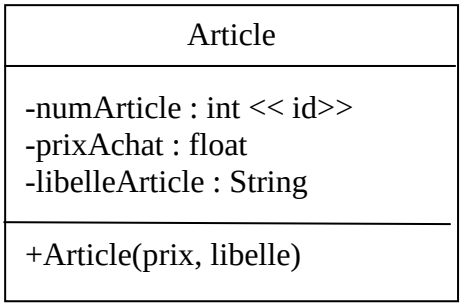
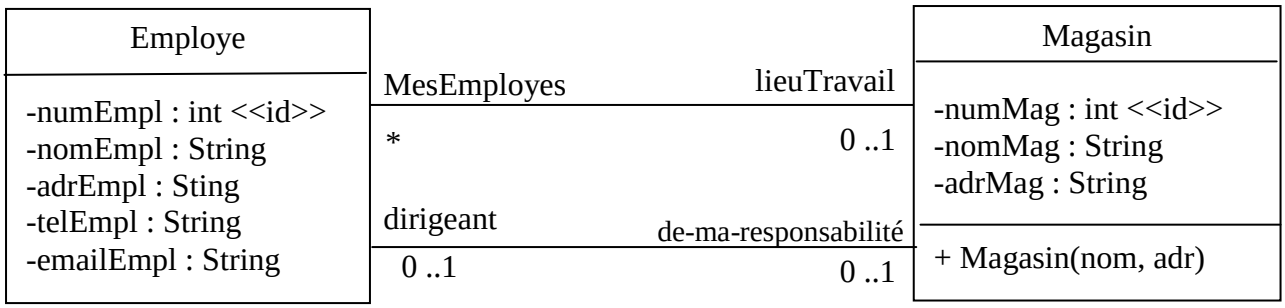
TD6 : Génération de Code en Java

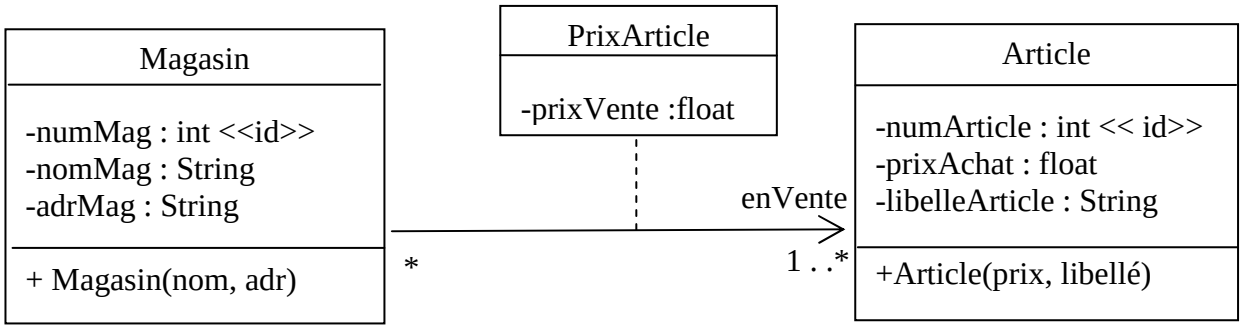
1. Implémentez la classe Article en Java



2. Implémentez la classe Employé et la Classe Magasin.



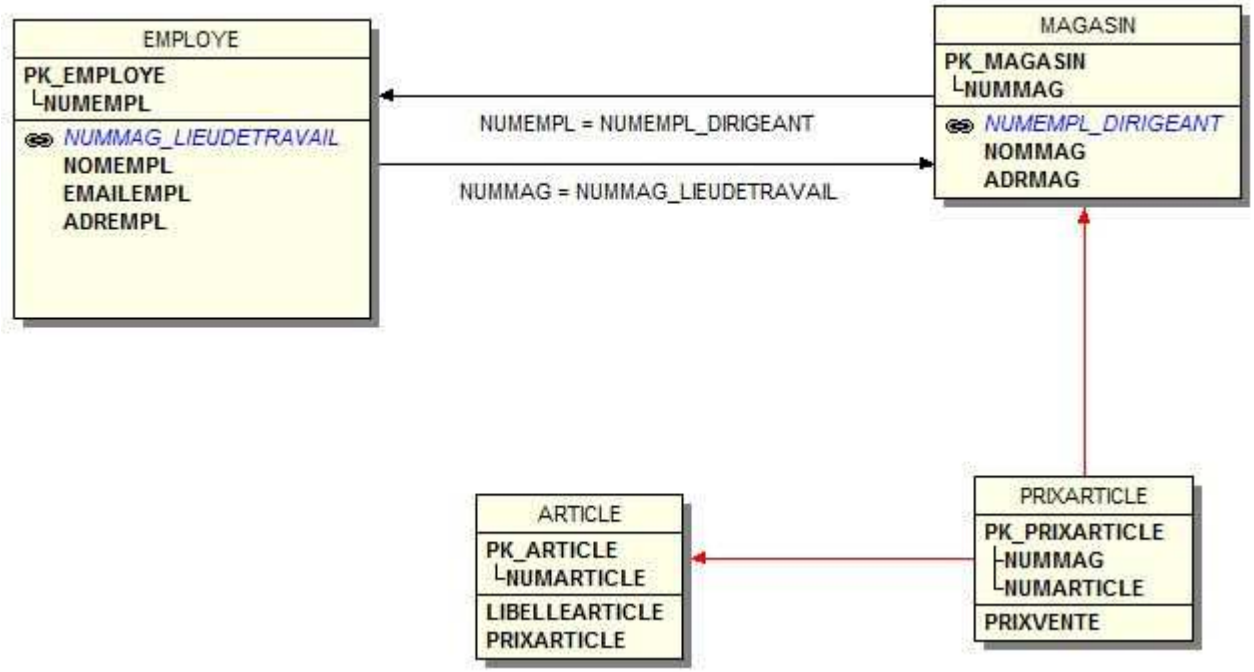
3. Complétez la classe Magasin en Java. Modifiez si nécessaire la classe Article en Java. Proposez une solution pour coder « prix-article ».



4. Votre code permet-il facilement de trouvez le magasin ayant le meilleur prix pour un article donné ?
5. Modifiez votre code et votre diagramme de classes pour faciliter ce calcul.
6. Donnez les inconvénients de votre nouvelle solution.

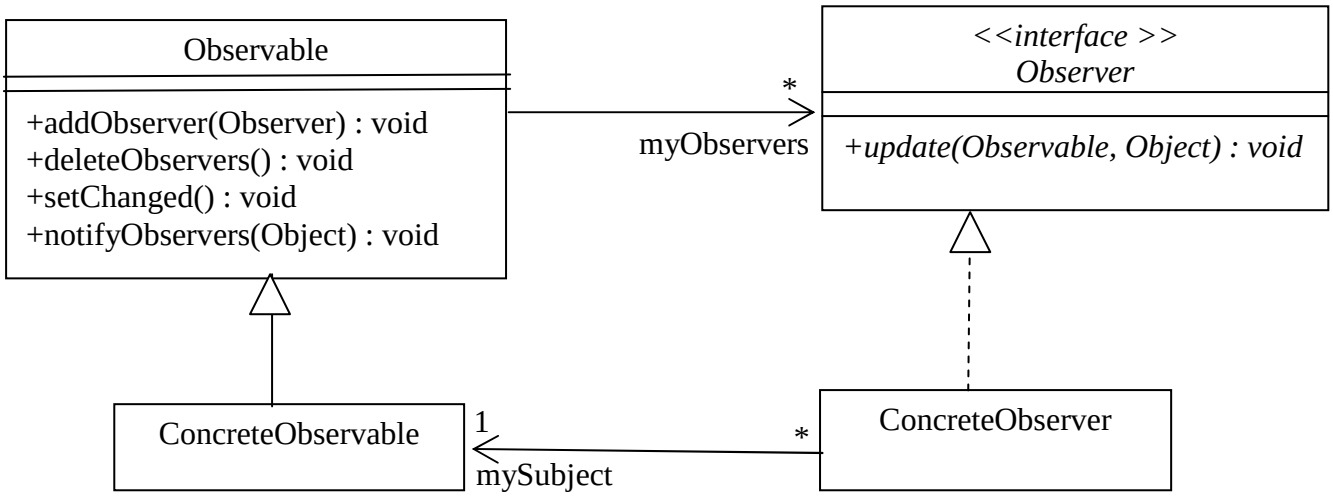
7. Quel est le nom du diagramme suivant ?

a. A quoi correspond la flèche entre PRIXARTICLE et MAGASIN ?
b. A quoi correspond la flèche entre PRIXARTICLE et ARTICLE ?
c. A quoi correspond la flèche d'étiquette « NUMMAG = NUMMAG_LIEUDETRAVAIL »
d. A quoi correspond la flèche d'étiquette « NUMEMP = NUMEMP_DIRIGEANT »

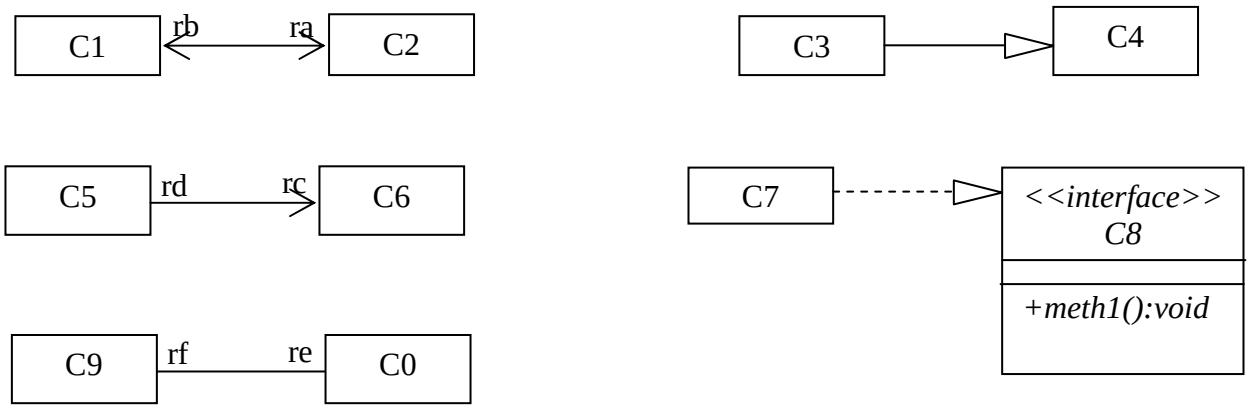


8. Codez les classes Observable, ConcreteSubject et ConcreteObserver en java

Modèle « observateur/observable » : Les observables envoient des notifications à leurs observateurs en cas de changement. En cas de notification, les observateurs exécutent leur méthode update ; ils peuvent obtenir des informations complémentaires par l'appel à la méthode getState de l'observable.



Questions de cours: expliquez, avec le maximum de détails, les différents diagrammes de Classes : expliquez la signification de ra, rb, rc, rd, re et rf. Donnez une idée générale de l'implémentation en Java des classes



Bataille navale

Comme son nom l'indique, la bataille navale est un jeu consistant à torpiller les navires de son adversaire. La flotte de chaque joueur est placée aléatoirement au début du jeu par l'application. Elle est composée :

* 1 porte-avion (5 cases consécutives, non diagonal)
* 2 croiseurs (4 cases consécutives, non diagonal)
* 2 sous-marins (3 cases consécutives)
* 2 torpilleurs (2 cases consécutives).

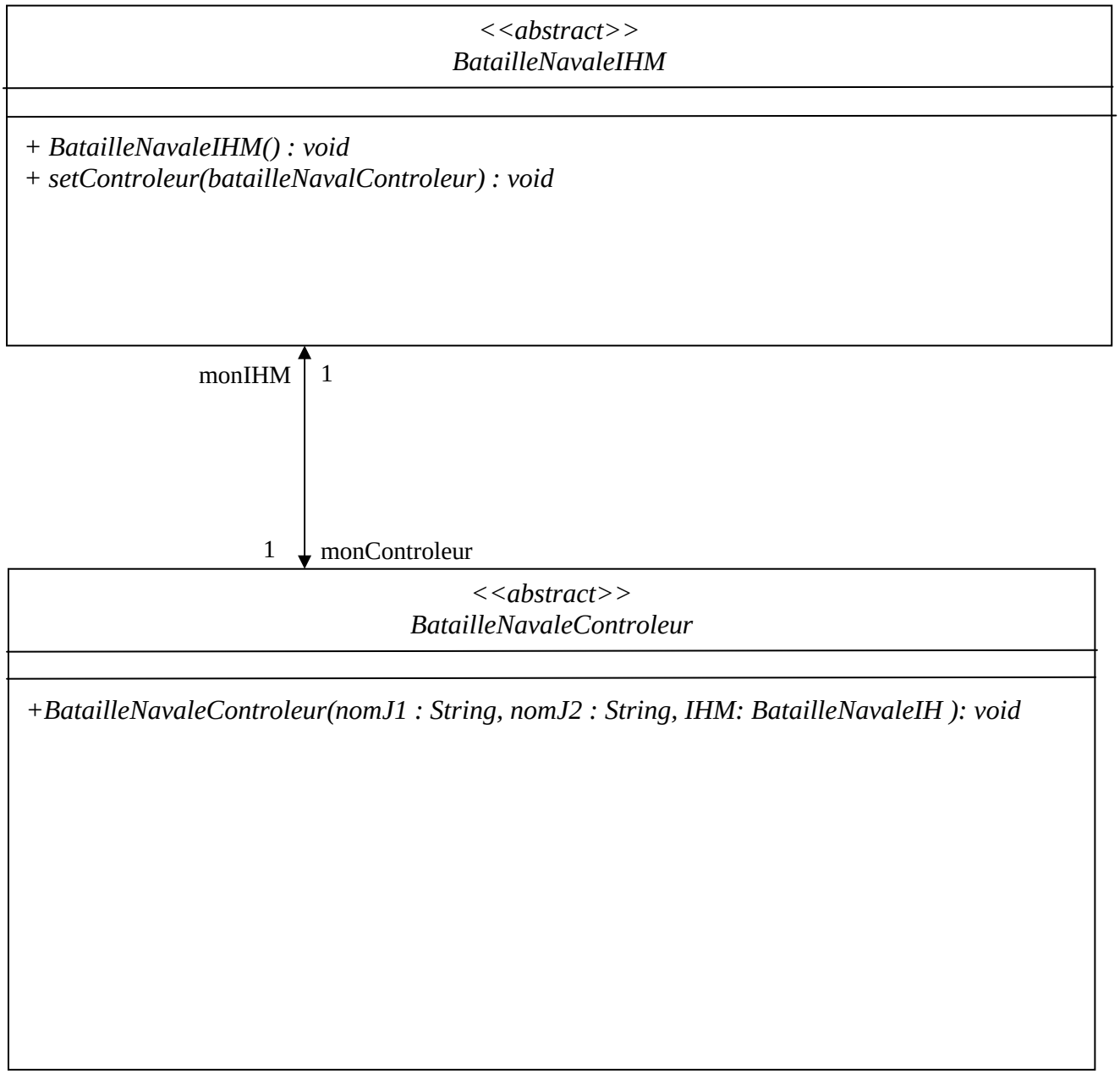
Ensuite, tour à tour, les joueurs lancent virtuellement des torpilles sur l'ennemi en indiquant les coordonnées d'un tir (par exemple B4). L'application indique si l'un de ses bateaux a été atteint ou pas. Le gagnant est celui qui parvient à torpiller complètement les navires de l'adversaire avant que tous les siens ne le soient.

La taille des grilles de jeu est toujours la même 10x10 : numérotée de 1 à 10 horizontalement et de A à J verticalement.

Chaque joueur voit une grille correspondant à la connaissance qu'il a acquise sur la flotte de son adversaire : une case sans pion est une case non torpillée, un pion blanc indique une torpille qui n'a pas touché un navire ; un pion rouge indique une touche.

1. Listez les événements externes et les événements résultats.
2. Complétez la classe abstraite `BatailleNavaleControleur` et l'interface `BatailleNavaleIHM` : listez les méthodes nécessaires au bon fonctionnement du logiciel.
3. Donnez la signature des méthodes.
4. Etablir le diagramme de classes de la partie gestion du jeu.

Remarque : l'interface `BatailleNavaleControleur` est la façade de la partie gestion du jeu.


<<abstract>>
BatailleNavaleIHM
+ BatailleNavaleIHM() : void
+ setControleur(batailleNavalControleur) : void
monIHM
1
1
monControleur
<<abstract>>
BatailleNavaleControleur
+BatailleNavaleControleur(nomJ1 : String, nomJ2 : String, IHM: BatailleNavaleIH): void