

### 1. Partie Orientée-objet (Java) [8 points][8% de votre note finale]

*We ask you to implement the DBSCAN algorithm in order to cluster the various trip records using the GPS coordinates of the starting points. Your program must be a Java application, named `TaxiClusters`, that is run by specifying the dataset filename and the values of the parameters `minPts` and `eps`. This program should produce as output the list of clusters contained in a csv file specifying, for each cluster, its position (average value of the GPS coordinates of its point set) and the number of points it contains. The outlier points are discarded.*

*Since this dataset is very large, we give you a reduced version of it containing all the trip records for January 15, 2009 between 12pm and 1pm.*

*Since this solution must follow the object-oriented paradigm, your program must be composed of a set of classes. Specifically, it must include, among others, the following classes:*

- *class `GPScoord`*
- *class `TripRecord` having the following attributes:*
  - *`pickup_DateTime` (String)*
  - *`pickup_Location` (GPScoord)*
  - *`dropoff_Location` (GPScoord)*
  - *`trip_Distance` (float)*
- *class `Cluster`*

*You will find on the Web, numerous implementations of the DBSCAN algorithm. You can use them to help you in the design of your solution but you must specify the source of all code you are using and for which you are not the author (use commented blocks). The document to be submitted must also include all references used to build your solution.*

*In addition to the source code of your solution, you must also submit a document that includes a UML diagram of all your classes (showing attributes, associations and methods). Do not use static methods, except for the `main` function.*

*You must use the object-oriented paradigm.*

On vous demande de programmer l'algorithme DBSCAN afin de grouper les différents enregistrements en utilisant les coordonnées GPS des points de départ. Votre programme doit être une application Java appelée `TaxiClusters` prenant en paramètre le nom du fichier contenant la base de données à analyser, suivi des paramètres `minPts` et `eps`. Le programme produira en sortie la liste des groupes dans un fichier csv donnant, pour chaque groupe, sa position (valeur moyenne des coordonnées de ses points) et son nombre de points. Les points isolés sont ignorés.

La base de données étant immense, nous vous donnons accès à une version réduite contenant les courses du 15 janvier 2009 entre 12h00 et 13h00.

Puisque cette solution doit être orientée objet, votre programme doit être un ensemble de classes; plus spécifiquement, il doit, parmi d'autres, inclure les classes suivantes :

- une classe `GPScoord`
- une classe `TripRecord` avec les attributs suivants :
  - `pickup_DateTime` (`String`)
  - `pickup_Location` (`GPScoord`)
  - `dropoff_Location` (`GPScoord`)
  - `trip_Distance` (`float`)
- une classe `Cluster`

Vous trouverez sur le Web de nombreuses implémentations de l'algorithme DB-SCAN. Vous pouvez vous inspirer de ces solutions mais vous devez clairement identifier dans votre code source la provenance de tout code dont vous n'êtes pas l'auteur (en utilisant des blocs commentés). Le document à soumettre doit aussi inclure des références à toute source d'information utilisée pour réaliser votre solution.

En plus de votre code source, vous devez aussi soumettre un diagramme UML de vos classes, montrant bien les associations, les attributs et les méthodes. Vous ne devez pas utiliser de méthodes statiques (sauf pour la fonction `main`).

Vous devez utiliser le paradigme orienté objet sans quoi vous serez pénalisé.