
TP2 : TABLEAU D'ÉTIQUETTES

Introduction

Le deuxième a pour objectifs de vous faire construire des `Iterator` et d'utiliser l'héritage d'un `ArrayList` à votre avantage. Vous allez ajouter des fonctionnalités d'annotation (étiquetage) à un tableau. Les étiquettes vont identifier un intervalle du tableau. Une classe `Etiquette` vous est donnée. Aussi, une description initiale de la classe de base `TableauEtiquette` avec la signature de ces méthodes est donnée.

Description

Une étiquette est simplement un identificateur, représenté par une chaîne de caractères, et un intervalle, représenté par deux valeurs : la position de la première case et de la dernière case. L'intervalle est inclusif. Cette classe contient aussi des comparateurs d'intervalles.

```
public class Etiquette implements Comparable< Etiquette > {
    public Etiquette( String a_id, int a_debut, int a_fin )
    public String id()
    public int debut()
    public int fin()
    public int compareTo( Etiquette a_deuxieme )
    public boolean equals( Object a_deuxieme )
}
```

Le code de la classe `Etiquette` est disponible sur **Moodle**. Voici le code de base pour la classe que vous devez compléter :

```
import java.util.ArrayList;
import java.util.Iterator;

public class ListeEtiquette< E > extends ArrayList< E > {
    public ListeEtiquette() {
        super();
    }
    public void ajouterEtiquette( Etiquette a_etiquette )
    public void supprimerEtiquette( Etiquette a_etiquette )
    public void supprimerEtiquettes( String a_id )
    public Iterator< Etiquette > iterator( int a_position )
    public Iterator< E > iterator( String a_id )
}
```

Votre classe hérite de la classe `ArrayList`, qui servira de contenant pour les éléments du tableau. Ce sont ces éléments qui seront étiquettes. Vous devez ajouter les champs pour contenir vos étiquettes. Voici le rôle de chaque méthode :

1. **public void** `ajouterEtiquette(Etiquette a_etiquette)`
Cette méthode va permettre d'ajouter une étiquette à la structure. Il n'y a pas de règle pour ces étiquettes. Elles seront considérées comme valide, peu importe leur contenu.
2. **public void** `supprimerEtiquette(Etiquette a_etiquette)`
Cette méthode permet de supprimer une étiquette de la structure. Utiliser `equals()` pour trouver l'étiquette à supprimer. Une seule étiquette est supprimée. Donc, s'il y a plusieurs étiquettes similaires, une seule de ces étiquettes est supprimée.
3. **public void** `supprimerEtiquettes(String a_id)`
Cette méthode permet de supprimer plusieurs étiquettes. Toutes les étiquettes ayant le même `id` sont supprimées.
4. **public** `Iterator< Etiquette > iterator(int a_position)`
Cette méthode retourne un itérateur sur des étiquettes. L'utilisateur donne la position d'une case du tableau (indice). L'itérateur va permettre de parcourir toutes les étiquettes qui contiennent cette case dans leur intervalle.
5. **public** `Iterator< E > iterator(String a_id)`
Finalement, cette méthode va retourner un itérateur qui parcourt les cases du tableau d'éléments. Par contre, seule les cases qui font partie d'un intervalle ayant l'`id` indiqué seront parcourues. Même si une case est contenue dans plusieurs intervalles, elle ne sera parcourue qu'une seule fois. Les cases doivent être parcourues dans le même ordre que nous les retrouvons dans le tableau.

Directives

1. Le tp est à faire seul.
2. Vous pouvez ajouter des méthodes aux classes. Vous ne pouvez pas modifier la signature des méthodes déjà données.
3. Commentaires :
 - a. Commentez l'entête de chaque classe et méthode. Ces commentaires doivent contenir la description de la méthode et le rôle de ces paramètres.
 - b. Une ligne contient soit un commentaire, soit du code, pas les deux.
 - c. Utilisez des noms d'identificateur significatif.
 - d. Utilisez le français.
4. Code :
 - a. Pas de `goto`, `continue`.
 - b. Les `break` ne peuvent apparaître que dans les `switch`.
 - c. Un seul `return` par méthode.
5. Indentez votre code. Assurez-vous que l'indentation est faite avec des espaces.

Remise

Remettre le tp par l'entremise de Moodle. Placez vos fichiers '*.java' dans un dossier compressé de **Windows**. Vous devez remettre l'archive. Le tp est à remettre avant le 2 juillet 23 :55.

Évaluation

- Fonctionnalité (8 pts) : des tests partiels vous seront remis. Un test plus complet sera appliqué à votre tp.
- Structure (2 pts) : veillez à utiliser correctement le mécanisme d'héritage et de méthode.
- Lisibilité (3 pts) : commentaire, indentation et noms d'identificateur significatif.