

TP4 : Tableaux à une dimension

Analyse de texte

Partie 1 : Classe DictionnaireStringInt

A partir du travail effectué en TD, réaliser la classe `DictionnaireStringInt`. Le squelette de la classe sera le suivant. Il contient les méthodes vues en TD ainsi que quelques nouvelles méthodes.

```
public class DictionnaireStringInt {  
  
    private String[] cles = new String[10];  
    private int[] valeurs = new int[10];  
    private int nbElements = 0;  
  
    public void ajouterModifier(String cle, int valeur ) {...}  
  
    public int rechercherValeur(String cle) {...}  
  
    private int rechercherIdx(String cle) {...}  
  
    private void agrandir() {...}  
  
    public int supprimer(String cle) {...}  
  
    public String toString() {...}  
  
    public int getNbElements() {...}  
  
    public String rechercherCleAvecValeurMax() {...}  
  
    /* Le tableau retourné aura une taille de nbElements */  
    public String[] construireTableauDesCles() {...}  
}
```

Partie 2 : Intégration avec l'interface graphique (c.f. TP3)

Pour intégrer le dictionnaire avec l'interface graphique, il faut réaliser les étapes suivantes :

1 – Déclarer un attribut compteur de type `DictionnaireStringInt` dans la classe `AnalyseOccurrence` (classe de l'interface graphique).

2- Ajouter la méthode suivante qui permet d'ajouter tous les mots de la chaîne de caractères passée en paramètre dans le dictionnaire.

```
public void analyseTexte(String txt) {

    Scanner sc = new Scanner(txt);
    compteur = new DictionnaireStringInt();

    /* expression régulière qui permet de définir
     * les limites des mots lus via la méthode next() du scanner.
     * (espaces, ponctuation, saut de ligne, etc.)
     */
    sc.useDelimiter("([^\p{L}\\p{Nd}]|\\s)+");

    while (sc.hasNext()) {
        String motLu = sc.next().toLowerCase();
        compteur.ajouterModifier(motLu, compteur.rechercherValeur(motLu)+1);
    }
    sc.close();
}
```

3- Rendre le composant `JTextArea` et les composants qui affichent des informations issues de l'analyse du texte accessibles. (Seulement pour Eclipse, sous Netbeans, les composants sont déjà des attributs)

En effet, tous les composants graphiques sont (avec eclipse) par défaut seulement visibles dans la méthode `initialize()`. Pour que l'on puisse les mettre à jour ou récupérer l'information qu'ils contiennent, il faut les déclarer comme attributs. Pour cela, il suffit de cliquer droit sur le composant à rendre visible, puis sélectionner « `exposeComponent` ». Laisser les options proposées par défaut et valider. Vous pourrez voir dans les sources que le composant est maintenant déclaré comme attribut et qu'une méthode d'accès à cet attribut a été automatiquement générée.

4- Ajouter le code à exécuter lorsque l'on clique sur le bouton « Analyser ».

Pour cela, il suffit de double cliquer sur le composant (ou il faut cliquer droit sur le composant, choisir « `add event handler` » → action → `actionPerformed`). L'onglet source de la classe s'affiche et le code suivant a été automatiquement ajouté :

```
btAnalyse.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // CODE A EXECUTER LORS DU CLIC
    }
});
```

Le code qui sera exécuté lors du clic sur le bouton est à mettre à l'endroit indiqué ci-dessus (i.e. dans la méthode `actionPerformed(...)`).

Pour récupérer le texte d'un objet de type `JTextArea`, il faut utiliser la méthode `getText()`. Pour mettre à jour le texte d'un `JLabel`, il faut utiliser la méthode `setText(String txt)`. La fonction `String.valueOf(int i)` permet de convertir un entier en chaîne de caractères.

Pour mettre à jour la liste des mots d'un `JComboBox`, il faut (1) tout supprimer via la méthode `removeAllItems()`, puis ajouter les nouveaux éléments un par un grâce à la méthode `addItem(String s)`.

5 – Ajouter le code à exécuter lors du changement d’item dans le composant JComBoBox.

Lorsque le mot sélectionné dans le combobox change alors sa valeur associée (i.e. le nombre d’occurrence affiché à côté) doit être mis à jour.

Pour ajouter du code à exécuter lors du changement de la valeur sélectionnée dans un JComBoBox, il faut cliquer droit sur le composant, choisir « add event handler » → action → actionPerformed. Ensuite, c’est le même fonctionnement que pour les boutons.