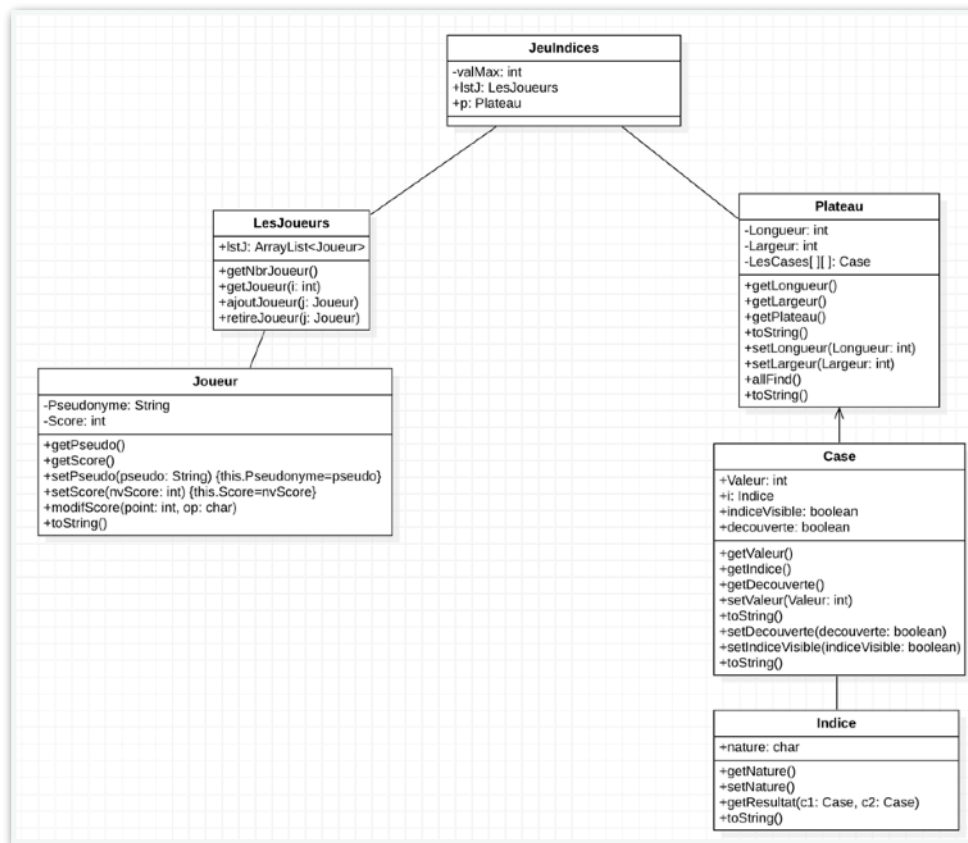


# Pré-rapport



- Le jeu des indices sera construit autour d'une classe principale **JeuIndices** qui permettra de mettre en place le jeu, on y définit une valeur maximum lors d'un tirage de type entier, créer la liste de type **LesJoueurs** qui « stockera » les joueurs ainsi que le plateau de jeu d'instance **Plateau**.
- La classe **LesJoueurs** permet de regrouper et de rajouter dynamiquement des joueurs grâce à une **ArrayList** qui regroupera donc des instances de type **Joueur**. Différentes méthodes permettent d'interagir avec cette **ArrayList** ainsi que les getters/setters pour accéder et définir les attributs.
- Une classe **Joueur** définit un joueur par son Pseudo de type **String** et de son score de type **int**. Une méthode **modifScore** permettra d'ajout ou de retirer des points du score du joueur en fonction de l'opérateur de type **char** rentré en paramètre. Une méthode **toString** permet de retourner les informations du joueur sous forme d'une chaîne de caractères.
- L'objet **Plateau** est composé de deux entiers pour les dimensions du plateau, plateau qui est modélisé grâce à une matrice (de taille définit par les attributs précédents) composé d'instances de type **Case**. Il ya bien évidemment les getters/setters. Une méthode **allFind** permet de parcourir la matrice de **Case** pour vérifier si elles ont tous été découverte permettant donc d'arrêter le jeu en conséquence. Une méthode **toString()** permet d'afficher une représentation du plateau grâce à deux boucles **for** pour parcours en longueur et largeur la matrice et afficher les Cases avec leur méthode **toString()**.

- Une case est composé d'une valeur qui devra être trouvé par le joueur ainsi que d'un indice qui pourra être afficher ou non selon un attribut IndiceVisible configuré en début de partie par le joueur. Un boolean permettra à l'instance Plateau de savoir si la case a été découverte et donc d'agir sur le jeu. Une méthode toString permet donc d'afficher la valeur de la case si découverte ainsi que son indice si configurer.
- La class indice est composé d'un attribut de type String(corrigé après le diagramme) pour stocker l'opérateur, des getters/setters pour configurer l'indice. Une méthode alea() permettrait d'attribuer un opérateur aléatoire choisis dans un tableau composé de tout indices disponibles. La méthode getResultat permettra de retourner l'indice qui aidera le joueur à trouver la valeur caché grâce à un switch qui récupérera l'indice associé à une des cases choisis par l'utilisateur et calculera en fonction de celui-ci.