

Développement orienté objet

TD5 : Bataille navale

A la bataille navale, le plateau de jeu est représenté par une grille rectangulaire de cases sur lesquelles on peut poser des bateaux. Les bateaux sont larges d'une case et longs d'un nombre variable de cases. Ils peuvent être posés verticalement ou horizontalement sur le plateau.

Le plateau est masqué au joueur qui *attaque* et celui-ci doit couler tous les bateaux du joueur qui *défend*. Pour cela, il propose une position du plateau qui désigne la case sur laquelle il *tire*. Plusieurs cas sont possibles :

- la case est vide : le défenseur annonce "dans l'eau"
- dans le cas contraire :
 - le défenseur annonce "touché" si toutes les cases occupées par le bateau ne sont pas touchées
 - le défenseur annonce "coulé" si toutes les cases occupées par le bateau sont touchées
- lorsqu'une case avait déjà été visée précédemment, la réponse est toujours dans "dans l'eau"

Diagramme de classes

- L'objectif premier du TD est de représenter les relations entre les différentes classes via un diagramme de classes UML utilisant les associations.
- Vous êtes encouragé à construire le diagramme de classes au fur et à mesure de l'avancement du TD.
- Si vous êtes bloqués par une implémentation, vous pouvez passer à la suite en laissant la méthode vide.
- Il faut qu'il y ait une cohérence entre votre diagramme de classes et votre implémentation.

Les bateaux

La classe **Bateau** permet de définir les données sur les bateaux. Un objet **Bateau** est défini par sa longueur. La longueur du bateau détermine son nombre de *points de vie* et est fixée à la création de l'objet. Lorsqu'il est touché (méthode **êtreTouché**), ce nombre diminue de 1. Un bateau est coulé quand son nombre de point de vie arrive à 0 (méthode **estCoulé** permet de savoir si c'est le cas). Le diagramme de classe de la classe **Bateau** est :



Question 1

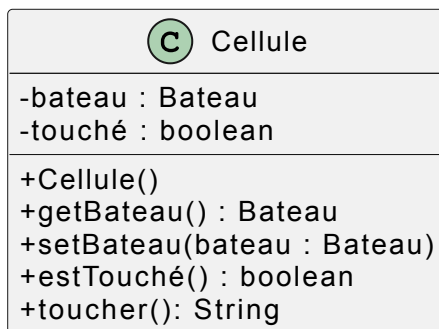
Donnez une implémentation de la classe **Bateau** conforme au diagramme ci-dessus.

Le plateau de jeu

La classe représentant le plateau de jeu s'appelle **Mer**. On décide de la représenter par un tableau à deux dimensions composé d'objet de type **Cellule**.

La classe **Cellule**

La classe **Cellule** permet de représenter une case du plateau de jeu. Le diagramme de classe initialement proposé est :



Indication : la méthode **toucher** retourne la chaîne de caractères "dans l'eau", "touché" ou "coulé" selon le cas.

Question 2

Comment proposez-vous de gérer la présence ou l'absence d'un bateau dans une cellule ?

Question 3


Donnez le code pour la méthode **toucher** de la classe **Cellule**.

Question 4

Implémentez la classe **Cellule** en tenant compte de votre réponse à la question 2.

Positions

Pour représenter les coordonnées des cases du plateau de jeu, on définit la classe **Position** dont le diagramme de classe est :

 Position
-x : int -y : int
+Position(x : int, y : int) +getX() : int +getY() : int +égal(p : Position) : boolean

Question 5

Donnez une implémentation de la classe **Position** en TypeScript.

La classe **Mer**

On doit définir maintenant la classe **Mer**. Pour rappel, on décide de la représenter par un tableau à deux dimensions composé d'objet de type **Cellule**. On souhaite notamment que :

- les dimensions du plateau sont fixées à la création de l'objet
- la classe **Mer** dispose d'une méthode **ajouterBateau(bateau : Bateau, position : Position)** qui permet d'ajouter une **unité** (un morceau) du bateau à la position indiquée. Pour respecter, les règles du jeu, les autres unités du bateau devront être posées verticalement ou horizontalement dans les cases adjacentes. Il ne vous est pas demandé de vérifier cela (mais vous pouvez le faire si vous avez le temps). Si la position est déjà occupée, il faut que la méthode lève une exception. De même, si la position n'est pas valide (en dehors du plateau), il faut que la méthode lève une exception.
- La classe **Mer** dispose d'une méthode **tirer(position : Position)** qui permet de tirer sur la case indiquée. La méthode retourne la chaîne de caractères "dans l'eau", "touché" ou "coulé" selon le cas. Si la position n'est pas valide (en dehors du plateau), il faut que la méthode lève une exception.

Question 6

Compléter votre diagramme de classes pour ajouter la classe **Mer**.

Question 7

Donnez une implémentation de la classe **Mer** en TypeScript.