

## 1 Le zoo

Voici la classe Zoo dont nous voulons comprendre le fonctionnement. Les commentaires ont volontairement été supprimés, ainsi que les détails non-pertinents pour l'exercice.

```
public class Zoo {
    private Lion theLion;
    private Gazelle theGazelle ;
    private Monkey theMonkey;

    public Zoo() {
        theLion = new Lion("Simba",4,167.5);
        theGazelle = new Gazelle("Lady Gaga");
        theMonkey = new Monkey("Cheeta");
    }

    public void display() {
        System.out.println(theLion.toString());
        System.out.println(theGazelle.toString());
        System.out.println(theMonkey.toString());
    }

    public void feed() {
        theLion.eat();
        theGazelle.eat();
        theMonkey.eat();
    }
}
```

1. Listez les méthodes et propriétés du zoo.
2. Considérez le *constructeur* Zoo et décrivez par un diagramme d'objets, les effets de son exécution ligne à ligne. Chaque objet est représenté par un rectangle, contenant le nom et la classe de l'objet, et la liste de ses propriétés et de leurs valeurs.
3. On considère le fragment de programme suivant, appartenant à une classe nommée Main :

```
Zoo z = new Zoo();
z.display();
System.out.println("At lunch!!!!");
z.feed();
z.display();
z.display();
```

Dessinez le diagramme de séquence correspondant. Chaque objet est représenté par une ligne de temps verticale. Les appels de méthodes sont représentés par des flèches horizontales étiquetées par le nom de la méthode et ses paramètres. La ligne de temps d'un objet est épaissie lorsque l'objet est actif.

## 2 Classes

Voici les classes des animaux contenus dans le zoo. Nous ne vous donnons qu'un extrait significatif.

```
public class Gazelle {

    private final String name;
    private int age;
    private double weight;
    private int hornsLength;

    public Gazelle(String name) {
        this.name = name;
        this.age = 1;
        this.weight = 12;
        this.hornsLength = 9;
    }

    public Gazelle(String name, int age, double weight, int hornsLength) {
        this.name = name;
        this.age = age;
        this.weight = weight;
        this.hornsLength = hornsLength;
    }

    public void eat() { weight = weight + 0.075; }

    public void sleep() { weight = weight - 0.050; }

    public void run() { weight = weight - 0.010; }

    public void jump() { weight = weight - 0.010; }

    public String toString() {
        return "I'm a Gazelle, my name is " + name + ", "
            + "I'm " + age + ", "
            + "I weigh " + weight + "kg "
            + "and my horns are " + hornsLength + "cm long.";
    }
}
```

```
public class Monkey {

    private final String name;
    private int age;
    private double weight;

    public Monkey(String name) {
        this.name = name;
        this.age = 1;
    }
}
```

```

    this.weight = 1;
}

public Monkey(String name, int age, double weight) {
    this.name = name;
    this.age = age;
    this.weight = weight;
}

public void eat() { weight = weight + 0.3; }

public void sleep() { weight = weight - 0.1; }

public void run() { weight = weight - 0.120; }

public void shout() { System.out.println("Wiwiwi!"); }

public String toString() {
    return "I'm a Monkey my name is " + name + ", "
        + "I'm " + age + ", "
        + "I weigh " + weight + "kg.";
}
}

```

```

public class Lion {

    private final String name;
    private int age;
    private double weight;

    public Lion(String name, int age, double weight) {
        this.name = name;
        this.age = age;
        this.weight = weight;
    }

    public void eat() { weight = weight + 1.9; }

    public void sleep() { weight = weight - 0.8; }

    public void run() { weight = weight - 0.5; }

    public String shout() { return "Roaaaaar !"; }

    public String toString() {
        return "I'm a Lion, my name is " + name + ", "
            + "I'm " + age + ", "
            + "I weigh " + weight + "kg.";
    }
}

```

1. Pour chaque classe, repérez ses propriétés, ses méthodes et ses constructeurs.
2. Expliquez ce que représente chacune des classes. Réalisez un diagramme de classe : représenter chaque classe par un rectangle avec le nom de la classe, la liste de ses propriétés et la liste de ses méthodes.
3. Quelles sont les méthodes de **Gazelle** utilisées par le zoo ?

### 3 L'éléphant

L'éléphant possède un nom, un âge, un poids et une longueur de trompe. Il peut crier (il barrit), manger (il grossit de 8kg), dormir (il perd 1kg), courir (il perd 1.3kg). On souhaite le représenter par une classe.

1. Quelles sont les propriétés et les méthodes d'un éléphant ? Représentez la classe des éléphants dans votre diagramme de classe.
2. En vous inspirant des autres animaux, écrire la classe **Elephant** dans la syntaxe de Java.
3. Modifier le zoo pour qu'il contienne un éléphant, et mettez vos diagrammes à jour pour tenir compte de ce nouvel objet.