Lab08. Interfacing with OS

Script Languages (INZ002025)

Wojciech Thomas

Summer 2020/2021

1 Learning goals

After this lab you should be able to:

1. Use command line arguments to control Python program,
2. Use environment variables,
3. Analyse CSV files,

2 Exercises

Artefacts to be uploaded to ePortal: - file: `app8.py`, - dataset used.

In this lab you will create Python application to read the dataset in CSV format, aggregate data read and finally prepare a report as an Excel file.

Warning: Do not use any external libraries (like *Pandas*) to analyse data. Use only `csv` module and perform all operations on known data structures (lists, dictionaries, tuples).

1. Download dataset in the CSV format from the web page https://www.kaggle.com/datasets. Choose dataset larger than 500KB.

   Use MS Teams / Script Languages (Lect/Thu15) (team) / General (channel) Lab8-dataset (channel tab) / Excel spreadsheet to coordinate your choices. Duplicated URLs will have an orange background.

   If you use the same dataset as other students, your lab grade will be divided by the total number of students who has used the same dataset.

   Put the URL of the dataset in the comment at the beginning of your application.

2. Pass a name of the dataset as an obligatory argument to the app, e.g.

   ```
   $ python app8.py dataset.csv
   ```

   Handle the following situations:

   - If the file name does not end with `.csv` extension stop the program with explanation of the problem.
   - If the dataset file does not exist also stop with the explanation.

3. Create a function that reads the content of the dataset and returns a data structure with the content of the csv file.
    - use a list, a tuple or a class to store a data from each row (your choice)
    - You do not have to read all data from datafile (if you don't need them).
4. Devise some operations on the choosen dataset (at least one for each of the following sections):
    - statistical (e.g. average, sum, median etc.)
    - aggregation (e.g. spending by country, number of tweets by a day of the week, etc.)
    - summary (e.g. number of items, countries, tweets; total spending etc.)
5. Use an optional argument `-o` to provide a name of the Excel file to be generated, e.g.

    ```
    $ python app8.py dataset.csv -o report.xlsx
    ```

    - If the argument `-o` is present, save results of all operations in the Excel file. Use text formatting (eg. font face, color etc.).
    - If the argument `-o` is not present, display only *summary* result on the screen.
6. Add an option `-h` to display help about the purpose of your application and all available options.