



Objectifs

- JAVA : manipuler des listes, utiliser et implémenter l'interface Comparable

Exercice 1 *Lecture de code*

```

1 public class Pokemon implements Comparable<Pokemon>{
2     private String nom;
3     private Integer force;
4     public Pokemon(String nom, int force){
5         this.nom = nom.toLowerCase();
6         this.force = force;
7     }
8     @Override
9     public String toString(){
10        return "("+this.nom +", "+this.force+")";
11    }
12    @Override
13    public int compareTo(Pokemon autrePokemon){
14        return this.nom.compareTo(autrePokemon.nom);
15    }
16 }

```

```

public class Executable{
    public static void main(String[] args){
        List<Pokemon> liste = Arrays.asList(
            new Pokemon("HYDRAGON", 14), new Pokemon("Meltan", 8),
            new Pokemon("Brindibou", 11), new Pokemon("Griknot", 14),
            new Pokemon("CHARCOR", 2), new Pokemon("Paras", 7));
        System.out.println(Collections.min(liste)); //ICI
    }
}

```

Réponds aux questions suivantes SANS TESTER CE CODE (mais tu as le droit de consulter la documentation) :

1.1 Que va afficher la ligne signalée //ICI ?

- ☐ (CHARCOR, 2) ☐ (charcor, 2) ☐ CHARCOR ☐ chartor ☐ 2
☐ (Brindillou, 11) ☐ (brindillou, 11) ☐ BRINDILLOU ☐ brindillou ☐ 11
☐ autre réponse (précise) :

1.2 Même question si on remplace la ligne 14 de la classe Pokemon par :

```

14     return this.force.compareTo(autrePokemon.force);

```

1.3 Même question si on remplace la ligne 14 de la classe Pokemon par :

```

14     return this.nom.length() - autrePokemon.nom.length();

```

1.4 Teste le code pour vérifier tes réponses.

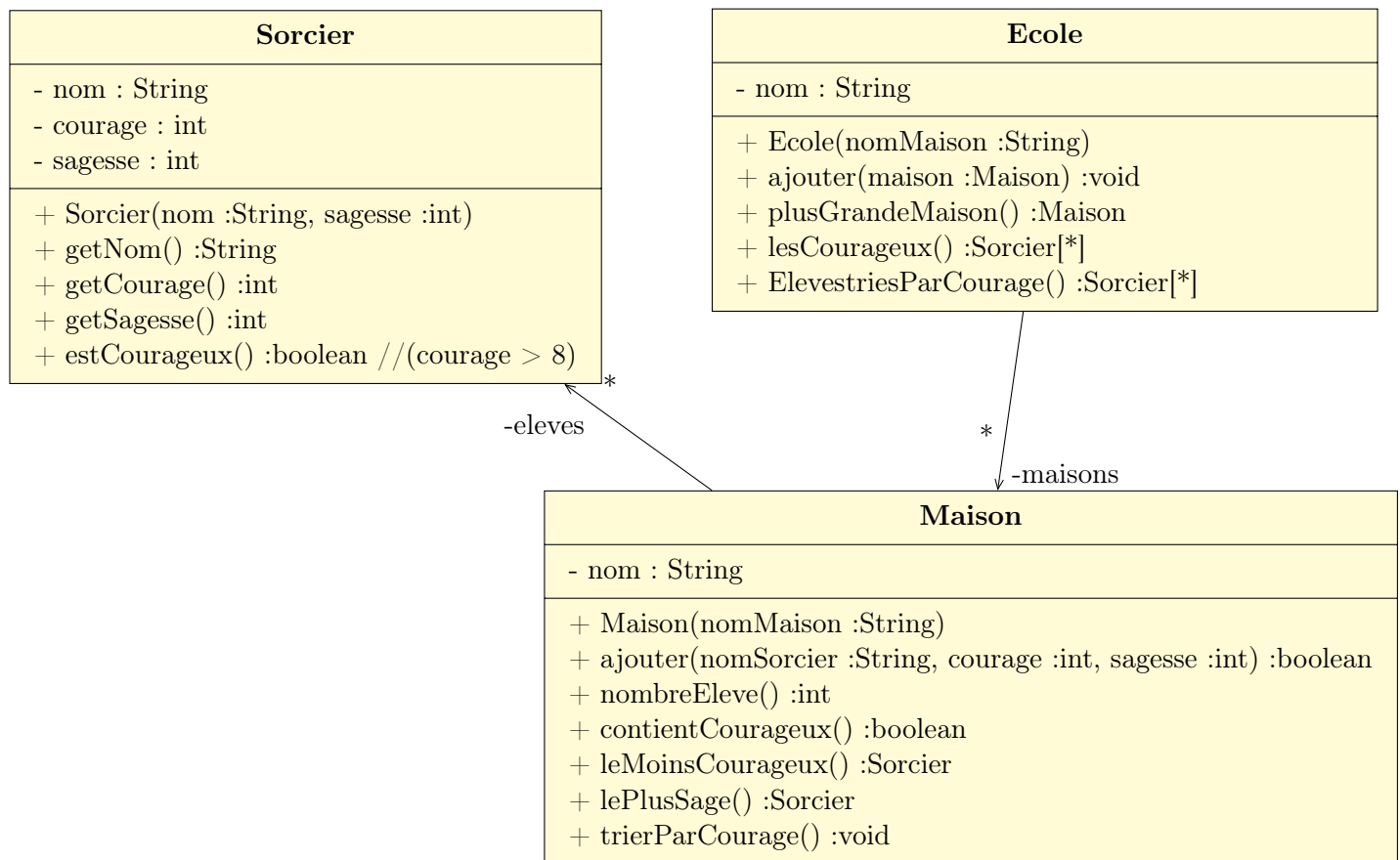
Exercice 2 *A l'école de Poudlard*

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	Neuveville	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