

## TP 4 : Trieurs

### 1. Sujet

On souhaite développer une application permettant de tester plusieurs types de trieurs sur des données comparables. Ces données implémentent l'interface de java Comparable.

### 2. Développement des classes d'application

#### 2.1. Interface Trieur

Afin d'intégrer les classes de tri dans une application permettant de les tester de façon évolutive, on définit une interface proposant un schéma commun à tous les trieurs.

Pour cela, on impose aux trieurs de stocker leurs données dans un ArrayList d'éléments comparables.

Écrire l'interface **Trieur** générique d'éléments comparables, disposant d'une méthode **sort** déclenchant le tri et deux accesseurs permettant un **getData** et un **setData** des données du trieur courant.

La méthode sort retourne un entier permettant de comptabiliser le nombre d'itérations (ou d'appels récursifs) pour essayer de comparer les différentes techniques de tri. On peut aussi comptabiliser le nombre de permutations de valeurs, mais dans ce cas ça ne va pas être utilisable pour toutes les techniques de tri.

Pour tester il faut exécuter dans l'ordre **setData** qui permet au trieur de connaître les données à trier, puis **trier** et enfin **getData** pour récupérer les données triées.

#### 2.2. Trieur sélection

Écrire la classe **TrieurSelection** qui implémente l'interface **Trieur** et qui met en oeuvre :

1. les méthodes **sort**, **getData** et **setData**
2. la méthode **toString**, pour permettre l'affichage des données à trier
3. la méthode **swap** qui permute deux éléments à partir de leur index

#### 2.3. Test du trieur

Écrire une classe **Main** qui contient la méthode main et qui permet de :

1. instancier le trieur
2. choisir le nombre d'éléments à trier
3. Initialiser les données à trier de façon aléatoire à l'aide de la classe Random du package java.util
4. lancer le tri

La classe doit pouvoir afficher les données avant et après le tri, ainsi que le temps d'exécution du tri. Pour cela, appeler la méthode **int System.currentTimeMillis()**, qui permet de récupérer la "date" courante en millisecondes, avant et après le tri.

#### 2.4. Trieur bulle

Écrire la classe **TrieurBulle** qui hérite de **TrieurSelection** de façon à réutiliser au maximum le code écrit pour **TrieurSelection**. Cette classe mettra en œuvre le **tri par la méthode des bulles**.

**Tester votre classe.**

#### 2.5. Trieur Shaker

Écrire la classe **TrieurShaker** qui hérite de **TrieurSelection** de façon à réutiliser au maximum le code écrit pour **TrieurSelection**. Cette classe mettra en œuvre le **tri par la méthode du skaker** (tri à bulles dans les deux sens, parcours de l'intervalle des valeurs restant à trier : dans le sens des index croissants pour "descendre" la valeur la plus grande, puis dans le sens des

index décroissants pour "monter" la valeur la plus petite).

**Tester votre classe.**

## 2.6. Trieur pivot

Écrire la classe **TrieurPivot** qui hérite de **TrieurSelection** de façon à réutiliser au maximum le code écrit pour **TrieurSelection**. Cette classe mettra en œuvre le **tri pivot**. Cette technique de tri récursive, partitionne les données en deux sous-ensembles, selon que les valeurs sont  $\leq$  à la valeur du pivot, ou bien  $>$  à la valeur du pivot. A la fin de la partition, la valeur du pivot est à sa place dans la structure de données. Le tri pivot est ensuite relancé récursivement sur les deux sous-ensembles, si leur taille est  $\geq 2$ .

**Tester votre classe.**

## 2.7. Bonus : Trieur fusion

Écrire la classe **TrieurFusion** qui hérite de **TrieurSelection** de façon à réutiliser au maximum le code écrit pour **TrieurSelection**. Cette classe mettra en œuvre le **tri fusion**. Cette technique considère qu'un ensemble de données est trié s'il contient 0 ou 1 élément. Sinon, il est partagé en deux sous-ensembles qui sont triés par l'appel récursif du tri fusion. Au retour des deux appels récursifs, les deux sous-ensembles sont fusionnés de façon à interclasser les éléments.

**Tester votre classe.**

## 2.8. Test des trieurs

Utiliser les deux classes fournies dans Moodle pour **tester votre code** : Main, et TestTrieurs