

TP n°4

Introduction à JFlex (encore)

Comme lors des séances précédentes, le filtrage s’effectue uniquement à l’aide d’expressions rationnelles jflex : **on n’aura ainsi besoin d’aucun test (if) ni d’aucune boucle (for, while) dans le code Java.**

Exercice 1 – Le fichier `ratp.csv` contient des informations concernant le réseau Métro/RER francilien. Chaque ligne du fichier correspond à une station et possède 10 “colonnes” séparées par des ; (point-virgule). Ce symbole est donc essentiel pour bien former vos expressions rationnelles (le ; ne doit pas être échappé en jflex).

- La 1e colonne contient la mention METRO ou RER ;
- La 2e colonne contient le nom de la station ;
- La 3e colonne contient la fréquentation de la station ;
- Chaque colonne entre la 4e et la 8e contient (éventuellement) l’identifiant (numéro ou lettre) d’une ligne passant par la station ;
- La 9e colonne contient la ville où est située la station ;
- La 10e colonne contient l’arrondissement où est située la station (pour celles situées à Paris).

Voici le début du fichier en version tableur et en version texte.

reseau	station	trafic	corr_1	corr_2	corr_3	corr_4	corr_5	ville	arrdt
METRO	HOTEL DE VILLE	12307363	1	11				PARIS	4
METRO	ESPLANADE DE LA DEFENSE	9883008	1	A				PUTEAUX	
METRO	FRANKLIN D. ROOSEVELT	9545487	1	9				PARIS	8

```
reseau;station;trafic;corr_1;corr_2;corr_3;corr_4;corr_5;ville;arrdt
METRO;HOTEL DE VILLE;12307363;1;11;;;PARIS;4
METRO;ESPLANADE DE LA DEFENSE;9883008;1;A;;;PUTEAUX;
METRO;FRANKLIN D. ROOSEVELT;9545487;1;9;;;PARIS;8
```

On note que, pour certaines questions, il peut être nécessaire d’extraire un lexeme plus grand que le motif recherché, par exemple une ligne toute entière qui contient le motif. Dans ce cas, il peut être utile de rappeler que le symbole `^` dénote le début de la ligne et le symbole `$` dénote sa fin.

Par ailleurs pour extraire un champ particulier d’une ligne (ou portion de ligne) séparée par des ;, une fonction `String cut(String s, int col)` est fournie : un appel `cut(s, col)` renvoie le champs d’indice `col` de la chaîne `s` (le premier champ porte l’indice 0).

Les cinq questions sont indépendantes.

1. Compléter le fichier `exo1_1.jflex` pour obtenir un analyseur syntaxique qui compte les stations situées dans le 1er arrondissement.

↪ vous devez obtenir 8.

Ensuite, dans le même fichier, mettre en commentaire la ligne de la réponse pour le comptage (sans l’effacer !) et modifier le fichier de sorte que l’analyseur syntaxique *affiche les noms* de ces stations ainsi que la moyenne de leurs fréquentations.

Pour chacune des questions $i \in \{2, 3, 4, 5\}$, compléter d'abord la spécification (`exo1_i.jflex`) de sorte que l'analyseur syntaxique *compte* les stations demandées. Ensuite, dans le même fichier, mettre en commentaire la ligne de la réponse qui effectue le comptage (sans l'effacer) et modifier la spécification de sorte que l'analyseur syntaxique *affiche les noms* de ces stations (pas de moyenne demandée pour ces questions).

2. Les stations du réseau Métro avec au moins une correspondance avec le RER.

→ au comptage vous devez obtenir 11.

3. Les stations ayant une fréquentation $\geq 10M$ et $< 15M$.

→ au comptage vous devez obtenir 7.

4. Les stations dont le nom est constitué d'un seul mot qui contient au plus trois consonnes. Attention au mot "PARIS" dans la colonne "ville", il contient au plus trois consonnes.

→ au comptage vous devez obtenir 38.

5. Les stations ayant exactement une correspondance (exactement deux lignes passent par la station). Commencer par définir une macro `ligne` correspondant à l'ensemble

$$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, A, B, C, D, 3bis, 7bis\}$$

en utilisant le moins possible de symboles `|` (union), et idéalement pas plus de deux fois.

→ au comptage vous devez obtenir 42.

Exercice 2 – On considère des phrases (comme celles du fichier `texte.txt` fourni) dont la structure est

`<article> <nom> <verbe> <complément>`

où `<verbe>` désigne un verbe du premier groupe conjugué au présent de l'indicatif. Pour simplifier, on ne considère que des caractères majuscules.

Chaque question i correspond à une action que doit effectuer l'analyseur syntaxique. Pour chacune, dessiner l'automate (avec `YYINITIAL` comme état initial) et écrire la spécification dans le fichier `exo2_i.jflex`.

1. Substitution du sujet `<article> <nom>` par la chaîne de caractères "Qui" et du point final par un point d'interrogation.

→ Qui emprunte le chemin de halage ?

→ Qui marchent à pas lents et feutrés ?

2. Conjugaison du verbe `<verbe>` au futur (en plus de la substitution traitée à la question 1, mais légèrement modifiée).

→ Qui donc empruntera le chemin de halage ?

→ Qui donc marcheront à pas lents et feutrés ?

3. Modalisation du `<verbe>` par l'auxiliaire *devoir* (en anglais, on utiliserait *have to* ou *must*, etc).

→ La cycliste doit emprunter le chemin de halage.

→ Des loups doivent marcher à pas lents et feutrés.

4. Renversement avec préfixage du `<complément>` (en plus de la modalisation du verbe traitée en 3).

→ Le chemin de halage, la cycliste doit emprunter.

→ À pas lents et feutrés, des loups doivent marcher.