

- dans `Vide`, `toString()` renvoie "."
- dans `Mur`, `toString()` renvoie "#"
- dans `Balle`, `toString()` renvoie "o"

Les cases `Balle`, en plus d'une *position*, possèdent une *direction* car elles seront amenées à se déplacer.

La direction sera codée par une classe `Direction`, sous la forme d'un couple d'entiers, chacun d'eux valant soit 1, 0 ou -1. Ce codage permettra dans le projet d'indiquer la variation à appliquer à la composante correspondante de la position : la valeur 1 indique que la valeur (en x ou en y) de la position doit être augmentée, la valeur -1 qu'elle doit être diminuée, et la valeur 0 qu'elle doit rester inchangée. Par exemple, avec un point de référence situé en haut à gauche, la direction *nord* est codée par le couple $(0, -1)$, et la direction *sud* par le couple $(0, 1)$. La direction *ouest* est codée par le couple $(-1, 0)$. La direction *sud-est* est codée par le couple $(1, 1)$. Etc.

Classe Damier

Un damier peut être instancié avec un nombre de lignes et de colonnes fournis en paramètre. La matrice de `Case` est un attribut de cette classe.

Les bords du damier sont des murs. L'intérieur est constitué de cases vides, sauf pour certaines qui sont des murs ou des balles disposés aléatoirement.

Une méthode `getRandomFreePosition()` doit permettre de retourner une `Position` correspondant à une case vide du damier, et choisie aléatoirement parmi les cases libres.

Une méthode `setRandMur()` permet de positionner une case `Mur` au hasard (selon une position choisie par `getRandomFreePosition()`). Même chose pour une case `Balle` avec une méthode `setRandBalle()`.

Enfin, le `toString()` de `Damier` construit une chaîne représentant le damier. Il suffit pour cela de parcourir le damier en invoquant le `toString()` de chaque case, qui par polymorphisme renvoie la chaîne la représentant. N'oubliez pas d'insérer dans la chaîne le caractère de saut de ligne `'\n'` à la fin de chaque ligne du damier.

Programme principal

Votre programme principal consistera simplement à instancier un damier, à y disposer aléatoirement des cases de type `Mur` et des cases de type `Balle`, puis à l'afficher (en affichant la chaîne construite par son `toString()`).

Par exemple, pour obtenir l'image du damier présenté en exemple au début de ce sujet, on a créé un damier de 22 lignes et 32 colonnes. Comme les bords sont des murs, il reste un espace de 20 lignes par 30 colonnes pour ajouter des obstacles (`Mur`) et des balles. Dans cet espace, on a disposé aléatoirement 30 cases de type `Mur` et 10 cases de type `Balle`. Les autres cases sont des cases de type `Vide`.