



MASTER LANGUE ET INFORMATIQUE

Travaux Pratiques n° 6 Héritage et arborescence de classes

1. VOS PREMIERES CLASSES DERIVEES

L'objectif de cette partie est de créer votre première arborescence de classes. Soit une classe abstraite Animal, avec 2 attributs, 1 constructeur et 2 méthodes

```
/** Création et gestion d'animaux */
public abstract class Animal{
    private String espèce;
    private int nb_pattes = 0;

    /** création d'une nouvelle instance de la classe Animal
     * @param type nom de l'espèce
     * @param pattes nombre de pattes
     */
    public Animal(String type, int pattes) {
        espèce=type;
        nb_pattes = pattes;
    }

    /** présentation des caractéristiques de l'animal */
    public void présente() {
        System.out.println("je suis un représentant de l'espèce des " +espèce+
            " et j'ai " +nb_pattes+ " pattes" );
    }

    /** cri de l'animal */
    public abstract void crie();
}
```

Exercice 1: Compléter le code de la classe Félin. Ecrire une classe de test.

```
/** Création et gestion de félins */
public class Félin extends Animal {
    // ajout d'attributs propres à la sous-classe
    protected boolean domestique = false;

    /** le constructeur de Félin fait appel au constructeur de la sur-classe Animal */
    public Félin(String type) { // les félins ont 4 pattes
        .....
    }

    /** présentation des caractéristiques du félin */
    public void présente() {
        // appel de la méthode de la sur-classe
        .....
        String etat = (domestique) ? "domestique" : "sauvage";
        System.out.println("je suis vraiment un animal " + etat) ;
    }

    /** cri du félin */
    .....
}
```

Exercice 2: Créer une classe Chat dérivée de la classe Félin et implémentant l'interface Domesticable. Ecrire une classe de test.

```
/** gestion de l'état de domestication d'un animal */
public interface Domesticable {

    /** Domestiquer un animal et lui donner un nom
     * @param nom nom de l'animal */
    public abstract void domestiquer(String nom);

    /** Rendre le nom de l'animal
     * @return nom de l'animal */
    public abstract String nom();
}
```

Exercice 3: Générer la documentation Javadoc

Exercice 4: Créer et tester une des trois arborescences suivantes de classes en ajoutant au moins un attribut et une méthode à chaque dérivation.

- 1) Animal (2 attributs, 2 méthodes) -> Reptile -> Crocodile
- 2) Métal (2 attributs, 2 méthodes) -> Bronze -> Charrue
- 3) Végétal (2 attributs, 2 méthodes) -> Arbre -> Peuplier