Programming and Data Processing
Homework assignment 4.
Files. File system. Data formats

General information

Deadline: November 12th, 23:59:59
Submission: through the ICEF information system as a PyCharm project

Description of tasks

In this assignment you will develop an application that loads data about currency exchange rates from multiple files of different formats and calculates a number of statistical characteristics for each user request.

- Register on quandl.com - an online resource with a rich database of financial information.
- Download a number of files that store **daily** exchange rates for different pairs of currencies. You need to use at least two different file formats.
- Load data into the Python program and form the following structure in memory:

```
{
   "EUR-USD": [
      ("2015-01-03", 0.76),
      ("2015-01-04", 0.75),
      ("2015-01-05", 0.78),
      ...
   ],
   "CZK-HUF": [
    ("2014-03-02", 11.2),
    ("2014-03-03", 11.3),
    ...
   ]
}
```

  The list for each currency pair should be sorted in the ascending order of dates (this can be used to optimize search in the next task)

  The program should make no initial assumption about the number of files available in the data directory.

- Build a request-response workflow: the user enters two currency codes, start and end dates and the program prints four statistical characteristics for the exchange rate in the corresponding period - minimal, maximal, mean ($\overline{x}$) values and the standard deviation (σ). Use the following formulas for mean value and standard deviation:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

Several formats should be supported for both start and end dates:
DD.MM.YYYY, MM.YYYY, YYYY. If a date is entered in the MM.YYYY format (without the day part), the first day of the month should be taken as the start date and the last day as the end date. If a date is entered with only the year part, 1st January should be taken as a start date and 31st December as the end date. Examples of possible period requests are shown below:
01.04.2015 - 10.2016, 2015 - 04.2016, 2014 - 2016

Additional requirements and guidelines

- Split application logic into multiple functions (and possibly modules). Try not to leave large functions (with more than 30 lines of code - extract a piece of code, which solves a narrow task and may be reused and define a new small function)
- Avoid global variables (prefixed with the global keyword). Use function parameters and return values instead
- Look for optimizations, avoid code repetition
- Follow the same naming convention throughout the program
- Test your program for all possible scenarios, especially for the case of the user entering incorrect data

Program flow example

User input is shown in blue:

```
Enter two currency codes separated by a space:  USD EUR
Enter a period:  05.09.2015 - 10.2016
Min=0.8 Max=0.95 Mean=0.87 StdDev=0.03[1])
Enter a period:  05.2015 - 01.2015
Error:  the end date should be larger than the start date
Enter a period:  05.2015 - 01.2016
Min=0.78 Max=0.92 Mean=0.83 StdDev=0.04)
Enter a period:  05.1990 - 05.1991
Error:  no information available for the requested dates
Enter a period:  Enter pressed
Enter two currency codes separated by a space:  USD HUF
Error:  no information available for the requested
currency pair
Enter two currency codes separated by a space:  Enter
pressed
```

[1] Real values will differ