

Exercices Heritage

Exercice 1 :

1. Ajouter la classe **Personne** à votre projet. Cette classe contient les attributs suivants : **ID, Nom, Salaire**.
2. Pour chaque attribut de la classe Personne, Ajouter des accesseurs et des modificateurs.
3. Modifier l'accesseur des attributs nom pour qu'il retourne la valeur de ces champs en majuscule.
4. Ajouter le constructeur à la classe Personne :
5. Ajouter à la classe Personne la méthode **AffichePersonne()** : Permet d'afficher les informations d'une personne.
6. Ajouter la classe Utilisateur à votre projet. Cette classe contient les attributs suivants : login ; password.
7. Modifier la classe Utilisateur pour qu'elle hérite de la classe Personne.
8. Pour chaque attribut de la classe Utilisateur, Ajouter des accesseurs et des modificateurs.
9. Ajouter le constructeur
10. Ajouter à la classe Utilisateur **AfficheUtilisateur()** qui permet d'afficher les informations d'un utilisateur.
11. Dans la méthode Main :
 - a. Créer et initialiser une liste d'utilisateurs.

- b. Proposer à utilisateur de saisir son login et son password et puis vérifier si ce login et password existe dans la liste déclarée :
- c. Si l'utilisateur existe, afficher le message : Authentification réussi et puis afficher tous les informations de cet utilisateur.
- d. S'il n'existe pas, afficher le message : Echec d'Authentification.

1.1.1 Exercice 3

- 1. Ecrire une classe appelée **Particulier** contient :
 - a. 3 variables d'instance privées :
 - i. numéro_CIN, (incrémenté auto)
 - ii. nom
 - iii. adresse
 - b. constructeurs :
 - c. 2 propriétés de lecture-écriture pour nom et adresse, et 1 accesseur en lecture seul pour numéro_CIN;
 - d. 1 méthode ToString().
- 2. Ecrire une classe appelée **Entreprise** contient :
 - a. 3 variables d'instance privées :
 - i. raison_sociale (de type string),
 - ii. capital (de type double)
 - iii. nombre_actions (de type int)
 - b. constructeurs
 - c. 3 accesseurs.

- d. 1 méthode `Prix_Action()` qui retourne la valeur d'une action d'une entreprise suivant la relation : Valeur d'une action = Capital / Nombre d'actions.
 - e. 1 méthode `ToString()`.
- 3. Ecrire une classe **Employé** représente les employés qui hérite de la classe `Particulier`, et contient :
 - a. 2 variables d'instance privées :
 - i. `matricule` (de type `string`) qui s'incrémente automatiquement.
 - ii. `entreprise` (de type `Entreprise`);
 - b. constructeur :
 - c. Une propriété de lecture-écriture pour `entreprise`, et 1 accesseur en lecture seul pour `matricule`;
 - d. 1 méthode `ToString()`.
- 4. Ecrire une classe **Cadre** représente les cadres qui hérite de la classe **Employé**, et contient :
 - a. Une variable d'instance privée : `grade` (de type `char`)
 - b. constructeur :
 - c. Une propriété lecture-écriture pour `grade`;
 - d. Une méthode `ToString()`.
- 5. Tester les classes que vous avez créés.