

1. Créer un package de nom `humain`,
2. Créer une classe `humain.Homme` ayant pour attributs un `prénom` et `nom` et des opérations comme `String getPrénom()`, `String getNom()`, `int getSexe()` (qui renvoie toujours 1), `String prononcer(String msg)` (qui renvoie la chaîne (`Voix grave`) suivie du `msg`).
3. Créer une classe `humain.Femme` ayant pour attributs un `prénom` et `nom` et des opérations comme `String getPrénom()`, `String getNom()`, `int getSexe()` (qui renvoie toujours 2), `String prononcer(String msg)` (qui renvoie la chaîne (`Voix aiguë`) suivie du `msg`).
4. Créer une interface `humain.Prénomable` dont on identifiera la ou les méthode(s). Identifier les classes correspondant à cette interface et établir la relation d'implémentation.
5. Créer une interface `humain.Bavard` dont on identifiera la ou les méthode(s). Identifier les classes correspondant à cette interface et établir la relation d'implémentation.
6. Créer un package `général`,
7. Créer une interface `général.EntitéNommée` dont on identifiera la ou les méthode(s). Identifier les classes correspondant à cette interface et établir la relation d'implémentation.
8. Créer une interface `humain.Humain` liée aux interfaces précédentes (un humain est nommé, prénommé et bavard, n'est-ce pas ?). Modifier les classes en conséquence.
9. Dans un `main` contenu dans une classe `Test` dans un package `prog`, fabriquer quelques femmes et hommes, et puis faire la liste des bavards et les faire causer (en leur demandant de prononcer un message quelconque).
10. Créer un package `animal`.
11. Créer une classe `animal.Chat` qui devra posséder un `nom`, un `sexe` (1 ou 2 - choisit à la «construction») et être capable de crier (`Crier()` qui affiche un message quelconque).
12. Faire apparaître une interface `général.Sexué` avec la ou les méthode(s) utiles. Établir les bonnes relations avec cette interface.
13. Dans le `main` créer des chats et fabriquer la liste de toutes les entités nommées et écrire une fonction capable d'afficher le nom de toutes ces entités.
14. Faire une modélisation UML (on pourra oublier les attributs et les méthodes) de la solution obtenue.