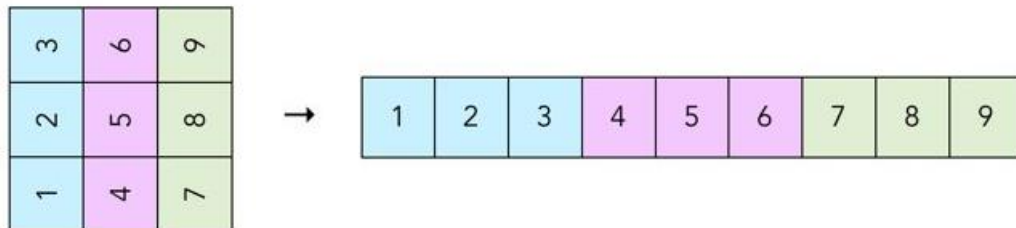


1. Créé une énumération Jeton avec les valeurs Vide, Jaune et Rouge.
2. Créé la classe Grille, qui va correspondre à la grille de jeu. Par soucis de simplicité et de performance, au lieu d'utiliser un tableau en 2 dimensions pour modéliser cette grille, on va utiliser un tableau en 1 dimension. Ainsi, les 6 première case vont correspondre aux 6 lignes de la première colonne, les 6 suivante aux cases de la deuxième colonne ...



- Attribut `_grille`, qui est un tableau de Jeton avec une taille de 42 (qui correspond à 6*7, car il y a 6 lignes et 7 colonnes)
- Un constructeur par défaut, qui initialise le tableau avec des jetons Vide
- La fonction `void put(int colonne, Jeton jeton)`, qui va venir mettre le jeton passé en paramètre en haut de la colonne passé en paramètre.
- La fonction `Jeton get(int pos)`, qui retourne la couleur du jeton à la position passée en paramètre.
- La fonction `void show()` qui permet d'afficher la grille sur le terminal, les Jeton vide seront représenté par `_`, les Jaunes par `o` et les Rouge par `x` : Voici l'apparence qu'elle doit avoir :

```
|_|_|_|_|_|_| |
|_|_|_|_|_|_|
|_|_|_|_|_|_|
|_|_|_|_|_|_|
|_|_|_|_|_|_|
|_|_|_|_|_|_|
|_|_|_|_|_|_|
|0|1|2|3|4|5|6|
```

Pour une grille vide

```
|_|_|_|_|_|_| |
|_|_|_|_|_|_|
|_|_|_|_|_|_|
|_|_|_|x|_|_|
|_|_|o|_|o|_|_|
|_|_|o|x|x|_|_|
|0|1|2|3|4|5|6|
```

Pour une grille partiellement remplie

3. Créé la classe Joueur :
 - Un attribut `nom` de type `string`
 - Un attribut `couleur` de type `Jeton`
 - Un constructeur qui prend en paramètre un `string` et un `Jeton`, pour initialiser les deux attributs.
 - Une fonction `string getNom()`
 - Une fonction `Jeton getCouleur()`
4. Créé la classe Jeu, qui va contenir la grille de jeu, les deux joueurs, et ce que l'on appelle le gameloop.
 - Un attribut `grille`, de type `Grille`
 - Deux attribut, `joueur1` et `joueur2`, de type `Joueur`

- Un attribut `joueurCourant` de type `*Joueur`
 - Un constructeur par défaut, qui initialise la grille de jeu, et va demander à chaque joueur leurs noms pour pouvoir initialiser les deux joueurs. On initialise aussi le pointeur `joueurCourant` sur le premier joueur.
 - Une fonction void `loop()`. Cette fonction va d'abord afficher la grille de jeu, demander au joueur courant dans quelle colonne il souhaite mettre son jeton, puis changer de joueur, afficher la grille, et demander dans quelle colonne il souhaite jouer. Puis on redémarre la boucle.
5. On va maintenant programmer la condition de victoire. Dans la classe Grille. On va créer la fonction `int CheckUp(int pos, Jeton jeton)`. Cette fonction sera récursive, c'est-à-dire que dans le corps de cette fonction, elle va s'appeler elle-même. Le but de cette fonction sera de retourner le nombre de jeton identique qu'il y a au-dessus de la position passée en paramètre. Voici le pseudo code de cette fonction :

```

Int CheckUp(int pos, Jeton jeton){
    If(jeton == grille.get(pos))
        Return 1 + CheckUp(pos+1,jeton)
    Else
        Return 0
}

```

Fait la même chose pour les 7 autres directions.

6. On peut maintenant faire la fonction de vérification de victoire. Dans la classe `loop`, rajoute la fonction `bool victoire(int pos, Jeton couleur)`. Elle va venir cumuler la somme des deux `check` dans les sens opposés (`checkUp + checkDown`), si c'est 4 ou plus, le joueur à gagner.