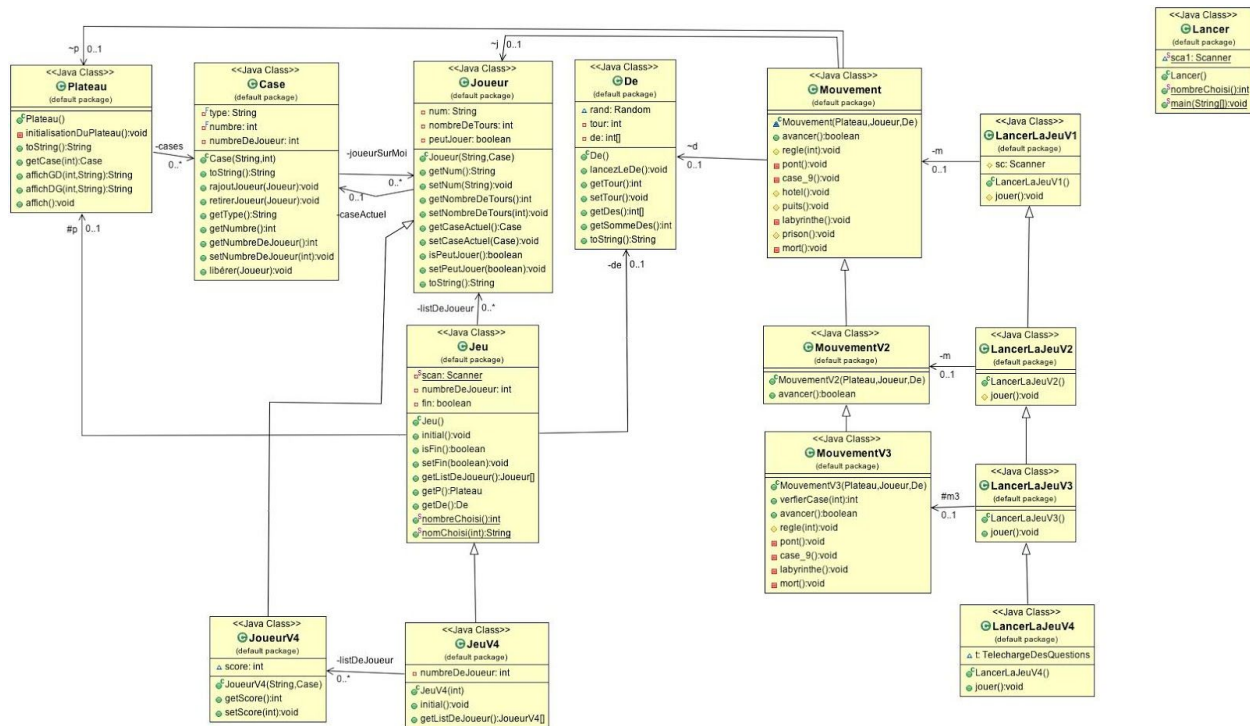


Tout d'abord, excusez-moi pour mon français, qui n'est pas encore très bon (je suis un étranger).



Mon programme est basé sur cinq classes qui ont fait ensemble la partie modèle, ces classes sont : Case, Plateau, De, Joueur, Jeu.

1. Classe Case : continent les variables suivant:

- final String type : le type de case (le type de boîte, si c'est une case normale ou un case special : Puits, Hôtel,...)
- final int nombre : contient le nombre de la case;
- LinkedList<Joueur> joueurSurMoi: liste chaînée de objet de type Joueur représenter les jouer sur cette case.

et elle contient aussi le méthodes suivantes:

- constructeur public `Case(String s, int n)`, qui initialise les valeur final de type, nombre.
- `ajouterJoueur(Joueur j)` qui ajoute le joueur j passé en paramètre dans la list `joueurSurMoi`.
- `retirerJoueur(Joueur j)` qui retire le joueur j passé en paramètre de la list `joueurSurMoi` de cette case
- `getType()` qui retourne le valeur de la variable type.
- `getNombre()` qui retourne le valeur de la variable nombre.
- `getNombreDeJoueur()` qui retourne le size de la list `joueurSurMoi` (le nombre de joueur sur cette case).
- `surMoi()` qui retourne le size de la list `joueurSurMoi` list de joueur sur cette case.
- `liberer()` qui redéfinit la valeur de la variable `peutJouer` de chaque joueur sur cette case à `true`;

2. Plateau: continent une variable:

- final ArrayList<Case> cases: liste chaînée de objet de type Case, représenter les case de plateau de la jeu.

et elle contient aussi le méthodes suivent:

- constructeur public Plateau() qui appelle la méthode privé initialisationDuPlateau().
- initialisationDuPlateau() qui initialise les cases de list cases, par imbriquer une boucl for (entre 0 et cases list size) et boucl switch pour donner le bon type de chaque case.
- getCase(int n) qui retourne la case numéro n de list cases.
- affichGD(int i, String s) qui ajoute la valeur de toString méthode appelé sur chaque case entre i + 1 et i + 8 dans le string s (ajoute un line de case de gauche à droite) .
- affichDG(int i, String s) qui ajoute la valeur de toString méthode appelé sur chaque case entre i + 7 et i dans le string s (ajoute un line de case de droite à gauche) .
- affich() qui appelle le deux méthodes précédent dans un boucle for pour ajouter la valeur de toString méthode appelé sur chaque case de tout list cases dans le variable s et affiche le sur la console en zigzag.

3. De: continent les variables suivent:

- Random rand: un object de type random.
- private int tour: initialise en 0, pour compter le nombre de tours qui ont joué.
- private int[] de: initialise en { 0, 0 }, qui tenir la valeur de deux de a chaque lancer.

et elle contient aussi le méthodes suivent:

- lancezLeDe() qui génère une valeur entre 1 et 6 de chacun des aléatoirement en utilisant la méthode nextInt de rand.
- getTour() retourne le nombre de tour qui ont joué.
- setTour() incrément la valeur de tour.
- getDes() retourne l'objet de.
- getSommeDes(): retourné le somme de valeur de deux des.
- toString affiche les valeur de des.

4. Joueur: continent les variables suivent:

- private String mon: le nom de joueur.
- private int nombreDeTours = 0; nombre de tour laquelle le joueur a déjà joué.
- private Case caseActuel la case ou le joueur est positionné.
- private boolean peutJouer = true; si le joueur à le droit a jouer ou pas.

et elle contient aussi le méthodes suivent:

- constructeur public Joueur(String n, Case c) qui initialise le nom et la caseActuel et appelle la méthode ajouterJoueur(this) sur la caseActuel pour ajouter le joueur dans la list de jour joueurSurMoi de la caseActuel.
- les méthode getter et setter pour toutes les variable de class.

5. Jeu: continent les variables suivent:

- private static Scanner scan = new Scanner(System.in); pour recevoir des réponses des utilisateurs du programme.
- private int nombreDeJoueur; le nombre de joueurs qui souhaite jouer.
- protected Plateau p = new Plateau(); objet plateau.
- private De de = new De(); objet de.
- private Joueur[] listDeJoueur; list d'objets joueur.
- private boolean fin = false; di la jeu est fini ou pas.

et elle contient aussi le méthodes suivent:

- constructeur public Jeu() qui appelle la méthode initial.
- initial() qui initialise les variable nombreDeJoueur et listDeJoueur en utilisant les deux méthode nombreChoisi() et nomChoisi()
- nombreChoisi() et nomChoisi() communiquer avec les utilisateurs.

après il y'a la parti controller de mon programme qui consiste en trois classes :

1. Class Mouvement c'est le base de logique et les règles de mouvement des joueurs sur le plateau.
2. Class LancerLaJeu qui l'a un boucle while qui gère la continuation de la jeu le temps qui l'est pas fini.
3. Class Lancer qui contient la méthode main et qui gère le lancement de programme et la variante choisi par l'utilisateur.

Malheureusement, j'avais eu plus de temps pour faire le jeu graphique et le jeu No Nomery (je travaille à temps plein)

Merci.