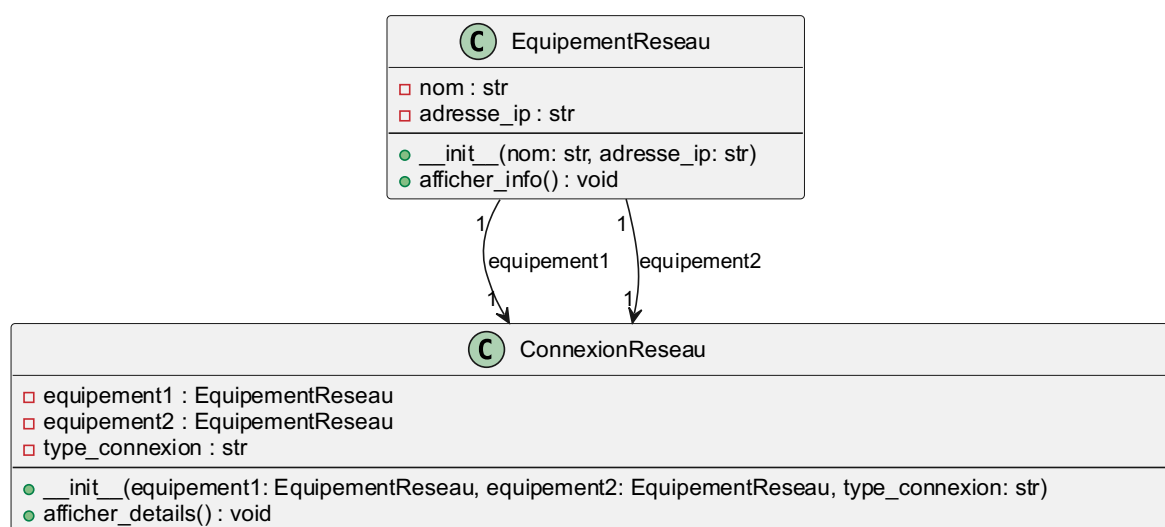


TRAVAUX PRATIQUES
La programmation en Python
<b>La programmation orientée objet</b>

BTS CIEL
2 <sup>ème</sup> année
Page 1 sur 3

## Exercice 1 : Gestion des connexions réseau

Vous devez implémenter deux classes selon le digramme suivant :



Crée deux objets de type `EquipementReseau`.

Crée un objet de type `ConnexionReseau` utilisant les deux équipements.

Affiche les détails de la connexion.

```
# Création des équipements réseau
equipement1 = EquipementReseau("Switch01", "192.168.1.1")
equipement2 = EquipementReseau("Router01", "192.168.1.254")

# Création d'une connexion réseau
connexion = ConnexionReseau(equipement1, equipement2, "Ethernet")

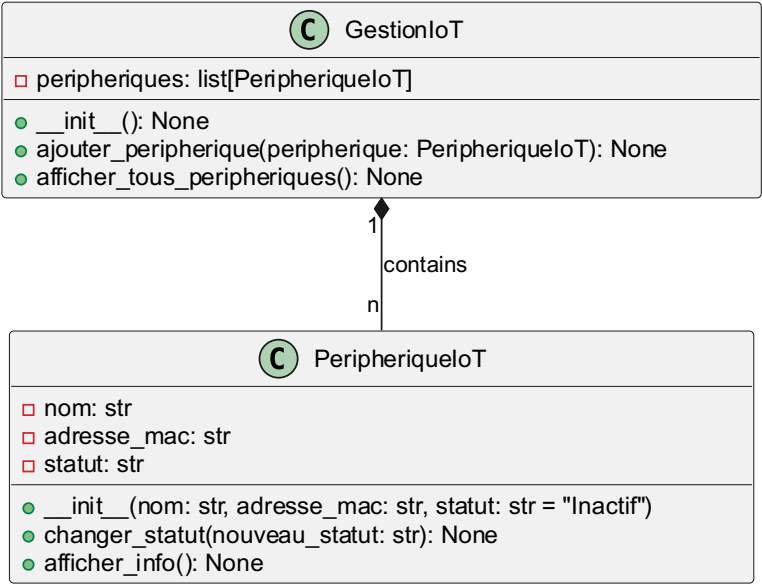
# Affichage des détails de la connexion
connexion.afficher_details()
```

Résultats attendus :

```
Détails de la connexion :
Équipement 1 :
Nom: Switch01, Adresse IP: 192.168.1.1
Équipement 2 :
Nom: Router01, Adresse IP: 192.168.1.254
Type de connexion : Ethernet
ikorri@MacBook-Air-de-ilyas cours_poo %
```

Exercice 2 : Gestion des périphériques IoT

Vous êtes chargé de développer une application pour gérer les **périphériques IoT** connectés dans un réseau. L'application doit permettre d'ajouter des périphériques, de modifier leur statut (actif/inactif), et d'afficher les informations de tous les périphériques enregistrés.



Assurez-vous que chaque classe inclut les attributs et méthodes spécifiés dans le diagramme.

Respectez les relations définies dans le diagramme, en particulier la relation de composition entre les classes.

Implémentez les classes en respectant les noms, attributs, et méthodes indiquées dans le diagramme.

Testez le programme en créant des objets et en affichant les résultats pour vérifier que les relations fonctionnent correctement.

```
iot1 = PeripheriqueIoT("Capteur Température", "00:1B:44:11:3A:B7")
iot2 = PeripheriqueIoT("Camera Sécurité", "00:1B:44:11:3A:B8", "Actif")
```

Le gestionnaire doit pouvoir manipuler plusieurs périphériques et fournir une vue d'ensemble claire de leurs états.

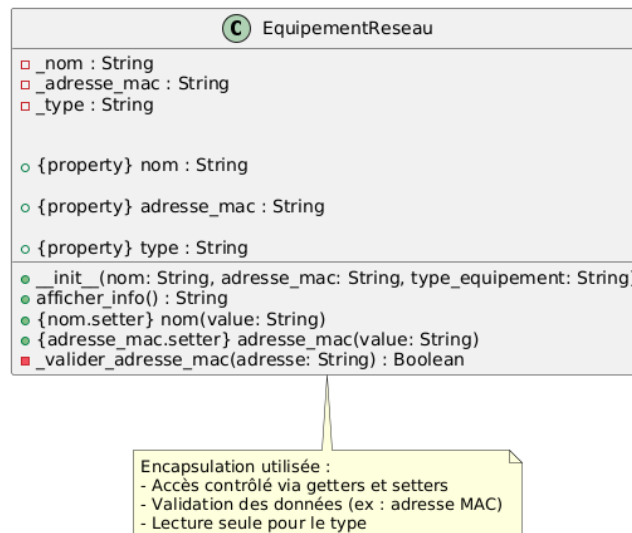
Résultats attendus

```
Nom: Capteur Température, Adresse MAC: 00:1B:44:11:3A:B7, Statut: Actif
Nom: Camera Sécurité, Adresse MAC: 00:1B:44:11:3A:B8, Statut: Actif
```

TRAVAUX PRATIQUES
La programmation en Python
<b>La programmation orientée objet</b>

BTS CIEL
2 <sup>ème</sup> année
Page 3 sur 3

Créer une classe EquipementReseau en respectant le diagramme de classe suivant :



Voici le code de la fonction principale :

```

# Test du programme principal
try:
    equipement1 = EquipementReseau("Switch Principal", "00:1A:2B:3C:4D:5E", "Infrastructure")
    print(equipement1.afficher_info())

    equipement1.adresse_mac = "01:23:45:67:89:AB"
    print(equipement1.afficher_info())

    # Cas invalide
    equipement2 = EquipementReseau("IoT Capteur", "ZZ:ZZ:ZZ:ZZ:ZZ:ZZ", "IoT")
except ValueError as e:
    print("Erreur :", e)
  
```

Résultats attendus :

```

Équipement: Switch Principal, Adresse MAC: 00:1A:2B:3C:4D:5E, Type: Infrastructure
Équipement: Switch Principal, Adresse MAC: 01:23:45:67:89:AB, Type: Infrastructure
Erreur : L'adresse MAC 'ZZ:ZZ:ZZ:ZZ:ZZ:ZZ' est invalide.
  
```