

TP JAVA n°2

Héritage

Exercice 1 : Emission musicale

On veut modéliser des émissions musicales radiophoniques. Pour ceci, on va définir les classes **Chanson**, **Emission** et **EmissionCommentee**.

Une **Chanson** est caractérisée par son titre (une chaîne de caractères), son texte (une chaîne de caractères) et les nombres de minutes et de secondes (deux entiers) qu'elle dure.

- 1) Écrire le code de la classe **Chanson**.
- 2) Implémenter un **constructeur** qui a 4 paramètres : le titre et le texte de la chanson ainsi que les nombres de minutes et de secondes qu'elle dure.
- 3) Écrire la méthode **void passe()** qui simule le passage d'une chanson (en se contentant d'afficher son texte).
- 4) Donner le code de la méthode **int duree()** qui retourne la durée totale de la chanson en nombre de secondes.
- 5) Générer de manière automatique l'accesseur **String getTitre()** (qui sera utile à la classe **EmissionCommentee**).

Une **Emission** est caractérisée par sa durée en minutes (un entier) et sa "playlist" (la liste des chansons qu'elle passe). La playlist peut être mémorisée dans un tableau dont la taille est égale à la durée de l'émission divisée par 2 (en considérant que toute chanson dure au moins deux minutes). Réfléchir à d'autres caractéristiques nécessaires pour gérer l'émission.

- 6) Écrire la classe **Emission**.
- 7) Implémenter un **constructeur** qui prend en paramètre la durée de l'émission en nombre de minutes.
- 8) Donner le code de la méthode **boolean ajoute(Chanson c)** qui ajoute une chanson c à la playlist.
- 9) Ajouter la méthode **void passe(int i)** qui passe la ième chanson de la playlist.
- 10) Créer la méthode **void passeTout()** qui passe toutes les chansons de la playlist dans l'ordre tant que la durée totale des chansons déjà passées ne dépasse pas la durée de l'émission.
- 11) Écrire l'accesseur **Chanson getChanson(int i)** qui retourne la ième chanson de la playlist (et qui sera utile à la classe **EmissionCommentee**)

Une émission musicale peut ne pas se contenter de passer des chansons. Certaines disposent d'un animateur qui commente l'émission. Nous appellerons cette classe **EmissionCommentee**.

- 12) Donner le code de la classe **EmissionCommentee** qui héritera d'**Emission**.

Pour les questions suivantes, vous devez redéfinir certaines méthodes de classe. **Aucune** nouvelle méthode n'est à définir. Vous ferez en sorte de réutiliser au maximum les méthodes écrites dans les classes précédentes afin d'éviter de réécrire inutilement les instructions déjà présentes dans ces méthodes.

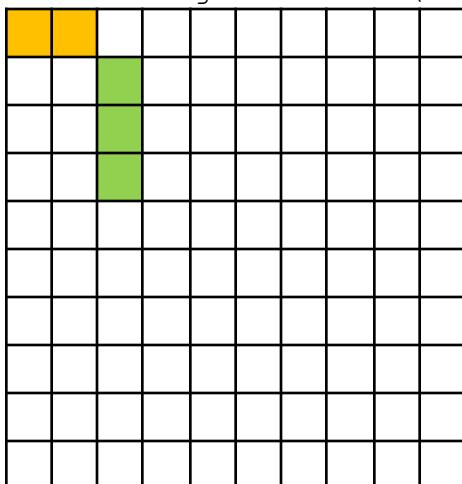
- 13) Faire en sorte que le titre de la chanson soit donné avant son passage.

- 14) Faire en sorte que lorsqu'on passe toutes les chansons, l'animateur salue les auditeurs avant la première chanson (on affiche "Bonjour") et après la dernière chanson (on affiche "Au revoir").
- 15) Ecrire la classe **Main** qui permet l'exécution du programme suivant : deux chansons sont créées, "La javanaise" dont le texte est "J'avoue j'en ai bavé..." et la durée 2m50s et "La java des bombes atomiques" dont le texte est "Mon oncle un fameux bricoleur..." et la durée 3m32s, puis une émission de 15 minutes est créée, à la playlist de laquelle on ajoute les deux chansons, puis on passe toutes les chansons de la playlist.

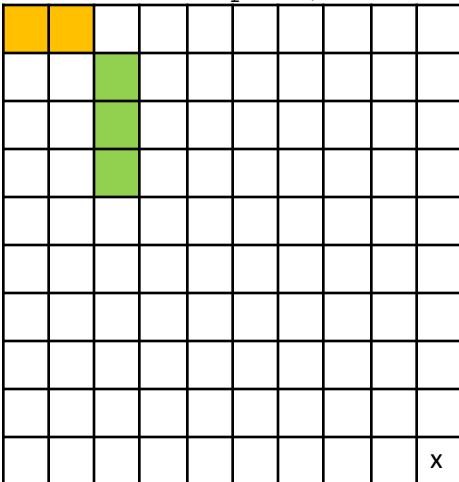
*Exercice 2 : Bataille Navale

On va créer des classes permettant de jouer à la bataille navale. Sur la mer, on dispose un certain nombre de bateaux dans un sens horizontal ou vertical en indiquant les coordonnées (entières) de leur coin supérieur gauche. Il y a quatre types de bateaux : le porte-avions, le croiseur, le cuirassé et le torpilleur, occupant respectivement 5, 4, 3 et 2 cases. Lorsqu'un joueur a joué un coup en indiquant des coordonnées, un bateau peut être touché si une des cases qu'il occupe à ces coordonnées et peut être coulé si toutes ses cases ont été touchées. Après chaque coup, le programme doit afficher "Touché" si un bateau est touché, "Coulé" si c'était la dernière case à toucher pour le couler, et "Gagné" si c'était le dernier bateau à couler (et "À l'eau" si aucune de ces actions ne se produit). Voici un exemple de déroulement de programme (le schéma est juste là pour la compréhension) :

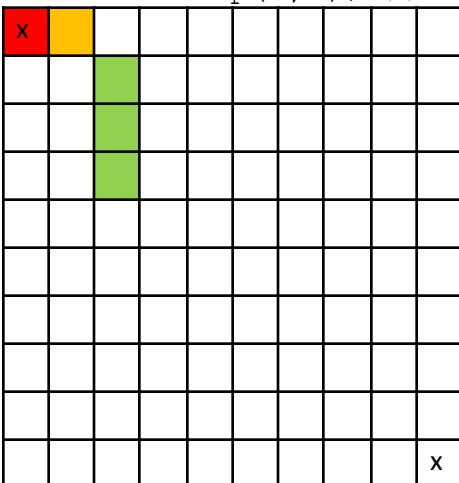
```
Mer mer = new Mer(2); // 2 bateaux à couler
// place un torpilleur horizontalement en (1,1) :
mer.ajouteBateau(new Torpilleur(1, 1, true));
// place un cuirassé verticalement en (3,2) :
mer.ajouteBateau(new Cuirasse(3, 2, false));
```



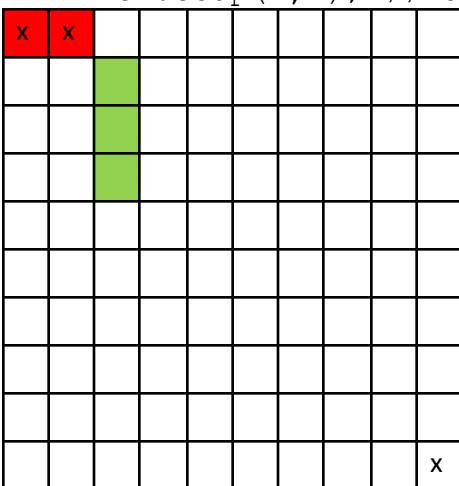
```
mer.coup(10,10); // affiche : A l'eau
```



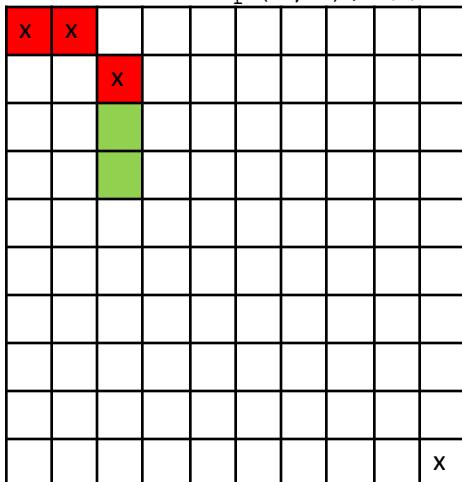
```
mer.coup(1,1); // affiche : Touché
```



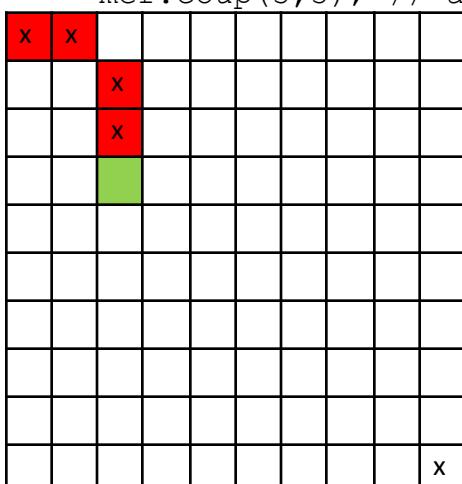
```
mer.coup(2,1); // affiche : Touché Coulé
```



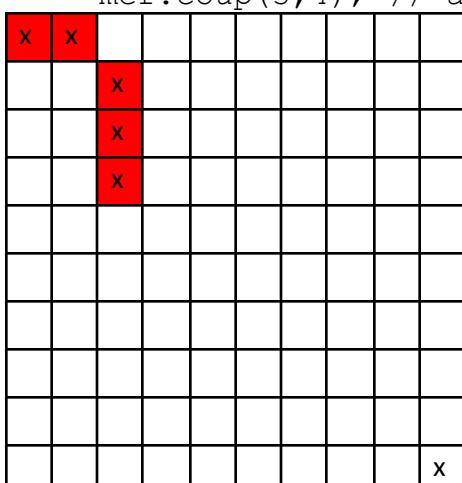
```
mer.coup(3,2); // affiche : Touché
```



```
mer.coup(3,3); // affiche : Touché
```



```
mer.coup(3,4); // affiche : Touché Coulé Gagné
```



- 1) Définir une hiérarchie de classes permettant de modéliser au mieux la mer et les bateaux tels qu'ils ont été décrits et de façon à ce que l'on puisse exécuter la suite

d'instructions donnée ci-dessus. Plus précisément, il vous est demandé de donner l'ensemble des classes à définir, les liens (d'héritage ou de composition) entre les classes, et la liste des méthodes de chacune de ces classes.

- 2) Donner les attributs des classes définies dans la question précédente, et implémenter leurs méthodes. Tester ensuite votre programme.