

## Projet rapport intermédiaire

### Diagramme de classes



— — — — — lien entre les classes

→ est une sous classe de

## Descriptif des classes

### La classe PartieDeJeu( ) :

- Cette classe conciste à regrouper tout les éléments nécessaire à une partie de jeu.
- Elle initialise les joueurs ; le plateau...
- Elle dicte les tours de jeu...
- Pour cela il faut créer d'autre classes qui se chargerons des méthodes associer et créer différents objets qui caractérise une partie de jeu.
- L'idée est que quand on créer un objet PartieDeJeu dans le main c'est comme si l'on appui sur un bouton play, ce qui veut dire que dans le constructeur il y aurai trois on quatre méthodes comme par exemple pour :
  - intialiser les joueurs de la partie – initJoueur()
  - initialiser le plateau – initPlateau()
  - faire des tours de jeu – Jouer()...

### La classe Plateau( ) :

- Cette classe à pour but de créer un plateau de jeu. Un plateau de jeu est un tableau de Case.
- on caractérise un plateau de jeu par : le nombre de ligne du plateau son nombbre de colonne, un tableau de cases puis la case courante que le joueur choisi de prendre lors de sa sélection de case au cours de la partie.
- On créer un constructeur par défaut ou le nombre de ligne et de colonne est fixé puis on décrit notre tableau ou à chaque "emplacement" on associe une Case.
- On créer également un contructeur standard avec en paramètrre le nombre de lignes et de colonnes de notre tableau, pour que le joueur puisse choisir ces préférences de jeu s'il le souhaite.
- Une méthode affichePlateau() est écrite pour afficher le plateau de jeu dans la console.
- Une méthode getCasePlateau() est écrite pour donner la case courante.
- Une méthode getCaseDecouverte() est écrite pour dire si toute les cases du plateau sont découvertes.

### La classe Indice( ) :

- La classe indice sera une classe abstraite qui définira ce qu'est un indice dans sa globalité.
- On caractérise l'Indice par : un nombre entier et deux cases du plateau (celle sélectionner par le joueur).
- Elle aura une méthode abstraite donneIndice().
- Elle admetra des sous classes qui représenterons un indice particulier par exemple une somme ou une multiplication.

### La classe IndiceSomme( ) :

- La classe IndiceSomme() est une sous classe de la classe abstraite Indice().
- On rajoute une caractérisation pour cette classe qui est le signe « + », on peut donc reprendre le constructeur de la classe Indice() et rajouter un attribut signe.

→ On a une méthode `donneIndice()` rendu obligatoire par la classe `Indice()`.

→ Une méthode `toString()` est écrite pour l'affichage de notre objet.

### La classe `Case()` :

→ Cette classe à pour but de représenter une case de notre plateau de jeu.

→ On caractérise une case par un entier (qui est la valeur à découvrir au long du jeu) et un boolean qui nous dit si le joueur a découvert cette case.

→ Deux constructeurs sont créés, un par défaut où la valeur de la case est donnée aléatoirement entre 0 et 9 et un standard où le joueur s'il souhaite peut choisir la valeur maximum des cases et donc la valeur à trouver sera entre 0 et cette valeur maximale.

→ Des accesseurs en écriture et en lecture...

→ Une méthode `toString()` pour afficher notre objet comme on le souhaite...(découvert ou non).

### La classe `Joueur()` :

→ Cette classe à pour but de représenter un joueur de la partie.

→ Un joueur est caractérisé par une chaîne de caractère qui est un pseudo et un nombre entier, son score.

→ Deux constructeurs un par défaut et un standard pour le joueur puisse avoir un pseudo choisit par l'utilisateur.

→ Une méthode `tourDeJeuJoueur()` est écrite pour demander si le joueur veut faire une supposition sur les valeurs des deux cases et si elles sont justes (les deux cases sont passées en paramètres de la méthode).

→ Une méthode `toString()`.

### La classe `LesJoueur()` :

→ Cette classe a pour but de gérer une liste de joueurs pour avoir plusieurs participants lors de la partie.

→ Elle peut les ajouter les retirer, rechercher un joueur en particulier...

→ Méthode `toString()` pour afficher la liste des joueurs créer.