**Travail Pratique 1**

**Serpent et Échelles**

Votre tâche est de créer un jeu de « Serpent et échelles ».

**Classes fournies :**

Les classes sont déjà créées, et les signatures des méthodes sont faites. Il vous reste à compléter le contenu des méthodes.  Vous ne pouvez pas modifier les méthodes qui vous sont fournies, mais vous pouvez créer plus de méthodes au besoin.

Les classes sont :

**Tableau de jeu** : Cette classe représente l’objet tableau de jeu, représenté par un tableau de 100 objets de type Tuile.

**Tuile** : Chaque case du tableau de jeu. Chaque case est numérotée par un **int**, qui indique son numéro de case entre 0 et 99.  
Les tuiles sont de 2 types, false ou true, indiqué par un booléen. Les tuiles de type false sont des tuiles « normales »,   sans serpent ni échelle.

Les tuiles true ont soit un serpent, soit une échelle. La destination de celle-ci est notée dans la variable de classe : « destination ». Pour une tuile de type false, la variable destination est laissée à 0.

**Joueur** : Un joueur. Cette classe a deux variables de classe : le nom du joueur et sa position. La position du joueur est un int qui indique la case du jeu sur laquelle il se trouve. Son nom est un int séquentiel (le premier joueur est 0, le 2e est 1 etc.). Une variable statique permet d’assurer la numérotation des joueurs.

Serpent et échelles : La classe principale du jeu. La méthode main simulera une partie.  Elle effectue les actions suivantes :

1)     Le nombre de joueurs voulu est envoyé au programme au moment de lancer l’exécution. Le programme le récupère donc dans args. Il n’y a donc pas d’interaction avec l’usager. Une exception est lancée si le paramètre passé au programme n’est pas un nombre entre 1 et 6.

2)     Les joueurs requis sont créés, et ajoutés à une liste chainée de joueurs. Vous pouvez utiliser la CircularlyLinkedList fournie avec le manuel de Goodrich et al., ou bien une liste chainée de Java.util. (Si vous utilisez la classe de Goodrich, vous pourrez devoir supprimer le nom du package, ou bien créer un package avant d’y placer la classe).

3)     Crée un tableau de jeu

4)     Crée une référence vers le premier joueur.

5)     Tant qu’aucun joueur n’a atteint ou dépassé la case 99, la boucle principale du jeu ​continue. En fait, si un joueur dépasse la case 99, les règles du jeu spécifient qu’il « rebondit » et recule dans l’une des cases 95-99.

**Travail à remettre :** Chacune des classes complétées, à remettre sur Moodle avant le 22 oct. (dans une archive zip). Inclure aussi un fichier noms.txt qui contient le(s) nom(s) des co-équipiers qui ont effectué le TP. Vous êtes libre de travailler seul, ou en équipe de 2 (maximum ferme).