



Projet animal

La hiérarchie de classes

Thèmes : l'héritage en Java, constructeur, super(), toString, méthode virtuelle, exercice de classification

1 Question : analysez et classifiez

Le contexte étudié : Des animaux, des mammifères, des chiens, des hommes, ... Avec un diagramme de classes (UML) et sur cette feuille, modélisez le contexte en utilisant l'héritage. Dans le but d'avoir une application minimaliste, l'unique variable d'instance est le « nom » et elle est située dans la classe Animal. Faites valider votre diagramme par le formateur :

.....

2 Test Driven Development (TDD) avec la classe TestAnimal fournie

Sous votre IDE, créez un nouveau projet animal et travaillez, par exemple, dans les packages animal.domain et animal.test

Dans le cadre d'un développement guidé par le test (TDD), votre responsable technique vous fournit la classe animal.test.TestAnimal plus bas, ainsi que son affichage (voir plus loin encore). Recopiez-la telle quelle et c'est à la fin qu'elle doit fonctionner avec les classes que vous allez écrire. Donc, gardez-la précieusement sans l'utiliser pour l'instant.

```
public class TestAnimal {
public static void main(String[] args) {
    Animal[] animaux = new Animal[5];
    animaux[0] = new Animal("Truc");
    animaux[1] = new Animal();
    animaux[2] = new Chien("Médor");
    animaux[3] = new Homme();
    animaux[4] = new Homme("Robert");
    for (int i = 0; i < animaux.length; i++) {
        System.out.println(animaux[i]);
    }
}}
```

Tournez la page SVP

3 Développement

3.1 Le grand principe

Au fur et à mesure du développement, vous testez dans le main de chaque classe les constructeurs et le toString.

3.2 Ecriture des classes et des constructeurs

Dans quel ordre allez-vous coder les classes ?

Avec le code dans l'encart plus haut (TestAnimal), notez les constructeurs dont vous avez besoin. Vous ne codez que les constructeurs strictement nécessaires.

Rappel : l'unique information est le « nom » et elle est située au niveau le plus abstrait c.a.d. dans la classe Animal.

Nota : Pour les constructeurs et si c'est nécessaire, vous pouvez utiliser le constructeur de la classe mère avec « super(...) »

3.3 Les toString et leurs affichages imposés

Notez la forme des toString dont vous avez besoin avec l'encart plus bas. En effet, au runtime de la classe TestAnimal, votre responsable technique vous impose l'affichage suivant:

<p>Je suis un animal de nom Truc. Je suis un animal. Je suis un chien de nom Médor. Je suis un homme. Je suis un homme de nom Robert.</p>

Nota : La méthode polymorphe toString est redéfinie (override) dans les différentes classes. Pensez à l'annotation. En Java les méthodes sont virtuelles et, donc, « C'est le toString, le plus adapté au type d'objet, qui est appelé ».

4 Au final, le runtime de la classe TestAnimal

Avec cette classe TestAnimal fournie, vous vérifiez l'affichage imposé (voir plus haut).

5 Un initialiseur de tableau et un for « foreach » java5

Modifiez le code de TestAnimal avec un initialiseur de tableau et un « foreach » pour l'affichage.

6 Programmation fonctionnelle

Dans TestAnimal, ajoutez un nouvel affichage (même aspect) en faisant de la programmation fonctionnelle (les collections et l'API Stream Java8).

