

COURS 420-920-MA

TRAVAIL PRATIQUE #3

Consignes



Ce travail peut être fait en équipe d'au maximum 3 étudiant(e)s.

CLASSES ABSTRAITES, FICHIERS D'OBJETS ET HASHMAP

ÉTAPES À FAIRE

Reprendre votre **TP2** et y apporter les modifications suivantes :

1. Ajouter une option **Sauvegarder** à votre menu, comme suit :

```
===== MENU DU ZOO =====  
1. Ajouter un animal  
2. Afficher tous les animaux  
3. Rechercher un animal  
4. Supprimer un animal  
5. Sauvegarder  
6. Quitter
```

2. Rendre votre classe **Animal abstraite** ayant comme méthode abstraite **faireDuBruit**.
3. Votre classe **Animal** doit avoir un attribut de plus soit l'id de l'animal **private int id**;
4. Vous allez remplacer votre **ArrayList** par un **HashMap<clé, valeur> listeAnimaux** dont la clé sera le **ID** de l'animal et la valeur une instance d'un animal spécifique.
5. Au départ vous allez lire votre fichier de données **animaux.txt** (faudra y ajouter une clé à chacun comme 1, 2, 3, ...) et charger les objets dans votre **listeAnimaux**.
6. Toutes les opérations du **CRUD** se feront sur **listeAnimaux**.
7. L'option Sauvegarder du menu va enregistrer votre map **listeAnimaux** dans un fichier d'objets nommé **animaux.obj**.
8. Votre méthode **chargerAnimaux** doit vérifier en premier si le fichier **animaux.obj** existe. Si tel est le cas alors vous aller charger votre map de ce fichier, sinon vous aller lire le fichier texte **animaux.txt** et créer les objets à placer dans votre map. Lorsque vous allez sauvegarder le fichier **animaux.obj** sera crée et le fichier **animaux** ne sera plus jamais utilisé.

ÉVALUATION

Éléments évalués (100 points)

Critère	Description	Points
Classe abstraite	Utilisation correcte d'une classe de base et de classes dérivées avec surcharge et méthode abstraite	15
Composition	Classe Enclos avec HashMap <clé, valeur> bien implémentée	15
Map	Bonne gestion de l'ajout, la suppression, la recherche, l'affichage	40
Fichier objets	Bonne utilisation de sauvegarde et récupération du map à partir du fichier d'objets animaux.obj.	20
Lisibilité	Code commenté, bien structuré, indentation, noms explicites	5
Ergonomie globale	Présentation des résultats, affichage des champs de saisie, couleurs, couleurs de fond, etc.	5

Toute option du menu qui n'est pas fonctionnelle vous fera perdre 50% des points attribués à cette option, les autres 50% sont consacrés au code développé.

CONSIGNES

- L'application se lance avec un enclos par défaut : Enclos Principal (capaciteMax = 10).
- Tous les messages doivent être en français.
- Aucune saisie en console (System.in), uniquement via Swing.

Rendu attendu

- Dossier du projet au complet (dans un seul dossier compressé .zip)

ANNEXE

Voici deux fichiers texte que vous pouvez utiliser comme **données de départ** pour votre application :

 animaux.txt (avec l'attribut id)

Contient une liste d'animaux avec leurs attributs spécifiques selon leur type :

Lion;1;Simba;5;190.5;20

Elephant;2;Dumbo;10;540.3;1.8

Singe;3;George;3;33.2;Capucin

Lion;4;Nala;4;180.0;18

Elephant;5;Babar;12;600.7;2.0

Singe;6;Kiki;2;28.9;Singe-araignée

Format :

TypeAnimal;ID;Nom;Age;Poids;AttributSpécifique

- **Lion** → nombreCriniere (int)
- **Elephant** → longueurTrompe (double)
- **Singe** → espece (String)

 [enclos.txt](#)

Liste des enclos disponibles avec leur nom et leur capacité maximale :

Enclos Principal;10

Savane Africaine;8

Forêt Tropicale;6

Format :

NomEnclos;CapaciteMax

Bon travail. 😊