Aix*Marseille université
Initiative d'excellence

Année 2020-21

Domaine Sciences et Technologies
Licence informatique

Programmation 2 : TP 2
Code UE : SIN3U02

Recherche d'un mot dans une grille de caractères

Le but de ce TP est de créer un programme qui recherche un mot donné par l'utilisateur dans une grille de caractères. Le mot peut apparaître horizontalement ou verticalement dans la grille. Dans ce TP, Vous apprendrez à utiliser des tableaux multidimensionnels et à lire et écrire à partir de fichiers.

1 Rappel de quelques commandes `git`

Comme pour le TP précédent, vous utiliserez le dépôt `git` du cours sur `https://etulab.univ-amu.fr/` (voir TP-1).

- Pour récupérer le fichiers du TP-2, il suffit de faire un `pull` à partir de votre dépôt `git`. Pour que cela fonctionne, il faut que vous ayez gardé le lien entre votre dépôt et le dépôt d'origine :
  - `git remote add base  https://etulab.univ-amu.fr/nasr/programmation2`
  - `git pull base master`
- À chaque modification de programme, faites un `commit` en sélectionnant (à l'aide de `git add`) les fichiers modifiés. Chaque `commit` doit contenir un message précisant la nature des modifications effectuées.
- À chaque tâche terminée, faites un `push` de votre travail.

2 La classe `Crossword`

Cette classe représente la grille de caractères. L'élément principal est un tableau bidimensionnel de caractères : `private char [][] array`

- Le constructeur de la classe `Crossword` remplit le tableau de caractères `array` avec les caractères lus à partir d'un fichier (un exemple est donné dans le fichier "WORD.txt"). Avant de créer le tableau, on doit calculer sa taille : le nombre de lignes et de colonnes.
- `private int rows` : doit contenir le nombre de lignes du tableau. Pour déterminer cette valeur, le programme doit compter le nombres de lignes dans le fichier.
- `private int columns` : doit contenir le nombre de colonnes du tableau. Pour déterminer cette valeur, le programme doit compter le nombres de caractères dans la première ligne du fichier. (On suppose que toutes les lignes contiennent le même nombres de caractères)
- Les méthodes `display` et `displayWord` sont fournies, elles permettent de visualiser le tableau en utilisant une interface graphique simple (classe `CrosswordGUI`).
  Voici un exemple d'affichage :

Attention : Ne modifiez pas le code dans le fichier `CrosswordGUI.java`.

3 Les classes `java.io.File` et `java.util.Scanner`

Le classe `Scanner` permet de lire des caractères à partir du terminal ou d'un fichier. La classe `File` permet de réaliser des operations sur un fichier. Vous trouverez la documentation dans les documents suivants :
`https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html`
`https://docs.oracle.com/javase/8/docs/api/java/nio/file/Files.html`

3.1 Comment lire à partir du terminal ?

Créer une instance de la classe `Scanner` avec `System.in` :
```
Scanner scan = new Scanner(System.in);
```
Ensuite utiliser les méthodes `Next()` ou `NextLine()`

3.2 Comment lire à partir d'un fichier ?

Utiliser les classes `Scanner` et `File` :
```
File file = new File(filename);
Scanner input = new Scanner(file);
```
On peut vérifier que le fichier `file` a bien été ouvert, en utilisant `file.exists()`

4 Tache 1 : Construction d'une instance de la classe `Crossword`

Constructeur du classe `Crossword` :
— Déterminer la valeur des variables d'instance `rows` et `columns`.
— Construire le tableau de caractère `array` de taille `rows` × `columns`
— Remplir `array` avec les caractères contenus dans le fichier donné (par exemple "WORDS.txt")
Méthode main() dans `Main.java` :
— Utiliser le constructeur de la classe `Crossword` pour en créer une instance.
— Vérifier le constructeur en utilisant la méthode `display()` qui permet de visualiser le tableau.

5 Tache 2 : Recherche d'un mot dans le tableau `Crossword`

Créer une méthode `boolean search(String word)` qui va chercher le mot `word`. La méthode retourne `true` si le mot existe dans le tableau de caractères, soit horizontalement soit verticalement. L'idée est la suivante :
— Chercher le premier caractère de `word` dans le tableau.
— Si vous avez trouvé ce caractère à la position `[i][j]`, alors le mot peut être present dans la ligne `i` ou bien dans la colonne `j`.
— Verifier les deux cas séparément (vous pouvez créer méthodes supplémentaires).
— Si aucune des deux conditions est vérifiée, on continue la recherche du premier caractère de `word` dans le reste du tableau à partir de la position `[i][j]`.

Notes :
— Une fois le mot trouvé, sauvegarder la position du mot dans les variables `(PositionY, PositionX, EndY, EndX)` qui donne les coordonnées de début et fin de mot.
— Pour tester votre code, vous pouvez utiliser la méthode `displayWord()` qui permet de visualiser le tableau avec le mot en surbrillance (comme dans l'image de la page précédente).

6 Tache 3 : Compter le nombre total d'occurrences d'un mot

Créer une méthode `int countWord(String word)` qui compte le nombre d'occurrences du mot `word` dans le tableau. Le méthode doit retourner un entier (`int`) qui contient ce nombre.

7 Tâches optionnelles

Si vous avez fini les tâches précédentes, vous pouvez améliorer votre projet en rajoutant les fonctionnalités suivantes.

— Modifier la méthode `search` de sorte qu'elle retrouve les mots écrits en diagonale dans la grille.