

# Examen NFP121 – Session 1

Jeudi 20 Avril 2019

## Partie 2 – 1h30 – Toute ressource autorisée

Dans tout l'exercice, le code fourni ne devra en aucun cas être altéré.

Une fois l'exercice terminé, vous enverrez vos dossiers "src" et "tests" préalablement archivés (zip) par email à l'adresse [anael.megna@gmail.com](mailto:anael.megna@gmail.com).

Avec un ami, vous voulez développer une application permettant de faire le suivi des séries Netflix que vous êtes en train de regarder, et de les noter.

| Titre                       | Progression | Note globale |
|-----------------------------|-------------|--------------|
| The Running Dead - S2       | 13 / 13     | 8.8          |
| The Running Dead - S3       | 5 / 16      | 8.2          |
| Breaking Good - S1          | 7 / 7       | 9.4          |
| The Running Dead - S1       | 6 / 6       | 9.0          |
| The Black Hole Theory - S1  | 17 / 17     | 8.4          |
| Yellow is the new Blue - S1 | 13 / 13     | 7.8          |
| Breaking Good - S2          | 13 / 13     | 7.5          |
| How I Met Your Father - S1  | 22 / 22     | 8.6          |

Titre de la série:  Note:  +1 Nb d'eps:  +

L'application se composera d'une petite interface récapitulant dans un tableau la liste des séries, leur avancement et leur note globale. À terme, vous stockerez les informations dans un fichier XML pour ne pas les perdre lorsque vous quitterez l'application, mais pour le moment, vous choisissez de ne pas vous focaliser dessus.

Vous décidez pour plus de simplicité de représenter une série de N saisons par N séries suffixées par leur numéro de saison. Par exemple, on aurait non pas une série "Breaking Good" de 5 saisons, mais 5 séries appelées respectivement "Breaking Good - S1", "Breaking Good - S2", etc...

## Partie 1 : (6 points | temps conseillé 30 min)

Avec votre ami, vous voulez faire les choses dans le bon ordre. Après avoir spécifié ensemble en quoi consiste un catalogue et une série, vous vous répartissez les tâches :

- À partir de la spécification informelle présente dans l'interface "Series", vous vous attachez à la rédaction d'une classe de test utilisant JUnit4. Vous pourrez créer autant d'exception que nécessaire. Votre ami quant à lui se charge de son côté de l'implémentation.\*
- Pour le catalogue, c'est vous qui codez la classe "CatalogImpl" implémentant l'interface. Votre classe devra passer tous les tests lancés par la classe de test "CatalogImplTest".

(\* l'implémentation de l'ami, c'est un fichier que vous ne possédez pas, et qui devra passer vos tests)

## Partie 2 : (4 points | temps conseillé 20 min)

Dans cette partie, on ne se focalisera pas sur les choix d'organisation du code. Tout le code à écrire se situe dans le fichier "NetflixTracker.java".

Une grosse partie du code a déjà été rédigée, mais il manque une partie de l'interface.

Ajouter les éléments graphiques correspondant au contrôleur permettant de regarder et noter un épisode d'une série préalablement sélectionnée (uniquement les éléments graphique pour le moment, zone manquante par rapport à l'image).

Aussi, complétez la Vue de la table des séries afin d'y afficher l'état courant du catalogue.

## Partie 3 : (9 pts + 1 au choix + 2 bonus | tps conseillé 40 min)

Vous choisissez d'utiliser le modèle Passif du patron de conception MVC.

- ✓ Faîtes en sorte que lorsqu'on clique sur le bouton "+1", la série sélectionnée progresse d'un épisode.
- ( Faîtes en sorte que lorsqu'on clique sur le bouton "+", une nouvelle série est ajoutée dans le tableau.
- +1 pt si on peut appuyer sur la touche Entrée depuis un champ de texte au lieu de cliquer sur les boutons;
- +1 pt si un champ de texte voit son background devenir rouge si sa valeur n'est pas correcte lors d'une validation (bouton cliqué ou touche Entrée pressée);
- +0.25 pt si un champ de texte prend le focus si la validation échoue;
- +0.25 pt si un champ de texte voit son contenu sélectionné si la validation échoue;
- +0.5 pt si un champ de texte redevient blanc après une validation réussie.