.isdigit() for char in parts[0]):
            print("Помилка: Прізвище та ім'я не повинні містити цифри")
            continue

        # Валідація групи (формат XX-00 або подібний)
        if len(parts) > 1:
            group = parts[1].strip()
            if not (2 <= len(group) <= 6) or not ('-' in group):
                print("Помилка: Група має бути у форматі 'XX-00'")
                continue

        with open(selected_path, 'a', encoding='utf-8', newline='') as csvfile: # відкриття файлу для додавання

```

```

writer = csv.writer(csvfile) # запис в файл
writer.writerow(parts[:3])   # Беремо перші 3 елементи

end = input("Продовжити додавання? (Enter - Так, n - Ні): ")
if end == '':
    continue
elif end == 'n':
    print("Додавання студентів завершено")
    break

```

Принцип роботи функції

Функція додає нових студентів у вибраний користувачем .csv файл. Вона дозволяє вводити дані одного або декількох студентів у форматі "Прізвище Ім'я, Група, Середній бал" і перевіряє правильність введених даних.

Тестові запуски

```

Оберіть функцію: 2
Оберіть файл для роботи:
Список доступних файлів:
1. CB-24.csv
2. CI-24.csv
3. CS-24.csv
4. new.csv

Введіть номер файлу для вибору: 4
Вибрано файл: new.csv

Додавання студентів:
Введіть дані студента у форматі 'Прізвище ім'я, група, середній бал': Олена Курилова, КБ-28, 200
Продовжити додавання? (Enter - Так, n - Ні): n
Додавання студентів завершено

```

Рисунок 3

Блок-Схема



Рисунок 4

Лістинг функції 4

```

def read_file(): # 4) Читання файлу
    """Функція для читання файлу обраного функцією derectory_reader()"""

    selected_path = derectory_reader() # вибір файлу для читання

    with open(selected_path, 'r', encoding='utf-8', newline='') as csvfile: #
відкриття файлу для читання
        reader = csv.reader(csvfile) # читання файлу
        print("\nДані студентів:")
        for student_num, row in enumerate(reader): # перебір рядків файлу
            if student_num == 0: # пропуск заголовка
                print(f"{row[0]} {row[1]} {row[2]}") # вивід заголовка
                print("-" * 40) # розділювач
            else:
                print(f"{student_num}. {row[0]} | {row[1]} | {row[2]}") # вивід даних
студентів
        return None
  
```

Принцип роботи функції

Функція зчитує та виводить вміст вибраного .csv файлу, який містить список студентів.

Тестові запуски функції

```

Оберіть функцію: 3
Оберіть файл для роботи:
Список доступних файлів:
1. CB-24.csv
2. CI-24.csv
3. CS-24.csv
4. new.csv

Введіть номер файлу для вибору: 2
Вибрано файл: CI-24.csv

Дані студентів:
  Прізвище Ім'я   Група   Середній бал
-----
1. Аналізенко Дмитро | КІ-24 | 85.4
2. Бекендов Максим | КІ-24 | 90.2
3. Вебович Артем | КІ-24 | 88.1
4. Датабазов Андрій | КІ-24 | 79.7
5. Запитова Олена | КІ-24 | 91.5
6. Кешован Марк | КІ-24 | 76.8
7. Логінова Катерина | КІ-24 | 84.3
8. Мереженко Ігор | КІ-24 | 89.9
9. Нейромережна Дарина | КІ-24 | 95.7
10. Оптимізенко Павло | КІ-24 | 82.1
11. Потоков Владислав | КІ-24 | 77.6
12. Рендерук Євген | КІ-24 | 80.4
13. Серверний Олексій | КІ-24 | 86.9
14. Тестувальська Марина | КІ-24 | 92.3
15. Фронтендов Олег | КІ-24 | 83.5
16. Хмаренко Назар | КІ-24 | 78.2
17. Циклович Ярослав | КІ-24 | 85.0
18. Шифрук Віталій | КІ-24 | 87.6
19. Юзерова Софія | КІ-24 | 93.1
20. Інтерфейсов Ілля | КІ-24 | 81.7

```

Рисунок 5

Блок-Схема



Рисунок 6

Лістинг функції 5

```
def search_student_name(): # 5)Пошук студента за прізвищем
    """функція для пошуку студента за прізвищем в файлі обраному функцією
    derectory_reader()
    та вивидення його даних"""

    selected_path = derectory_reader() # вибір файлу для пошуку

    search = input("Введіть прізвище студента: ")
    with open(selected_path, 'r', encoding='utf-8', newline='') as csvfile: #
відкриття файлу для читання
        reader = csv.reader(csvfile) # читання файлу
        for student_num, row in enumerate(reader): # перебір рядків файлу
            if student_num == 0: # пропуск заголовка
                continue
            if search in row[0]: # пошук студента за прізвищем
                print(f"Студент {row[0]} з групи {row[1]} має середній бал {row[2]}")
# вивід даних студента
                return {
                    'path': selected_path,
                    'student_num': student_num,
                    'name': row[0],
                    'group': row[1],
                    'average': row[2]
                }
```

Принцип роботи функції

Функція `search_student_name()` дозволяє шукати студента за прізвищем у вибраному файлі та виводити його дані. Спочатку викликається функція `derectory_reader()`, яка дозволяє вибрати файл для пошуку. Користувач вводить прізвище студента, якого хоче знайти. Потім файл відкривається для читання, і кожен рядок файлу перевіряється на наявність введеного прізвища. Перший рядок, що містить заголовки колонок, пропускається. Якщо в одному з рядків знайдено прізвище, виводяться дані цього студента, зокрема його прізвище, група та середній бал. Також функція повертає словник, який містить шлях до файлу, номер рядка студента, його ім'я, групу та середній бал. Якщо студент не знайдений, функція просто завершить виконання без виведення будь-яких повідомлень.

Тестові запуски

```
Оберіть функцію: 4
Оберіть файл для роботи:
Список доступних файлів:
1. CB-24.csv
2. CI-24.csv
3. CS-24.csv
4. new.csv

Введіть номер файлу для вибору: 3
Вибрано файл: CS-24.csv
Введіть прізвище студента: Мироненко
Студент Мироненко Ярослав з групи КН-24 має середній бал 98
```

Рисунок 7

Блок-Схема



Рисунок 8

Лістинг функції 6

```
def search_student_num(): # 6)Пошук студента за номером
    """Функція для пошуку студента за номером в файлі обраному функцією
derectory_reader()
та вивидення його даних"""

    selected_path = derectory_reader() # вибір файлу для пошуку

    search = int(input("Введіть номер студента: "))
    with open(selected_path, 'r', encoding='utf-8', newline='') as csvfile: #
відкриття файлу для читання
        reader = csv.reader(csvfile) # читання файлу
        for student_num, row in enumerate(reader): # перебір рядків файлу
            if student_num == 0: # пропуск заголовка
                continue
            if search == student_num: # пошук студента за номером
                print(f"Студент {row[0]} з групи {row[1]} має середній бал {row[2]}")
# вивід даних студента
        return { # повернення даних студента
            'path': selected_path,
            'student_num': student_num,
            'name': row[0],
            'group': row[1],
            'average': row[2]
        }
```

Принцип роботи функції

Функція `search_student_num()` шукає студента за номером у вибраному файлі та виводить його дані. Спочатку викликається функція `derectory_reader()`, яка дозволяє вибрати файл для пошуку. Користувач вводить номер студента, якого хоче знайти. Потім файл відкривається для читання, і кожен рядок файлу перевіряється на відповідність введеному номеру. Перший рядок із заголовками колонок пропускається. Якщо знайдено рядок, номер якого збігається з введеним числом, то виводяться дані цього студента: його прізвище, група та середній бал. Також функція повертає словник, що містить шлях до файлу, номер рядка студента, його ім'я, групу та середній бал. Якщо студент із таким номером не знайдений, функція завершує виконання без виведення повідомлень.

Тестові запуски

```
Оберіть функцію: 5
Оберіть файл для роботи:
Список доступних файлів:
1. CB-24.csv
2. CI-24.csv
3. CS-24.csv
4. new.csv

Введіть номер файлу для вибору: 3
Вибрано файл: CS-24.csv
Введіть номер студента: 20
Студент Мельник Дмитро з групи КН-24 має середній бал 89.7
```

Рисунок 9



Рисунок 10

Лістинг функції 7

```

def editing_student(): # 7) Редагування студента
    """Функція для редагування всіх даних про студента
    з можливістю вибору по номеру та прізвищу"""

    print("\nВиберіть спосіб пошуку студента:")

    menu = { # меню вибору способу пошуку
        1: "Пошук за номером",
        2: "Пошук за прізвищем"
    }

    for key, value in menu.items(): # вивід меню
        print(f"{key}: {value}")

    choice = int(input("Оберіть спосіб пошуку \nта введіть відповідний номер: "))

    while True:
        if choice == 1:
            student = search_student_num() # вибір студента за номером
            break
        elif choice == 2:
            student = search_student_name() # вибір студента за прізвищем
            break
        else:
            print("Виберіть відповідний номер")
  
```

```

        choice = int(input("Оберіть спосіб пошуку \nта введіть відповідний номер:
"))

        continue

# Перевірка чи знайдено студента
if not student:
    print("Не вдалося знайти студента")
    return None

current_student = { # дані студента
    'name': student['name'], #
    'group': student['group'],
    'average': student['average']
}
print("\nВибраний студент:")
print(f"Прізвище та ім'я: {current_student['name']}") # вивід даних студента
print(f"Група: {current_student['group']}")
print(f"Середній бал: {current_student['average']}")

print("\nРедагування даних студента:")
new_name = input(f"Прізвище та ім'я ({current_student['name']}): ") or
current_student['name'] # введення нових даних
new_group = input(f"Група ({current_student['group']}): ") or
current_student['group']
new_score = input(f"Середній бал ({current_student['average']}): ") or
current_student['average']

new_student = [new_name, new_group, new_score] # нові дані студента

print("\nНові дані студента:") # вивід нових даних
print(f"Прізвище та ім'я: {new_name}")
print(f"Група: {new_group}")
print(f"Середній бал: {new_score}")

confirm = input("Зберегти зміни? (Enter - Так, n - Ні): ") # підтвердження
збереження змін
if confirm == '':
    with open(student['path'], 'r', encoding='utf-8', newline='') as csvfile: #
відкриття файлу для читання
        reader = csv.reader(csvfile) # читання файлу
        rows = list(reader) # перетворення файлу в список
        rows[student['student_num']] = new_student # заміна даних студента на
нові

```

```
with open(student['path'], 'w', encoding='utf-8', newline='') as csvfile: #  
відкриття файлу для запису  
    writer = csv.writer(csvfile) # запис в файл  
    writer.writerows(rows) # запис рядків у файл  
    print("Зміни збережено")  
else:  
    print("Зміни не збережено")
```

Принцип роботи функції

Функція `editing_student()` дозволяє редагувати всі дані про студента, надаючи можливість вибрати його за номером або прізвищем. Спочатку користувач обирає спосіб пошуку: за номером або прізвищем. Якщо вибрано пошук за номером, викликається функція `search_student_num()`, якщо за прізвищем – `search_student_name()`. Далі перевіряється, чи знайдено студента. Якщо ні, виводиться повідомлення і функція завершує виконання.

Якщо студент знайдений, його поточні дані виводяться на екран, після чого користувач може змінити їх, ввівши нові значення. Якщо користувач залишає поле порожнім, залишаються старі значення. Після цього відображаються оновлені дані студента, і користувач підтверджує збереження змін. Якщо зміни підтверджені, файл відкривається для читання, його вміст записується у список, у ньому замінюються дані вибраного студента, а потім файл переписується з оновленими даними. Якщо користувач не підтвердив зміни, функція завершується без збереження.

Тестові запуски

```
Оберіть функцію: 6

Виберіть спосіб пошуку студента:
1: Пошук за номером
2: Пошук за прізвищем
Оберіть спосіб пошуку
та введіть відповідний номер: 2
Оберіть файл для роботи:
Список доступних файлів:
1. CB-24.csv
2. CI-24.csv
3. CS-24.csv
4. new.csv

Введіть номер файлу для вибору: 3
Вибрано файл: CS-24.csv
Введіть прізвище студента: Мельник
Студент Мельник Дмитро з групи КН-24 має середній бал 89.7

Вибраний студент:
Прізвище та ім'я: Мельник Дмитро
Група: КН-24
Середній бал: 89.7

Редагування даних студента:
Прізвище та ім'я (Мельник Дмитро): Мельник Дмитро
Група (КН-24): КН-24
Середній бал (89.7): 98

Нові дані студента:
Прізвище та ім'я: Мельник Дмитро
Група: КН-24
Середній бал: 98
Зберегти зміни? (Enter - Так, n - Ні):
Зміни збережено
```

Рисунок 11

Блок-Схема

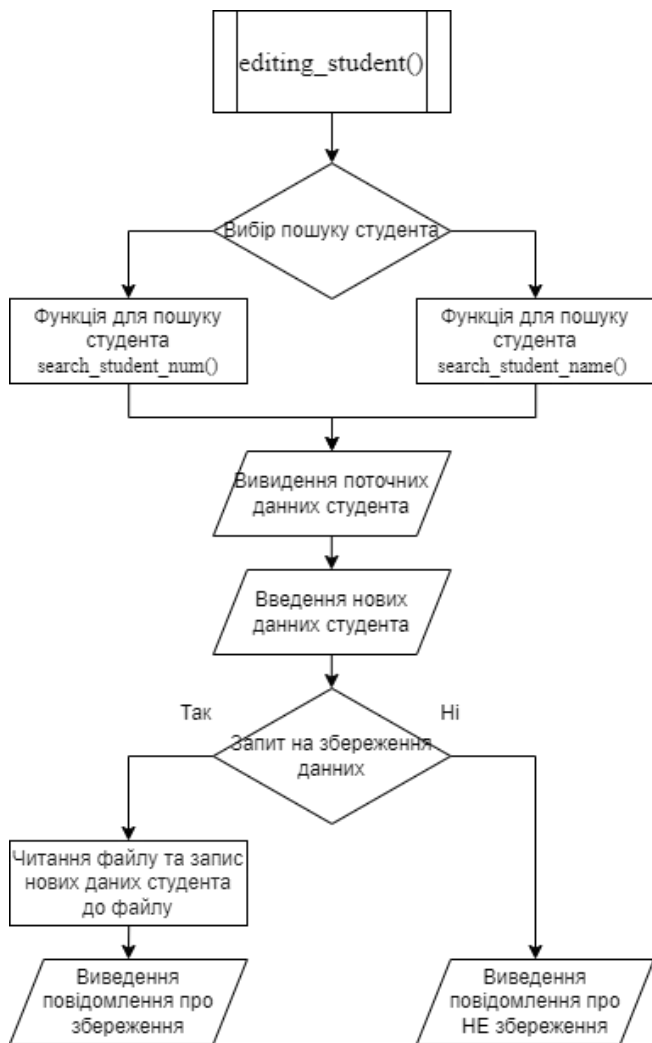


Рисунок 12

Лістинг функції 8

```
def sort_average(): # 8)Сортування студентів за середнім балом
    """функція для сортування студентів за середнім балом по зростанню та спаданню в
    файлі обраному функцією derectory_reader() """

    selected_path = derectory_reader() # вибір файлу для сортування

    menu = {
        1: "Сортування за середнім балом по зростанню",
        2: "Сортування за середнім балом по спаданню"
    }

    for key, value in menu.items(): # вивід меню
```

```

print(f"{key}: {value}")

choice = int(input("Оберіть спосіб сортування \n1 - за зростанням, 2 - за спаданням середнього балу: "))

if choice == 1:
    with open(selected_path, 'r', encoding='utf-8', newline='') as csvfile: # відкриття файлу для читання
        reader = csv.reader(csvfile) # читання файлу
        rows = list(reader) # перетворення файлу в список
        rows = rows[1:] # пропуск заголовка
        rows.sort(key=lambda x: float(x[2])) # сортування студентів за середнім балом по зростанню

        print("\nСортування студентів за середнім балом по зростанню:")
        print("Прізвище | Ім'я | Група | Середній бал")

        for row in rows: # вивід даних студентів
            print(f"{row[0]} | {row[1]} | {row[2]} | {row[3]}") # вивід даних студентів

elif choice == 2:
    with open(selected_path, 'r', encoding='utf-8', newline='') as csvfile: # відкриття файлу для читання
        reader = csv.reader(csvfile) # читання файлу
        rows = list(reader) # перетворення файлу в список
        rows = rows[1:] # пропуск заголовка
        rows.sort(key=lambda x: float(x[2]), reverse=True) # сортування студентів за середнім балом по спаданню

        print("\nСортування студентів за середнім балом по спаданню:")
        print("Прізвище | Ім'я | Група | Середній бал")

        for row in rows: # вивід даних студентів
            print(f"{row[0]} | {row[1]} | {row[2]} | {row[3]}") # вивід даних студентів
else:
    print("Виберіть відповідний номер")
    choice = int(input("Оберіть спосіб сортування \n1 - за зростанням, 2 - за спаданням середнього балу: "))

```

Принцип роботи функції

Функція `sort_average()` виконує сортування студентів за середнім балом у файлі, який користувач обирає за допомогою `directory_reader()`. Після вибору файлу користувач обирає спосіб сортування: за зростанням або за спаданням середнього балу.

Якщо вибрано сортування за зростанням, відкривається файл, його вміст зчитується у список, заголовок пропускається, а решта даних сортується за середнім балом у порядку зростання. Потім виводиться заголовок таблиці та відсортовані дані студентів.

Якщо вибрано сортування за спаданням, процес аналогічний, але сортування проводиться у зворотному порядку.

Якщо користувач вводить некоректний номер, його просять повторити вибір.

Тестові запуски

```
Оберіть функцію: 7
Оберіть файл для роботи:
Список доступних файлів:
1. CB-24.csv
2. CI-24.csv
3. CS-24.csv
4. new.csv

Введіть номер файлу для вибору: 1
Вибрано файл: CB-24.csv
1: Сортування за середнім балом по зростанню
2: Сортування за середнім балом по спаданню
Оберіть спосіб сортування
та введіть відповідний номер: 1

Сортування студентів за середнім балом по зростанню:
Прізвище Ім'я | Група | Середній бал
Скріншотов Дмитро | КБ-24 | 70.8
Сітонет Юрій | КБ-24 | 72.8
Лаговська Софія | КБ-24 | 74.2
Баганов Михайло | КБ-24 | 76.9
Фреймвордов Роман | КБ-24 | 78.6
Гігавольт Марина | КБ-24 | 79.3
Апдейтова Вероніка | КБ-24 | 80.2
Піксельчук Данило | КБ-24 | 81.4
Клавішенко Артем | КБ-24 | 82.5
Алгоритмов Тарас | КБ-24 | 84.7
Коденко Олександра | КБ-24 | 85.6
Хакеренко Катерина | КБ-24 | 86.3
Байтов Артем | КБ-24 | 88.5
Джаваскриптова Ірина | КБ-24 | 88.9
Чіпізатор Олеся | КБ-24 | 89.0
Максименко Ігор | КБ-24 | 90.1
Компілов Стас | КБ-24 | 91.3
Деплойко Владислав | КБ-24 | 92.7
Програменко Владислав | КБ-24 | 95.2
Гігабайтович Євгеній | КБ-24 | 97.1
```

Рисунок 13

Блок-Схема

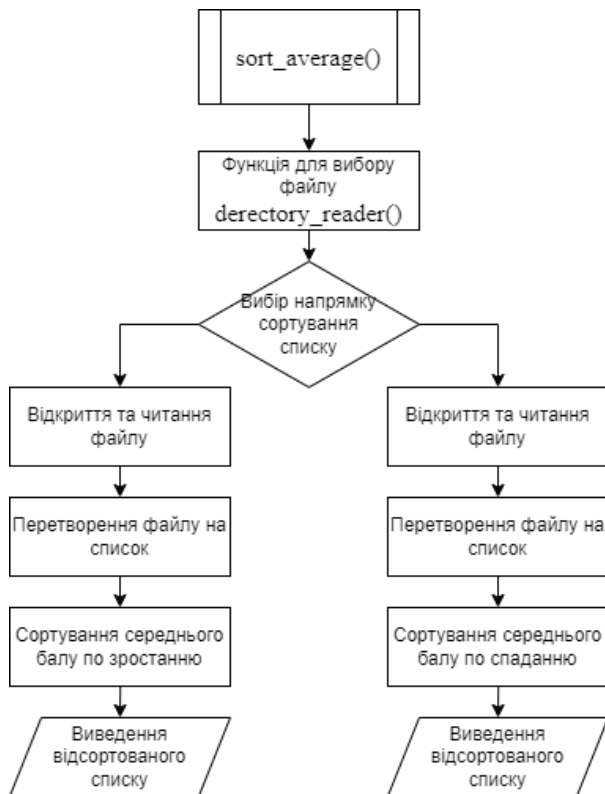


Рисунок 14

Лістинг функції 9

```
def remove_student():  
    """Функція для видалення студента з файлу за номером  
    з попереднім відображенням списку студентів"""  
  
    # Отримуємо шлях до файлу  
    selected_path = derectory_reader()  
    if not selected_path:  
        print("Спочатку виберіть файл.")  
        return None  
  
    try:  
        # Читаємо дані прямо тут, без виклику read_file()  
        with open(selected_path, 'r', encoding='utf-8', newline='') as csvfile:  
            reader = csv.reader(csvfile)  
            rows = list(reader)    # Зчитуємо всі рядки  
  
            if len(rows) <= 1:  
                print("У файлі немає студентів для видалення.")  
                return None
```

```

# Виводимо список студентів
print("\nСписок студентів для видалення:")
print(f"{rows[0][0]} {rows[0][1]} {rows[0][2]}") # Заголовок
print("-" * 40)

for i in range(1, len(rows)):
    print(f"{i}. {rows[i][0]} | {rows[i][1]} | {rows[i][2]}")

# Запитуємо номер студента для видалення
student_num = int(input("\nВведіть номер студента для видалення: "))

# Перевіряємо валідність номера (номери починаються з 1)
if student_num < 1 or student_num >= len(rows):
    print(f"Помилка: Студента з номером {student_num} не існує.")
    return None

# Отримуємо дані студента для підтвердження
student_data = rows[student_num]
print(f"\nВибраний студент для видалення:")
print(f"Прізвище та ім'я: {student_data[0]}")
print(f"Група: {student_data[1]}")
print(f"Середній бал: {student_data[2]}")

# Запитуємо підтвердження
confirm = input("\nВи впевнені, що хочете видалити цього студента? (Enter
- Так, n - Ні): ")
if confirm != '':
    print("Видалення скасовано.")
    return None

# Видаляємо студента (student_num вже відповідає індексу в rows)
deleted_student = rows.pop(student_num)

# Записуємо оновлений список назад у файл
with open(selected_path, 'w', encoding='utf-8', newline='') as csvfile:
    writer = csv.writer(csvfile)
    writer.writerows(rows)

print(f"\nСтудент {deleted_student[0]} успішно видалений.")

except ValueError:
    print("Помилка: Введіть правильний номер студента.")
except IndexError:
    print("Помилка: Неправильний індекс студента.")
except Exception as e:

```

```
print(f"Помилка при видаленні студента: {e}")

return None
```

Принцип роботи функції

Ця функція `remove_student()` дозволяє видаляти студента з файлу, спочатку відображаючи список усіх студентів для зручності вибору. Вона перевіряє, чи вибрано файл, а також чи є в ньому студенти. Далі виводить список студентів з їхніми номерами, після чого користувач вводить номер студента для видалення. Функція перевіряє коректність введеного номера та запитує підтвердження перед видаленням. Якщо користувач підтвердив, студент видаляється, і оновлений список зберігається у файл. Реалізовано перевірку на помилки, щоб уникнути неправильного введення або технічних збоїв.

Тестові запуски

```
Оберіть файл для роботи:
Список доступних файлів:
1. CB-24.csv
2. CI-24.csv
3. CS-24.csv
4. new.csv

Введіть номер файлу для вибору: 1
Вибрано файл: CB-24.csv

Список студентів для видалення:
Прізвище Ім'я Група Середній бал
-----
1. Байтов Артем | КБ-24 | 88.5
2. Гігавольт Марина | КБ-24 | 79.3
3. Деллойд Владислав | КБ-24 | 92.7
4. Коденко Олександра | КБ-24 | 85.6
5. Лаговська Софія | КБ-24 | 74.2
6. Максименко Ігор | КБ-24 | 90.1
7. Піксельчук Данило | КБ-24 | 81.4
8. Програменко Владислав | КБ-24 | 95.2
9. Сітонет Юрій | КБ-24 | 72.8
10. Хакеренко Катерина | КБ-24 | 86.3
11. Чипізатор Олесь | КБ-24 | 89.0
12. Алгоритмов Тарас | КБ-24 | 84.7
13. Баганов Михайло | КБ-24 | 76.9
14. Компілов Стас | КБ-24 | 91.3
15. Андейтова Вероніка | КБ-24 | 88.2
16. Фреймвордов Роман | КБ-24 | 78.6
17. Гігабайтович Євгеній | КБ-24 | 97.1
18. Клавішенко Артем | КБ-24 | 82.5
19. Скріншотов Дмитро | КБ-24 | 70.8
20. Джаваскриптова Ірина | КБ-24 | 88.9

Введіть номер студента для видалення: 9

Вибраний студент для видалення:
Прізвище та ім'я: Сітонет Юрій
Група: КБ-24
Середній бал: 72.8

Ви впевнені, що хочете видалити цього студента? (Enter - Так, n - Ні):

Студент Сітонет Юрій успішно видалений.
```

Рисунок 15

Лістинг main файлу

```
from func_modul import *

def main():
    """Основна функція яка викликає інші функції з модуля\n
    func_modul та виводить результат їх роботи"""

    main_menu = { # Список головного меню
        1: "Створити новий файл .csv",
        2: "Додати студента до списку",
        3: "Вивести вміст файлу",
        4: "Знайти студента за прізвищем",
        5: "Знайти студента за номером у списку",
        6: "Відредагувати данні про студента",
        7: "Сортувати список студентів",
        8: "Видалити студента зі списку",
        9: "Вихід"
    }

    while True:
        print("\nВас вітає програма для роботи з файлами .csv\n"
              "В цій програмі ви можете обрати одну з наступних дій\n"
              "які в свою чергу виконають обрану функцію\n")

        for key, value in main_menu.items():
            print(f"{key}: {value}")

        menu = (input("\nОберіть функцію: "))

        # Валідація введення
        if not menu.isdigit(): # перевірка чи введене число
            print("Номер повинен бути числом від 1 до 10"
                  " Продовжити роботу з програмою? (Введіть n для завершення або Enter\n"
                  "для продовження)")
            if input() == "n":
                print("До побачення!")
                break
            else:
                continue

        menu = int(menu) # перетворення введеного значення в число

        # Валідація введення
        if menu not in main_menu:
            print("Не вірний номер функції виберіть від 1 до 8")
```

```

        " Продовжити роботу з програмою? (Введіть n для завершення або
Enter для продовження) ")
        if input() == "n":
            print("До побачення!")
            break
        else:
            continue

# Виклики функцій
if menu == 1:
    new_file()

# Запит на продовження роботи з програмою
    print("Продовжити роботу з програмою? (Введіть n для завершення або Enter
для продовження) ")
    if input() == "n":
        print("До побачення!")
        break
    else:
        continue

elif menu == 2:
    add_student()

# Запит на продовження роботи з програмою
    print("Продовжити роботу з програмою? (Введіть n для завершення або Enter
для продовження) ")
    if input() == "n":
        print("До побачення!")
        break
    else:
        continue

elif menu == 3:
    read_file()

# Запит на продовження роботи з програмою
    print("Продовжити роботу з програмою? (Введіть n для завершення або Enter
для продовження) ")
    if input() == "n":
        print("До побачення!")
        break
    else:
        continue

```



```
elif menu == 4:
    search_student_name()

    # Запит на продовження роботи з програмою
    print("Продовжити роботу з програмою? (Введіть n для завершення або Enter для продовження)")
    if input() == "n":
        print("До побачення!")
        break
    else:
        continue

elif menu == 5:
    search_student_num()

    # Запит на продовження роботи з програмою
    print("Продовжити роботу з програмою? (Введіть n для завершення або Enter для продовження)")
    if input() == "n":
        print("До побачення!")
        break
    else:
        continue

elif menu == 6:
    editing_student()

    # Запит на продовження роботи з програмою
    print("Продовжити роботу з програмою? (Введіть n для завершення або Enter для продовження)")
    if input() == "n":
        print("До побачення!")
        break
    else:
        continue

elif menu == 7:
    sort_average()

    # Запит на продовження роботи з програмою
    print("Продовжити роботу з програмою? (Введіть n для завершення або Enter для продовження)")
    if input() == "n":
        print("До побачення!")
        break
```

```

        else:
            continue

    elif menu == 8:
        remove_student()

    # Запит на продовження роботи з програмою
    print("Продовжити роботу з програмою? (Введіть n для завершення або Enter для продовження)")

    if input() == "n":
        print("До побачення!")
        break
    else:
        continue

elif menu == 9:
    print("Вихід з програми")
    break

break

if __name__ == "__main__": # Перевірка чи файл був запущений напряму
    main() # Виклик основної функції

```

Контрольні питання

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

Запис даних у файл здійснюється за допомогою функції `open()` у режимі запису ('w', 'a' або 'x'). При використанні 'w' файл створюється або перезаписується, 'a' додає дані до існуючого файлу, а 'x' створює новий файл, але видає помилку, якщо файл уже існує.

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

Читання даних із файлу здійснюється за допомогою `open()` у режимі читання ('r'). Файл можна прочитати повністю або построково.

3. Як здійснюється копіювання файлу?

Копіювання файлу можна здійснити за допомогою модуля `shutil`, який містить функцію `shutil.copy()`.

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

Функція `open()` використовується для відкриття файлу і має такий синтаксис

```
open(file, mode='r', encoding=None, newline=None,  
closefd=True, opener=None)
```

5. Які бувають режими роботи з файлами?

У Python існують такі режими роботи з файлами:

'r' (read) – читання файлу (за замовчуванням)

'w' (write) – запис у файл, перезаписує вміст

'a' (append) – додавання до файлу

'x' (exclusive) – створює новий файл, якщо він уже існує – викликає помилку

'b' (binary) – відкриває файл у бінарному режимі

't' (text) – відкриває файл у текстовому режимі (за замовчуванням)

'r+' – читання і запис у файл

'w+' – читання і запис, але файл перезаписується

'a+' – читання і додавання