

EXERCICE RECENSEMENT POPULATION - INTRO

Objectif du TP

Maintenant que vous mis en oeuvre les quelques lignes de code qui permettent de lire un fichier, nous allons passer à l'exploitation des données qu'il contient !

Le but de cet TP va être de créer une application simple qui permette d'exploiter les données de ce fichier pour afficher par exemple :

- 1. Population d'une ville donnée
- 2. Population d'un département donné
- 3. Population d'une région donnée
- 4. Afficher les 10 régions les plus peuplées
- 5. Afficher les 10 départements les plus peuplés
- 6. Afficher les 10 villes les plus peuplées d'un département
- 7. Afficher les 10 villes les plus peuplées d'une région
- 8. Afficher les 10 villes les plus peuplées de France
- 9. Sortir

Description du TP Recensement Population

- Creez le package **fr.diginamic.recensement**.
- Creez une classe **exécutable Application** qui sera notre point d'entrée
- Pour réaliser cette application, il existe de nombreuses possibilités avec leurs avantages et leurs inconvénients. Nous vous proposons plusieurs options. A vous de choisir.
- **Modèle objet simple**
 - Toutes les villes de notre fichier seront stockées dans une `ArrayList<Ville>`
 - La classe **Ville** a les attributs suivants :
 - code région
 - nom de la région
 - code département
 - code de la commune
 - nom de la commune
 - population totale

- Dans ce cas la première étape va être de lire le fichier et de s'en servir pour construire une liste contenant 35800 instances de villes.
- Etape 1 :
 - Dans la classe **Application**, mettez en place le code de lecture du fichier, comme vous l'avez vu dans le TP guidé.
 - Pour chaque ligne contenu dans le fichier, que vous récupérez sous forme de String :
 - découpez la ligne en tableaux de chaînes de caractères (cf. SUPPORT page 3 en cas de blocage),
 - servez vous en pour créez une instance de Ville.
 - *Attention certaines infos dans le fichier ne sont pas forcément pertinentes, comme le code canton par exemple.*
 - placez cette instance dans la liste
- Etape 2 :
 - Redéfinissez la méthode **toString()** dans la classe **Ville**, ce qui servira par la suite lorsqu'on souhaitera afficher une ville.
- Etape 3 :
 - Recherchez dans la liste la ville de Montpellier et affichez toutes les informations la concernant
- Etape 4 :
 - Exploitez les données dont vous disposez pour afficher la population de tout le département de l'Hérault.
- Etape 5 :
 - Affichez la plus petite ville du département
- Etape 6 :
 - Affichez les 10 plus grandes villes du département
 - Affichez les 10 plus petites villes du département
- Etape 7 :
 - Affichez la population de toute la région Occitanie
- Etape 8 (plus difficile):
 - Affichez les 10 villes les plus importantes de la région Occitanie
 - Affichez le département le plus peuplé de la région Occitanie
- Etape 9 (plus difficile) :
 - Affichez les 10 régions les plus peuplées de France
 - Affichez les 10 département les plus peuplés de France
 - Affichez les 10 villes les plus peuplées de France.

Support :

Si vous êtes bloqué à l'étape de parsing du fichier, voir le code ci-après

Si vous ne voyez pas comment faire du comptage, pensez à consulter le document appelé « Java POO – Support comptage »

Commitez vos développements sur GitHub

ANNEXES

Traiter une ligne de données du recensement :

```
// On commence par découper la ligne en morceaux sur la base du caractère
séparateur « ; » . De plus on ne récupère que les morceaux qui nous intéressent. En
l'occurrence on ignore les morceaux 3 et 4 dont on a pas besoin dans le TP
String[] morceaux = ligne.split(";");
String codeRegion = morceaux[0];
String nomRegion = morceaux[1];
String codeDepartement = morceaux[2];
String codeCommune = morceaux[5];
String nomCommune = morceaux[6];
String population = morceaux[7];

// Pour la population, avant la conversion en int, il faut d'abord supprimer les
// espaces qui se trouvent à l'intérieur.
int populationTotale = Integer.parseInt(population.replace(" ", "").trim());

// On cree enfin la ville avec toutes les données utiles
Ville ville = new Ville(codeRegion, nomRegion, codeDepartement, codeCommune,
nomCommune, populationTotale);
```

Pour les opérations de comptage, par exemple l'affichage des 10 régions les plus peuplées :

- pensez à consulter le document « Java POO – Support comptage »
- pensez à revoir le cours sur les opérations de tri.