

# TD7 – GUI pour application de comptage d’une lettre dans un fichier

4ASTI – Programmation Java – B.Nguyen

## Problème :

Vous disposez de plusieurs fichiers texte. Il s’agit de sélectionner un ensemble de ces fichiers, et compter combien de fois une lettre particulière apparaît dans cet ensemble. Chaque tâche de comptage de fichier peut être réalisée de manière autonome et parallèle. On souhaite afficher toutes ces informations dans une interface graphique.

## Apparence de la GUI :

La GUI doit permettre de sélectionner plusieurs fichiers en utilisant un FileChooser, lié à un bouton « Charger », les noms des fichiers doivent apparaître dans une structure JTextArea (voir fichier ExempleJTextArea.java). On ne s’embarasse pas ici du layout, on utilisera donc un simple FlowLayout.

La GUI doit aussi avoir un JTextField qui représente la lettre à compter, un autre représentant le résultat global, un bouton « compter » permettant de lancer l’exécution. Lors du lancement de l’exécution, on démarre une tâche pour chaque fichier pour lequel on va compter combien de fois la lettre apparaît, et effectuer le calcul global, qui sera affiché dans un JLabel.

N’oubliez pas de mettre les JLabels explicatifs devant chaque champ (par exemple le champ « lettre à compter »



Figure 1- GUI cible

## Modélisation MVC :

**Modèle :** les données manipulées sont les fichiers (et plus précisément les contenus des fichiers), la lettre qu’on veut compter, et le compte du nombre de fois que cette lettre apparaît pour chaque fichier. Nous allons nommer cette classe ModeleCompte.

**Vue :** Les informations pertinentes de la vue sont indiquées dans la partie *apparence de la GUI*.

**Controler :** Il faut gérer la sélection de l’ensemble de fichiers via un FileChooser. Il faut également gérer le lancement des tâches de comptage déclenché par un appui sur le bouton « compter », le comptage de chaque fichier étant exécuté dans un Worker Thread. Nous décidons de n’utiliser qu’une seule instance de la classe Controler, il faudra donc que le controler sache quelle est l’origine de l’action, en utilisant `getActionCommand()`

## Gestion des exceptions :

Les exceptions devront être capturées et affichées dans une fenêtre pop-up.

## Indications pour commencer :

Commencez en créant :

- Une classe Vue comprenant juste un JButton pour charger les fichiers et un JTextArea pour afficher les noms de fichiers chargés. Le JButton devra avoir comme actionCommand la valeur Controleur.CHARGER.
- Une classe ModeleCompte comprenant un File[] correspondant aux fichiers qui seront sélectionnés, un String[] correspondant au contenu de chaque fichier
- Une classe Controleur comprenant un ActionListener qui lance un JFileChooser (voir fichier exemple) lorsque l'action déclenchée est Controleur.CHARGER puis affiche dans le JTextArea les fichiers sélectionnés.
- Pensez à bien ajouter au bouton de la classe vue l'ActionLisener (le controleur) que vous venez de définir !

C'est très important de bien voir quelles classes vont avoir besoin de quelles références (par exemple Vue doit connaître l'instance de ModeleCompte et de Controleur). De même, le Controleur doit aussi connaître l'instance de Vue et le ModeleCompte ...

Ensuite :

- Rajoutez à la Vue le bouton compter, le JTextField pour remplir la lettre et les JLabels restants.
- Créez une classe CompteTask qui hérite de SwingWorker<Integer, Integer>. Vous lierez via le constructeur CompteTask à l'instance de Vue (pour pouvoir afficher les résultats) et à l'instance de ModeleCompte (pour manipuler les informations). Programmez dans la méthode `protected Integer doInBackground()` le code permettant de charger 1 fichier, compter le nombre de lettres égal à la valeur dans le champ de la GUI et qui affiche le résultat dans la GUI
- Rajoutez à la classe Controleur une action qui se déclanchera lorsqu'on appuiera sur le bouton Compter : ça déclanchera une CompteTask pour le premier fichier sélectionné

Enfin, gérez le fait qu'on veut charger plusieurs fichiers et effectuer le comptage dans des tâches en parallèle.