

Devoir Structures de Données

EPSI SN1 BTS

I. Objectif

Réaliser un solveur de Sudoku en PHP

II. Exemple

```
~/www/epsi-wis/SolveurSudoku git:(master) ±3 (0.755s)  
php solve.php
```

```
Chargement du fichier grid1.json
```

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

```
Début de la recherche de solution
```

```
Reussite :
```

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

```
Durée totale : 1 ms
```

III. Consignes

Pour démarrer :

- Récupérer le code source de démarrage et le dézipper dans le dossier de travail
- Architecture des dossiers :
 - **grids** : Contient des grilles d'exemple de plusieurs difficultés, ainsi qu'une grille déjà complétée
 - **src** : Contient les classes qui seront à développer, ainsi que les interfaces à implémenter
 - **tests** : Contient les tests unitaires de la classe **SudokuGrid**
- Dans le terminal, se positionner dans le dossier concerné, et exécuter "**composer install**" afin de télécharger PHPUnit
- Créer un fichier **index.php** à la racine qui vous permettra d'instancier les classes, charger les grilles et tester leur résolution. Ce fichier fera l'affichage à l'utilisateur dans le terminal.
- Pour exécuter les tests unitaires et vérifier que votre classe **SudokuGrid** est correcte, exécuter cette commande à la racine du dossier : **./vendor/bin/phpunit --bootstrap vendor/autoload.php tests**

Votre projet devra comporter les points suivants :

- Fonctions utilitaires :
 - Chargement d'une grille à partir d'un fichier JSON
 - Affichage de grille
 - Getter/Setter de valeur
 - Getter de ligne, colonne et bloc
 - Parcours de grille
 - Test de validité d'une valeur pour une cellule
 - Test de validité de la grille
- Algorithme de résolution
 - Récursivité
 - Test de toutes les permutations (brute force)

IV. Rendu

Ce travail sera à rendre via **360 Learning**, au plus tard 2 semaines après la dernière séance de cours. Aucune exception ne sera tolérée (pas de code par zip, pas d'envoi par mail, pas de rendu au-delà de la limite).

V. Critères de notation

Vous serez évalués sur un total de 20 points, selon le barème suivant :

Fonctionnalité		Valeur en points
Fonctions	Chargement de grille	1
	Affichage de grille	1
	Getter de valeur	1
	Setter de valeur	1
	Getter de ligne	1,5
	Getter de colonne	1,5
	Getter de bloc	2
	Parcours de grille	2
	Test de validité d'une valeur dans une cellule	1,5
	Test de validité de la grille	1,5
Algorithme	Récurtivité	3
	Brute force	3
Total		20

N.B. : Pas de demi-points pour une fonctionnalité partiellement OK

N.B. 2 : La beauté et l'esthétisme étant des critères subjectifs vous ne serez pas évalués sur ce point, seulement sur le respect des fonctionnalités