TP authentification

Christophe Viroulaud

Terminale - NSI

Lang 07

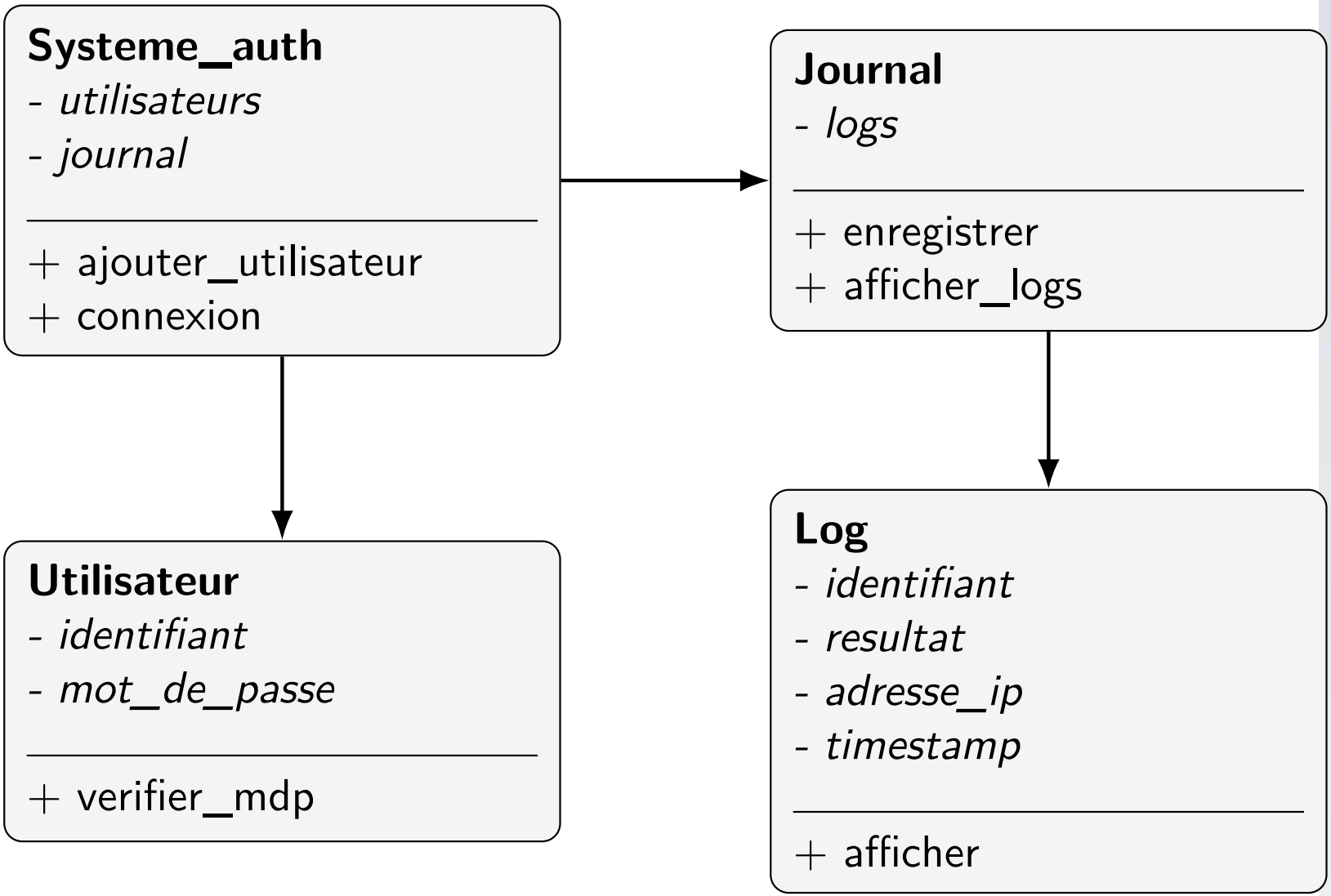

Figure 1 – **Rappel :** Modélisation de l'authentification

Activité 1 :

1. Télécharger et extraire l'annexe.
2. Ouvrir le fichier **`classes.py`**

Remarque

- On retrouve la classe **`Utilisateur`**.
- Les imports permettront de gérer les dates simplement.

Sommaire

Classe `Log`

Activité 2 :

1. Compléter le constructeur de la classe **`Log`**
2. Quel est le rôle de la méthode **`afficher`** ?

Avant de regarder la correction

- Prendre le temps de réfléchir,
- Analyser les messages d'erreur,
- Demander au professeur.

```
1  class Log:
2      def __init__(self, ident: str, res: str, adr: str):
3          self.identifiant = ident
4          self.resultat = res
5          self.adresse_ip = adr
6          self.timestamp = datetime.now().strftime("%d %B
       %Y, %H:%M -")
```

Observation

- La bibliothèque **datetime** permet de gérer les dates.
- La ligne suivante formate la date actuelle

```
1  datetime.now().strftime("%d %B %Y, %H:%M -")
```

- La méthode **afficher** construit une chaîne de caractère à partir de l'évènement de connexion.

Sommaire

Classe `Journal`

Activité 3 : Créer la classe **`Journal`** en s'appuyant sur les signatures suivantes :

- Le constructeur initialisera l'attribut **`logs`** par un tableau vide.
- La méthode **`enregistrer`** acceptera un paramètre de type **`Log`** et ne renverra rien. Elle ajoutera ce log dans le tableau **`logs`**.
- La méthode **`afficher_logs`** n'acceptera aucun paramètre et renverra une chaîne de caractère. Cette chaîne sera composée de plusieurs lignes où chaque ligne représentera les informations d'un log.

Remarque

Le caractère $\backslash n$ est un retour à la ligne.

Avant de regarder la correction

- Prendre le temps de réfléchir,
- Analyser les messages d'erreur,
- Demander au professeur.

```
1 class Journal:
2     def __init__(self):
3         self.logs = []
```

Code 1 – Initialisation

```
1 def enregistrer(self, log: Log) -> None:
2         self.logs.append(log)
```

Observation

Une instance de **Log** est un objet qu'on peut manipuler, stocker, au même titre que tous les objets Python natifs.

```
1  def afficher_logs(self) -> str:
2      res = ""
3      for log in self.logs:
4          res = res + log.afficher() + "\n"
5      return res
```

Observation

Le tableau `logs` contient des objets de type `Log`. On peut donc utiliser les méthodes de ces objets.

Sommaire

Classe `Systeme_auth`

Activité 4 : La classe **`Systeme_auth`** possède deux attributs :

- **`utilisateurs:`** un dictionnaire qui associe l'identifiant de l'utilisateur à son instance,
- **`journal:`** une instance de **`Journal`**

1. Compléter le code de la classe.
2. Prendre le temps de comprendre le code de la méthode **`connexion`**

Avant de regarder la correction

- Prendre le temps de réfléchir,
- Analyser les messages d'erreur,
- Demander au professeur.

```python
class Systeme_auth:
    def __init__(self):
        self.utilisateurs = {}
        self.journal = Journal()

    def ajouter_utilisateur(self, util: Utilisateur) -> None:
        self.utilisateurs[util.get_identifiant()] = util

    def connexion(self, ident: str, mdp: str, ip: str) -> str:
        resultat = "Échec"
        if ident in self.utilisateurs:
            utilisateur = self.utilisateurs[ident]
            # Vérification du mdp
            if utilisateur.verifier_mdp(mdp):
                resultat = "Succès"
        # Création du log de connexion
        log = Log(ident, resultat, ip)
        # Enregistrement dans le journal de log
        self.journal.enregistrer(log)
        return resultat == "Succès"
```

Sommaire

Programme principal

Activité 5 : Écrire le programme principal qui respecte l'algorithme :

- Créer une instance du système d'authentification.
- Ajouter deux utilisateurs :
  - alice, mdp : azerty00
  - bob, mdp : 12345678
- Tenter trois connexions :
  - "alice", "12345678", "192.168.0.1"
  - "bob", "12345678", "10.0.0.42"
  - "eve", "12345678", "172.16.5.5"
- Afficher le journal de connexion

```python
from classes import *

auth = Systeme_auth()
auth.ajouter_utilisateur(Utilisateur("alice", "azerty"))
auth.ajouter_utilisateur(Utilisateur("bob", "12345678"))

auth.connexion("alice", "12345678", "192.168.0.1")
auth.connexion("bob", "12345678", "10.0.0.42")
auth.connexion("eve", "12345678", "172.16.5.5")

print(auth.journal.afficher_logs())
```