

# TP n4 : Sudoku

---

Le but de ce TP est de créer un jeu qui permet de jouer au Sudoku sur l'ordinateur. Pour cela, on va découper le problème en différents sous-problèmes, en utilisant le découpage par objet. Le projet se rapproche beaucoup du **TP 15** que l'on a fait en C++, donc pensez à réutiliser le code que vous avez fait (et de le mettre au bon endroit)...

Nous allons utiliser une classe qui s'appelle grille :

Grille
- plateau : int[9][9] - nombresNonModifiable : bool[9][9]
+ Grille() : void + Grille(string) : void + afficheGrille() : void + modifieGrille() : void + grillePleine() : bool

## Première partie

Avant de se lancer tête baissé dans le codage de la classe, répondez aux questions suivantes (je passerais voir si vous y avez répondu...) :

- Pourquoi avoir deux tableaux : l'un nommé `plateau`, et l'autre `nombresNonModifiable` ?
- Quel lien existe-t-il entre les deux tableaux et l'initialisation ?

Comme vous l'avez remarqué, il y a deux constructeurs pour cette classe : l'un qui ne prend pas de paramètre et l'autre qui prend un string... Dans le premier constructeur, vous demanderez à l'utilisateur de rentrer les chiffres initiaux et leurs positions dans la grille, tandis que dans le constructeur avec un argument, vous chargerez la grille à partir d'un fichier. Je vous conseille de faire un fichier de ce type :

posX1 posY1 valeur1  
posX2 posY2 valeur2  
posX3 posY3 valeur3  
....



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

Pensez bien à remplir le tableau `plateau` et de mettre à jour le tableau `nombresNonModifiable` pour que l'utilisateur ne puisse pas modifier les nombres initiaux.

## Seconde partie

Maintenant que la grille est initialisée, il faut pouvoir l'afficher. Pour cela, vous implémenterez la fonction `afficheGrille()`.

De plus, il faut pouvoir mettre des chiffres dans les cases... Dans la fonction `modifieGrille()`, vous commencerez par demander à l'utilisateur la coordonné du nombre à insérer. Ensuite, il y a plusieurs choses à vérifier :

- La coordonnée n'est pas sur un chiffre non modifiable (utilisez pour cela le deuxième tableau)
- Les coordonnées ne "sortent" pas du tableau (elle doit être comprise entre 0 et 8 (les tableaux commencent à 0!))

Si ces deux conditions sont réalisées, vous demandez alors à l'utilisateur un nombre. Il y a là aussi plusieurs choses à vérifier :

- Le nombre qu'il veut entrer est compris entre 1 et 9
- Il n'existe pas déjà ce nombre sur la même ligne (créez et utilisez la fonction `bool existeLigne(int numLigne, int numero)` qui renvoie `vrai` si il existe une case avec la valeur `numero` sur la ligne `numLigne` du tableau).
- Il n'existe pas déjà ce nombre sur la même colonne (créez et utilisez la fonction `bool existeColonne(int numCol, int numero)` qui renvoie `vrai` si il existe une case avec la valeur `numero` sur la colonne `numCol` du tableau).
- Il n'existe pas déjà ce nombre dans le même carré (vous n'êtes pas obligé de le faire...)

Si toutes ces conditions sont réalisées, alors vous mettez le chiffre dans la grille. Sinon, à vous de voir : soit vous ne faites rien en affichant un message d'erreur, soit vous redemandez à l'utilisateur de saisir les infos...

Enfin, faites la fonction `grillePleine` qui teste si la grille est pleine. Soit vous testez qu'il n'y a pas de cases à 0 dans le tableau, soit vous faites la somme de toutes les cases, et si le total est égal à 405, c'est que c'est fini...

## Troisième partie

Maintenant que l'on a une classe qui marche bien (ou du moins, on l'espère...) faites un `main` où vous créez une partie, puis vous proposerez de charger un plateau à partir d'un fichier ou de créer un plateau de toute pièces. Ensuite, vous demanderez à l'utilisateur de mettre un chiffre dans la grille, jusqu'à ce qu'il remplisse le tableau.

## Quatrième partie

Ajoutez à la classe une fonction `enregistrePlateau` qui permet de sauvegarder les valeurs de l'utilisateur et la fonction `chargePlateau` qui permet de charger une partie précédemment enregistrée.