

## \*\*\* Mode Opérateur de l'application Tableur sous Java \*\*\*

### Organisation du programme :

Les différentes actions réalisées par le programme sont :

- Afficher la table au format littéral (commande "a" dans le terminal)
- Modifier ou saisir le contenu d'une cellule (commande "m" dans le terminal)
- Effacer le contenu d'une cellule (commande "e" dans le terminal)
- Importer une table issue d'un fichier texte avec le format du fichier de la consigne (cf fichier de test table.txt, commande "if" dans le terminal)
- Afficher la table au format numérique (commande "an")
- Sauvegarder le tableur dans un fichier (commande "ef" dans le terminal)
- Afficher le tableur dans une fenêtre graphique (commande "ig")

### Voici un descriptif des différentes classes et de leurs fonctions/ méthodes :

Program.java :

- La classe Programme qui contient la fonction main() et sert à exécuter le programme
- Vous trouverez le code du menu afin d'appeler les objets et fonctions pour réaliser les actions demandées par l'utilisateur
- Il y a notamment les fonctions loadFile et saveFile permettant respectivement de charger et d'enregistrer la table dans un fichier txt
- On instancie également la classe Grid et lui passe les informations rentrées par l'utilisateur

Grid.java:

- la classe grid représente le tableur et stocke la position des différentes cellules grâce au tableau ArrayList
- Il y a notamment les fonctions : display(), assignCell(), deleteCell(), arrayToGrid(), calculate() permettant de réaliser les opérations demandées par l'utilisateurs

Node.java et Value.java :

- les classes Node et Value servent à stocker les coordonnées des cellules Nodes (attributs : value, right, down) ainsi que les valeurs : Value (attributs : sval : texte, dval : décimal, tag)

Test.java :

- la classe Tests permet de déclarer les fonctions de test afin qu'elles puissent être exécutées quand elles seront appelées

CalcGUI.java :

- la classe CalcGUI qui représente l'interface graphique pour afficher le tableur dans la fenêtre d'une application

Pour tester le programme, voici les différentes actions à faire :

Dans un premier temps, il faut renseigner le chemin absolu du fichier lorsque vous l'importez ou l'enregistrer, exemple mettre : /Users/XXX/Desktop/\_NFA035/\_projet/ + table.txt

Ensuite, exécuter le fichier Program.java.

Un menu s'affiche dans le terminal avec les différents choix d'opérations pour l'utilisateur.

-> L'utilisateur choisit d'Afficher le tableur -> entrer la commande "a" dans le terminal puis "entrée"

-> L'utilisateur choisit de modifier une cellule -> entrer la commande "m" dans le terminal puis "entrée"

Le programme demande à l'utilisateur de choisir les coordonnées de la cellule à modifier :

- le numéro de la ligne : row, le numéro de la colonne : col puis la valeur de la cellule : values

- pour vérifier que la cellule a bien été modifiée, on peut demander au programme d'afficher le tableur (commande "a")

et constater que la cellule a bien été modifiée dans le Tableur

-> L'utilisateur choisit d'effacer le contenu d'une cellule -> entrer la commande "e" dans le terminal puis "entrée"

Le programme demande à l'utilisateur de choisir les coordonnées de la cellule à effacer :

- le numéro de la ligne : row et le numéro de la colonne : col

- pour vérifier que le contenu de la cellule a bien été effacée, on peut demander au programme d'afficher le tableur (commande "a")

et constater que le contenu de la cellule a bien été effacée dans le Tableur

-> L'utilisateur choisit d'importer un fichier -> entrer la commande "if" dans le Terminal puis "entrée"

Le programme demande à l'utilisateur le nom du fichier à importer

- l'utilisateur rentre le chemin absolu du fichier (exemple :

/Users/XXX/Desktop/\_NFA035/\_projet/ + table.txt)

- puis, le programme affiche à l'utilisateur le contenu du fichier qui a été importé

- pour vérifier que le fichier a bien été importé, on peut demander au programme d'afficher le tableur (commande "a") et constater que les données ont bien été importées dans le Tableur

-> L'utilisateur choisit d'afficher le tableur avec les valeurs numériques -> entrer la commande "an" dans le Terminal puis "entrée"

-> L'utilisateur choisit d'enregistrer le tableur dans un fichier -> entrer la commande "ef" dans le terminal puis "entrée"

Le programme demande à l'utilisateur le nom du fichier dans lequel il veut enregistrer le tableur.

- l'utilisateur rentre le chemin absolu du fichier (exemple :

/Users/XXX/Desktop/\_NFA035/\_projet/ + toto.txt)

- le programme crée un nouveau fichier avec les valeurs des cellules du Tableur séparés par " & "

- Pour vérifier que cela a bien fonctionné, on peut ouvrir le fichier qui a été créé

-> L'utilisateur veut afficher l'interface graphique -> entrer la commande "ig" dans le Terminal puis "entrée"

- le programme ouvre une fenêtre sur l'ordinateur et affiche le tableur avec la valeur des cellules

## Parties 4.2 et 5 :

Je n'ai pas réussi à implémenter les parties 4.2 et 5, voici les explications de comment est ce que j'aurais fait :

- Pour la partie 4.2 : Dépendances entre cellules :

Concernant la modélisation des liens entre les cellules, pour gérer les dépendances, pour chaque cellule (ou de la classe Value), j'aurais créé un attribut de type liste avec les coordonnées des cellules qui dépendent de C. Pour détecter un cycle, il faut vérifier, à chaque fois que l'utilisateur rentre une valeur, s'il y a une référence déjà présente dans la liste des cellules qui dépendent de C. Si c'est le cas, ne pas autoriser à la renseigner.

Idem, pour la mise à jour automatique d'une cellule. A chaque fois qu'une cellule est modifiée, il faut mettre à jour les cellules qui sont renseignées dans la liste des cellules

qui dépendent d'elle et ainsi de suite (la valeur du calcul est stockée dans l'attribut dval de l'objet Value et la valeur texte (formule) dans l'attribut sval ce qui permet de pouvoir recalculer la formule).

Afin de remonter le cycle et les différentes cellules j'aurais appliqué une fonction récursive.

- Dans le point 5 du projet, il est demandé que le programme prenne en charge les références de cellules relatives avec le signe \$.

Pour calculer, j'aurais pris la position d'une cellule ayant une référence relative \$(1,2) dans la cellule (2,3) par exemple est la position absolue : (1,1) 1 ligne plus haute et 2 colonnes vers la gauche). Idem que dans l'affichage du contenu de la cellule référencée de manière absolue, j'aurais affiché sa valeur dans la cellule de destination pour pouvoir faire le calcul, je ferai donc la même technique que celle utilisée pour les références absolues c'est à dire avec le tableau des références qui dépendent de la cellules