

Université Alioune Diop



UFR : SATIC - Département : TIC - Filière : SRT2 - EC : Java
Responsable du cours : Dr Lamine Yade

TP/TD N°2 : Tableaux & Chaînes de caractères & POO

Exercice 1 : Manipuler les chaînes de caractères et extraire des informations.

Écrire un programme qui prend une URL (ex: "https://uadb.sn") et affiche :

- Le protocole (https)
- Le nom de domaine (uadb)
- L'extension (sn)

☞ Utilisez les méthodes de la classe String : split(), substring(), indexOf()

Exercice 3: Utiliser les chaînes et conditions

Un utilisateur doit définir un mot de passe sécurisé. Le programme vérifie que le mot de passe :

- Contient au moins **8 caractères**
- Contient au moins **une lettre majuscule**
- Contient au moins **un chiffre**

1. Créer une classe Utilisateur qui a deux attributs privés **login** et mot de passe **mdp**. Créer les **Getter/Setter** des attributs. Ajouter une méthode **afficher()** qui affiche les infos d'un utilisateur.
2. Créer une autre classe représentant le programme principal. En plus de la méthode main, elle contient deux autres méthodes static:
 - `checkPassword(String mdp)` : qui vérifie si le mot de passe est valide ou non.
 - `createUser(String login, String mdp)` : qui crée et affiche un utilisateur.

☞ Utilisez les méthodes :

- `length()` : retourne la longueur de la chaîne.
- `toCharArray()` : transforme une chaîne en tableau de caractères.
- `Character.isUpperCase(char)` : vérifie si le caractère est une majuscule.
- `Character.isDigit(char)` : vérifie si c'est un chiffre.

Exercice 3:

1. Créez une classe **Livre** avec les attributs privés suivants :
 - *isbn* : identifiant unique du livre
 - *titre* : titre du livre
 - *annee* : année de publication
2. Créez le constructeur et des *getters/setters* pour chaque attribut. Setter de *annee* doit vérifier que l'année est > 1900.
3. Créez une classe **Auteur** avec les attributs : nom, prénom.
Un livre peut être écrit par plusieurs auteurs et un auteur peut écrire plusieurs livres.
4. Modifiez la classe *Livre* pour contenir une liste d'auteurs.
5. Modifiez la classe *Auteur* pour contenir une liste de livres.
6. Créez une méthode dans *Livre* pour ajouter un auteur, et dans *Auteur* pour ajouter un livre.
7. Dans la classe principale, affichez les livres d'un auteur donné, et les auteurs d'un livre donné.

Exercice 4: Héritage et Spécialisation

1. Créez une classe mère *Chantier* avec les attributs suivants : nom, lieu, région, pays.
2. Créez trois classes filles :
 - *ChantierConstruction* avec un attribut *typeBatiment*
 - *ChantierRehabilitation* (à compléter selon votre choix)
 - *ChantierArcheologique* avec les attributs *surface*, *epoque*, *typeBatiment*
3. Implémentez dans chaque classe une méthode *toString()* qui affiche toutes les informations d'un chantier.
4. Dans la classe principale, un tableau qui contient un chantier de réhabilitation, 2 chantiers de construction et un chantier archéologique.
5. Testez l'affichage.

Exercice 5: Classe Abstraite et Polymorphisme

1. Créez une classe abstraite **Personne** (*nom*, *prénom*) avec une méthode abstraite *afficherRole()*.
2. Créez une classe **Etudiant** héritant de **Personne**, avec l'attribut *niveau*.
3. Créez une classe **Enseignant** héritant aussi de **Personne**, avec *specialite*.
4. Implémentez *afficherRole()* pour chaque classe.
5. Dans la classe principale, créez une liste contenant des objets *Etudiant* et *Enseignant* et affichez leurs rôles de manière polymorphe.