Lab 04 - Classes and objects

Script Languages

Learning goals

1. Create your own class and use objects. Know what is a purpose of constructor.
2. Use basic magic methods. Know how to convert an object into a string.
3. Create a new class using inheritance.
4. Use properties.

Exercises

1. Introduction

1. Download Apache access log from ePortal.
2. Read about classes `IPv4Address` and `IPv4Network`. How can you check if an IPv4 address belongs to IPv4 network using these classes?
3. Read about class `datetime` and functions `strftime` and `strptime`.
4. Read about Data Classes (https://docs.python.org/3/library/dataclasses.html).

2. Defining classes

1. Crate a function that takes as an argument timestamp from the log file (e.g. 18/Oct/2020:01:30:42 +0200) and returns a datetime object.
2. Create your class to store the HTTP request. What attributes should contain this class? Include at least:
    1. constructor (what arguments are required?),
    2. a method to convert an object into a string,
    3. a method to return the request method,
    4. a method to return the requested resource.
3. Create your own class to store log entry. Create at least:
    1. constructor,
    2. a method to convert object into string.

4. Create classes inheriting from the class created in the previous task for storing:
    1. Malformed HTTP request,
    2. Correct HTTP request.
5. Add utility properties using `@property` decorator e.g. `is_successful` or `is_error.`
6. Convert into dataclass a class to store HTTP requests.
7. Place class definitions into separate files and import them into the main file of your application.

3. Enriching classes

1. Add basic magic methods to created classes: `__str__, __repr__, __eq__, __lt__, __gt__, __bool__.` Add `__len__` method if necessary. Demonstrate their purpose in your code.
2. Create a function that:
    1. takes as an argument one line of the log file,
    2. returns a log entry object.
3. Create a function that:
    1. reads the content of the logfile,
    2. returns a list of log entry objects.
4. Create a function that displays all requests between two given moments in time. Both moments should be passed as arguments. If the second argument is earlier than the first one display the error message. The displayed list should be sorted without explicitly specifying sorting key.