

Національний Університет Біоресурсів і Природокористування України
Факультет інформаційних технологій

Програмування Python

Лабораторна робота №5

Виконав

Студент групи ІПЗ-23008бск

Постумент Денис Андрійович

Київ 2025

Тема: Робота з файлами.

Мета роботи:

1. Створити віртуальне оточення (ім'я оточення - прізвище студента). В цьому оточенні створити проект Python, що складається з трьох програм.

2. **Програма 1 (Правильне сортування).** В коді програми задається список (кількість елементів списку - не менше 10) текстових рядків (слів або фраз).

2.1. Обовязкові елементи списку:

- Англійські слова (або фрази), що починаються з маленьких літер;
- Англійські слова (або фрази), що починаються з Великих літер;
- Українські слова (або фрази), що починаються з маленьких літер;
- Українські слова (або фрази), що починаються з Великих літер;
- Українські слова (або фрази), що починаються з маленьких літер "і","ї","е";
- Українські слова (або фрази), що починаються з великих літер "І","Ї","Є".

2.2. Створити функцію, або метод сортування, який сортує за наступними правилами: Українські літери, незалежно від регістру, латинські літери, незалежно від регістру.

2.3. Робота програми: Програма виводить заданий список на екран. Потім відсортований список.

Приклад роботи програми:

Заданий список:

['English', 'інформація', 'android', 'Windows', 'Добрий день', 'матриця', 'актова зала', 'бюоресурси', 'єдиний', 'кава']

Відсортований список:

['актова зала', 'бюоресурси', 'Добрий день', 'єдиний', 'інформація', 'кава', 'матриця', 'android', 'English', 'Windows',]

3. **Програма 2 (запис в json кириличних літер).** В коді програми задається словник (кількість записів - не менше 10) наступної структури: {"Прізвище": ["Ім'я", "По батькові", "рік народження"]}. Всі записи українською мовою.

Програма повинна записати створений словник в json файл. Вимоги до файлу: кодування: UTF-8, всі літери повинні бути записані як літери, а не коди. Тобто якщо відкрити цей файл в текстовому редакторі, то можна бачити всі слова українською мовою, які в нього записані.

4. **Програма 3 (читання із json кириличних літер).** Програма повинна прочитати json файл, створений в попередньому пункті (кодування: UTF-8, ПІБ та рік народження українською) і вивести в консоль всі записи в коректному кодуванні.
5. Створити файл requirements.txt, в якому зберігаються всі встановлені модулі та пакети.
6. Створити звіт, що відповідає структурі наведеної нижче.
7. Завантажити проект на сервіс GitHub.

Хід роботи

Код програми 1 (сортування):

```
def custom_sort_key(s):  
    s_lower = s.lower()  
    if not s_lower:  
        return (2, "")  
  
    first_char = s_lower[0]  
  
    if 'а' <= first_char <= 'я' or first_char in 'їїер':  
        group = 0  
    elif 'а' <= first_char <= 'з':
```

```

group = 1

else:

    group = 2


return (group, s_lower)

initial_list = [
    'English', 'інформація', 'android', 'Windows',
    'Добрий день', 'матриця', 'актова зала', 'біоресурси',
    'єдиний', 'кава', 'Історія', 'Їжак', 'apple'
]

print("Заданий список:")
print(initial_list)

sorted_list = sorted(initial_list, key=custom_sort_key)

print("\nВідсортований список:")
print(sorted_list)

```

Результат виконання програми 1 (сортування):

```
(Postument) PS D:\Postument python\lr5> python sort.py
Заданий список:
['English', 'інформація', 'android', 'Windows', 'добрий день', 'матриця', 'актова зала', 'біоресурси', 'єдиний', 'кава', 'Історія', 'Їжак', 'apple']

Відсортований список:
['актова зала', 'біоресурси', 'добрий день', 'кава', 'матриця', 'єдиний', 'інформація', 'Історія', 'Їжак', 'android', 'apple', 'English', 'Windows']
(Postument) PS D:\Postument python\lr5> 
```

Рис. 1 - Результат сортування

Код програми 2 (запис в json):

```
import json
```

```
student_data = {  
    "Шевченко": ["Тарас", "Григорович", 1814],  
    "Франко": ["Іван", "Якович", 1856],  
    "Косач-Квітка": ["Лариса", "Петрівна", 1871],  
    "Скворода": ["Григорій", "Савич", 1722],  
    "Хмельницький": ["Богдан", "Михайлович", 1595],  
    "Корольов": ["Сергій", "Павлович", 1907],  
    "Довженко": ["Олександр", "Петрович", 1894],  
    "Вернадський": ["Володимир", "Іванович", 1863],  
    "Амосов": ["Микола", "Михайлович", 1913],  
    "Ступка": ["Богдан", "Сильвестрович", 1941]  
}
```

```
file_name = 'people.json'
```

```
with open(file_name, 'w', encoding='utf-8') as f:  
    json.dump(student_data, f, ensure_ascii=False, indent=4)
```

```

print(f"Дані успішно записано у файл '{file_name}'")
print(f"Перевірте файл '{file_name}' у вашому текстовому редакторі.")

```

Результат виконання програми 2 (запис в json):

```

(Postument) PS D:\Postument python\lr5> python write_json.py
Дані успішно записано у файл 'people.json'
Перевірте файл 'people.json' у вашому текстовому редакторі.
(Postument) PS D:\Postument python\lr5>

```

Рис. 2 - Запис даних у JSON файл

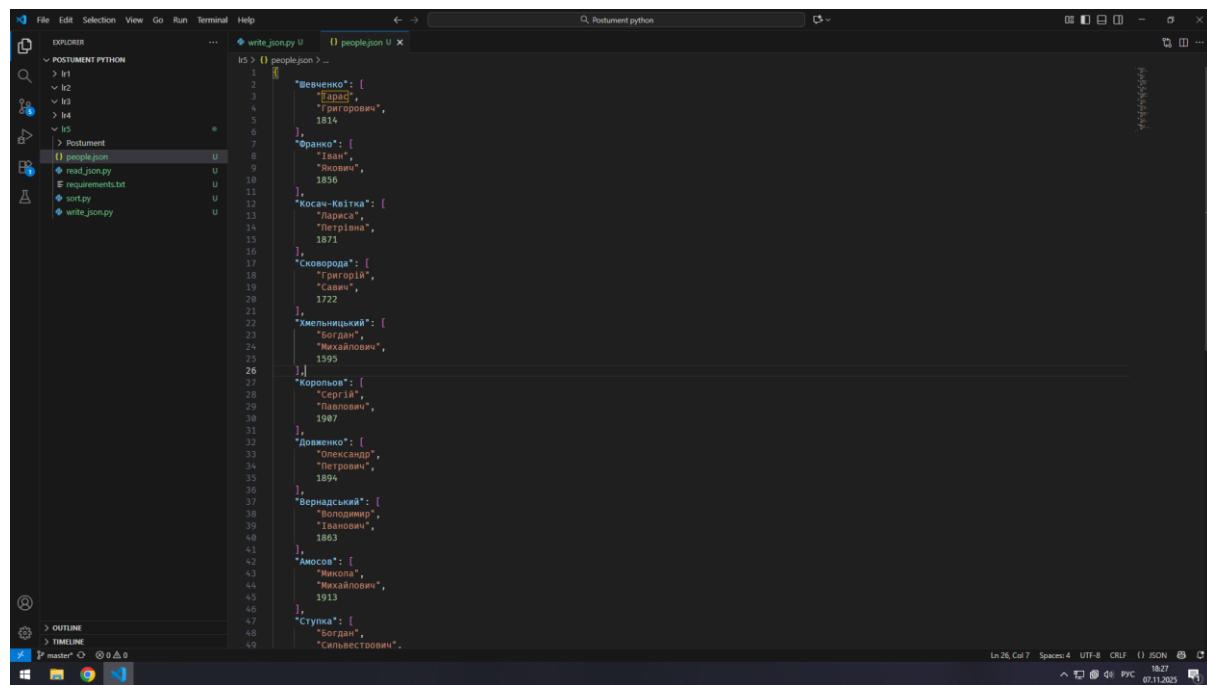


Рис. 3 – JSON файл

Код програми 3 (читання json):

```
import json
```

```
file_name = 'people.json'
```

```
try:
```

```
    with open(file_name, 'r', encoding='utf-8') as f:
```

```
        data = json.load(f)
```

```
    print(f"Дані, успішно прочитані з файлу '{file_name}':\n")
```

```
for surname, info in data.items():
```

```
    print(f"Прізвище: {surname}")
```

```
    print(f" Ім'я: {info[0]}")
```

```
    print(f" По батькові: {info[1]}")
```

```
    print(f" Рік народження: {info[2]}")
```

```
    print("-" * 25)
```

```
except FileNotFoundError:
```

```
    print(f"Помилка: Файл '{file_name}' не знайдено.")
```

```
    print("Будь ласка, спочатку запустіть 'program2_write_json.py'.")
```

```
except json.JSONDecodeError:
```

```
    print(f"Помилка: Не вдалося прочитати JSON з файлу '{file_name}'.")
```

Результат виконання програми 3 (читання json):

The screenshot shows a Python development environment with several files in the Explorer pane: lr1, lr2, lr3, lr4, lr5, Postument, people.json, read_json.py, requirements.txt, sort.py, and write_json.py. The Terminal pane displays the command `python read_json.py` and its output, which reads data from `people.json` and prints it to the console. The Data pane shows the JSON data being processed.

```
(Postument) PS D:\Postument\python\lr5> python read_json.py
[{"Прізвище": "Вєнчанко", "Ім'я": "Тарас", "По батькові": "Григорович", "Рік народження": 1814}, {"Прізвище": "Франко", "Ім'я": "Іван", "По батькові": "Якович", "Рік народження": 1856}, {"Прізвище": "Косач-Квітка", "Ім'я": "Лавр", "По батькові": "Петрівна", "Рік народження": 1871}, {"Прізвище": "Скворода", "Ім'я": "Григорій", "По батькові": "Савич", "Рік народження": 1722}, {"Прізвище": "Хмельницький", "Ім'я": "Богдан", "По батькові": "Михайлович", "Рік народження": 1595}, {"Прізвище": "Корольов", "Ім'я": "Сергій", "По батькові": "Валентин", "Рік народження": 1987}, {"Прізвище": "Довженко", "Ім'я": "Олеся", "По батькові": "Петрович", "Рік народження": 1894}, {"Прізвище": "Борисівський", "Ім'я": "Володимир", "По батькові": "Іванович", "Рік народження": 1863}, {"Прізвище": "Амосов", "Ім'я": "Микола", "По батькові": "Михайлович", "Рік народження": 1913}, {"Прізвище": "Ступка"}]
```

Рис. 4 - Читання JSON файлу

The screenshot shows a terminal window with the command `python --version` and its output, indicating Python 3.13.7.

```
(Postument) PS D:\Postument\python\lr5> python --version
Python 3.13.7
(Postument) PS D:\Postument\python\lr5>
```

Рис. 5 - Версія Python у віртуальному оточенні

The screenshot shows a terminal window with the command `pip list` and its output, which shows the package pip with version 25.2.

```
(Postument) PS D:\Postument\python\lr5> pip list
Package Version
-----
pip    25.2
(Postument) PS D:\Postument\python\lr5>
```

Рис. 6 - Список завантажених пакетів (нічого не завантажувалось)

Посилання на гітхаб: <https://github.com/DenysSheppard/python-labs/tree/master/lr5>

Висновки

Під час виконання цієї лабораторної роботи я успішно закріпив навички роботи з файлами в Python. Я навчився реалізовувати складні алгоритми сортування за допомогою ключових функцій, а також правильно зберігати та читати дані у форматі JSON, забезпечуючи коректне кодування кириличних символів.