

1) Les Animaux

Commencez par écrire une classe qui s'appelle Animal. Celle-ci doit comporter uniquement :

- une méthode qui s'appelle cri qui ne prend aucun paramètre en entrée et qui retourne une chaîne de caractères vide.
- une redéfinition de la méthode toString qui retourne simplement la chaîne de caractères "Je suis un animal".

Maintenant, créez une classe qui s'appelle Chat qui hérite de la classe Animal et qui :

- redéfinit la méthode cri afin de retourner la chaîne de caractères "Miaaaw !".
- redéfinit la méthode toString afin de retourner la chaîne de caractères "Je suis un chat !".
- Enfin, créez une troisième classe qui s'appelle Chien qui hérite, elle aussi, de la classe Animal et qui contient :
 - une redéfinition de la méthode toString qui retourne la chaîne de caractères "Je suis un chien !".
 - une méthode qui s'appelle aboyer qui ne prend aucun paramètre en entrée et qui retourne la chaîne de caractères "Haw haw !".

Testez vos trois classes grâce au bout de code suivant. Que remarquez-vous ?

```
public class TestAnimaux {
    public static void main(String[] args){

        // Création d'un animal, un chat et un chien.
        Animal animal = new Animal();
        Chat chat = new Chat();
        Chien chien = new Chien();

        // Affichage des trois animaux (toString)
        System.out.println(animal);
        System.out.println(chat);
        System.out.println(chien);

        // Affichage des cris des animaux
        System.out.println(animal.cri());
        System.out.println(chat.cri());
        System.out.println(chien.cri());
        System.out.println(chien.aboyer());

        // On considère le chat et le chien comme étant des animaux
        Animal achat = chat;
        Animal achien = chien;

        System.out.println(achat);
        System.out.println(achien);

    }
}
```

2) Les Pokémons

Les Pokémon sont certes de très mignonnes créatures, mais ils sont également un bon exemple pour illustrer l'héritage.

Je vous propose donc de commencer par créer une classe Pokemon qui contient (entre autres) :

- un attribut **nom** qui représente le nom du Pokémon.
- un attribut **hp** (pour Health Points) qui représente les points de vie du Pokémon.
- un attribut qui s'appelle **atk** qui représente la force de base de l'attaque du Pokémon.
- un constructeur pour instancier des Pokémon adéquatement.
- des getters (accesseurs) qui permettent de consulter les attributs du Pokémon.
- une méthode **isDead()** qui retourne un boolean pour indiquer si un Pokémon est mort ($hp == 0$) ou non.
- une méthode **attaquer(Pokemon p)** qui permet au Pokémon appelant d'attaquer le Pokémon passé en paramètre. L'attaque déduit atk points de la vie hp du Pokémon attaqué p .
- une redéfinition de la méthode `toString` qui affiche les informations du Pokémon.

En plus des Pokémon normaux (décrits à travers la classe Pokemon) on recense trois types de Pokémon. Les Pokémon de type Feu, les Pokémon de type Eau et les Pokémon de type Plante:

- les Pokémon de type Feu sont super efficaces contre les Pokémon de type Plante et leur infligent deux fois plus de dégâts ($2*atk$). Par contre, ils sont très peu efficaces contre les Pokémon de type Eau ou de type Feu et ne leur infligent que la moitié des dégâts ($0.5*atk$). Ils infligent des dégâts normaux aux Pokémon de type Normal.
- les Pokémon de type Eau sont super efficaces contre les Pokémon de type Feu et leur infligent deux fois plus de dégâts ($2*atk$). Par contre, ils sont très peu efficaces contre les Pokémon de type Eau ou de type Plante et ne leur infligent que la moitié des dégâts ($0.5*atk$). Ils infligent des dégâts normaux aux Pokémon de type Normal.
- enfin, les Pokémon de type Plante sont super efficaces contre les Pokémon de type Eau et leur infligent deux fois plus de dégâts ($2*atk$). Par contre, ils sont très peu efficaces contre les Pokémon de type Plante ou de type Feu et ne leur infligent que la moitié des dégâts ($0.5*atk$). Ils infligent des dégâts normaux aux Pokémon de type Normal.

Créez trois classes `PokemonFeu`, `PokemonEau` et `PokemonPlante` qui héritent de la classe `Pokemon` et qui représentent les trois types de Pokémon susmentionnés. Ensuite, amusez-vous à faire des combats de Pokémon.

Remarque : Connaître le nom de la classe d'un objet

Afin de connaître la classe à laquelle un objet o appartient, vous pouvez utiliser la méthode `getClass()` (appel : `o.getClasse()`). Pour connaître le nom de cette classe, vous pouvez utiliser la méthode `getName()` (`o.getClass().getName()`) qui retourne le nom complet de la classe (y compris le nom du

package auquel elle appartient) ou la méthode `getSimpleName()` (`o.getClass().getSimpleName()`) qui retourne seulement le nom de la classe sans le nom du package.