

# Poulailler

Soit le code suivant (incomplet) permettant de représenter la vie d'un poulailler.

A chaque tour, lors de l'invocation de la méthode `tourSuivant()` de la classe `Poulailler`, toutes les poules pondent aléatoirement 0, 1 ou 2 oeufs, puis tous les oeufs éclosent et donnent naissance à de nouvelles poules qui sont rajoutées au poulailler.

```
interface Oeuf {
    Object eclore();
}

// Un oeuf (générique) peut eclore et
// faire naître une nouvelle entité

public class Poule
{
    private String nom;
    private int nbPontes;

    // Nom de la poule
    // Nombre d'oeufs pondus

    public Poule(String s) {
        nom = s;
    }

    public Oeuf pondre() {
        // a completer
    }

    // Ponte d'un oeuf (de Poule)
}

public class Poulailler
{
    private int tour;
    private LinkedList poules = new LinkedList();
    private Random random = new Random();

    // Numéro de tour dans la simulation
    // Poules du poulailler
    // Générateur aléatoire

    public void ajouter(Poule p) {
        poules.add(p);
    }

    // Ajout d'une poule dans le poulailler

    public void tourSuivant() {
        // a completer
    }

    // Nouveau tour (ponte + eclosion)

    public static void main(String[] args) {
        Poulailler p = new Poulailler();

        // Démarrer la simulation
        // d'un poulailler

        p.ajouter(new Poule("Cocotte"));
        p.ajouter(new Poule("Rosette"));

        // possède au départ 2 poules

        for (int i = 0; i < 10; i++)
            p.tourSuivant();

        // sur 10 tours
    }
}
```

L'exécution du programme doit produire un résultat similaire à:

```
-- Tour #1
Cocotte pond son oeuf #1
Cocotte pond son oeuf #2
Rosette pond son oeuf #1
Cocotte1 sort de l'oeuf #1 de Cocotte
Cocotte2 sort de l'oeuf #2 de Cocotte
Rosette1 sort de l'oeuf #1 de Rosette
-- Tour #2
Cocotte pond son oeuf #3
Cocotte2 pond son oeuf #1
Cocotte2 pond son oeuf #2
Rosette1 pond son oeuf #1
Cocotte3 sort de l'oeuf #3 de Cocotte
Cocotte21 sort de l'oeuf #1 de Cocotte2
Cocotte22 sort de l'oeuf #2 de Cocotte2
Rosette11 sort de l'oeuf #1 de Rosette1
...
```

1. Ecrire le code de la méthode `public Oeuf pondre()` de la classe `Poule` en considérant que le *i*-ème oeuf d'une poule donnée donnera naissance à une poule qui prendra le nom de sa mère concaténé avec *i*.  
Par exemple, le 4ème oeuf de la poule `Cocotte12` donnera naissance à la poule `Cocotte124`.
2. Ecrire le code de la méthode `public void tourSuivant()` de la classe `Poulailler` en prenant garde à d'abord faire pondre aléatoirement 0, 1 ou 2 oeufs à toutes les poules existantes et seulement ensuite faire éclore les oeufs.