LP - Exercício em trios

Instruções

- O exercício deve ser feito em trios formados pelos alunos e enviado pelo Eclass até dia 30/09 às 23:59.
- Para a entrega, vocês devem se inscrever nos grupos pelo eclass. Uma vez inscritos nos grupos, será possível visualizar a entrega de atividade.
- Para a implementação em grupo, sintam-se à vontade para utilizar o repl.it ou quaisquer outras plataformas.

Atividade

Vocês devem utilizar o arquivo python em anexo para implementar a classe *FileDict*, filha da classe *dict* do python. A classe deverá implementar um dicionário com funcionalidade de arquivos. Para cada dicionario do tipo *FileDict*, deve haver um arquivo correspondente em que cada linha representam um elemento, sendo que a chave e o valor ficam em uma mesma linha delimitadas por um separador escolhido pelo grupo, podendo ser espaço, traço ou dois pontos, por exemplo.

Há um arquivo "dicio.txt" de exemplo com um dicionário em que adotamos dois pontos como separador (Seu grupo deve adotar um separador para usar em todos os dicionários de forma fixa). A classe deve sobrescrever as funções *__init__*, *__setitem__*, *pop* e *__del__*, acrescentando as funcionalidades descritas abaixo:

- *__init__*: O construtor deve receber o caminho/nome de um arquivo e, possivelmente, um dicionário no formato do construtor da classe dicionário do Python. Se o arquivo em questão já existir, deve-se ler o arquivo e carregar o seu conteúdo em um dicionário padrão chamando a classe *super*(). Caso o arquivo não exista, deve-se criar o arquivo, chamar o construtor da classe mãe e salvar os dados do dicionário (possivelmente vazio) passados como segundo argumento desta função.
- *__setitem__*: Este método deve permitir realizar a seguinte operação em um dicionário - $dicionario['chave'] = valor$ - em que será incluir o valor caso a chave não exista ou atualiza-lo caso já exista. O comportamento espelha

exatamente o do $\_\_setitem\_\_$ da classe dicionário, exceto pelo fato de que deve-se atualizar também o arquivo referente ao $FileDict$ em questão.

- *pop*: Além de realizar a remoção do item com a chamada da classe mãe, deve atualizar o arquivo referente com a exclusão da respectiva linha.

- $\_\_del\_\_$: Deve remover o dicionario, excluindo também o arquivo referente.

Note que todas essas implementações visam ampliar as funções previamente existentes da classe *dict*. Ou seja, as funcionalidades de dicionários Python devem permanecer operantes no objeto $FileDict$.

Ademais, todo o trabalho deve ser documentado com Docstrings (modelo NumPy) e Sphinx. A documentação das funções será avaliada juntamente com a resolução. Veja mais sobre o Sphinx aqui.