



Thread

Les Processus

On appelle Processus un programme en cours d'exécution.

Chaque Processus est indépendant des autres et possède sa propre zone mémoire.

Les processus peuvent être associés à des applications "visibles" ou à des applications exécutées en tâche de fond.

Dans un système monoprocesseur, les processus semblent être exécutés simultanément mais sont en réalité exécutés successivement par le système d'exploitation.



Thread

Un processus peut contenir et exécuter plusieurs thread (processus légers), qui partagent la même zone mémoire du processus.

Les threads donnent l'impression d'être exécutés simultanément.

En fait L'OS leur attribue une durée d'exécution à tour de rôle.



Thread

Java possède une classe **Thread** qui permet de créer et gérer les Thread.

Quelques méthodes:

public void start () : L'appel de cette méthode provoque l'exécution de la méthode run() du Thread qui contient le code à exécuter.

public void sleep(long millis) : Interrompt l'exécution du Thread pendant une durée déterminée.



Thread

Création: 1^{ère} méthode

Classe associée au Thread

```
public class MaClasse extends Thread{  
    public void run(){  
        ....//code executé au lancement du Thread  
    }  
}
```

Classe principale

```
public class EssaiThread{  
    public static void main (String [ ]args){  
        MaClasse monObjet = new MaClasse();  
        monObjet.start(); // ! appel non bloquant  
        // la suite  
    }  
}
```



Thread

Création: 1^{ère} méthode

Remarques:

- Le Thread est démarré à l'appel de la méthode `run()` par la méthode `start()`. Il sera arrêté (tué) à la sortie de la méthode `run()`
- Cette méthode a un inconvénient, car l'héritage multiple en Java n'existant pas, la classe du Thread ne peut pas hériter d'une autre classe



Thread

Création: 2ème méthode

Classe associée au Thread

```
public class MaClasse implements Runnable{  
    public void run(){  
        ....//code executé au lancement du Thread  
    }  
}
```

Classe principale

```
public class EssaiThread{  
    public static void main (String [ ]args){  
        MaClasse monObjet = new MaClasse();  
        Thread monThread = new Thread(monObjet);  
        monObjet.start();  
    }  
}
```