IUT2 Université Grenoble Alpes — Département INFO

R1.01 – TP6(A)
Recherche séquentielle dans un vecteur en ArrayList

L'objectif de ce TP est de :
- *parfaire la déclaration et l'utilisation de vecteurs en ArrayList*
- *définir un ordre sur des objets instanciant une classe*
- *appliquer (et assimiler) les algorithmes de recherche séquentielle dans un vecteur trié*

CONTEXTE : PAYS DU MONDE (STATISTIQUES 2019)

Dans ce TP, un pays est représenté par son *nom*, le *continent* où il se situe, son *nombre d'habitants* et sa *superficie*.

Vous allez manipuler des `ArrayList` contenant des objets représentant des pays du monde : ces objets sont des instances d'une classe `Pays` que vous allez compléter.

Nous vous fournissons dans un dossier `TP6A_Files` :
- un fichier texte nommé *Monde2019.txt* qui contient les caractéristiques de 253 pays
- une classe java `InitMonde` qui contient une fonction permettant de générer un `ArrayList<Pays>` intialisé avec les données du fichier *Monde2019.txt*
- une classe java `Pays` dont le code est à compléter

Avant de commencer...

- Ouvrez un terminal et placez-vous dans votre répertoire `R1.01`
  - ✓ Exécutez la commande : `cp -r /users/info/pub/1a/R1.01/TP6A_Files .`
  - ✓ Lancez **IJ** et créez un projet `TP6_A`
- Ouvrez le dossier `TP6A_Files` que vous avez copié dans votre répertoire R1.01 et effectuez les actions suivantes :
  1. **copie dans votre projet** `TP6_A` **du fichier** *Monde2019.txt*
     - copiez le fichier *Monde2019.txt* (clic sur le fichier + CTRL + C)
     - dans la fenêtre de votre projet TP6_A, sélectionnez avec le bouton droit de la souris, l'icône du projet 
     - collez le contenu du presse-papiers (CTRL + V) puis validez (clic sur OK)
  2. **copie dans le répertoire** `src` **de votre projet** `TP6_A` **de la classe java** `InitMonde`
     - copiez le fichier `InitMonde.java` (clic sur le fichier : CTRL + C)
     - dans la fenêtre de votre projet TP6_A, sélectionnez avec le bouton droit de la souris, l'icône du répertoire src
     - collez le contenu du presse-papiers (CTRL + V) puis validez (clic sur OK)
  3. **copie dans le répertoire** `src` **de votre projet** `TP6_A` **de la classe java** `Pays`
     - copiez le fichier `Pays.java` (clic sur le fichier + CTRL + C)
     - dans la fenêtre de votre projet TP6_A, sélectionnez avec le bouton droit de la souris, l'icône du répertoire src
     - collez le contenu du presse-papiers (CTRL + V) puis validez (clic sur OK)

1. Complétion de la classe Pays

On définit un ORDRE NATUREL sur la classe **Pays** :
Les objets de type **Pays** sont ordonnés sur le nom des pays qu'ils représentent

1.1. Dans le projet TP6_A, complétez la classe `Pays` qui implémente l'interface `Comparable` pour le type `Pays` (cf. cours 6 – Partie 1 : pages 9 et suivantes)

- Complétez le **constructeur** :
```
public Pays(String nom, String continent, int population, int superficie) {
```
- Complétez les **getters** : `getNom()` / `getContinent()` / `getPopulation()` / `getSuperficie()`
- Complétez la fonction de comparaison entre les objets de type `Pays` (cf. encadré ci-dessus)

2. Initialisation et affichage d'un vecteur de Pays

2.1. Créez une classe `Monde` et ajoutez-y une procédure `main` dans laquelle vous :

- copierez la déclaration : `final ArrayList<Pays> leMonde = InitMonde.creerMonde();`
- écrirez les instructions nécessaires à l'affichage ligne à ligne, des pays du vecteur **`leMonde`**

  NOTE : la méthode **`toString`** de la classe `Pays` permet d'afficher proprement toutes les caractéristiques d'un objet de `Pays` avec `System.out.Print(nom_de_l'objet_de_type_pays)`

2.2. Exécutez et vérifiez (à l'œil) que le vecteur `leMonde` est trié par nom de pays

2.3. Commentez les instructions d'affichage des éléments du vecteur **`leMonde`**

3. Création d'un vecteur de continents à partir du vecteur leMonde

3.1. Créez une classe `Utilitaire`

3.2. Dans cette classe, ajoutez et codez la fonction suivante :

```
public static int indString(ArrayList<String> vString, String uneString) {
    // {vString trié} => { résultat = -1 si uneString est déjà dans vString,
    //                     sinon, résultat = indice où il faudrait
    //                     placer uneString dans vString (en respect du tri)}
```

3.3. Dans le corps de la procédure `main` de la classe `Monde`, ajoutez les instructions suivantes :

- déclaration d'un `ArralyList` de `String`, nommé `lesContinents`
- boucle de parcours du vecteur `leMonde` qui initialise le vecteur `lesContinents` :

  PRINCIPE : pour chaque élément du vecteur **`leMonde`**, utiliser **`indString`** pour tester si le continent de l'élément courant est déjà présent dans le vecteur **`lesContinents`** et, s'il ne l'est pas, l'insérer à la bonne place dans ce vecteur

- affichage du vecteur `lesContinents`
- Exécutez et vérifiez (à l'œil) que le vecteur **`lesContinents`** est trié selon l'ordre lexicographique

4. Vecteurs de pays par continent

4.1. Dans la classe `Utilitaire` déclarez et codez les fonctions suivantes :

a) Vérification de l'appartenance d'une chaîne à un vecteur de chaînes trié

```
public static boolean existString(ArrayList<String> vString, String uneString) {
    //  { vString trié } => { résultat = vrai si uneString appartient à vString }
```

b) Saisie contrôlée d'un continent (cette fonction doit utiliser la précédente)

```
public static String saisieCont(ArrayList<String> vCont) {
    // { vCont trié, non vide } =>
    //              { résultat = nom d'un élément de vCont
    //                saisi par l'utilisateur }
```

c) Création d'un vecteur contenant tous les pays d'un continent donné

```
public static ArrayList<Pays> paysDeCont(ArrayList<Pays> vPays, String cont) {
    // { cont est le continent d'au moins un pays de vPays } =>
    //        { résultat = vecteur contenant les pays de vPays
    //          dont le continent est cont }
```

4.2. Dans le corps de la procédure `main` de la classe `Monde`, ajoutez les instructions suivantes :

- déclaration d'une chaîne `unCont` (pour le nom d'un continent), initialisée par appel de `saisieCont`
- déclaration et initialisation d'un vecteur de `Pays`, contenant tous les pays du continent `unCont`, figurant dans le vecteur `leMonde`
- affichage du nom du premier et du dernier pays de ce vecteur, ainsi que du nombre de pays

4.3. Exécutez et testez

EXEMPLE DE TRACES D'EXÉCUTION DE CETTE PARTIE - *en vert, les saisies de l'utilisateur*

Donnez le nom d'un continent : *Afrique*
- Premier Pays : Afrique du Sud
- Dernier Pays : Zimbabwe
- Nombre de Pays : 63

5. Recherche d'un pays dans un vecteur de pays trié

5.1. Dans la classe `Utilitaire` déclarez et codez la fonctions suivante :

```
public static int rechPaysT(ArrayList<Pays> vPays, String nomP) {
    // { vPays trié sur le nom } =>
    //  { résultat = indice dans vPays du pays de nom nomP s'il existe,
    //    vPays.size() si pas de pays de nom nomP dans vPays }
```

5.2. Dans le corps de la procédure `main` de la classe `Monde`, ajoutez les instructions suivantes :

- saisie du nom d'un pays
- Si ce pays existe dans le vecteur précédemment créé, affichage des caractéristiques de ce pays, sinon affichage d'un message indiquant que le pays n'existe pas dans le continent `unCont` (cf. 4.2)

5.3. Exécutez et testez

EXEMPLE DE TRACES D'EXÉCUTION DE CETTE PARTIE - *en vert, les saisies de l'utilisateur*

Donnez le nom d'un pays : *Botswana*
Botswana
- continent : Afrique
- population : 2325082
- superficie : 581726

7. EXTENSIONS

7.1. Dans la classe Utilitaire déclarez et codez les procédures suivantes

a) Pays le(s) moins peuplé(s) du monde

```
public static void paysMoinsPeuples(ArrayList<Pays> vMonde) {
    //{ vMonde non vide } =>
    //     {les caractéristiques du ou des pays le(s) moins peuplé(s)
    //      dans vMonde ont été affichées}
```

b) Continent(s) comptant le plus grand, respectivement le plus petit nombre de pays

```
public static void contExtremes(ArrayList<Pays> vMonde,
                                ArrayList<String> vCont) {
    // { vMonde non vide } =>
    //      { le nombre de pays et le nom du ou des continents comptant
    //        le plus grand nombre, respectivement le plus petit nombre
    //        de pays ont été affichés }
```

7.2. Dans le corps de la procédure `main` de la classe `Monde`, **testez** ces procédures

EXEMPLE DE TRACE D'EXÉCUTION DE CETTE PARTIE

*** Pays le(s) moins peuplé(s) :
Géorgie du Sud-et-les îles Sandwich du Sud
- continent : Antarctique
- population : 35
- superficie : 4190
*** Continent(s) comptant le plus de pays (63 pays) :
- Afrique
*** Continent(s) comptant le moins de pays (3 pays) :
- Antarctique