***Couche de contrôle***

Comme nous l'avons vu ensemble, il y a de multiples façons d'interagir avec un utilisateur. Nous allons préparer l'architecture de notre application Blu-Ray Store à cette éventualité.

Ajoutez un sous-package controller et ajoutez y la classe com.mycompany.bluray.controller.MovieController

La classe disposera d'une méthode addUsingConsole dans laquelle on déplacera le code de saisie qui se trouve dans App.

Plutôt que de créer une variable de type MovieService dans le code de la méthode addUsingConsole, vous devrez ajouter à la classe MovieController une propriété de type MovieService. N'oubliez pas d'instancier la variable.

La classe App devra désormais instancier MovieController et invoquer sa méthode addUsingConsole .

***Repository alternatif***

Lors de la mise en production, il n'est plus question de mettre les films en mémoire via un ArrayList comme nous le faisions auparavant. Ces derniers doivent persister sur le disque dur.

Créez un repository alternatif via la classe com.mycompany.bluray.repository.MovieRepositoryWithText qui dispose lui aussi de la méthode add.

Vous devrez alors enregistrer les informations liées à un dans un fichier movies.txt qui devra se situer sur votre disque local C ou dans vos documents si vous êtes sur Mac.

Le fichier contiendra une ligne par Film, et vous séparerez les propriétés par des « ; »

Le contenu du fichier devrait ressembler à ça :

- Training Day;Policier
- American History X Man;Drame

Vous adapterez le code suivant pour écrire dans votre fichier text :

```
FileWriter writer;
try{
        writer=new FileWriter("C:\\temp\\movies.txt",true);
        writer.write("blabla\n");
        writer.close();
}
catch (IOException e){
e.printStackTrace();
}
```

Vous appelerez alors ce repository depuis la classe Service.