

TD N°8– M2103

Semaine 13

Objectifs du TD

Utiliser des interfaces

implémenter deux tris simples : tri par sélection, tri par insertion

Rappels de cours : Les interfaces (cours n°8)

- Une interface Java est une spécification d'un type, sous la forme d'un nom de type et d'un ensemble de signatures de méthodes.
- L'interface *Comparable*

1- La classe Pays : utilisation de d'une interface existante

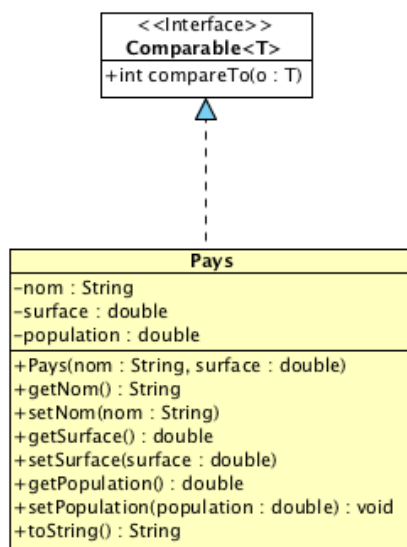
La classe Pays représente des données pour un pays : sa surface, sa population.

Notre objectif est de pouvoir manipuler un ensemble de pays et en particulier de les trier selon la taille de leur surface ou le nombre d'habitants.

Pour comparer un objet de type Pays avec un autre nous allons utiliser l'interface Java **Comparable**.

Cette interface ne spécifie qu'une méthode : **int compareTo(T o)** qui retourne un entier négatif, 0 ou positif selon que l'objet soit plus petit, égal ou plus grand que l'objet spécifié en paramètre.

Le diagramme UML indiquant que la classe Pays va implémenter l'interface est le suivant :



On remarque qu'en UML les méthodes spécifiées dans l'interface n'apparaissent pas dans la boîte de la classe qui implémente l'interface.

On remarque également que pour utiliser Comparable il faut préciser le type (T) des élément à comparer.

Question 1 :

Dans un premier temps on réalisera la comparaison de deux pays par rapport à leur surface.

Donner le code Java de la classe Pays (sans donner le code des accesseurs et modificateurs qui pourra être écrit en TP) .

Question 2 :

Ecrire une classe scénario qui permet de créer deux Pays et de les comparer.

2 - Implémentation de tris simples en Java

L'objectif maintenant est de pouvoir trier un ensemble de Pays avec un tri simple, c'est-à-dire un tri par sélection ou un tri par insertion.

Rappel du principe du tri par sélection :

Sur un tableau de n éléments (numérotés de 1 à n), le principe du tri par sélection est le suivant :

- rechercher le plus petit élément du tableau, et l'échanger avec l'élément d'indice 1 ;
- le second plus petit élément du tableau, et l'échanger avec l'élément d'indice 2 ;
- continuer de cette façon jusqu'à ce que le tableau soit entièrement trié.

Bien sûr si on veut utiliser un tableau Java, les indices iront de 0 à n-1 pour trier n éléments.

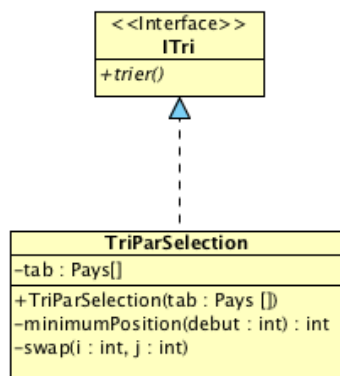
2.1 Définition d'une interface de tri

Comme il existe beaucoup d'algorithmes de tri, nous avons décidé de créer une nouvelle interface **ITri** qui spécifie la méthode **trier()** que l'on veut retrouver dans tous les algorithmes.

- donner le code Java de l'interface ITri représentée dans le diagramme ci-dessous.

2.2 Le tri par sélection en Java

Nous avons également décidé d'appliquer l'algorithme du tri par sélection pour trier des pays. Le diagramme présentant la spécification de notre classe **TriParSelection** est présenté ci-dessus.



Le constructeur reçoit le tableau à trier et le met dans l'attribut **tab** de la classe **TriParSelection**. L'algorithme de tri est réparti dans plusieurs méthodes pour décomposer le problème à résoudre :

- La méthode privée **swap(int i, int j)** échange deux entrées du tableau d'indices i et j.
- La méthode privée **int minimumPosition(int debut)** retourne la position du plus petit élément dans tab[debut] jusqu'à tab[tab.length-1].
- La méthode **trier()** implémente la boucle principale de l'algorithme.

- Donner le code Java de la classe **TriParSelection** .

TP N°8– M2103

Semaine 13

1- Implémentation des classes vues en TD**La classe Pays**

Implémenter en Java la classe **Pays** du TD sans oublier la javadoc

Un tri simple en Java

- 1) implémenter et compiler l'interface **ITri**
- 2) implémenter la classe **TriParSélection** sans oublier la javadoc.

3- Tester le tri simple de Pays

Dans la classe de scénario vous devez créer un tableau de Pays.

Voici un exemple que vous pouvez copier/coller :

```
Pays[] tabPays = new Pays[10];
tabPays [0] = new Pays ("Cuba" , 11423952, 110860);
tabPays [1] = new Pays ("Chile" , 16454143, 756950);
tabPays [2] = new Pays ("Russia" , 140702094, 17075200);
tabPays [3] = new Pays ("Norway" , 46444457, 323802);
tabPays [4] = new Pays ("Nigeria" , 138283240, 923768);
tabPays [5] = new Pays ("Paraguay" , 6831306, 406750);
tabPays [6] = new Pays ("Oman" , 3311640, 212460);
tabPays [7] = new Pays ("Yemen" , 23013376, 406750);
tabPays [8] = new Pays ("Togo" , 5858673, 56785);
tabPays [9] = new Pays ("France" , 64057790 , 643427);
```

Astuce :

Il existe une classe `java.util.Arrays` qui permet d'imprimer un tableau passé en paramètre (regarder l'API Java) .

Il faut penser à importer cette classe et utiliser son message `toString()`.

Exemple :

```
System.out.println (Arrays.toString(tabPays)) ;
```

3- Vers une bibliothèque d'algorithmes

Implémenter d'autres algorithmes de tri que vous avez vu au premier semestre. Vous devez garder le découpage en méthodes utilisé précédemment et bien sûr utiliser l'interface **ITri**.