IFT1025 – Programmation 2
TP2 : Mini correcteur d'orthographe
Remise : avant minuit le 22 décembre 2017
À faire en groupe de 2 personnes (ou seul si vous le voulez)
Ce TP correspond à 15% dans la note globale.

But
Ce TP vise à intégrer les éléments présentés dans ce cours, notamment, l'utilisation des structures de données internes, traitements sur les fichiers, et l'interface graphique. On ne vous impose pas une façon de structurer votre programme. C'est à vous de le faire en utilisant les différents éléments du langage Java (notamment la programmation OO) d'une façon la plus raisonnable possible.

Problème
Ce TP traite le problème de correction d'orthographe. Pour le but de ce TP, on considère une correction très restreinte : Seuls les mots stockés dans un dictionnaire (dans un fichier) sont considérés corrects et on ne considère pas la grammaire. Dans un texte entré par un utilisateur ou lu d'un fichier, il peut y avoir des mots qui ne sont pas dans le dictionnaire. Ces mots sont « inconnus ». Le rôle du correcteur est (1) d'identifier ces mots « inconnus », (2) de proposer des mots du dictionnaire les plus proches à l'utilisateur pour le remplacer.

Pour ce correcteur, on ne tient pas compte des règles grammaticales (qu'un vrai correcteur devrait faire). On se contente de traiter chaque mot de façon isolé. Un dictionnaire (liste de mots) en français vous est fourni.

Tâches
Pour ce TP, vous devez concevoir une interface graphique qui permet à l'utilisateur d'exécuter les tâches suivantes :

1. Lire un dictionnaire (une liste de mots triée) d'un fichier;
2. Entrer un texte ou lire un texte d'un fichier;
3. Lancer la vérification d'orthographe qui vérifie si chaque mot est un mot connu. Si non, le mot sera surligné en rouge.
4. Si l'utilisateur clique sur un mot en rouge, on doit voir apparaître les mots les plus proches dans un petit menu à côté du mot. L'utilisateur peut alors choisir un des mots proposés pour le remplacer.
5. Finalement, une fois la correction est faite, l'utilisateur peut stocker le texte dans un fichier.

Interface
Menu :
L'interface que vous programmez doit avoir un menu, dans lequel il y a « Fichier », « Dictionnaire » et « Vérifier ».

- Quand on clique sur le menu « Fichier », on verra apparaître les options « Ouvrir » et « Enregistrer » pour respectivement lire un texte d'un fichier et sauvegarder le texte dans un fichier. Le texte lu sera affiché dans la zone de texte (un TextArea - voir plus bas).
- Quand on clique sur le menu « Dictionnaire », on veut charger un dictionnaire à partir d'un fichier. Ce dictionnaire sera alors stocké dans une structure de données internes (e.g. une liste chaînée comme ce que vous avez utilisé dans l'exercice 3), un tableau, ou un HashSet (voir les exemples à https://www.cs.cmu.edu/~adamchik/15-121/lectures/Hashing/hashing.html).

Pour ces deux opérations liées au fichier, vous devez utiliser la classe FileChooser ((ou JFileChooser) pour demander le nom du fichier. Voir le tutoriel https://docs.oracle.com/javase/tutorial/uiswing/components/filechooser.html

- Cliquer sur « Vérifier » va lancer la vérification du texte, qui change les mots inconnus en rouge.

TexteArea :
Votre interface doit contenir une zone d'affichage de texte (TextArea ou JTextArea), qui affiche le texte (lu d'un fichier, ou entré directement par l'utilisateur). L'utilisateur peut voir apparaître les mots en rouges après avoir cliqué sur « Vérifier ». Il peut alors cliquer sur un mot en rouge, et votre programme doit lui proposer une liste de 5 mots les plus proches dans un menu à côté (voir distance d'édition plus bas). Si l'utilisateur choisit un de ces mots, le mot inconnu sera remplacé par le mot choisi.

Les opérations sur le texte :
Pour surligner les mots inconnus, vous aurez besoin d'identifier chaque mot dans le texte, et s'il est inconnu, de le surligner en rouge. Un exemple TextAreaHighlight.java, qui surligne un mot (le mot « public ») dans un TexArea, est fourni. Vous pouvez vous inspirer de cet exemple pour surligner les mots.

Quand l'utilisateur clique sur un mot dans le texte, vous aurez besoin d'identifier le mot en question. Vous pouvez vous inspirer de l'exemple TextAreaTest.java, qui est également fourni. Cet exemple affiche un petit texte, et quand on clique sur un mot, il afficher sur l'écran les coordonnées de la position cliquée. C'est la méthode `public void mouseClicked(MouseEvent e)` qui détecte le mot sur lequel l'utilisateur a cliqué. Vous pouvez vous inspirer de cet exemple pour identifier le mot cliqué.

Pour modifier un mot inconnu, vous aurez besoin de la méthode void replaceRange(String, int, int) de la classe TextArea (voir un tutoriel http://java.sun.com/docs/books/tutorial/uiswing/components/textarea.html).

Les mots les plus proches
Pour déterminer les mots du dictionnaire les plus proches à un mot inconnu, on utilise la distance d'édition (edit distance), ou la distance Levenshtein. Cette distance correspond au nombre d'insertion, de remplacement et d'enlèvement de caractère pour transformer un mot en un autre. Par exemple, la distance d'édition entre « extraction » et « axtrations » est 3 : 1 remplacement (a par e), 1 enlèvement (c) et une insertion (s). Les mots les plus proches sont les mots du dictionnaire ayant la distance minimale avec le mot inconnu. Dans ce TP, vous faîtes afficher 5 mots les plus proches. Il peut y avoir plusieurs mots de même distance. Dans ce cas, vous pouvez choisir à afficher n'importe quels candidats équivalents.

Vous pouvez trouver une implantation de cette distance : https://gist.github.com/gabhi/11243437
Vous pouvez utiliser cette implantation directement dans votre TP.

Conception et organisation
Vous êtes demandé à concevoir vos programmes vous-mêmes, en exploitant le plus possible les outils existants (les classes de Java et les devoirs et TP antérieurs). Certains outils que vous allez utiliser ne sont pas présentés dans la classe. Vous devez chercher la documentation sur le Web. Ceci est voulu afin de vous encourager à aller chercher des solutions possibles plus largement que dans les supports du cours. C'est aussi ce qui se passe souvent dans la vie réelle d'un programmeur.

Évaluation

Ce TP compte pour 15% dans la note finale. Ils sont attribués aux éléments suivants :

- Conception et affichage de l'interface graphique, activation de menus et interactions avec le texte : 5 points
- Lecture du dictionnaire pour le transformer dans une structure interne : 2 point
- Lecture d'un fichier texte : 1 point
- Identification des mots inconnus dans le texte : 3 point
- Correction d'un mot inconnu (afficher les mots proches et remplacement du mot inconnu : 3 points
- Conception et organisation globale: 1 point

La date limite de remise est avant minuit le 22 décembre.