

1. Créer, dans un package `humain`, une classe `Personne` et y définir une méthode `void bonjour()` qui affiche le message `Bonjour`. Créer dans un package `principal`, une classe `Test` contenant une fonction `main` permettant de tester la réponse d'une personne lorsqu'on tente de lui dire bonjour (ie. `p.bonjour()`).
2. (extends) Créer dans le package `humain` une sous-classe de `Personne`, `PersonneFamiliere` qui lorsqu'on lui dit bonjour répond `Salut`. Tester dans le programme principal.
3. (méthode polymorphe) Modifier le programme principal de sorte à implémenter une méthode statique `void direBonjour(Personne p)` qui sollicite la personne de sorte qu'elle réponde à la sollicitation (ie. `p.bonjour()`). Modifier le `main` de sorte à appeler `direBonjour` sur une personne ordinaire ou un individu familial.
4. (extends) Créer dans le package `humain` une classe `Grognon` qui répond aux bonjours par des grommellements (`Grrr hmmm...`). Modifier le programme principal de sorte à tester ces nouveaux individus mal léchés.
5. (méthode polymorphe) Créer dans la classe de test une méthode prenant en paramètre deux personnes et faire en sorte qu'elles se disent bonjour... Tester avec différentes combinaisons.
6. (structure polymorphe) Modifier le programme principal de sorte à avoir un petit menu permettant de créer à volonté des individus ordinaires, familiaux ou grognons et à les stocker dans une liste. Le menu permettant d'obtenir la réponse au bonjour de tous les individus de la liste :

```
1 - créer personne normale
2 - créer individu familial
3 - créer personne grognon
4 - dire bonjour à tout le monde
5 - quitter
```

7. (délégation de constructeur) Modifier la classe `personne` de sorte qu'elle possède un attribut `prenom` qui sera initialisé à la construction et modifier la méthode `bonjour` de sorte qu'une personne se présente sous la forme "`(xxxx) bonjour`" ou "`(xxxx) salut`", etc (où `xxxx` est le prénom de la personne). Attention : cela aura des implications sur les sous-classes.