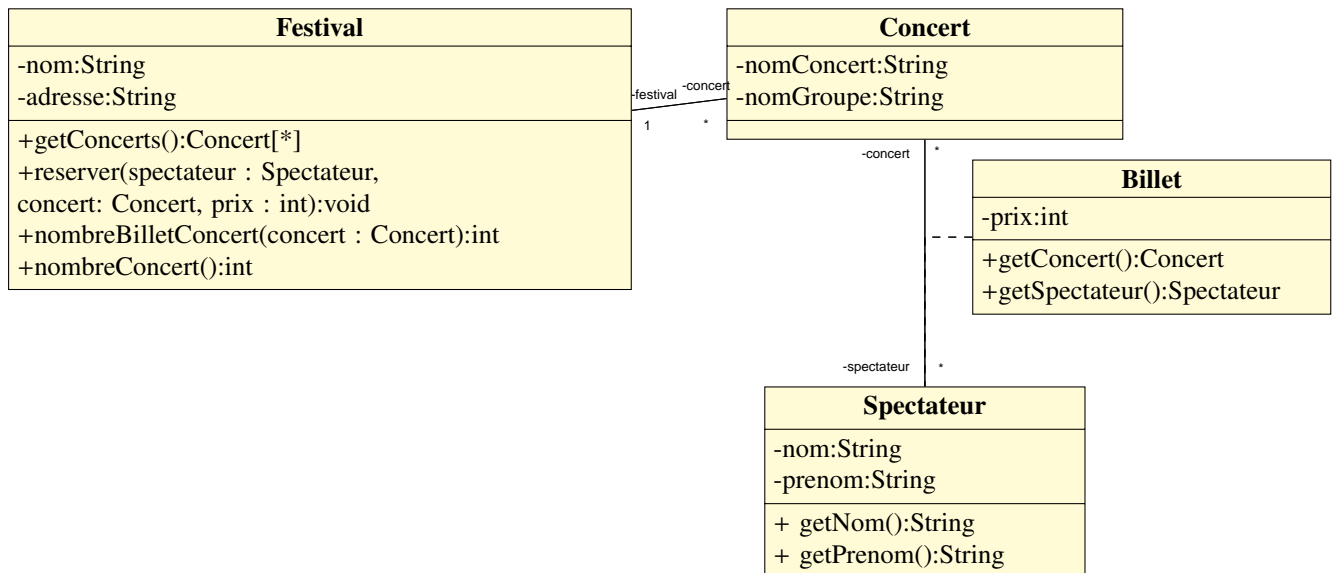


Exercice 1. Festival

1.1 Compléter le diagramme de classes avec les multiplicités.

1.2 Les classes association n'existent pas en Java. Comment allez vous traduire cette classe association ?

1.3 On vous donne le code du programme exécutable. Écrivez le code minimal des classes, pour qu'elles compilent. Complétez les méthodes au fur et à mesure.

```

public class Executable{

    public static void main(String[] args) {

        Spectateur paul = new Spectateur("Smith","Paul");
        Spectateur julie = new Spectateur("Martin","Julie");

        Concert concert1 = new Concert("divide","u2");
        Concert concert2 = new Concert("alive","daft punk");
        Concert concert3 = new Concert("memento","depeche mode");

        Festival fest1 = new Festival("festival");

        fest1.ajouterConcert(concert1);
    }
}
  
```

```

        fest1.ajouterConcert(concert2);
        fest1.ajouterConcert(concert3);

        fest1.reserver(paul, concert2, 50);
        fest1.reserver(julie, concert1, 40);
        fest1.reserver(julie, concert2, 40);
        fest1.reserver(paul, concert3, 90);

        System.out.println(fest1.nombreConcert());
        //3
        System.out.println(fest1.nombreBilletConcert(concert2));
        //2
    }
}

```

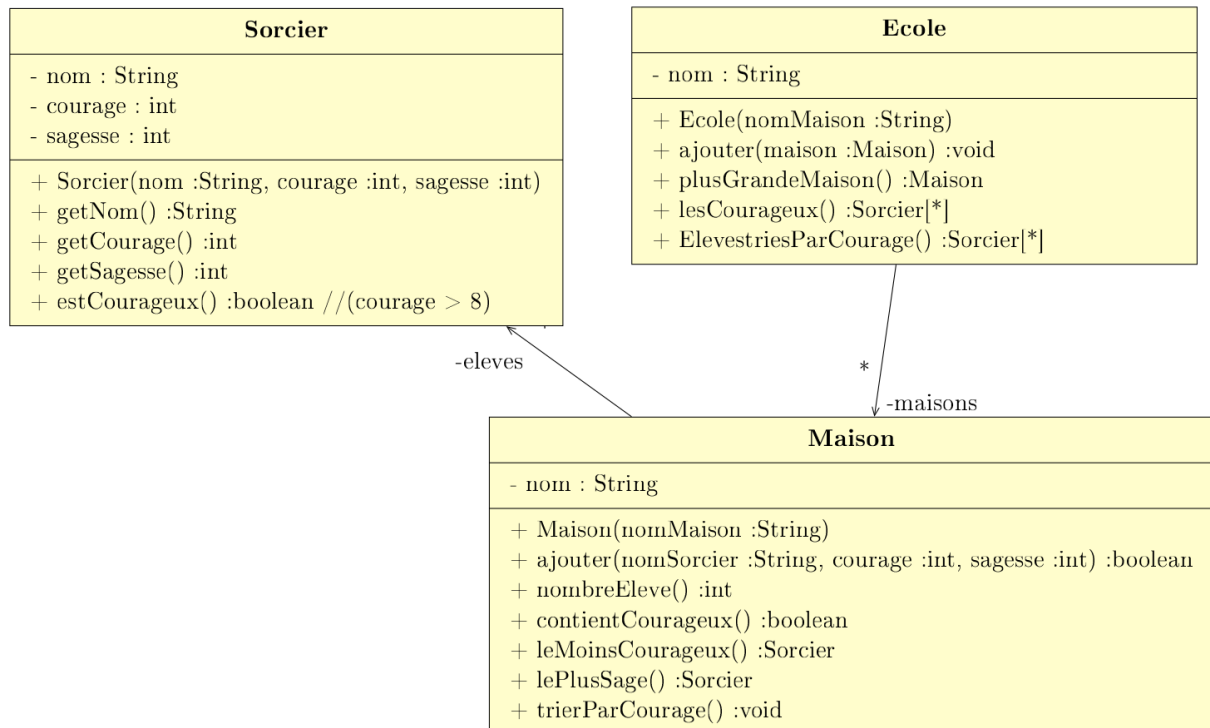
Exercice 2. A l'école de Poudlard

L'école de Poudlard accueille chaque année des élèves sorciers, répartis en plusieurs maisons et Dumbledore voudrait bien une application pour gérer ces élèves.

Voici les élèves accueillis cette année à Poudlard :

Nom	Courage	Sagesse	Maison	Nom	Courage	Sagesse	Maison
Adrian	9	7	Serpentar	Neuville	10	4	Griffondor
Hermione	8	6	Griffondor	Pansy	4	10	Serpentar
Luna	2	9	Serdaigle	Gregory	6	7	Serpentar
Drago	6	5	Serpentar	Gilderoy	7	9	Serdaigle
Norbert	3	7	Poufsouffle	Dean	9	4	Griffondor

On décide d'utiliser le modèle suivant :



2.1 Commence par écrire un exécutable qui reprend le scénario proposé (avec 10 élèves)

2.2 Écris le code minimal¹ des classes **Sorcier**, **Ecole** et **Maison**. Vérifie pour que le projet compile.

2.3 Choisis une méthode à implémenter (tu peux en ajouter si tu le souhaites)

2.4 Ajoute dans l'exécutable des tests permettant de vérifier le code de cette méthode.

2.5 Vérifie que tes tests échouent

2.6 Code la méthode choisie puis vérifie que les tests passent (y compris les tests des méthodes précédentes)

2.7 Recommence la question **2.3** jusqu'à ce qu'il n'y ait plus aucune méthode à implémenter.

2.8 Refactorise éventuellement ton code (pour améliorer la complexité de certaines méthodes ou la lisibilité du code par exemple)

2.9 Vérifie que ton code est suffisamment documenté et génère la javadoc.

¹. Le code attendu ici est le code des constructeurs, le code getteurs/setteurs éventuels. Les autres méthodes pour le moment ne font RIEN