

# TD5 – Streams

Inspiré du TD créé par Frédéric Rallo & Peter Sander

This assignment will give you practice with streams.

**Part 1: Beginning with a list of restaurants.** \_\_\_\_\_ **1**

**Part 2 : Towards a Scrabble** \_\_\_\_\_ **2**

## Part 1: Beginning with a list of restaurants.

Dans cette partie, vous allez apprendre en regardant quand cela vous est nécessaire les solutions qui vous sont données.

1. Placez le fichier « mcdonalds.csv » dans le « resource folder »
2. Créez une classe **Restaurant** qui correspond au contenu du fichier : latitude, longitude, ville, état, etc.
3. Créez une classe **McDoReader** telle que le constructeur par défaut crée une `List<Restaurant>` à partir de la lecture du fichier *mcdonalds.csv*
4. Définir les méthodes qui répondent aux questions suivantes :
  1. Quel est le nombre de villes distinctes qui disposent d'au moins un restaurant McDonald ?
  2. Afficher les restaurants dans une ville donnée
  3. Quel est le nombre de MacDo dans une ville donnée ?
  4. Afficher le nombre de MacDo dans un état donné
  5. Afficher les villes distinctes d'un état dans l'ordre alphabétique qui ont des MacDo
  6. Dans quelle ville se trouve le plus de restaurants McDonald ?
  7. Donner le nom d'une ville qui a le moins de MacDo.
  8. Quelles sont les 4 villes avec le plus de MacDo et le nombre de restaurants dans ces villes ?
  9. Afficher la liste des villes qui ont exactement un restaurant McDonald triée par ordre alphabétique inverse et limitée à 15.

**Voici une trace possible.**

*The number of cities that have a McDonald : 4926*

*The city that have the most MacDo restaurants : Houston*

*The top 4 cities with the most MacDo : [Houston=135, Chicago=124, Las Vegas=84, Los Angeles=80]*

*One city with fewest restaurants : Coshocton*

*Short List of cities with one McDo : [zionsville, zion, zillah, zeeland, zebulon, zapata, zachary, yucaipa, yreka, york county, yerington, yemassee, yelm, yazoo city, yarmouth]*

*Number of McDo in Oxford : 8*

*Restaurants in Oxford : [Restaurant{latitude=-85.83459, longitude=33.60032, name='McDonalds-Oxford in AL', address='92 Plaza Lane [WM #0809] ', city='Oxford', state='AL', phone='(256) 835-7912'},*

*Restaurant{latitude=-85.82799, longitude=33.60845, name='McDonalds-Oxford in AL', address='1015 Hwy 21 S', city='Oxford', state='AL', phone='(256) 831-7561'}, ...*

*Number of McDo in Coshocton : 1*

*Restaurants in Coshocton : [Restaurant{latitude=-81.87034, longitude=40.2751, name='McDonalds-Coshocton in OH', address='101 Chestnut St', city='Coshocton', state='OH', phone='(740) 622-4300'}]*

*The cities in AK : [Anchorage, Eagle River, Fairbanks, Homer, Kenai, North Pole, Palmer, Soldotna, Wasilla]*

*Number of McDo in AK : 25*

## Part 2 : Towards a Scrabble

*Bien évidemment dans cet exercice, vous utilisez des Stream autant que possible.*

*On désire savoir si Shakespeare était un bon joueur de scrabble (anglais)!*

*Plus précisément, on va s'intéresser au fichier contenant l'intégralité du vocabulaire utilisé dans son œuvre.*

1. Téléchargez les fichiers suivants à l'adresse :  
<https://introcs.cs.princeton.edu/java/data/> (Ils vous sont donnés aussi)
  1. [words.shakespeare.txt](#) (renommez-le shakespeare.txt)
  2. [ospd.txt](#) (le dictionnaire des mots anglais autorisés au scrabble)
2. Placez ces fichiers dans le « resource folder »
3. Définissez la classe **EnglishScrabbleUtil**
  1. Elle contient les informations propres à la version anglaise du jeu, d'une part le nombre de points que valent chaque lettre (par exemple 1 point pour a, 10 points pour z) et d'autre part la distribution des lettres dans le jeu (par exemple, 9 a, 1 z)

```
static final int[] scrabbleENScore = {  
    // a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z  
    1, 3, 3, 2, 1, 4, 2, 4, 1, 8, 5, 1, 3, 1, 1, 3, 10, 1, 1, 1, 1, 4, 4, 8, 4, 10 };  
  
private static final int[] scrabbleENDistrib = {  
    // a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z  
    9, 2, 2, 1, 12, 2, 3, 2, 9, 1, 1, 4, 2, 6, 8, 2, 1, 6, 4, 6, 2, 2, 1, 2, 1 };
```

2. Enregistrer dans *EnglishScrabbleUtil*, la liste des mots autorisés (ils sont définis dans [ospd.txt](#))
3. Définissez une méthode qui, à partir d'une lettre vous donne sa distribution dans le jeu. Testez-la.
4. Définissez une méthode qui, à partir d'une lettre vous donne son nombre de points dans le jeu. Testez-la.
5. Définissez une méthode qui vérifie si un mot est autorisé ou non. Testez-la.
6. Définissez une méthode qui calcule le nombre de points d'un mot. Testez-la.
  - a. Vous aurez peut-être besoin de `chars()` qui appliquée à une string renvoie une stream of Char (`IntStream`)
  - b. Bien évidemment, implémentez cette méthode avec une stream !
7. Définissez une méthode qui vérifie qu'une String peut bien être écrite avec la distribution des lettres, en utilisant une stream!
8. Testez votre classe bien évidemment.

### Exemple de tests à adapter à votre code :

```
assertEquals(0, EnglishScrabbleUtil.computeScoreOfWord(""));  
assertEquals(0, EnglishScrabbleUtil.computeScoreOfWord(" "));  
assertEquals(8, EnglishScrabbleUtil.computeScoreOfWord("Hello"));  
assertEquals(29, EnglishScrabbleUtil.computeScoreOfWord("Buzzards"));  
assertEquals(87, EnglishScrabbleUtil.computeScoreOfWord("abcdefghijklmnopqrstuvwxyz"));  
  
assertTrue(EnglishScrabbleUtil.isWordAllowed("Hello"));  
assertTrue(EnglishScrabbleUtil.isWordAllowed("inspirit"));  
assertTrue(EnglishScrabbleUtil.isWordAllowed("Buzzards"));  
assertFalse(EnglishScrabbleUtil.isWordAllowed("abcdefghijklmnopqrstuvwxyz"));
```

```
assertTrue(EnglishScrabbleUtil.isWordAvailable("abcdefghijklmnopqrstuvwxy"));
assertTrue(EnglishScrabbleUtil.isWordAvailable("hello"));
assertFalse(EnglishScrabbleUtil.isWordAvailable("Buzzards"));
```

Attention, cette classe va être très largement utilisée. Si vous n'avez pas confiance dans vos codes, si vous avez des bugs, il vous sera difficile de déboguer dans les streams sur des volumes un peu importants de données ! Pensez à tester aux limites.

4. Définissez la classe **Shakespeare** (attention vous devez définir seul les variables nécessaires, les constructeurs, etc.)
  1. Mémorisez les mots utilisés par Shakespeare en les extrayant de « shakespeare.txt »
  2. Associez à chaque score, les mots qui ont ce score
    - a. Retournez le résultat sous la forme d'un Dictionnaire clé=score / valeur=liste de mots.
  3. Évaluez la différence entre une utilisation de *parallel stream* ou non
    - a. Commencer par enregistrer le temps :
      - **long** start = System.currentTimeMillis();
    - b. Exécutez la même action au moins 100 fois pour lisser les différences
    - c. Enregistrez le temps final
    - d. Calculez et affichez la différence
    - e. Comparer les temps obtenus par l'utilisation des deux types de streams.
  4. Calculez le nombre de mots par score dans l'œuvre de Shakespeare.
    - a. Retournez le résultat sous la forme d'un Dictionnaire clé=score / valeur=nombre de mots.
  5. Retournez tous les mots avec leur score, dont le score est supérieur à un score en paramètre par exemple 36
    - a. Retournez le résultat sous la forme d'une liste de Pair (mot, score)
  6. Affichez les mots qui existent dans le dictionnaire officiel et dont le score est supérieur à un nombre qu'on passera en paramètre (par exemple >28)
  7. Déterminer le nombre de mots dans l'œuvre de Shakespeare qui
    - a. Ne peuvent pas être écrit avec la distribution des lettres du scrabble ;
    - b. Ne sont pas des mots acceptés ;
    - c. Peuvent être écrits et sont acceptés.
    - d. Déduire le nombre de mots de son vocabulaire, avec lesquels on ne peut pas jouer.
5. Pour jouer, à présent, nous devons tenir compte de nouvelles informations. Il est de votre responsabilité de choisir comment vous les implémentez et dans quelle classe.
  1. Au scrabble, il existe deux « blancs » qui peuvent prendre la valeur de la lettre de son choix et avec un score de 0.
    - a. Implémenter une méthode qui renvoie le nombre de blancs nécessaires pour écrire un mot, par exemple pour "buzzards", il vous faut un blanc de plus car il n'y a qu'un z dans la distribution.
      - TIP : Vous aurez besoin pour calculer le score d'un mot qui contient des blancs de savoir combien il y a de chacune des lettres, par exemple 2 z dans buzzards. Pensez à définir une méthode intermédiaire qui vous donne cette information.
    - b. Implémenter une méthode qui calcule le score d'un mot selon le principe suivant :
      - Le mot doit être autorisé
      - La distribution des lettres doit permettre de l'écrire, éventuellement en utilisant un ou deux blanc,

- Si les deux conditions précédentes ne sont pas vérifiées le score est de 0.
- Un blanc à une valeur de 0.

Par exemple le mot "buzzards" a un score de 19

2. Affichez les mots de l'œuvre de Shakespeare dont la distribution des jetons du jeu de scrabble en anglais permet d'écrire et dont le score est supérieur à un nombre qu'on passera en paramètre (par exemple >24)
3. Amusez-vous.