

La programmation Orientée Objet

1. Compréhension de programmes

Exercice 1 — *Résultat d'exécution*

Quel est le résultat de l'exécution du programme suivant :

```
public class Exo1{
    public static void main(String[] args){
        B b = new B(5);
    }
}
class A{
    public A(){
        System.out.println("Je_suis_dans_A");
    }
}
class B extends A{
    public B(int x){
        System.out.println("Je_suis_dans_B");
    }
}
```

Exercice 2 — *Affichage de programme*

Qu'affiche le programme suivant :

```
public class Exo2{
    class A{
        int i = 0;
        A(int j){
            this.i = j;
        }
        void setI(int k){
            this.i=k;
        }
        void setI(A a){
            this.i = a.i;
        }
    }
}

class B extends A{
    int i = 1;
    B(){
        super(2);
    }
    void setI(int l){
        this.i = l;
    }
}
```

```

    void setI(A a){
        super.i = a.i;
    }

    void setI(B b){
        this.i = b.i;
    }
}
public static void main(String[] truc){
    A a = new A(5);
    B b = new B();
    System.out.println("a.i="+a.i+",_b.i="+b.i+"_ou_"+((A) b).i);
    b.setI(3);
    b.setI(a);
    System.out.println("a.i="+a.i+",_b.i="+b.i+"_ou_"+((A) b).i);
}
}

```

2. Analyse et Conception Orientée Objet

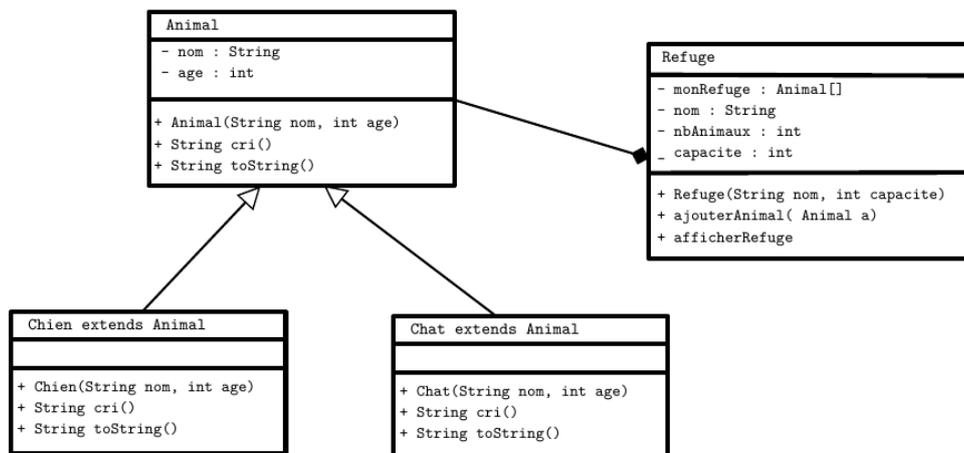
Exercice 3 — *Un élevage de volaille*

Un éleveur de volaille élève des canards et des poulets jusqu'à ce qu'ils aient la taille nécessaire à leur commercialisation. Une volaille est caractérisée par son poids et un numéro d'identification reporté sur une bague qu'elle porte a sa patte. Les volailles arrivent à l'élevage à l'âge de trois semaines. Elles sont baguées et enregistrées dans le système informatique. Il y a deux sortes de volailles : des canards et des poulets. Le prix du canard et celui du poulet sont deux prix différents, exprimés en euros par kilo. En revanche, le prix est le même pour tous les individus de la même espèce. Ce prix varie chaque jour. Le poids auquel on abat les bêtes est différents pour les canards et les poulets, mais c'est le même pour tous les poulets (respectivement, tous les canards).

Donnez la modélisation UML de la classe des volailles avec deux sous-classes des poulets et des canards.

Exercice 4 — *Conception d'un refuge*

Dans cet exercice, on va créer un refuge avec différents animaux. Chaque `Animal` est caractérisé par son `nom` et son `age`. L'application est modélisé par le diagramme UML suivant :



- (1) La classe `Animal` a deux attributs, `nom` de type `String` et `age` de type `int`. Vous coderez un constructeur qui prend en paramètre le nom et l'âge de l'animal. Dans cette classe vous coderez :

- la méthode `String cri()` qui retourne une chaîne de caractères vide.
 - une redéfinition de la méthode `toString` qui retourne la chaîne de caractères "Mon nom est `nom` et j'ai `age` ans."
- (2) la classe `Chat` qui hérite de la classe `Animal` et qui :
- redéfinit la méthode `cri()` afin de retourner la chaîne de caractères "Miaouuu!".
 - redéfinit la méthode `toString ()` afin de retourner la chaîne de caractères "Mon nom est `nom` et j'ai `age` ans. Je suis un chat! Miaouuu!".
- (3) la classe `Chien` qui hérite de la classe `Animal` et qui :
- redéfinit la méthode `cri()` afin de retourner la chaîne de caractères "Ouaf Ouaf!".
 - redéfinit la méthode `toString ()` afin de retourner la chaîne de caractères "Mon nom est `nom` et j'ai `age` ans. Je suis un chien! Ouaf Ouaf!".
- (4) Créez une classe `Refuge` qui caractérise un refuge qui est composé d'un tableau `refuge` dont les éléments sont de type `Animal`, d'un `nom`, d'une capacité maximale `capacite`, et d'une capacité courante `nbAnimaux`. Vous coderez les méthodes suivantes :
- `public Refuge(String nom, int capacite)` qui crée une instance de `Refuge`,
 - `public void ajouterAnimal(Animal a)` qui ajoute un l'`Animal a` au `refuge`, et qui renvoie un message d'erreur si la capacité maximale du refuge est atteinte,
 - `public void afficherRefuge()` qui affiche chaque `Animal` présent dans `refuge`.
- (5) Créez une classe `RefugeMain`, où vous exécuterez les instructions suivantes :
- Déclarez 2 variables de type `Animal` que vousinstancierez pour chacune des espèces `Chien` et `Chat`.
 - Déclarez une variable de type `Refuge`, instanciez la et ajoutez lui les 2 animaux que vous venez de créer.
 - Affichez le `Refuge` ainsi créé.