Assignment 1. Working with directories and files in Python

(upload tomoodle and protect until 19.09.2025)

If Assignment is loaded and protected late,
to -10%of each task,
that is, if everything is completed, the max score is 90 points.

Task 1: (25b) Working with folders and files

1. Creating a project structure

- Write a script that creates a project directory with subfolders:
  - data
  - results

2. Working with CSV files

- In the data folder, save the students.csv file with data about students (full name, group, grades).
- Read data from students. csv and calculate the average score for each student.

3. Save results

- Save aggregated results in JSON format (results/report.json).
- Check: if the file already exists, display a warning and ask for confirmation for overwriting.

4. Archiving

- Automatically create an archive results.zip, where the results folder is saved.

5. Working with pathlib

- Implement a check to see if report.json exists. If yes, print its size and last modification date.

Task 2: (25b) Search for a word in a text file

Generate 5 text files (name_.txt_.txt), put them in the same folder. The files contain any text. Write a program that accepts a search query and outputs the names of text files containing the desired substring.

Input format

String containing the search query. It can be either a single word or a phrase combination. For example: mathematics, probability theory.

Output format

A list of text files containing the substring entered by the user. Or a message that the search query is missing in these files.

Task 3: (25b) File Information о файле

There is a file **available** file.txt (the file is attached to the task). Write a program that outputs the following text statistics:

- the number **of letters** of the Latin alphabet.
- number of words.
- number of lines.

Task 4: (25b) Forchanging words

There are two files **words.txt.txt** and **forbidden_words.txt_words.txt** (the files are attached to the task). In the file**words.txt.txt** contains text. In a text file **forbidden_words.txt** forbidden words are stored separated by a space character.

Write a program that replaces in the file**words.txt.txt** all forbidden words from the file **forbidden_words.txt_words.txt** asterisks * (the number of asterisks is equal to the number of letters in the word).

The program must replace all forbidden words wherever they occur, even in the middle of another word. Case-insensitive substitution is performed: if the file **forbidden_words.txt** If it contains the forbidden word **exam**, then the words **exam**, **Exam**, **ExaM**, **EXAM** and **exAm** should be replaced with ****.

Output format

Text edited according to the task condition.

Questions for self-monitoring

1. How can I open a file for reading and writing in Python?
2. What is the difference between the file opening modes 'r', 'w', 'a','rb','wb'?
3. What is the advantage of using the with open(...)construct? as f: when working with files?
4. How to close a file correctly and why is this important?
5. What are the main functions **of the os module** used for working with directories?
6. How can I use Python to create a new folder or delete an existing one?
7. What is the difference between os. getcwd() and os. chdir()?
8. How do I get a list of files and folders in a directory?
9. Для чего используется модуль **What is the os.path module used for**?
10. How do I check if a file or folder exists?
11. How is pathlib more convenient than **pathlib** по сравнению с **os. path**?
12. How do I combine file paths in a cross-platform style?
13. What Python methods can be used to read a CSV file?
14. How **is the csv module useful** compared to the usual line-by-line reading of a file?
15. How do I serialize Python data to **JSON format**?
16. What is the difference between working with JSON and CSV?
17. What Python data types can be directly stored in JSON?
18. What is **ленивая загрузка (lazy loading)** and why is it needed?
19. How can I read a large file line-by-line without loading it completely into memory?

20. What is the use of the readline() method and iterating over a file object?
21. When should I use **generators** when working with files?
22. What Python modules can be used to work with archives (.zip, .tar)?
23. How do I create a ziparchive using Python?
24. How do I unzip the archive to the specified directory?
25. What are the features of working with binary files (for example, .png, .exe)?