Homework 4 OOP

This time the homework itself will be really small:

1. Play with the **sys** and **os** modules. Try passing arguments to a python script, as well as exploring your filesystem. Send me the resulting python programs.

2. Play with the requests library. Learn using API's: e.g. https://swapi.co/

3. Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle

4. Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle

5. Define a class named Bulgarian which has a static method called printNationality(e.g. "Bulgarian").

6. Extend our Employee example we created in class. Add another classsmethod for creating objects from a different format.

HOWEVER, if you are willing to to practice, I give you two bonus project tasks. They are not mandatory!

BONUS TASKS:

These tasks use and build upon what you have learned in the past month. They are designed to also help you learn new things and push yourselves to think like a programmer. Don't worry if you cannot complete them. I would like to see that you've made the effort.

Task 1

The Song List App

You will create a simple application that will allow a user to create a list of favorite songs, play the songs, and view other data about the songs. It'll be like Spotify/iTunes/etc, but worse.

This app will have two classes, Song and List, and a list can contain many songs.

A. A Song should have the following attributes: 1. title - The name of the song. 2. artist - The creator of the song. This will be represented using a string. 3. duration - The length of a song in seconds. This will be represented by a number. 4. lyrics - The lyrics of the song. This will be represented with a string.

One should be able to call the title, artist, duration, and lyrics methods on a particular Song instance to view any of those pieces of information.

B. A Song should also have a play method, which will "play" the song. For now “playing” means just printing the lyrics. And yes, the play method could just be the same as the lyrics one. But you would be fixing this little quirk sometime in the future.

C. A Song should have a method called "friendly_duration" which should return the duration of the Song in minutes and seconds. For example, if this method is called on a Song with a duration of 150 seconds, it should return: "2 minutes, 30 seconds."

D. Songs can be added to a List. Create a method called "add_song" to do so.

E. A List should a have a method called "play", which will play all of the songs in the List.

F. A List should have a method called "shuffle" which will play all of the songs in the List in random order.

G. A List should have a method called "duration" which will return the total number of seconds of the entire list, based on the sum of the durations of all the Songs in that List.

Task 2

The ToDo App

Hey, <your name>, make me a program that like helps keep track of stuff I have to do.
Or whatever.

Hint: Implement a class ToDoList. A user should be able to create instances of the class which represent separate lists. Each of the lists will contain a number of TaskEntries(maybe another class,huh?) and will have methods for adding, updating, removing and searching TaskEntries.
Addittionally: Add a method which sorts the TaskEntries in a ToDoList by date.