IUT2 Université Grenoble Alpes – Département INFO

R1.01 – TP6(B)

Vecteurs en ArrayList – Recherche séquentielle et dichotomique

L'objectif de ce TP est de :

- *parfaire l'utilisation de vecteurs en ArrayList et la définition d'un ordre naturel sur une classe*
- *parfaire l'acquisition des algorithmes de recherche séquentielle et dichotomique dans un vecteur trié (ici sur deux critères)*

CONTEXTE : ROMANS POLICIERS

Dans ce TP, un roman policier est représenté par l'année de sa première parution, le nom de son auteur et son titre

Vous allez manipuler des `ArrayList` contenant des objets représentant des romans policiers, instances d'une classe `Polar` que vous allez définir.

Nous vous fournissons dans un dossier `TP6B_Files` :

- un fichier texte nommé *Polars.txt* qui contient les caractéristiques de 193 romans policiers
- une classe java `InitBibPolars` qui contient une fonction qui contient une fonction permettant de générer un `ArrayList<Polar>` intialisé avec les données du fichier *Polars.txt*
- une classe `Polars` 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/TP6B_Files .`
  - ✓ Lancez **IJ** et créez un projet `TP6_B`
- Ouvrez le dossier `TP6B_Files` que vous avez copié dans votre répertoire R1.01 et effectuez les actions suivantes :
  1. **copie dans votre projet** `TP6_B` **du fichier** *Polars.txt*
     - copiez le fichier *Polars.txt* (clic sur le fichier + CTRL + C)
     - dans la fenêtre de votre projet TP6_B, sélectionnez avec le bouton droit de la souris, l'icône du projet TP6_B
     - collez le contenu du presse-papiers (CTRL + V) puis validez (clic sur OK)
  2. **copie dans le répertoire** `src` **de votre projet** `TP6_B` **de la classe java** `InitBibPolars`
     - copiez le fichier `InitBibPolars.java` (clic sur le fichier : CTRL + C)
     - dans la fenêtre de votre projet TP6_B, 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_B` **de la classe java** `Polars`
     - copiez le fichier `Pays.java` (clic sur le fichier + CTRL + C)
     - dans la fenêtre de votre projet TP6_B, 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 Polar

On définit un ORDRE NATUREL sur la classe **`Polar`**, appelé ordre (`annee`, `auteur`)

Les objets de type **`Polar`** sont ordonnés sur l'année de parution et, à année de parution égale, sur le nom de l'auteur.

1.1. Dans le projet `TP6(B)`, **complétez** le code de la classe `Polar` qui implémente l'interface `Comparable` pour le type `Polar` (cf. cours 6 – Partie 1 : pages 9 et suivantes)

- Complétez le **constructeur** :

```
public Polar(int annee, String auteur, String titre) {
```

- Complétez les **getters** : `getAnnee()` / `getAuteur()` / `getTitre()`
- Complétez la **fonction de comparaison** entre les objets de type `Polar` selon l'ordre (`annee`, `auteur`)
- Complétez la fonction **de traduction en chaîne de caractères** d'un objet de type `Polar`

2. Déclaration, initialisation et affichage d'un vecteur de romans policiers

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

- la déclaration : `final ArrayList<Polar> lesPolars = InitBibPolars.lesPolars();`
- les instructions nécessaires à l'affichage de tous les romans policiers du vecteur **`lesPolars`**

2.2. Exécutez et vérifiez (à l'œil) que le vecteur **`lesPolars`** est trié dans l'ordre (`annee, auteur`)

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

2.4. Créez une classe `Utilitaire`

DANS LES PARTIES SUIVANTES :

- vous coderez dans la classe **`Utilitaire`**, des fonctions et des procédures utiles pour effectuer des recherches dans des vecteurs de **`Polar`** triés selon l'ordre (**`annee`**, **`auteur`**) .
- vous testerez au fur et à mesure dans la procédure `main` de la classe **`Polars_Main`**, les fonctions ou les procédures codées dans la classe **`Utilitaire`**
- des exemples de traces d'exécution de la procédure `main` vous sont proposés en fin de chaque partie

ATTENTION !
Il est indispensable que vous commentiez clairement les algorithmes de la classe **`Utilitaire`** ainsi que les tests effectués dans la procédure `main` de la classe **`Polars_Main`**

---

3. 1ères recherches dans un vecteur de romans policiers

A – Saisie d'un entier compris entre deux entiers

3.1. Dans la classe **`Utilitaire`**, ajoutez et codez la fonction suivante :

```
public static int saisirIntMinMax(int min, int max) {
    //{ min <= max } => { résultat = entier compris entre min et max ]
```

3.2. Dans la procédure `main` de la classe **`Polars_Main`**, ajoutez :

- la déclaration d'un entier **`anMin`**, que vous initialiserez avec la plus petite valeur de l'attribut **`annee`**, parmi les éléments du vecteur **`lesPolars`**
- la déclaration d'un entier **`anMax`**, que vous initialiserez avec la plus grande valeur de l'attribut **`annee`**, parmi les éléments du vecteur **`lesPolars`**
- la déclaration d'un entier **`anTest`** et son initialisation avec **`saisirAnnee`** *appliquée aux paramètres effectifs* **`anMin`** et **`anMax`**

3.3. **Exécutez et testez** en "*jouant à l'utilisateur qui ne lit pas les consignes…*"

B – Recherche séquentielle dans un vecteur de `Polar`

3.4. Ajoutez dans la classe **`Utilitaire`**, la fonction suivante :

```
public static int rechPremIndSeq(ArrayList<Polar> vPolar, int an, String aut) {
    // {vPolar trié dans l'ordre (annee, auteur)} =>
    //  {*  s'il y a dans vPolar au moins un élément d'année an et d'auteur aut,
    //      résultat = indice du premier de ces éléments
    //   *  sinon, résultat = -1}
    // LA RECHERCHE EST SÉQUENTIELLE !!!
```

3.5. Complétez la procédure `main` de la classe **`Polars_Main`**, avec :

- la déclaration d'une variable de type **`String`**, nommée **`auteurTest`**
- l'initialisation de cette variable par saisie
- les instructions permettant de tester la fonction **`rechPremIndSeq`** en vérifiant si dans le vecteur **`lesPolars`**, il existe au moins un roman de **`auteurTest`** pour l'année **`anTest`**
  - ✓ *s'il en existe un*, afficher l'indice dans **`lesPolars`** et le titre du premier roman de **`auteurTest`** pour l'année **`anTest`**
  - ✓ *sinon*, afficher un message approprié (EXEMPLE : aucun roman de … pour …)

3.6. Exécutez et testez plusieurs fois, avec des saisies différentes de **anTest** et de **auteurTest**

3.7. ***Vérification…*** Dans la procédure `main` de la classe **Polars_Main** :

- déclarez un nouveau vecteur de **Polar**, nommé **vPolarsVide** et ne contenant aucun élément
- ajoutez les instructions permettant de vérifier que la fonction **rechPremIndSeq** appliquée à **vPolarsVide**, **anTest** et **auteurTest** retourne le résultat attendu (cf. postcondition de la fonction)

C – Recherche dichotomique dans un vecteur de Polar

3.8. Ajoutez dans la classe **Utilitaire** la fonction suivante :

```
public static int rechPremIndDicho(ArrayList<Polar> vPolar, int an, String aut) {
    // {vPolar trié dans l'ordre (annee, auteur)} =>
    //  {*  s'il y a dans vPolar au moins un élément d'année an et d'auteur aut,
    //      résultat = indice du premier de ces éléments
    //   *  sinon, résultat = -1}
    // LA RECHERCHE EST DICHOTOMIQUE !!!
```

3.9. Complétez la procédure `main` de la classe **LesPolars** avec les instructions permettant de tester si la fonction **rechPremIndDicho** donne les même résultats que la fonction **rechPremIndSeq** :

- ✓ quand elle est appliquée à **lesPolars**, **anTest** et **AuteurTest**
- ✓ quand elle est appliquée à **vPolarsvide**, **anTest** et **AuteurTest**

3.10. Exécutez et testez plusieurs fois, avec des saisies différentes de **anTest** et de **auteurTest**

EXEMPLES DE TRACES D'EXÉCUTION (PARTIE 3) - *en vert, les saisies de l'utilisateur*

```
---------------------------
    Entrez un entier compris entre 1971 et 2018 : 1971
    Donnez le nom d'un auteur : POUY

    RECHERCHE SÉQUENTIELLE D'UN ROMAN ÉCRIT PAR POUY en 1971
    * Aucun roman de POUY pour 1971
    RECHERCHE SÉQUENTIELLE DANS UN VECTEUR VIDE D'UN ROMAN ÉCRIT PAR POUY en 1971
    * Aucun roman de POUY pour 1971 (vecteur vide)
    RECHERCHE DICHOTOMIQUE D'UN ROMAN ÉCRIT PAR POUY en 1971
    * Aucun roman de POUY pour 1971
    RECHERCHE DICHOTOMIQUE DANS UN VECTEUR VIDE D'UN ROMAN ÉCRIT PAR POUY en 1971
    * Aucun roman de POUY pour 1971 (vecteur vide)
---------------------------
    Entrez un entier compris entre 1971 et 2018 : 2010
    Donnez le nom d'un auteur : POUY

    RECHERCHE SÉQUENTIELLE D'UN ROMAN ÉCRIT PAR POUY en 2010
    * Roman trouvé à l'indice 174 dans le vecteur - titre : Holiday
    RECHERCHE SÉQUENTIELLE DANS UN VECTEUR VIDE D'UN ROMAN ÉCRIT PAR POUY en 2010
    * Aucun roman de POUY pour 2010 (vecteur vide)
    RECHERCHE DICHOTOMIQUE D'UN ROMAN ÉCRIT PAR POUY en 2010
    * Roman trouvé à l'indice 174 dans le vecteur - titre : Holiday
    RECHERCHE DICHOTOMIQUE DANS UN VECTEUR VIDE D'UN ROMAN ÉCRIT PAR POUY en 2010
    * Aucun roman de POUY pour 2010 (vecteur vide)
---------------------------
    Entrez un entier compris entre 1971 et 2018 : 2000
    Donnez le nom d'un auteur : VARGAS

    RECHERCHE SÉQUENTIELLE D'UN ROMAN ÉCRIT PAR VARGAS en 2000
    * Roman trouvé à l'indice 101 dans le vecteur - titre : Les quatre fleuves
    RECHERCHE SÉQUENTIELLE DANS UN VECTEUR VIDE D'UN ROMAN ÉCRIT PAR VARGAS en 2000
    * Aucun roman de VARGAS pour 2000 (vecteur vide)
    RECHERCHE DICHOTOMIQUE D'UN ROMAN ÉCRIT PAR VARGAS en 2000
    * Roman trouvé à l'indice 101 dans le vecteur - titre : Les quatre fleuves
    RECHERCHE DICHOTOMIQUE DANS UN VECTEUR VIDE D'UN ROMAN ÉCRIT PAR VARGAS en 2000
    * Aucun roman de VARGAS pour 2000 (vecteur vide)
```

4. Extension : autres recherches dans un vecteur de Polars

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

(a) *Existence de romans d'un auteur donné dans un intervalle d'années*

```
public static boolean existPolar(ArrayList<Polar> vPolar,
                                 int an1, int an2, String unAuteur) {
    // { * vPolar non vide et trié dans l'ordre (annee, auteur)
    //   * an1 <= an2
    //   * an1 est >= à la plus petite valeur de l'attribut annee dans vPolar
    //   * an2 est <= à la plus grande valeur de l'attribut année dans vPolar }
    //  => { résultat = vrai s'il existe dans vPolar, au moins un roman
    //          dont l'année est comprise entre an1 et an2 et d'auteur unAuteur'}
```

(b) *Affichage des titres des romans d'un auteur donné pour chaque année d'un intervalle d'années*

```
public static void lesPolarsDe(ArrayList<Polar> vPolar,
                               int an1, int an2, String unAuteur) {
    // { * vPolar non vide et trié dans l'ordre (annee, auteur)
    //   * an1 <= an2
    //   * an1 est >= à la plus petite valeur de l'attribut annee dans vPolar
    //   * an2 est <= à la plus grande valeur de l'attribut année dans vPolar }
    //   * vPolar contient au moins un roman dont l'auteur est unAuteur
    //      et dont l'année est dans l'intervalle [an1, an2} }
    //  => { chaque année A comprise entre an1 et an2 est affichée, suivie :
    //       * d'un tiret, si aucun roman n'a été écrit l'année A par unAuteur
    //       * de l'affichage ligne à ligne des titres des romans écrits par
    //         unAuteur, l'année A (chaque titre précédé d'un tiret) }
```

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

- saisie d'une année `an1` comprise entre `anMin` et `anMax`
- saisie d'une année `an2` comprise entre `an1` et `anMax`
- saisie du nom d'un auteur
- si, dans le vecteur `lesPolars`, il existe au moins un roman pour l'auteur saisi, dont l'année est comprise entre `an1` et `an2` :
  - ✓ affichage des titres de ces romans sur l'intervalle d'années [**an1** , **an2**]
  - ✓ *sinon* : affichage d'un message approprié

4.3. Exécutez et testez

EXEMPLE DE TRACE D'EXÉCUTION (PARTIE 4) – *en vert, les saisies de l'utilisateur*

```
--------------------------
    Entrez un entier compris entre 1971 et 2018 : 2016
    Entrez un entier compris entre 2009 et 2018 : 2018
    Donnez le nom d'un auteur : MANCHETTE

    Aucun roman de MANCHETTE entre 2016 et 2018
--------------------------
    Entrez un entier compris entre 1971 et 2018 : 2016
    Entrez un entier compris entre 2016 et 2018 : 2018
    Donnez le nom d'un auteur : POUY

    Romans de POUY entre 2016 et 2018
    * année 2016
     - Le Merle d'Arthur Keelt
    * année 2017
     -
    * année 2018
     - Ma ZAD
     - Mes soirs sans tweet
     - Lord Gwynplaine
```