Marseille, 05/12/2018 - Achref El Mouelhi

Cas pratique

Énoncée

Considérons une classe `Stagiaire` ayant les attributs suivants :

- `__nom` : un attribut privé de type chaîne de caractères
- `__notes` : un attribut privé de type tableau de float

1. Créez la classe `Stagiaire` et définissez un constructeur avec deux paramètres.
2. Définissez les getters et setters des deux attributs ainsi que la méthode `__str__(self)`.
3. Écrivez la méthode `calculer_moyenne()` qui permet de retourner la moyenne de notes d'un stagiaire.
4. Implémentez l'opérateur `__gt__()` qui permet de comparer deux stagiaires selon leurs moyennes.
5. Écrivez les méthodes `trouver_max()` et `trouver_min()` qui permettent de retourner respectivement les notes max et min d'un stagiaire.

Considérons maintenant une classe `Formation` avec les attributs suivants :

- `__intitulé` : un attribut privé de type `str`
- `__nbr_jours` : un attribut privé de type `int`
- `__stagiaires` : une liste d'objets de type `Stagiaire`

6. Créez la classe `Formation` avec un constructeur à trois paramètres.
7. Définissez les getters et setters de chaque attribut.
8. Modifiez le constructeur et le setter pour lever une exception, de type `FormationError` (à créer), dans le cas où le nombre de jours est négatif.
9. Écrivez une méthode `calculer_moyenne_formation()` qui retourne la moyenne d'un objet de type formation (la moyenne des moyennes des stagiaires, vous pouvez utilisez `map` et `average`).
10. Écrivez une première version de la méthode `trouver_moyenne_superieure(self, min: int)` qui retourne la liste des stagiaires ayant une moyenne supérieure à la valeur passée en paramètre (vous pouvez utilisez `map` et `filter`).
11. Créez un décorateur `@temps_calcul` qui permet, en utilisant le module `time`, de calculer le temps d'exécution d'une méthode en secondes.
12. Écrivez une méthode `trouver_index_max()` qui retourne l'indice du stagiaire dans le tableau `stagiaires` ayant la meilleure moyenne de la formation (vous pouvez utilisez l'opérateur `__gt__()` défini dans la classe `Stagiaire`).
13. Écrivez une méthode `trouver_nom_max()` qui retourne le nom du premier stagiaire ayant la meilleure moyenne d'une formation.
14. Écrivez une méthode `trouver_min_max()` qui retourne la note minimale du premier stagiaire ayant la meilleure moyenne d'une formation.
15. Écrivez une méthode `trouver_moyenne_par_nom(self, nom: str)` qui retourne la moyenne du premier stagiaire dont le nom est passé en paramètre.

16. Créez un indexeur sur stagiaires.
17. Écrivez une méthode `enregistrer_stagiaires()` qui sauvegarde les stagiaires dans un fichier `stagiaires.txt` (un par ligne, un espace pour séparer les différentes valeurs).
18. Rajoutez des commentaires de documentation (**DocString**) aux différentes classes (et générez la documentation).
19. Vérifiez avec **Pylint** que le code respecte les propositions **PEP 8**.
20. Dans un fichier `main.py`, créez une formation et trois stagiaires avec des valeurs saisies par l'utilisateur et testez toutes les méthodes réalisées dans les questions précédentes.