

Programmation Orientée Objet en Java

Licence 3 GLSI

Projet de Fin d'étude : Solveur de Sudoku

Le Sudoku est un jeu de réflexion basé sur une grille de 9x9, divisée en neuf sous-grilles de 3x3 cases. L'objectif est de remplir chaque ligne, chaque colonne et chaque sous-grille avec les chiffres de 1 à 9, de manière à ce que chaque chiffre apparaisse une seule fois par ligne, colonne et sous-grille. Cependant, certaines cases de la grille sont déjà remplies, et il faut compléter la grille en respectant ces contraintes.

Dans ce projet, vous allez concevoir un programme en Java qui résout des grilles de Sudoku. L'application devra être en mode console et permettra de charger une grille partiellement remplie depuis un fichier texte, puis de résoudre cette grille en affichant d'abord l'état initial et ensuite la grille résolue.

L'objectif du projet est donc de développer un solveur de grilles de Sudoku en utilisant la programmation orientée objet.

1. Spécifications Fonctionnelles

1.1 Chargement de la grille

Le programme devra charger la grille initiale depuis un fichier texte dont le nom est fourni en argument lors de l'exécution. Ce fichier contiendra :

- 9 lignes, chacune contenant 9 chiffres entre 0 et 9.
- Le chiffre 0 représente une case vide à remplir par le solveur.

Exemple de fichier d'entrée (grille.txt)

```
5 3 0 0 7 0 0 0 0
6 0 0 1 9 5 0 0 0
0 9 8 0 0 0 0 6 0
8 0 0 0 6 0 0 0 3
4 0 0 8 0 3 0 0 1
7 0 0 0 2 0 0 0 6
0 6 0 0 0 0 2 8 0
0 0 0 4 1 9 0 0 5
0 0 0 0 8 0 0 7 9
```

Si aucun fichier n'est fourni en argument, l'utilisateur devra saisir la grille manuellement, ligne par ligne, en respectant le format ci-dessus (valeurs séparées par des espaces).

1.2 Affichage de la grille

Avant la résolution, la grille chargée ou saisie doit être affichée à l'écran en utilisant les caractères spéciaux de dessin de boîte ("box-drawing characters") pour bien structurer l'affichage. A noter que des cases vides devront être affichées en lieu et place des valeurs 0.

Exemples d'affichage en console :

8	5				2	4		
7	2							9
		4						
			1		7			2
3		5				9		
	4							
				8			7	
	1	7						
				3	6		4	

Exemple d'affichage 1

2	5	8	4	1	7	6	9	3
6	1	7	9	2	3	8	5	4
9	3	4	8	6	5	1	7	2
3	2	5	7	8	1	4	6	9
8	9	6	3	5	4	2	1	7
7	4	1	6	9	2	5	3	8
4	6	9	1	3	8	7	2	5
5	7	3	2	4	6	9	8	1
1	8	2	5	7	9	3	4	6

Exemple d'affichage 2

1.3 Résolution du Sudoku

Le programme doit résoudre la grille en respectant les règles du Sudoku :

1. Chaque ligne contient les chiffres 1 à 9 sans répétition.
2. Chaque colonne contient les chiffres 1 à 9 sans répétition.
3. Chaque sous-grille 3x3 contient les chiffres 1 à 9 sans répétition.

L'algorithme de résolution peut être basé sur la technique de **Backtracking**, simple à implémenter et efficace pour un projet de ce niveau.

1.4 Affichage de la solution

Après résolution, le programme doit afficher la grille solutionnée en conservant le même format de présentation (cf. [section 1.2](#)).

2. Contraintes Techniques

- Le programme devra être structuré en plusieurs classes que vous identifierez
- Le programme doit au moins gérer les erreurs suivantes :
 - Fichier introuvable.
 - Mauvais format du fichier (ex. nombre incorrect de lignes ou de colonnes).
 - Grille invalide (ex. valeurs incorrectes).
- Le code devra être bien commenté et structuré.

3. Évaluation du projet

Le projet sera évalué sur la :

1. **Qualité de l'implémentation** (code source bien structuré avec commentaires explicatifs).
2. **Présentation des résultats** (affichage clair en mode console).

4. Livrables attendus

Les livrables ci-après devront être envoyés par mail à envoip@gmail.com avec comme objet : **Projet_Sudoku_Gx_L3GLSlz, x** étant le numéro du groupe et **z € {A ou B}**. Les membres du groupe (4 étudiants par groupe) devront également être précisés dans le contenu du mail.

- Le code source du projet (fichiers **.java**) comme pièces jointes au mail (à ne pas zipper)
- Lien YouTube de la vidéo de présentation du code et des choix techniques (classes utilisées, algorithme de résolution, etc.), ainsi que du logiciel développé (maximum 10 minutes)

Le non-respect de ces contraintes rendra le travail irrecevable.

5. Date de rendu du projet

Au plus tard le **25/03/2025**