



TP Animal Polymorphisme

Thème : diagramme de classes, polymorphisme, packaging, librairie

1 Question : Diagramme de classe sous un AGL

Sous Rose (par exemple), et dans la <Logical View < Main, élaborer le diagramme de classes qui documente le projet Animal.

Vous complétez ce modèle avec les questions demandées.

2 Question : nouveau projet

Vous dupliquez le projet animal_interface précédent dans le projet animal_polymorphisme.

Dans le main de TestAnimal, vous supprimez des tableaux l'objet Animal "Truc"

Cela donne :

```
public class TestAnimal {
    public static void main(String[] args) {
        Animal[] animaux = new Animal[2];
        animaux[1] = new Chien("Médor");
        animaux[2] = new Homme("Robert");
        for (int i = 0; i < animaux.length; i++) {
            System.out.println(animaux[i]);
        }
    }
}
```

3 Question : Polymorphisme

On souhaite faire parler les animaux en mettant en œuvre le polymorphisme.

Faites l'évolution nécessaire pour avoir l'affichage suivant :

```
Médor parle :waouf waouf
Robert parle :Vive Le polymorphisme!
```

4 Question : la méthode parle pour le polymorphisme

Commentez la méthode parle de la classe Animal. Etes-vous obligé d'avoir une méthode parle dans la classe Animal ?

Mettez la méthode parle d'Animal en « abstract ». Que se passe-t-il pour la classe Animal ?

Expliquez :

Eventuellement, adaptez son main en commentant les lignes.

Quel serait l'avantage de mettre la méthode parle d'Animal en « abstract » ?

Obliger à coder... parle()

Pour la suite, vous ne gardez pas la méthode parle en abstract.

5 Question : une collection à la place du tableau

5.1 Mise en place

Utilisez une collection de type « ensemble ordonné » ou « tableau dynamique » et chargez dedans les mêmes éléments que pour le tableau. Indiquez votre choix de collection:

.....Affichez son contenu et faites parler les animaux.

5.2 Question : Triez dans les 2 sens la collection

Vous triez ascendant la collection puis affichez les éléments.

Vous triez descendant la collection puis affichez les éléments.

Indiquez les instructions ici : *Collection.sort*
.....*Collection.reverse*

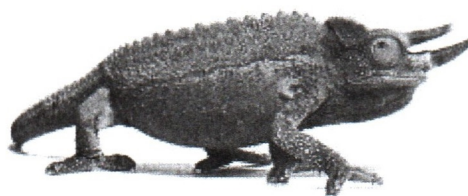
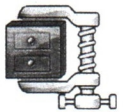
5.3 Question : programmation fonctionnelle

(Pour ceux qui avancent vite) Faites un autre affichage de la collection avec l'API Stream Java8. Faites parler l'ensemble des éléments en utilisant la programmation fonctionnelle. Supprimez Robert et Médor de collection puis affichez.

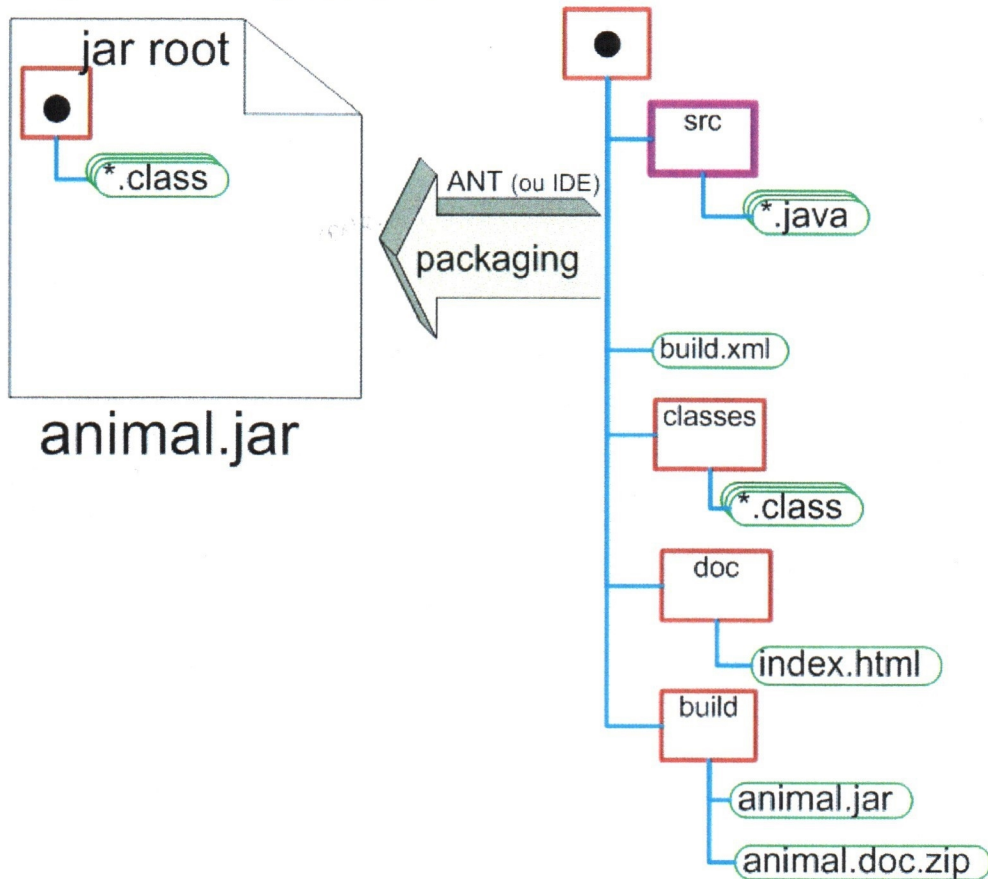
6 Question : Packaging d'un livrable

Grâce à Eclipse, packagez votre dernière application animal_polymorphe pour produire la librairie animal.jar (vous n'avez pas besoin que le JAR soit exécutable (runnable), ni qu'il contienne les tests, ni qu'il contienne les sources). Donnez cette librairie à votre voisin qui vous donnera la sienne.

Vous devez aussi fournir la javadoc. Soit cette javadoc est fourni « à part » sous forme d'un ZIP animal_doc.zip, soit elle est embarquée dans le jar (sous un répertoire doc). Voir l'illustration plus loin dans les annexes de ce document.



7 Annexe : Illustration pour la librairie animal.jar et la javadoc à produire



8 Annexe : Le diagramme de classes

