

## 1 Objectifs d'apprentissage

On termine la conception détaillée et on passe à l'implémentation des fonctionnalités

## 2 Travail à réaliser

On progresse dans le cycle en V :

Analyse des besoins

Recette

Analyse

Conception

Test intégration

**Conception détaillée**

Tests unitaires

**Réalisation**

### Livrables attendus :

1. Un *diagramme de classe* mis à jour correspondant au lot 1 ;
2. Les *diagrammes de séquence* complets (cas nominaux et levées des exceptions) et actualisés d'`addMember()`, `addItemFilm()` et `reviewItemFilm()`.
3. Un *code Java* incomplet correspondant à ces diagrammes
4. Une *trace d'exécution des tests* de `addMember()`, `addItemFilm()` et `reviewItemFilm()` montrant l'avancement de votre travail. On attend qu'au moins `addMember()` fonctionne sans erreur.

Ces livrables seront affinés lors des prochaines séances.

## 3 Recommandations

- Si vous utilisez un éditeur UML qui génère du code Java, attention à ne générer que du code utile :  
Lorsque vous créez des classes, des associations, des attributs, etc. prenez soin de choisir judicieusement les options associées de façon à éviter la production de code parasite (accesseurs par exemple).
- Produisez un code clair et lisible :  
Repérez et supprimez le code superflu. Complétez les bannières Javadoc (`/** ... */`) en utilisant les tags<sup>1</sup> appropriés.
- Insérer des commentaires à destination des développeurs  
Rappel : les commentaires Javadoc sont destinés aux développeurs qui souhaitent *utiliser* les classes dans leur propre code. Les autres commentaires (`/* ... */` ou `//...`), eux, sont

1 <https://www.oracle.com/technetwork/articles/java/index-137868.html#tag> Javadoc tag conventions

destinés aux développeurs qui sont amenés à *comprendre* le code d'une classe, notamment les mainteneurs qui doivent la modifier.

- Lisez exhaustivement le *guide de bonnes pratiques en génie logiciel* qui vous a été fourni et mettez en œuvre les pratiques qui concernent votre activité dans le moment. Notez celles qui s'appliqueront dans la suite de votre développement.
- Comme d'habitude, travaillez *incrémentalement*. Vous commencerez par déduire le *squelette* Java vos classes à partir des informations fournies par vos diagrammes UML. Puis vous devez faire en sorte que `addMember()` passe le test afférent. Les itérations suivantes porteront sur la gestion des films, puis des livres.