

1. Storytelling

L'actrice Yang se tient devant le miroir, arrangeant lentement sa longue chevelure. Aujourd'hui, elle n'est pas Yang. Elle est Raiponce, l'incarnation de l'optimisme à la chevelure flottante.

À chaque apparition sur scène, elle se présente sous le nom de son personnage, sans jamais prononcer le nom "Yang". Elle croit que le véritable art de l'interprétation, c'est d'oublier qui l'on est pour devenir pleinement un autre.

Mais lorsqu'aucun rôle ne lui est attribué, elle se présente alors calmement : « Je m'appelle Yang. » À ce moment-là, elle n'appartient à aucun monde de conte de fées, elle n'est qu'une actrice ordinaire, en attente d'un rôle. La scène est vide, les rôles ne sont pas encore distribués, mais elle ne cesse jamais de se préparer, car elle le sait : l'histoire ne s'arrête jamais.

De l'autre côté de la scène, le destin des mondes féeriques est lui aussi en train de changer. Autrefois, chaque histoire de princesse se terminait dès que le prince l'avait sauvée, figée dans le traditionnel « Ils vécurent heureux pour toujours ».

Mais aujourd'hui, une nouvelle règle est née :
« Si je suis sauvée, et que la série à laquelle j'appartiens n'a pas encore de suite, alors écrivez-en une pour moi. »

Ainsi, lorsque Ariel fut sauvée par le prince Éric, la série La Petite Sirène débloqua automatiquement un nouveau chapitre : La Petite Sirène – La suite.

Mais si la série possède déjà une suite, comme La Petite Sirène – La suite, même si Ariel est sauvée à nouveau, aucune nouvelle suite ne sera ajoutée. Chaque conte de fées n'a droit qu'à une seule renaissance.

Puis, d'autres princesses furent aussi sauvées :
Quand Belle fut délivrée par Adam, l'histoire de *La Belle et la Bête* reçut, elle aussi, une nouvelle vie, devenant *La Belle et la Bête – La suite*.

Chaque sauvetage d'une princesse n'est plus une fin, mais un nouveau départ.
Chaque montée sur scène d'une actrice n'est plus seulement une performance, mais une immersion totale, une transformation complète.
Chaque présentation de soi, chaque délivrance, n'est plus une conclusion, mais le commencement d'un autre chapitre.

2.Design

(diagramme UML de classes + design patterns adaptés aux nouveaux problèmes).

Nous avons mis en œuvre deux design patterns adaptés :

- Observer
- Proxy

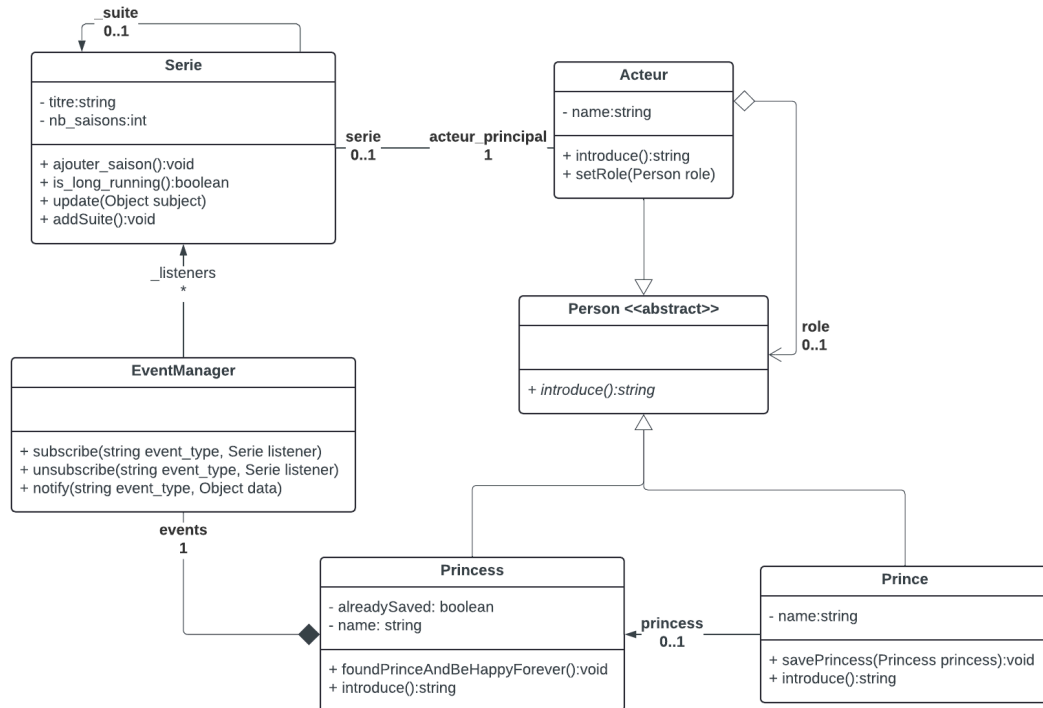


Diagramme UML de classes

Le système repose sur les classes suivantes :

Person (abstract) - Design pattern utilisé : Proxy

- Classe abstraite pour toutes les entités pouvant se présenter (`introduce()`).

Série - Design pattern utilisé : Observer

- Représente une série pouvant recevoir des événements.
- `+update(Object data)`
- `+addSuite()` : crée une suite si elle n'en a pas encore
- `suite: Serie`

EventManager - Design pattern : Observer

- Joue le rôle de dispatcher événementiel.
- `subscribe(event_type, Serie listener)`
- `notify(event_type, Object data)`
- `unsubscribe(...)`
- Cette classe découple la logique métier (princesse sauvée) de la logique de notification (Serie qui réagit).

Diagramme détaillé pour proxy

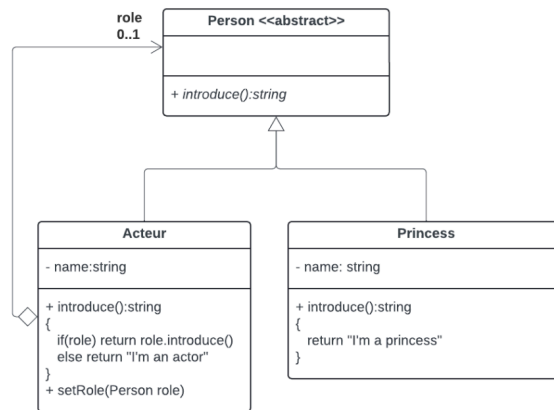
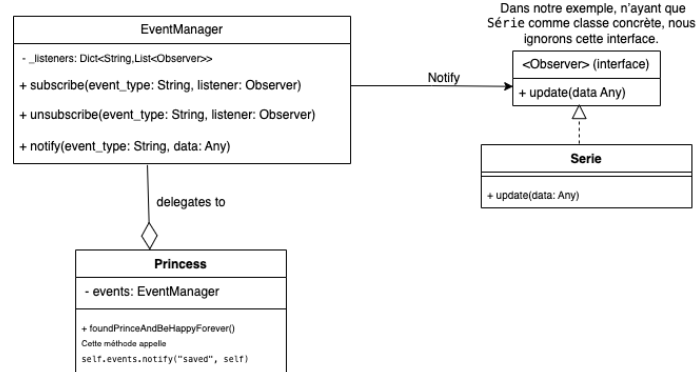


Diagramme détaillé pour observer



3. User stories (US) après fusion en Gherkin (Given/When/Then) avec critères d'acceptation

Critères d'acceptation (version courte)

Le premier critère est :

- Étant donné role = None
- Quand l'acteur·rice se présente
- Alors l'affichage est : « Je suis un·e acteur·rice, mon nom est <actor> »

Le deuxième critère est :

- Étant donné un rôle <role> nommé <roleName>
- Quand l'acteur·rice joue ce rôle
- Alors affiche « Je suis un·e <role>, mon nom est <roleName> »

```

@tag
Feature: Introduce oneself
  As an actor
  I want that ce soit mon personnage qui se présente à ma place
  So that je puisse vraiment m'immerger dans le rôle et l'interpréter de la façon la plus authentique

  Scenario: Actrice Yang se présente en tant que Rapunzel quand il joue le rôle de Rapunzel
    Given une actrice nommée Yang
    And un personnage qui est une princesse nommée Rapunzel
    When Yang joue le rôle de Rapunzel
    Then Yang devrait se présenter en tant que Rapunzel

  Scenario: Actrice Yang se présente en tant qu'elle-même quand elle ne joue pas aucun rôle
    Given une actrice nommée Yang
    When Yang ne joue aucun rôle
    Then Yang devrait se présenter en tant que Yang

  Scenario Outline: Un acteur se présente en tant que personnage qu'il joue
    Given un acteur/actrice nommé <actor>
    And un personnage qui est <role> nommé <roleName>
    When <actor> joue le rôle de <roleName>
    Then <actor> devrait se présenter en tant que <roleName>, sinon il/elle se présente en tant que

  Examples:
    | actor | role | roleName | expectedIntroduction |
    | Yang | Princess | Rapunzel | Je suis une princesse, mon nom est Rapunzel |
    | Flynn | Prince | Flynn | Je suis un prince, mon nom est Flynn |
    | Yang | None | None | Je suis un acteur, mon nom est Yang |

```

Le premier critère est :

- Étant donné que sérieA est abonnée à une princesse et que sérieA.suite est Null
- Quand princesse.foundPrinceAndBeHappyForever() est appelé
- Alors la méthode update(...) dans la série A est invoquée : cette série va créer une suite.

Le deuxième critère est :

- Étant donné que sérieB est abonnée à une princesse et que sérieB.suite n'est pas null
- Quand cette princesse appelle foundPrinceAndBeHappyForever()
- Alors la suite de sérieB reste inchangée.

```

# Author: yang.yang@uauprime.eu
@tag
Feature: Observer - Création de suite de série après sauvetage de la princesse
  En tant que princesse
  Je souhaite que mon histoire ne se termine pas après avoir été sauvée par le prince, et que les séries qui m'abonnent
  créent automatiquement une suite
  Afin de faire en sorte que seules les séries sans suite reçoivent une nouvelle saison quand je suis sauvée.

  Scenario: Une série sans suite crée une suite après que la princesse est sauvée
    Given une princesse nommée "Ariel" qui n'a pas encore été sauvée
    And un prince nommé "Eric" qui n'a pas de princesse
    And une série intitulée "La Petite Sirène" sans suite enregistrée
    And la série "La Petite Sirène" est abonnée à la princesse "Ariel"
    When le prince "Eric" sauve la princesse "Ariel"
    Then la série "La Petite Sirène" devrait avoir une suite intitulée "La Petite Sirène - Suite"

  Scenario: Une série qui a déjà une suite n'en crée pas de nouvelle
    Given une princesse nommée "Ariel" qui n'a pas encore été sauvée
    And un prince nommé "Eric" qui n'a pas de princesse
    And la série "La Petite Sirène" possède déjà une suite intitulée "La Petite Sirène - Suite"
    And la série "La Petite Sirène" est abonnée à la princesse "Ariel"
    When le prince "Eric" sauve la princesse "Ariel"
    Then la série "La Petite Sirène" ne devrait pas ajouter de suite supplémentaire

  Scenario Outline: Plusieurs séries réagissent correctement
    Given une princesse nommée "<princesse>" qui n'a pas encore été sauvée
    And un prince nommé "<prince>" qui n'a pas de princesse
    And une série intitulée "<serie>" sans suite enregistrée
    And la série "<serie>" est abonnée à la princesse "<princesse>"
    When le prince "<prince>" sauve la princesse "<princesse>"
    Then la série "<serie>" devrait avoir une suite intitulée "<serie> - Suite"

  Examples:
    | princesse | prince | serie |
    | Belle | Adam | La Belle et la Bête |

```

Après lancer “behave” :

```
@tag
Feature: Introduce oneself # features/Introduce.feature:2
  As un acteur
  I want que ce soit mon personnage qui se présente à ma place
  So that je puisse vraiment m'immerger dans le rôle et l'interpréter de la façon la plus authentique possible
Scenario: Actrice Yang se présente en tant que Rapunzel quand il joue le rôle de Rapunzel # features/Introduce.feature:7
  Given une actrice nommée Yang # features/steps/introduce_steps.py:18 0.000s
  And un personnage qui est une princesse nommée Rapunzel # features/steps/introduce_steps.py:22 0.000s
  When Yang joue le rôle de Rapunzel # features/steps/introduce_steps.py:26 0.000s
  Then Yang devrait se présenter en tant que Rapunzel # features/steps/introduce_steps.py:30 0.000s

Scenario: Actrice Yang se présente en tant qu'elle-même quand elle ne joue pas aucun rôle # features/Introduce.feature:13
  Given une actrice nommée Yang # features/steps/introduce_steps.py:18 0.000s
  When Yang ne joue aucun rôle # features/steps/introduce_steps.py:35 0.000s
  Then Yang devrait se présenter en tant que Yang # features/steps/introduce_steps.py:39 0.000s

Scenario Outline: Un acteur se présente en tant que personnage qu'il joue -- @1.1 # features/Introduce.feature:26
  Given un acteur/actrice nommé Yang # features/steps/introduce_steps.py:44 0.000s
  And un personnage qui est Princess nommé Rapunzel # features/steps/introduce_steps.py:48 0.000s
  When Yang joue le rôle de Rapunzel # features/steps/introduce_steps.py:26 0.000s
  Then Yang devrait se présenter en tant que Rapunzel, sinon il/elle se présente en tant que Yang : Je suis une princesse, mon nom est Rapunzel # features/steps/introduce_steps.py:62 0.000s

Scenario Outline: Un acteur se présente en tant que personnage qu'il joue -- @1.2 # features/Introduce.feature:27
  Given un acteur/actrice nommé Flynn # features/steps/introduce_steps.py:44 0.000s
  And un personnage qui est Prince nommé Flynn # features/steps/introduce_steps.py:48 0.000s
  When Flynn joue le rôle de Flynn # features/steps/introduce_steps.py:57 0.000s
  Then Flynn devrait se présenter en tant que Flynn, sinon il/elle se présente en tant que Flynn : Je suis un prince, mon nom est Flynn # features/steps/introduce_steps.py:62 0.000s

Scenario Outline: Un acteur se présente en tant que personnage qu'il joue -- @1.3 # features/Introduce.feature:28
  Given un acteur/actrice nommée Yang # features/steps/introduce_steps.py:44 0.000s
  And un personnage qui est None nommé None # features/steps/introduce_steps.py:48 0.000s
  When Yang joue le rôle de None # features/steps/introduce_steps.py:57 0.000s
  Then Yang devrait se présenter en tant que None, sinon il/elle se présente en tant que Yang : Je suis un acteur, mon nom est Yang # features/steps/introduce_steps.py:62 0.000s
```

```
@tag
Feature: Observer – Création de suite de série après sauvetage de la princesse # features/AutoCreationSuite.feature:3
  En tant que princesse
  Je souhaite que mon histoire ne se termine pas après avoir été sauvée par le prince, et que les séries qui m'abonnent créent automatiquement une suite
  Afin de faire en sorte que seules les séries sans suite reçoivent une nouvelle saison quand je suis sauvée.
Scenario: Une série sans suite crée une suite après que la princesse est sauvée # features/AutoCreationSuite.feature:10
  Given une princesse nommée "Ariel" qui n'a pas encore été sauvée # features/steps/auto_creationSerie.py:6 0.000s
  And un prince nommé "Eric" qui n'a pas de princesse # features/steps/auto_creationSerie.py:10 0.000s
  And une série titrée "La Petite Sirène" sans suite enregistrée # features/steps/auto_creationSerie.py:14 0.000s
  And la série "La Petite Sirène" est abonnée à la princesse "Ariel" # features/steps/auto_creationSerie.py:19 0.000s
  When le prince "Eric" sauve la princesse "Ariel" # features/steps/auto_creationSerie.py:37 0.000s
  Then la série "La Petite Sirène" devrait avoir une suite intitulée "La Petite Sirène – Suite" # features/steps/auto_creationSerie.py:46 0.000s

Scenario: Une série qui a déjà une suite n'en crée pas de nouvelle # features/AutoCreationSuite.feature:18
  Given une princesse nommée "Ariel" qui n'a pas encore été sauvée # features/steps/auto_creationSerie.py:6 0.000s
  And un prince nommé "Eric" qui n'a pas de princesse # features/steps/auto_creationSerie.py:10 0.000s
  And la série "La Petite Sirène" possède déjà une suite intitulée "La Petite Sirène – Suite" # features/steps/auto_creationSerie.py:26 0.000s
  And la série "La Petite Sirène" est abonnée à la princesse "Ariel" # features/steps/auto_creationSerie.py:19 0.000s
  When le prince "Eric" sauve la princesse "Ariel" # features/steps/auto_creationSerie.py:37 0.000s
  Then la série "La Petite Sirène" ne devrait pas ajouter de suite supplémentaire # features/steps/auto_creationSerie.py:55 0.000s

Scenario Outline: Plusieurs séries réagissent correctement -- @1.1 # features/AutoCreationSuite.feature:36
  Given une princesse nommée "Belle" qui n'a pas encore été sauvée # features/steps/auto_creationSerie.py:6 0.000s
  And un prince nommé "Adam" qui n'a pas de princesse # features/steps/auto_creationSerie.py:10 0.000s
  And une série titrée "La Belle et la Bête" sans suite enregistrée # features/steps/auto_creationSerie.py:14 0.000s
  And la série "La Belle et la Bête" est abonnée à la princesse "Belle" # features/steps/auto_creationSerie.py:19 0.000s
  When le prince "Adam" sauve la princesse "Belle" # features/steps/auto_creationSerie.py:37 0.000s
  Then la série "La Belle et la Bête" devrait avoir une suite intitulée "La Belle et la Bête – Suite" # features/steps/auto_creationSerie.py:46 0.000s
```

4. Exemple de couverture de code


a. Couverture de tests unitaires

| ===== tests coverage ===== | | | |
|--|-------|------|-------|
| coverage: platform darwin, python 3.13.5-final-0 | | | |
| Name | Stmts | Miss | Cover |
| src/Acteur.py | 21 | 21 | 0% |
| src/Kingdom.py | 37 | 4 | 89% |
| src/Person.py | 5 | 1 | 80% |
| src/Serie.py | 38 | 7 | 82% |
| src/_init_.py | 0 | 0 | 100% |
| TOTAL | 101 | 33 | 67% |

b. Couverture (features) peut être consultée dans le document **htmlcov** sur GitHub :

<https://github.com/yangjoanne216/M-thodesAgiles>

Coverage report: 95%



Files

Functions

Classes


☐ hide covered

coverage.py v7.9.1, created at 2025-06-27 15:00 +0200

| File ▲ | statements | missing | excluded | coverage |
|---------------------------------------|------------|-----------|----------|------------|
| features/steps/auto_creation_serie.py | 42 | 3 | 0 | 93% |
| features/steps/introduce_steps.py | 44 | 0 | 0 | 100% |
| features/steps/save_princess_steps.py | 33 | 0 | 0 | 100% |
| features/steps/serie_steps.py | 37 | 0 | 0 | 100% |
| src/Acteur.py | 21 | 2 | 0 | 90% |
| src/Kingdom.py | 37 | 2 | 0 | 95% |
| src/Person.py | 5 | 1 | 0 | 80% |
| src/Serie.py | 38 | 4 | 0 | 89% |
| Total | 257 | 12 | 0 | 95% |

coverage.py v7.9.1, created at 2025-06-27 15:00 +0200

Coverage report: 95%



Files

Functions

Classes

☐ hide covered

coverage.py v7.9.1, created at 2025-06-27 15:00 +0200

| File ▲ | class | statements | missing | excluded | coverage |
|---------------------------------------|----------------|------------|-----------|----------|------------|
| features/steps/auto_creation_serie.py | (no class) | 42 | 3 | 0 | 93% |
| features/steps/introduce_steps.py | IntroduceLogic | 15 | 0 | 0 | 100% |
| features/steps/introduce_steps.py | (no class) | 29 | 0 | 0 | 100% |
| features/steps/save_princess_steps.py | PrincessStep | 10 | 0 | 0 | 100% |
| features/steps/save_princess_steps.py | (no class) | 23 | 0 | 0 | 100% |
| features/steps/serie_steps.py | SerieLogic | 15 | 0 | 0 | 100% |
| features/steps/serie_steps.py | (no class) | 22 | 0 | 0 | 100% |
| src/Acteur.py | Acteur | 12 | 2 | 0 | 83% |
| src/Acteur.py | (no class) | 9 | 0 | 0 | 100% |
| src/Kingdom.py | EventManager | 6 | 2 | 0 | 67% |
| src/Kingdom.py | Princess | 7 | 0 | 0 | 100% |
| src/Kingdom.py | Prince | 7 | 0 | 0 | 100% |
| src/Kingdom.py | (no class) | 17 | 0 | 0 | 100% |
| src/Person.py | Person | 1 | 1 | 0 | 0% |
| src/Person.py | (no class) | 4 | 0 | 0 | 100% |
| src/Serie.py | Serie | 18 | 4 | 0 | 78% |
| src/Serie.py | (no class) | 20 | 0 | 0 | 100% |
| Total | | 257 | 12 | 0 | 95% |

coverage.py v7.9.1, created at 2025-06-27 15:00 +0200