

# **Université de Franche-Comté - UFR SLHS - Examen terminal de 2nd session Année universitaire 2023-2024**

**Semestre** : 2nd session – 2nd semestre

**Année** : Master 1

**Unité d'enseignement** : VGT8U1 Mémoire et logiciel : techniques de programmation 2

**Responsable UE** : Iana ATANASSOVA

**Nom des correcteurs** : Nicolas Gutehrle

**Durée de l'épreuve** : 2H

**Nature de l'épreuve** : ECRIT

**Public concerné** : TOUS

**Documents autorisés** : Supports de cours (partie 2)

Tous les exercices sont à réaliser dans ce notebook. Les questions de l'exercice 1 peuvent être répondues dans un fichier à part.

La première partie est à rendre sur feuille. La seconde partie est à renvoyer par mail à [nicolas.gutehrle@univ-fcomte.fr](mailto:nicolas.gutehrle@univ-fcomte.fr)

Le plagiat, c'est-à-dire la copie directe des supports de cours sera sanctionnée.

L'utilisation d'Internet est interdite.

**L'utilisation des supports de cours est autorisée pour la seconde partie de ce travail.**

Le travail est strictement personnel. L'évaluation portera sur la compréhension des notions du cours qui sera démontrée par l'étudiant.

---

## Partie 1 : Questions de cours (10 pts)

Répondez aux six questions ci-dessous :

1. Selon le paradigme de Programmation Orienté Objet, qu'est-ce qu'une classe ? Une instance de classe ? (2pts)
2. A quoi servent les **getters** et les **setters** ? (1pt)
3. Qu'elle est la différence entre une méthode et une fonction ? (1pt)
4. Qu'est-ce que l'héritage en Programmation Orienté Objet ? En Python, que faut-il faire pour qu'une classe hérite d'une autre ? (2pts)
5. En Python, comment peut-on faire en sorte qu'un programme continue de fonctionner, même s'il rencontre des erreurs ? A l'inverse, comment déclencher volontairement une erreur en Python ? (2pts)
6. A quoi servent les environnements virtuels en Python ? Pourquoi sont-ils nécessaires ? (2pts)

---

## Partie 2 : Programmation (10 pts)

Implémentez la classe "Memoire", en respectant les consignes ci-dessous :

- Elle possède un attribut public "auteur" de type str
- Elle possède un attribut privé "titre" de type str également. Cet attribut doit avoir un getter et un setter pour pouvoir y accéder et le modifier.
- Elle doit avoir une méthode pour représenter l'instance sous forme de str. Cette méthode doit retourner le message "Mémoire {VALEUR DE L'ATTRIBUT TITRE} par {VALEUR DE L'ATTRIBUT TITRE}" (ce qui est entre accolade doit être remplacé par les valeurs des variables correspondantes)
- Elle doit avoir une méthode "afficher\_details". Cette méthode retourne un dictionnaire dont les clés sont "auteur" et "titre". La clé "auteur" a pour valeur l'attribut "auteur". La clé "titre" a pour valeur l'attribut "titre".

Créez une instance de cette classe avec les valeurs suivantes :

- auteur : votre prénom
- titre : "PremierTitre"

Puis réalisez les actions suivantes :

- Affichez cette instance sous forme de str
- Appelez la méthode **afficher\_details**
- Changez la valeur de l'attribut privé "titre" pour la valeur "NouveauTitre". Vous vous assurez que la modification a été prise en compte en affichant la valeur de l'attribut privé "titre".