



1. Créer la classe **Animal**, qui est la superclasse abstraite de tous les animaux.
  - (a) Déclarer un attribut entier de visibilité *protected* appelé `pattes`, qui désigne le nombre de pattes de cet animal.
  - (b) Définir un constructeur de visibilité *protected* qui initialise l'attribut `pattes`.
  - (c) Déclarer une méthode abstraite **manger**.
  - (d) Déclarer une méthode **marcher** concrète qui affiche quelque chose sur la façon dont les animaux marchent (inclure le nombre de pattes).
2. Créer la classe **Araignee**.
  - (a) La classe **Araignee** étend la classe **Animal**.
  - (b) Définir un constructeur par défaut qui appelle le constructeur de la superclasse pour spécifier que toutes les Araignees ont huit pattes.
  - (c) Implémenter la méthode **manger** qui affiche un message "l'Araignee mange des insectes".
3. Créer l'interface **AnimalCompagnie** spécifiée par le diagramme UML en y ajoutant une méthode par défaut pour afficher un message générique.

4. Créer la classe **Chat** qui étend **Animal** et implémente **AnimalCompagnie**.
  - (a) Cette classe doit inclure un attribut **String** pour stocker le nom de l'animal.
  - (b) Définir un constructeur qui prend un paramètre **String** qui spécifie le nom du chat. Ce constructeur doit également appeler le constructeur de la superclasse pour spécifier que tous les chats ont quatre pattes.
  - (c) Définir un autre constructeur qui ne prend aucun paramètre. Demander à ce constructeur d'appeler le constructeur précédent (en utilisant le mot clé **this**) et de passer une chaîne vide comme argument.
  - (d) Implémenter les méthodes d'interface **AnimalCompagnie**.
  - (e) Implémenter la méthode **manger**.
5. Créer la classe **Poisson**. Remplacer les méthodes de la classe **Animal** pour spécifier que les poissons ne peuvent pas marcher et n'ont pas de pattes.
6. Créer un programme **TestAnimaux**. Demander à la méthode principale de créer et de manipuler des instances des classes créées ci-dessus. Commencer avec:

```
Poisson d = new Poisson();  
Chat c = new Chat("Fluffy");  
Animal a = new Poisson();  
Animal e = new Araignee();  
AnimalCompagnie p = new Chat();
```

Expérimenter en appelant les méthodes dans chaque objet