

# POCA — TP noté n°3

## Java concurrence

Jean-Baptiste Yunès  
Jean-Baptiste.Yunes@u-paris.fr

21 novembre 2023

### Rendu final

L'intégralité du code produit, les tests, etc doit être fourni à la date indiquée par l'enseignant. Il est important de ne fournir que le code source, et de prendre bien soin que rien ne manque. Le code sera fourni sous la forme d'une archive `tar.gz` ou `zip` (pas autre chose). Le code devra comprendre un fichier `README.txt` contenant le nom et le prénom (dans cet ordre!) de l'élève concerné. Les modalités de rendu seront donnés par l'enseignant mais se feront sur `moodle.u-paris.fr` dans le cours POCA et sous aucune autre forme.

### Généralités

Dans ce TP, on vise à utiliser divers techniques de concurrence pour accélérer un calcul. Le calcul consiste à compter les occurrences d'un mot dans un texte. Pour cela on découpe un fichier en tronçons. Le traitement de chaque tronçon pourra être effectué indépendamment en concurrence. Pour cela il est demandé d'utiliser les classes `FileSplitter`, `Chunk`, `ChunkReader` fournies. `FileSplitter` permet d'obtenir des tronçons, chacun représenté par un `Chunk`. `ChunkReader` permet d'effectuer le calcul du nombre d'occurrence d'un mot dans un `Chunk`. Le tout de la manière suivante :

```
public class Test {
    public static final String fileName = "/tmp/discoursMedium.txt";
    public static final String word = "homme";
    public static void main(String[] args) throws Exception {
        // On découpe en 10 morceaux
        List<Chunk> chunks = FileSplitter.getChunks(fileName, 10);

        // On calcule le nombre d'occurrence pour chaque morceau
        long n = 0;
        for (var c: chunks) {
            n += new ChunkReader(fileName, c, word).compute(true); // print debug
            info
        }

        System.out.println("Le mot "+word+" apparait "+n+" fois dans le fichier "+
            fileName);
    }
}
```

qui produit le résultat suivant (exemple) :

```
+Chunk (0,18673169 : 18673169)
-Chunk (0,18673169 : 18673169) 4668
+Chunk (18673169,37346309 : 18673140)
-Chunk (18673169,37346309 : 18673140) 4663
+Chunk (37346309,56019480 : 18673171)
-Chunk (37346309,56019480 : 18673171) 4589
+Chunk (56019480,74692647 : 18673167)
-Chunk (56019480,74692647 : 18673167) 4673
+Chunk (74692647,93365790 : 18673143)
-Chunk (74692647,93365790 : 18673143) 4575
+Chunk (93365790,112038898 : 18673108)
-Chunk (93365790,112038898 : 18673108) 4668
+Chunk (112038898,130712029 : 18673131)
-Chunk (112038898,130712029 : 18673131) 4663
+Chunk (130712029,149385200 : 18673171)
-Chunk (130712029,149385200 : 18673171) 4589
+Chunk (149385200,168058374 : 18673174)
-Chunk (149385200,168058374 : 18673174) 4673
+Chunk (168058374,186731008 : 18672634)
-Chunk (168058374,186731008 : 18672634) 4575
Le mot homme apparaît 46336 fois dans le fichier /tmp/discoursMedium.txt
```

**Il est demandé de ne pas pas modifier une quelconque de ces classes.** Le code source de ces classes ainsi que le fichier texte sont à récupérer sur moodle ou sur la page de l'enseignant.

## 1 Question n°1

Calculer le temps nécessaire pour compter le nombre de fois qu'apparaît le mot "homme" sans concurrence. Pour cela, utiliser `System.currentTimeMillis()` pour capturer le temps à l'instant  $t$ .

## 2 Question n°2

Écrire un test (classe `Test1` permettant de multithreader le calcul à l'aide d'un `ExecutorService` de type `fixedThreadPool` contenant 10 threads. Calculer le temps.

## 3 Question n°3

Dans une classe `Test2` permettant de multithreader le calcul à l'aide d'un `ExecutorService` de `virtualThreads`. Calculer le temps.

## 4 Question n°4

Dans une classe `Test3` permettant de multithreader le calcul à l'aide de `FutureCompletables` combinés de façon adéquate. Calculer le temps.

## 5 Question n°5

Dans une classe `Test4` permettant de multithreader le calcul à l'aide de `Thread` standards mais en utilisant une `ArrayBlockingQueue` pour récupérer les résultats. Calculer

le temps.

## 6 Question n°6

Dans une classe `Test5` permettant de multithreader le calcul à l'aide d'un `Stream` parallèle de `Chunks`. Calculer le temps.

## 7 Question n°7

Dans une classe `Test6` permettant de multithreader le calcul à l'aide d'une `ForkJoinTask`. Calculer le temps.