



# Pair programming ping pong

# TDD

- écrire un test
- vérifier qu'il échoue
- écrire juste le code suffisant pour passer le test
- vérifier que le test passe
- puis réusiner

# Pair programming

2 dev = 1 poste de travail

# L'association des 2 : Pair programming ping pong

- A écrit un nouveau test et vérifie que ça échoue
- B implémente le code voulu pour faire passer le test
- B écrit le nouveau test et vérifie que ça échoue
- A implémente le code voulu pour faire passer le test



# Jouons

D'après *Test driven development, a practical guide* de David  
Astels

# Gestion de films dans une liste

- 2 entités : les films et la liste
- Exercices découpés en match. Matches sur la liste avec les films et matches uniquement sur les films
- Matches découpés en tests à implémenter et auxquels répondre

# Match 1

Créer une liste dynamique de films

**Les films sont identifiés par leur nom.**

Avoir une liste dans laquelle on peut  
ajouter des films.

Savoir combien de films contient la liste.

Savoir si tel ou tel film est dans la liste.

**Les films sont identifiés par leur nom.**

# Match 1

## Une liste dynamique de films (avec 1 nom)

- **Test 1** : Une liste vide doit avoir une taille de zéro
- **Test 2** : Quand on ajoute un film à la liste, on doit avoir une liste avec une taille de 1
- **Test 3** : Quand on ajoute 2 films, on doit avoir une liste avec une taille de 2
- **Test 4** : Quand on ajoute à la liste le film *Terminator* et qu'on lui demande ensuite si elle contient le film *Terminator*, la liste doit répondre "oui" (true)
- **Test 5** : Quand on ajoute à la liste le film *Terminator* et qu'on lui demande ensuite si elle contient le film *Bambi*, la liste doit

# Match 2

## Renommer les films

Pouvoir autoriser les films à changer de nom.

**Travail à faire sur les films eux-mêmes uniquement.**

**Ne pas tenir compte de la liste.**



# Match 2

## Renommer les films

- **Test 1** : Après avoir créé un film, quand on change le nom de ce dernier, on doit ensuite pouvoir récupérer le bon nom.
- **Test 2** : Créer un film avec un nom de valeur null doit lever une exception
- **Test 3** : Créer un film avec un nom vide doit lever une exception
- **Test 4** : Renommer un film par un nom de valeur null doit lever une exception
- **Test 5** : Renommer un film par un nom vide doit lever une exception

# Match 3

Les films sont uniques

Un film apparaît une seule fois dans la liste.

Les noms dans la liste sont uniques.

Tenter de dupliquer un film dans la liste doit échouer.

**Etant donnée que nous n'avons que le nom du film, cela signifie que nous ne pouvons pas avoir 2 films avec le même nom.**

**Retour au travail sur la liste**

# Match 3

Les films sont uniques

- **Test 1** : Ajouter un film déjà existant (même nom) à la liste doit lever une exception et laisser la liste inchangée (de même taille)

# Match 4

Ajouter une note à un film

Chaque film peut avoir une seule note modifiable.

**Travail à faire sur les films eux-mêmes uniquement.**

**Ne pas tenir compte de la liste.**

# Match 4

## Ajouter une note à un film

- **Test 1** : Quand on demande à un film, qui n'a pas été noté, s'il a été noté, il doit répondre "non" (false).
- **Test 2** : Après avoir créé et noté un film, quand on demande à ce dernier s'il a été noté, il doit répondre "oui" (true).
- **Test 4** : Après avoir créé et noté un film, quand on demande la note de ce dernier, il doit la donner.
- **Test 5** : Quand on demande la note d'un film qui n'a pas été noté, une exception est levée.

# Match 5

Ajouter des catégories au film (catégories identifiées  
par un nom)

Les films appartiennent à des catégories  
comme *Horreur*, *Comédie*, ...

**Travail à faire sur les films eux-mêmes  
uniquement.**

**Ne pas tenir compte de la liste.**

# Match 5

Ajouter des catégories au film (catégories identifiées  
par un nom)

- **Test 1** : Un film à qui on n'a pas ajouté de catégorie doit répondre qu'il n'a pas de catégorie quand on lui demande sa catégorie (catégorie "uncategorized")
- **Test 2** : Si un film a eu une catégorie, il doit la donner quand on la lui demande
- **Test 3** : Créer un film avec une catégorie invalide (non présente dans une liste définie au préalable) lève une exception

A white ping pong ball is positioned in the center, resting on the red rubber surface of a ping pong paddle. The paddle has a black handle and is lying horizontally. The background is a blue surface with white lines, typical of a ping pong table. A semi-transparent blue rectangle is overlaid on the lower part of the image, containing the text.

**Debrief : So ?**