Exercices Object oriented programmation - EDPIF

1 Simple example : bibliography

A book is described by its title, author and year of publication (to keep it simple). Write a class `Book` that store those information. Write the `__repr__` method.

A bibliography is mainly a list of books. Write a class `Bibliography` that store the list of book (the class has an attribute which is a list).

At the end we want to be able to use the code as follow ::

```
book1 = Book("A very nice book", "F. Dupont", 2014)
book2 = Book("A very smart book", "A. Einstein", 1923)
book3 = Book("A very stupid comics", "D. Duck", 1937)

bibliography = Bibliography([book1, book2, book3])
```

Now that the data are store in object, there are many methods you can think of :

- Write a method `filter_by_year` to create a new bibliography, keeping books only from a specific year.
- Write a method `to_latex` for a book and a bibliography that returns a string that can be inserted in LaTeX

The string returns by the `to_latex` on the first book should be :

```
\bibitem{Dupont2014}
F. Dupont (2014) \emph{A very nice book}
```

For the bibliography, it should look like :

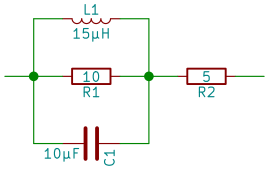
```
\begin{thebibliography}{9}
\bibitem{Dupont2014}
F. Dupont (2014) \emph{A very nice book}

\bibitem{Einstein1923}
A. Einstein (1923) \emph{A very smart book}
\end{thebibliography}
```

2 Pendulum

This is an example of using object for simulation

- Create an object Pendulum that describes a simple pendulum (length, mass)
- Write a method to save the parameters in a json file
- Write a method that returns the period for small oscillations
- Write a method that calculates the trajectory given initial values
- Write a method that plots the trajectory given initial values

3 Complex impedance of a bipolar circuit



The purpose of this problem is to represent a bipolar circuit made with resistor, capacitor and inductor in the computer. The goal is then to calculate it's impedance.

We will use a formal representation of the circuit. A circuit will be a tree whose leaves are the components and nodes the combination of those devices (parallel or serial). A component is defined by it type (resistor, capacitor, …) and the value of the component. In Python, the type is the class and the value and attribute of the object.

The class structure will be the following :

[5]:

```
class BipolarCircuit(object):
    pass

class Combination(BipolarCircuit):
    pass

class Serial(Combination):
    pass

# idem for parallel

class Device(BipolarCircuit):
    pass

class Resistor(Device):
    pass

# idem for capacitor and inductor
```

Write those classes so that we can use it as follow :

```
my_circuit = Serial(Parallel(Resistor(10),Capacitor(1E-5),Inductor(15E-6)),
                    Resistor(5))
```

or using binary operators:

```
my_circuit = (Resistor(10)|Capacitor(1E-5)|Inductor(15E-6))+Resistor(5)
```

where the `|` (`__or__`) represents the parallel combination and the `+` (`__add__`) the serial combination.

The goal is then to implement a method `impedance(omega)` for each kind of circuit. This method is straight forward for resistor, capacitor and inductor. For combination, the method will be recursive.