

TP – Classe Pays

Utilisation d'un TreeSet

Le but du TP est de coder une classe pour représenter un pays. Les informations pertinentes pour un pays seront le nom du pays et l'ensemble de ses pays limitrophes stockés en tant que chaînes de caractères.

Cette classe pourrait par exemple être utilisée pour développer un jeu permettant de tester les connaissances en géographie d'un joueur. Le jeu pourrait lui demander d'établir la liste des pays voisins d'un pays, ou lui faire deviner un pays à partir de ses pays limitrophes.

Fichier à récupérer sur Moodle : *TestPays.java*

Classe Pays

Pour représenter les pays limitrophes, la structure de données la plus adaptées est un ensemble. En effet, un pays ne peut pas apparaître 2 fois en tant que pays voisin. Si l'on se place dans l'optique d'utiliser la classe pour le jeu évoqué ci-dessus, nous serons amenés à effectuer des parcours de l'ensemble des pays limitrophes, pour vérifier qu'il coïncide avec les pays proposés par le joueur. La classe *TreeSet* conviendrait bien pour cet usage.

Les opérations à prévoir dans la classe *Pays* sont les suivantes :

- un **constructeur** avec en argument le nom du pays à créer. Par défaut, l'ensemble des pays limitrophes sera initialisée à vide. Si le nom du pays est invalide, une exception sera levée.
- un **constructeur** avec en argument le nom du pays à créer, et un tableau de chaînes de caractères contenant les noms des pays voisins. Si le nom du pays à créer ou l'un de ses voisins est invalide, une exception sera levée.
- une méthode *toString* qui renverra la description du pays courant : son nom et la liste de ses voisins. Indication : la méthode *toString* appliquée à un objet de type *TreeSet* renvoie une chaîne contenant les éléments de l'ensemble mis bout à bout. Cette représentation conviendra pour la liste des pays voisins.

Exemples de chaînes renvoyées par la méthode *toString* :

France a pour voisin : [Italie, Allemagne, Luxembourg, Suisse, Espagne, Belgique]
Espagne a pour voisin : [France, Portugal]

- une méthode qui aura en paramètre un nom de pays et qui **ajoutera** celui-ci aux pays limitrophes. Si ce pays figure déjà dans cet ensemble, l'opération sera sans effet. Si le nom du pays est invalide, une exception sera levée.
- une méthode qui détermine si le pays dont le nom est donné en paramètre **est un pays limitrophe** du pays courant

- une méthode qui renverra **le nombre de pays limitrophes** du pays courant
- une méthode qui aura en paramètre une liste de noms de pays (de type *ArrayList*) et qui déterminera si **cette liste correspond à l'ensemble des pays limitrophes**.
- une méthode qui aura en paramètre une liste de noms de pays (de type *ArrayList*) et qui déterminera **combien de pays de cette liste sont des pays limitrophes** du pays courant.
- un **accesseur sur le nom** du pays et un autre **accesseur sur l'ensemble des pays limitrophes**

Un nom de pays sera considéré comme valide, s'il n'est pas vide. Dans cette version de la classe, c'est la seule vérification qui est demandée.

Questions

- 1) Commencer à coder la classe **Pays** avec les attributs, les 2 constructeurs et la méthode *toString*.
- 2) A l'aide de la classe **TestPays**, tester ces méthodes.
- 3) Coder la méthode permettant d'ajouter un pays limitrophe et la tester avec la méthode prévue dans la classe **TestPays**.
- 4) Coder la méthode permettant de déterminer si le pays donné en argument fait partie des pays limitrophes du pays courant. Tester avec la méthode prévue dans la classe **TestPays**.
- 5) Coder la méthode permettant de déterminer le nombre de pays limitrophes du pays courante. Tester avec la méthode prévue dans la classe **TestPays**.

- 6) Coder la méthode permettant de déterminer si la liste des pays donnée en paramètre coïncide avec l'ensemble des pays limitrophes du pays courant. Par exemple, si le pays courant est la France, on considèrera que les listes suivantes coïncident :

```
[Italie, Allemagne, Luxembourg, Suisse, Espagne, Belgique]
[Iталie, Belgique, Allemagne, Luxembourg, Suisse, Espagne]
[Iталie, Belgique, Allemagne, Luxembourg, Suisse, Espagne, Belgique]
```

Par contre, les suivantes ne coïncideront pas :

```
[Italie, Allemagne, Luxembourg, Suisse, Espagne]
[Iталie, Allemagne, Luxembourg, Suisse, Espagne, Belgique, Pologne]
```

Tester avec la méthode prévue dans la classe **TestPays**.

- 7) Coder la méthode permettant de déterminer combien il y a de pays communs entre la liste des pays donnée en paramètre et l'ensemble des pays limitrophes du pays courant. Par exemple, si le pays courant est la France :

```
[Italie, Allemagne, Luxembourg, Suisse, Espagne, Belgique]    => 6 pays
[Iталie, Belgique, Allemagne, Luxembourg, Suisse, Espagne]    => 6 pays
[Iталie, Belgique, Allemagne, Luxembourg, Suisse, Espagne, Belgique] => 6 pays
[Iталie, Allemagne, Luxembourg]                                => 3 pays
[Iталie, Pologne, Allemagne, Luxembourg, Roumanie]            => 3 pays
```

Tester avec la méthode prévue dans la classe **TestPays**.

- 8) Ajouter les accesseurs et les tester.