

# TP 2: Java Threads indépendants

Master 2 STIC

Module : Les applications réparties

# Exercice 1 : Des threads indépendants

Un "compteur" a un nom (Amine par exemple) et il compte de 1 à n (nombre entier positif quelconque). Il marque une pause aléatoire entre chaque nombre (de 0 à 5000 millisecondes par exemple).

- Un compteur affiche chaque nombre (Amine affichera par exemple, "Amine : 3") et il affiche un message du type "\*\*\* Toto a fini de compter jusqu'à 10" quand il a fini.
- Ecrivez la classe compteur et testez-la en lançant plusieurs compteurs qui comptent jusqu'à 10. Voyez celui qui a fini le plus vite.

```
public class Compteur extends Thread { // Cr ation d'un thread
    protected String nom; // Un objet protected est accessible dans tout le package et dans les sous classes
    protected int max;

    public Compteur(String nom, int max) {
        this.nom = nom; // Le mot-cl  this d signe, dans une classe, l'instance courante de la classe elle-m me.
        this.max = max;
    }

    public Compteur(String nom) {
        this(nom, 10);
    }
}
```

```
public void run() {
    for (int i = 1; i <= max; i++) {
        try {
            sleep((int)(Math.random() * 5000));
        }
        catch(InterruptedException e) {
            System.err.println(nom + " a ete interrompu.");
        }
        System.out.println(nom + " : " + i);
    }
    System.out.println("*** " + nom + " a fini de compter jusqu'à " + max);
}
```

```
public static void main(String[] args) {
    Compteur[] compteurs = {
        new Compteur("Mohamed"),
        new Compteur("Amine"),
        new Compteur("Karim"),
        new Compteur("Soufiane")
    };
    for (int i = 0; i < compteurs.length; i++) {
        compteurs[i].start();
    }
}

} // Compteur
```