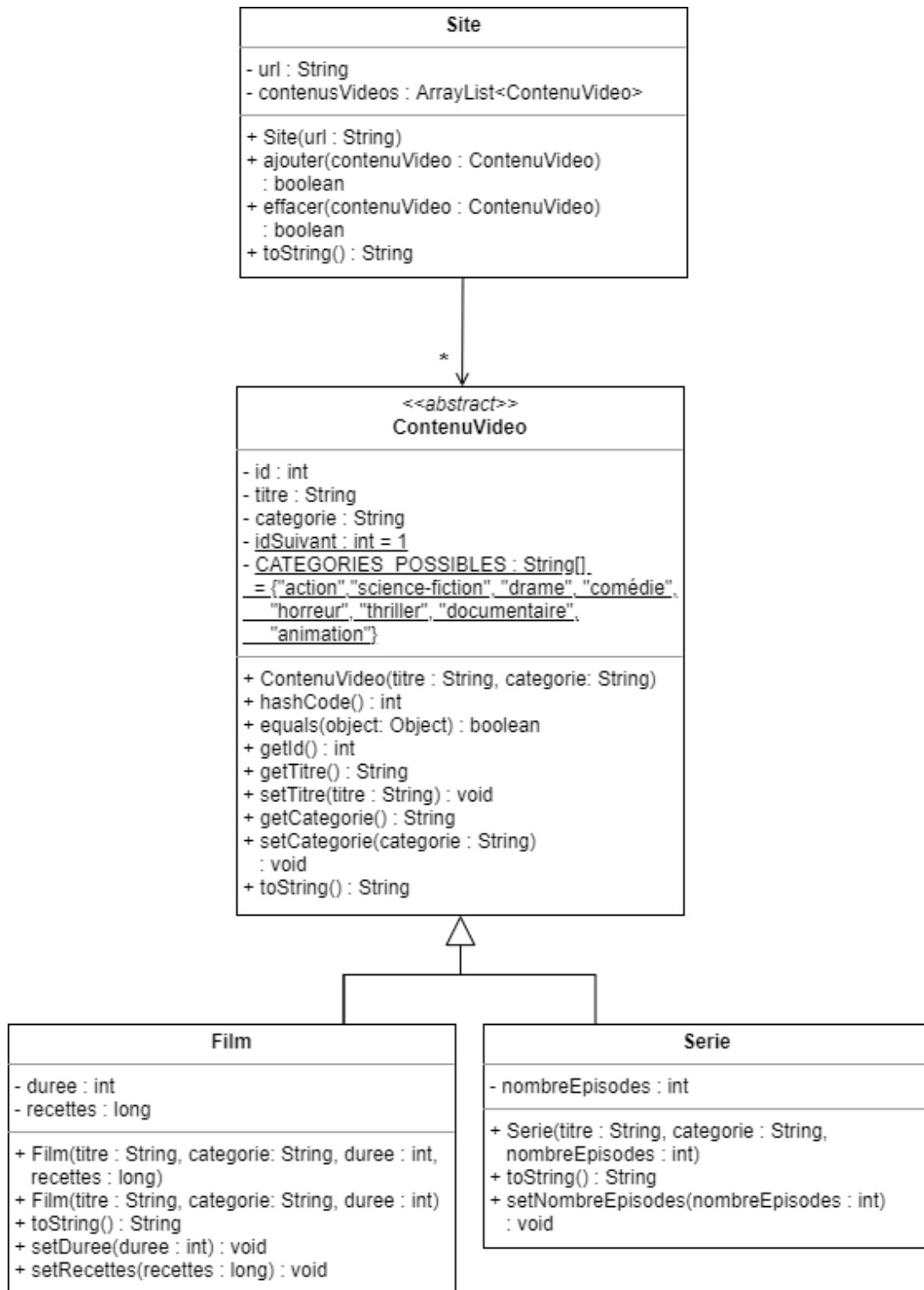


EXERCICES RÉCAPITULATIFS – PARTIE 1

1. Site pour contenu vidéo

- a) On vous demande, sur base du diagramme de classes ci-dessous et des consignes de créer le début d'une application permettant la gestion de données associées à des contenus vidéos d'un ou plusieurs sites web :



Créez, dans votre répertoire APOO, un projet intitulé **APOO_recapitulatifs_1** dans lequel vous implémenterez les classes `ContenuVideo`, `Film`, `Serie` et `Site` conformément au diagramme de classes et aux instructions qui suivent :

- Les méthodes et constructeurs devront lancer une `IllegalArgumentException` en cas de paramètre invalide. Dans le cas où la catégorie passée en paramètre n'est pas une des catégories possibles, le message à passer lors de la création de l'exception est « catégorie inexistante ».
- Pour la classe `ContenuVideo` :
 - Un contenu vidéo est identifié par son `id`.
 - Le constructeur attribue automatiquement l'`id` sur base de `idSuivant`.
- Pour la classe `Film` :
 - La durée est donnée en minutes.
 - Lorsque les recettes (total des sommes d'argent reçues) ne sont pas spécifiées à l'instanciation, elles doivent être initialisées à 0.
 - Un des constructeurs doit invoquer l'autre constructeur de la classe.
- Pour le `toString` des classes `ContenuVideo`, `Film` et `Serie`, référez-vous au `toString` attendu pour le site.
- Pour la classe `Site` :
 - La méthode **`toString()`** :

```
Site [url=www.supercontent.com]
```

```
Liste de contenus vidéos
```

```
Aucun contenu vidéo
```

- Elle renvoie les informations sous ce format pour un site web dont l'url est `www.spercontent.com` et ayant déjà des contenus vidéos :

```
Site [url=www.supercontent.com]
```

```
Liste de contenus vidéos
```

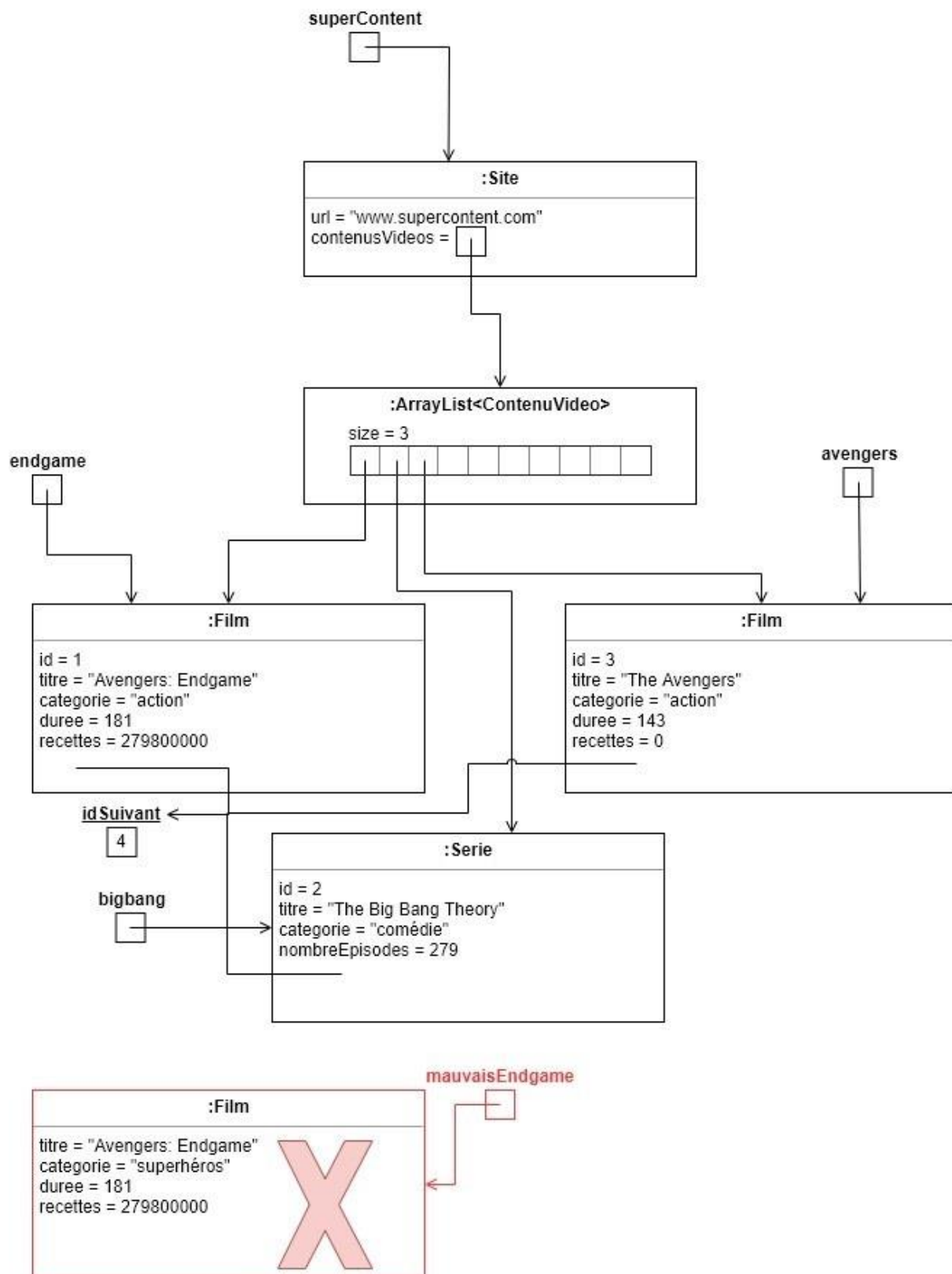
```
Film [id=1, titre=Avengers: Endgame, catégorie=action, durée=181, recettes=279800000]
```

```
Série [id=2, titre=The Big Bang Theory, catégorie=comédie, nombre d'épisodes=279]
```

```
Film [id=3, titre=The Avengers, catégorie=action, durée=143, recettes=0]
```

- Les méthodes **`ajouter()`** et **`effacer()`** :
 - permettent d'ajouter ou effacer un contenu vidéo à un site web ;
 - l'ajout d'un contenu vidéo à un site web n'est possible que si le contenu vidéo n'est pas déjà associé à ce site web ;
 - le retrait d'un contenu vidéo à un site web n'est possible que si le contenu vidéo est associé à ce site web ;
 - renvoient **`true`** ou **`false`** en fonction du succès de l'opération.

- b) Créez votre classe `TestSite`, sur base de la représentation des objets en mémoire ci-dessous :



Au sein de votre code, veuillez implémenter, dans l'ordre, les scénarios de tests décrits ci-dessous :

1. Scénario 1 : test de l'exception "Catégorie inexistante"

Essayez de créer l'instance `mauvaisEndgame` de votre site Web. Vérifiez que cela soulève une `IllegalArgumentException` et assurez-vous que votre programme continue sa bonne exécution (c'est à dire le scénario de tests qui suit) suite à l'affichage du message de l'exception dans la console.

2. Scénario 2 : création à la chaîne d'instances

Veuillez créer les instances des classes `Site`, `Film` et `Serie` conformément à la représentation mémoire, à l'exception de l'instance `mauvaisEndgame`.

Veuillez afficher l'état du site web `superContent` dans la console.

3. Scénario 3 : test d'ajout de contenus vidéos à un site web

Veuillez ajouter les contenus vidéos au site web `superContent` conformément à la représentation mémoire.

Veuillez afficher dans la console l'état du site web `superContent`.

Si vous avez le temps, complétez votre classe test pour tester l'ajout d'un contenu déjà présent dans le site, la méthode effacer, ...

2. Gestion d'un championnat de football

Une fédération de football veut clarifier son site de gestion des clubs de football. Votre expertise est requise pour établir le diagramme des classes de son domaine.

L'application est centrée sur les clubs de football (`Club`) et les résultats des championnats entre ses clubs.

Un club possède un matricule (un entier attribué automatiquement lors de l'instanciation : le premier club créé aura le numéro 1, le deuxième 2, ...), un nom, un entraîneur, une liste de joueurs, un terrain et un terrain d'entraînement. Pour instancier un club, son nom et son terrain sont indispensables. Un terrain d'entraînement peut être aussi spécifié lors de l'instanciation ou ajouté par la suite. Un club peut engager un entraîneur, licencier un entraîneur, engager ou licencier un joueur. Enfin on peut parcourir les joueurs du club (avec un `foreach`) et connaître leur nombre. Attention un club ne peut pas engager un joueur/entraîneur si celui-ci est déjà dans un club.

Un membre de la fédération possède un nom, un prénom et une nationalité. On doit pouvoir vérifier s'il est déjà dans un club, préciser qu'il a été engagé par un club ou encore dire qu'il a été licencié. Il n'est pas possible d'instancier un membre. Un membre est soit un joueur soit un entraîneur.

Un joueur possède une position sur le terrain qui peut être changée. Il faut vérifier que la position corresponde à une position permise à savoir : « G » pour gardien, « D » pour défenseur, « M » pour milieu et « A » pour attaquant. Si ce n'est pas le cas, il faut lancer une `IllegalArgumentException`. Un entraîneur possède une tactique (sous forme de chaîne de caractères). Que ce soit un joueur ou un entraîneur, il ne fait pas partie d'un club lors de sa

création. Les autres informations du joueur (ou d'un entraîneur) seront fournies lors de sa création.

Un terrain possède une longueur, une largeur. Il faut aussi savoir si la pelouse est naturelle ou non et connaître la date du dernier renouvellement de la pelouse. Lors de la création du terrain, ses mesures et le fait de savoir si la pelouse est naturelle ou non sont nécessaires. Par contre, la date de renouvellement correspond au moment de cette création. On peut changer la pelouse du terrain. Lors du changement, il faut préciser si la nouvelle pelouse est naturelle ou non.

Un championnat comporte un nom et la liste des équipes. On peut commencer le championnat, ajouter une équipe au championnat (tant que celui-ci n'a pas commencé), vérifier si le championnat a commencé, terminer le championnat, donner l'équipe qui est en tête du championnat (si celui-ci a commencé), donner le nombre d'équipes et donner le classement sous forme de texte. Pour instancier un championnat, seul son nom est nécessaire.

Une équipe comporte un club et un nombre de points. Seul le club est nécessaire lors de l'instanciation, les points étant mis à 0 automatiquement. On doit pouvoir ajouter un nombre de points à l'équipe.

Donnez le diagramme de classes UML de cette application. Vous ne devez pas mettre les getters, ni les toString.