Projet Structures de données : avancé

1 Objectif

Le fichier `flight.xml` contient des données à propos du trafic aérien. Ce fichier contient des informations sur des aéroports, des compagnies aériennes et des vols.

Pour ce projet, en utilisant ce fichier xml, nous vous demandons d'implémenter un programme java qui permettra de calculer un itinéraire pour aller d'un aéroport à un autre. Vous devrez implémenter deux façons de calculer un itinéraire :

1. qui minimise le nombre de vols,
2. qui minimise le nombre de kilomètres parcourus.

Votre programmera sauvegardera les itinéraires calculés dans un fichier au format xml. Ci-dessous, voici un exemple d'itinéraire minimisant le nombre de vol entre l'aéroport de Bruxelles et l'aéroport de Papeete (Faa'a International Airport). Cet itinéraire passe par Milan et Tokyo.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<trajet depart="Brussels Airport" destination="Faa'a International Airport"
        distance="19856.873091749196">
        <vol compagnie="Brussels Airlines" nombreKm="664.6542978073214"
                numero="1">
                <source iataCode="BRU" pays="Belgium" ville="Brussels">Brussels Airport
                </source>
                <destination iataCode="MXP" pays="Italy" ville="Milano">Malpensa
                        International Airport</destination>
        </vol>
        <vol compagnie="Alitalia" nombreKm="9748.034581687887" numero="2">
                <source iataCode="MXP" pays="Italy" ville="Milano">Malpensa
                        International Airport</source>
                <destination iataCode="NRT" pays="Japan" ville="Tokyo">Narita
                        International Airport</destination>
        </vol>
        <vol compagnie="Japan Airlines" nombreKm="9444.184212253987"
                numero="3">
                <source iataCode="NRT" pays="Japan" ville="Tokyo">Narita International
                        Airport</source>
                <destination iataCode="PPT" pays="French Polynesia" ville="Papeete">
                        Faa'a International Airport</destination>
        </vol>
</trajet>
```

Voici un exemple d'itinéraire minimisant le nombre de km parcouru en avion entre Bruxelles et Papeete. Cet itinéraire passe par Manchester, Las Vegas et Los Angeles.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<trajet depart="Brussels Airport" destination="Faa'a International Airport"
        distance="15692.738812872138">
        <vol compagnie="United Airlines" nombreKm="535.6183574786736"
                numero="1">
                <source iataCode="BRU" pays="Belgium" ville="Brussels">Brussels Airport
                </source>
                <destination iataCode="MAN" pays="United Kingdom" ville="Manchester">Manchester
                        Airport</destination>
        </vol>
        <vol compagnie="Condor Flugdienst" nombreKm="8166.04679602851"
                numero="2">
                <source iataCode="MAN" pays="United Kingdom" ville="Manchester">Manchester
                        Airport</source>
                <destination iataCode="LAS" pays="United States" ville="Las Vegas">McCarran
                        International Airport</destination>
```

```
        </vol>
        <vol compagnie="Delta Air Lines" nombreKm="379.9759915689007"
                numero="3">
                <source iataCode="LAS" pays="United States" ville="Las Vegas">McCarran
                        International Airport</source>
                <destination iataCode="LAX" pays="United States" ville="Los Angeles">Los
                        Angeles International Airport</destination>
        </vol>
        <vol compagnie="Air Tahiti Nui" nombreKm="6611.097667796053"
                numero="4">
                <source iataCode="LAX" pays="United States" ville="Los Angeles">Los Angeles
                        International Airport</source>
                <destination iataCode="PPT" pays="French Polynesia" ville="Papeete">
                        Faa'a International Airport</destination>
        </vol>
</trajet>
```

On remarque bien que ce deuxième itinéraire contient plus de vols (4 au lieu de 3) mais parcourt une distance moindre (15692 km au lieu de 19856 km).

2 Sur Moodle

Pour mener à bien votre projet, nous vous fournissons plusieurs fichiers :

- `flight.xml` contenant les informations sur les vols.
- `flight.xsd`, un schéma xml permettant de valider `flight.xml`.
- la classe `Util.java` possédant une méthode statique permettant de calculer la distance entre deux coordonnées géographiques. Une coordonnée géographique est composée d'une longitude et d'une latitude.
- une classe `Main.java` que vous ne pouvez pas modifier. Le code de cette classe est présenté ci-dessous. La classe à construire `Graph` devra contenir deux méthodes pour calculer un itinéraire. Ces deux méthodes prennent trois paramètres. Les deux premiers paramètres sont les codes iata des aéroports source et destination. Un code iata permet d'identifier un aéroport. Le troisième paramètre est le nom du fichier où il faudra sauvegarder l'itinéraire.

```java
import java.io.File;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

public class Main {
        public static void main(String[] args) {
                try {
                        File inputFile = new File("flight.xml");
                        SAXParserFactory factory = SAXParserFactory.newInstance();
                        SAXParser saxParser = factory.newSAXParser();
                        SAXHandler userhandler = new SAXHandler();
                        saxParser.parse(inputFile, userhandler);
                        Graph g = userhandler.getGraph();
                        g.calculerItineraireMiniminantDistance("BRU", "PPT", "output.xml");
                        g.calculerItineraireMinimisantNombreVol("BRU","PPT", "output2.xml");
                } catch (Exception e) {
                        e.printStackTrace();
                }
        }
}
```

3 Tâches à effectuer

Nous vous demandons de rendre la classe `Main` fonctionnelle en réalisant les différentes tâches suivantes :

- Implémenter un parseur SAX permettant la lecture du fichier XML et la construction du graphe en mémoire en utilisant les structures de données adéquates.
- Implémenter les deux algorithmes pour calculer les itinéraires en utilisant la théorie sur les graphes. S'il est impossible d'aller de l'aéroport source à l'aéroport destination, votre programme lancera une exception.
- Sauvegarder ces itinéraires dans des fichiers XML.

De plus, nous vous demandons d'écrire la DTD `flight.dtd` qui permet de valider les mêmes documents que `flight.xsd` (tout en étant plus permissive évidemment).

4 Tâches bonus

Si vous avez fini trop tôt et que vous vous embêtez durant les séances, vous pouvez faire une ou plusieurs tâches Bonus. L'objectif est de faire le projet durant les séances, ne prenez pas du temps à la maison pour faire ces tâches bonus ; privilégiez les autres cours comme le projet PAE. On ne s'attend pas à que vous fassiez ces tâches bonus.

Voici les différentes tâches Bonus :

1. Implémentez un parseur DOM pour créer le graphe.
2. Donnez la DTD permettant de valider les itinéraires calculés. Pour ajouter la dtd dans un fichier xml, voici une petite aide : https://stackoverflow.com/questions/13553614/how-to-add-doctype-in-xml-document-using-dom-java
3. Ajoutez une troisième façon de calculer un itinéraire en minimisant le nombre de compagnies aériennes différentes utilisées dans un itinéraire. Si deux itinéraires utilisent le même nombre de compagnies, privilégiez l'itinéraire parcourant le moins de km.

5 Organisation et livrables

Ce projet se déroule du 5 mars 2018 au 23 mars 2018 par groupe de deux étudiants. Vous composerez vous-même les groupes en respectant la règle que tous les étudiants d'un même groupe doivent être dans la même série. Vous nous communiquerez la composition des groupes au début de la semaine du 5 mars. Durant tout le projet, la présence aux cours est obligatoire.

Si une série comporte un nombre impair d'étudiant, nous accepterons un seul groupe de 3 étudiants. Ce groupe de 3 étudiants devra également faire les deux premières tâches bonus.

Le projet est à remettre via « Moodle » pour le vendredi 23 mars 2018 à 23h59. Nous ne demandons pas de rapport. Si certaines choses méritent des explications, vous pouvez les fournir en commentaire dans les différents fichiers.