

Exercice 3 : Programmation, POO et Base de Données

Partie A

1.

a a la valeur `[10, 8, 9, 9, 8, 10, 6, 7, 8, 8]`.

b a la valeur `"Fondation"`.

2.

```
def titre_livre(dico, id_livre):
    for i in range(len(dico['id'])):
        if dico['id'][i] == id_livre :
            return dico['titre'][i]
    return None
```

3.

```
def note_maxi(dico):
    m = 0
    for i in range(len(dico['id'])):
        if dico['note'][i] > m:
            m = dico['note'][i]
    return m
```

4.

```
def livres_note(dico, n):
    livres = []
    for i in range(len(dico['id'])):
        if dico['note'][i] == n:
            livres.append(dico['titre'][i])
    return livres
```

5. En utilisant les deux fonctions précédentes :

```
def livre_note_maxi(dico):
    return livres_note(dico, note_maxi(dico))
```

Partie B

6. La classe **Livre** possède 5 attributs (**id**, **titre**, **auteur**, **ann_pub** et **note**) et 4 méthodes en excluant le constructeur (**get_id**, **get_titre**, **get_auteur**, **get_ann_pub**).

7.

```
def get_note(self):
    return self.note
```

8.

```
bi = Bibliotheque() # Instanciation de la bibliothèque
bi.ajout_livre(Livre(8, "Blade Runner", "K.Dick", 1968, 8))
```

9.

Réponse à privilégier étant donné les définitions des accesseurs de la classe **Livre** :

```
def titre_livre(self, id_livre):
    for livre in self.liste_livre :
        if livre.get_id() == id_livre :
            return livre.get_titre()
    return None
```

Étant donné qu'en python il n'y a pas de restriction d'accès aux attributs, une autre réponse acceptable serait :

```
def titre_livre(self, id_livre):
    for livre in self.liste_livre :
        if livre.id == id_livre :
            return livre.titre
    return None
```

Partie C

10. Un auteur peut avoir écrit plusieurs livres et ne peut donc pas permettre d'identifier un livre de manière unique en jouant le rôle de clé primaire.

11.

titre
Ubik
Blade Runner

12.

```
SELECT titre FROM livres WHERE auteur = 'Asimov' AND ann_pub > 1950;
```

13.

```
UPDATE livres SET note = 10 WHERE id = 4 ;
```

14. Séparer les informations dans deux tables permet d'éviter les redondances. Les données sur les auteurs n'apparaissent ainsi qu'une fois au lieu d'être répétées pour chacun de leurs livres.

15. L'attribut **id_auteur** de la table **livres** est une clé étrangère qui fait référence à la clé primaire **id** de la table **auteurs**.

16.

```
SELECT nom, prenom FROM auteurs JOIN livres ON auteurs.id = livres.id_auteur
WHERE ann_pub > 1960 ;
```

17. La requête renvoie les titres des livres pour lesquels les auteurs avaient moins de 30 ans lors de leur publication.

18. La création d'une telle base de données contenant des données personnelles devra faire l'objet d'une déclaration préalable auprès de la Commission Nationale Informatique et Libertés (CNIL). De plus, le recueil de ces données personnelles est soumis au consentement préalable des personnes concernées et à l'exercice de leurs droits d'accès et de rectification d'après le Règlement Général sur la Protection des Données (RGPD) de l'Union Européenne.