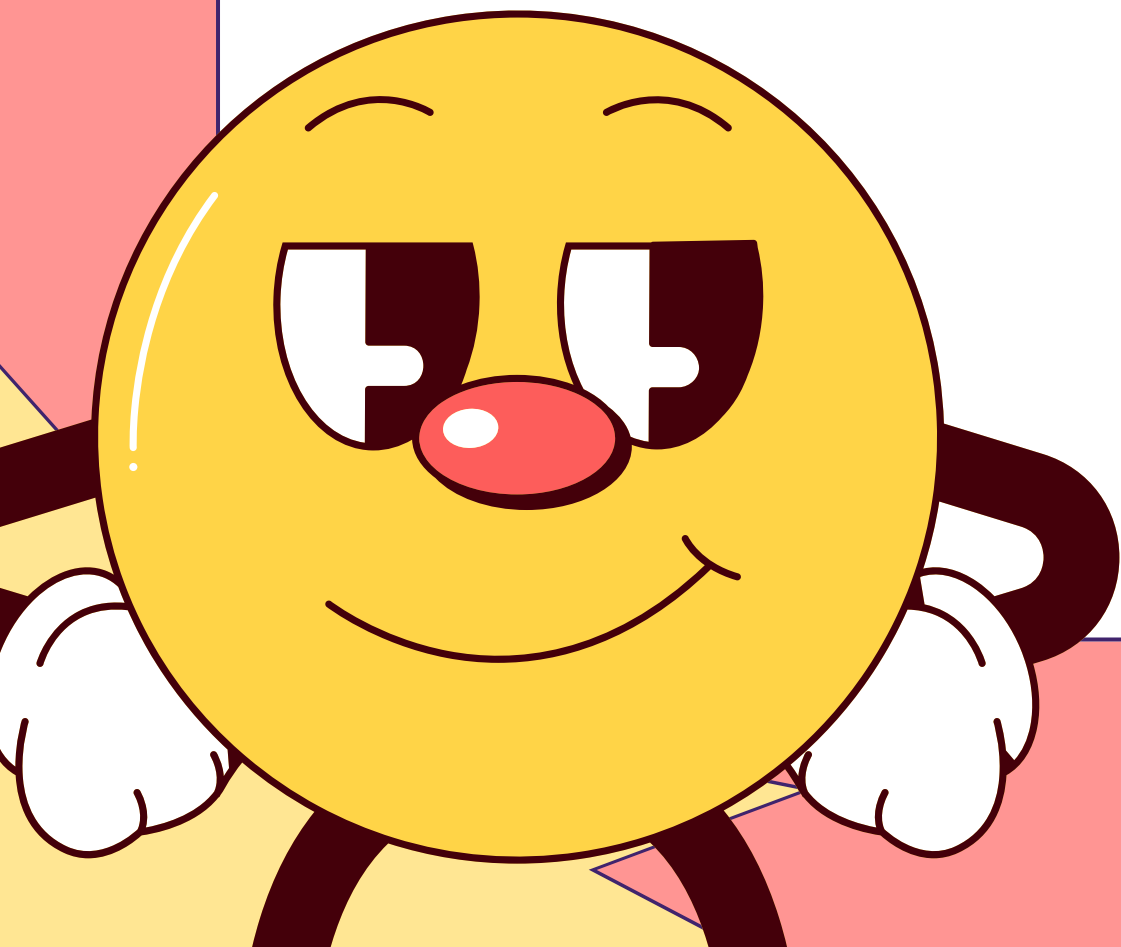




Princesse & Série



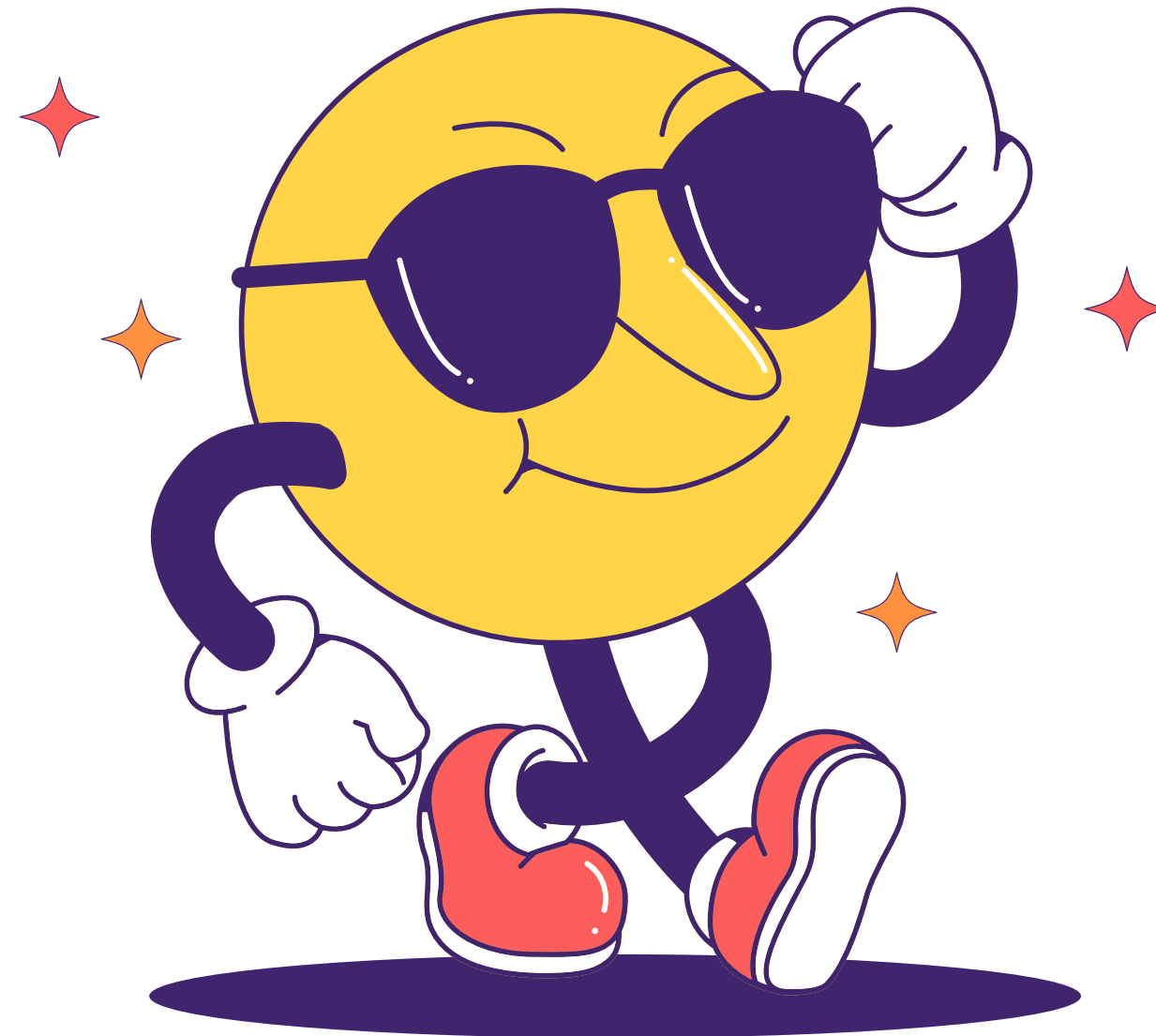
CHEN Yiqing
Ayman YAYA

YANG Yang
Binh Minh NGUYEN

Sommaire

Storytelling, US & Tests
originaux

Diagramme de classe
projet original



Storytelling, US & Tests
après merge + design
patterns

Diagramme de classe
projet complet

Storytelling & US originaux

Serie et Acteur

US01

Associer un acteur principal à une série

Feature: Association d'un acteur principal à une série

En tant que gestionnaire de production,
Je veux pouvoir associer un acteur principal à une série,
Afin de gérer les informations artistiques.

Scenario: Associer un acteur à la série Astérix

Given une série intitulée "Astérix" avec 4 saison(s)
And un acteur nommé "Roger Carel"
When j'associe l'acteur à la série
Then le nom de l'acteur principal de la série est "Roger Carel"

Scenario Outline: Associer un acteur à d'autres séries

Given une série intitulée "<titre>" avec <saisons> saison(s)
And un acteur nommé "<acteur>"
When j'associe l'acteur à la série
Then le nom de l'acteur principal de la série est "<acteur>"

Exemples:

titre	saisons	acteur
Lupin	1	Omar Sy
Dark	2	Anna Müller
The four seasons	3	Camille Cottin

US02

Ajouter une saison à une série

Feature: Gestion des saisons d'une série

En tant que gestionnaire de séries,
Je veux pouvoir ajouter des saisons à une série,
Afin de suivre l'évolution de sa production.

Scenario: Une série ajoute une saison

Given une série intitulée "Astérix" avec 4 saison(s)
When j'ajoute une nouvelle saison
Then la série doit avoir 5 saison(s)

Scenario Outline: Ajouter une saison à une série existante

Given une série intitulée "<titre>" avec <saisons> saison(s)
When j'ajoute une nouvelle saison
Then la série doit avoir <attendu> saison(s)

Exemples:

titre	saisons	attendu
Astérix	4	5
Lupin	5	6
The four seasons	1	2

Storytelling & US originaux

Princesse et Prince

Feature: Sauver Princess

As a prince

I want sauver ma princesse

So that vivre heureux ensemble pour toujours

Scenario: Flynn sauve Rapunzel avec succès

Given une princesse nommée Rapunzel qui n'a pas encore été sauvée

And un prince nommé Flynn qui n'a pas de princesse

When Flynn sauve Rapunzel

Then Flynn et Rapunzel sont ensemble et Rapunzel est sauvée

Scenario Outline: Un prince sans princesse sauve une princesse qui n'a pas encore été sauvée

Given une princesse nommée <princesse> qui n'a pas encore été sauvée: <alreadySaved>

And un prince nommé <prince> et sa princesse est: <princessDuPrince>

When le prince sauve <princesse>

Then prince et princesse devraient être ensemble et la <princesse> devrait être <saved>

Examples:

princesse	alreadySaved	prince	princessDuPrince	saved	
Rapunzel	False	Flynn	None	True	
Olive	True	Brutus	None	False	
Annie	False	Nick	Rapunzel	False	

US01

**En tant que prince, je veux
sauver une princesse**

US02

**Ne pas sauver une
princesse déjà sauvée**

Tests originaux

```
tag
eature: Sauver Princess # features/SavePrincess.feature:3
  As a prince
  I want sauver ma princesse
  So that vivre heureux ensemble pour toujours
Scenario: Flynn sauve Rapunzel avec succès # features/SavePrincess.feature:8
  Given une princesse nommée Rapunzel qui n'a pas encore été sauvée # features/steps/save_princess_steps.py:17
  And un prince nommé Flynn qui n'a pas de princesse # features/steps/save_princess_steps.py:21
  When Flynn sauve Rapunzel # features/steps/save_princess_steps.py:25
  Then Flynn et Rapunzel sont ensemble et Rapunzel est sauvée # features/steps/save_princess_steps.py:29

Scenario Outline: Un prince sans princesse sauve une princesse qui n'a pas encore été sauvée -- @1.1 # features/SavePrincess.feature:10
  Given une princesse nommée Rapunzel qui n'a pas encore été sauvée: False # features/steps/save_princess_steps.py:17
  And un prince nommé Flynn et sa princesse est: None # features/steps/save_princess_steps.py:21
  When le prince sauve Rapunzel # features/steps/save_princess_steps.py:25
  Then prince et princesse devraient être ensemble et la Rapunzel devrait être True # features/steps/save_princess_steps.py:29

Scenario Outline: Un prince sans princesse sauve une princesse qui n'a pas encore été sauvée -- @1.2 # features/SavePrincess.feature:11
  Given une princesse nommée Olive qui n'a pas encore été sauvée: True # features/steps/save_princess_steps.py:17
  And un prince nommé Brutus et sa princesse est: None # features/steps/save_princess_steps.py:21
  When le prince sauve Olive # features/steps/save_princess_steps.py:25
  Then prince et princesse devraient être ensemble et la Olive devrait être False # features/steps/save_princess_steps.py:29

Scenario Outline: Un prince sans princesse sauve une princesse qui n'a pas encore été sauvée -- @1.3 # features/SavePrincess.feature:12
  Given une princesse nommée Annie qui n'a pas encore été sauvée: False # features/steps/save_princess_steps.py:17
  And un prince nommé Nick et sa princesse est: Rapunzel # features/steps/save_princess_steps.py:21
  When le prince sauve Annie # features/steps/save_princess_steps.py:25
  Then prince et princesse devraient être ensemble et la Annie devrait être False # features/steps/save_princess_steps.py:29

feature passed, 0 failed, 0 skipped
scenarios passed, 0 failed, 0 skipped
6 steps passed, 0 failed, 0 skipped, 0 undefined
ook 0m0.021s
```

```
Scenario: Associer un acteur à la série Astérix # features/acteur.feature:7
  Given une série intitulée "Astérix" avec 4 saison(s) # features/steps/serie_steps.py:29
  And un acteur nommé "Roger Carel" # features/steps/serie_steps.py:34
  When j'associe l'acteur à la série # features/steps/serie_steps.py:38
  Then le nom de l'acteur principal de la série est "Roger Carel" # features/steps/serie_steps.py:46

Scenario Outline: Associer un acteur à d'autres séries -- @1.1 # features/acteur.feature:22
  Given une série intitulée "Lupin" avec 1 saison(s) # features/steps/serie_steps.py:29
  And un acteur nommé "Omar Sy" # features/steps/serie_steps.py:34
  When j'associe l'acteur à la série # features/steps/serie_steps.py:38
  Then le nom de l'acteur principal de la série est "Omar Sy" # features/steps/serie_steps.py:46

Scenario Outline: Associer un acteur à d'autres séries -- @1.2 # features/acteur.feature:23
  Given une série intitulée "Dark" avec 2 saison(s) # features/steps/serie_steps.py:29
  And un acteur nommé "Anna Müller" # features/steps/serie_steps.py:34
  When j'associe l'acteur à la série # features/steps/serie_steps.py:38
  Then le nom de l'acteur principal de la série est "Anna Müller" # features/steps/serie_steps.py:46

Scenario Outline: Associer un acteur à d'autres séries -- @1.3 # features/acteur.feature:24
  Given une série intitulée "The four seasons" avec 3 saison(s) # features/steps/serie_steps.py:29
  And un acteur nommé "Camille Cottin" # features/steps/serie_steps.py:34
  When j'associe l'acteur à la série # features/steps/serie_steps.py:38
  Then le nom de l'acteur principal de la série est "Camille Cottin" # features/steps/serie_steps.py:46

Feature: Gestion des saisons d'une série # features/serie.feature:1
  En tant que gestionnaire de séries,
  Je veux pouvoir ajouter des saisons à une série,
  Afin de suivre l'évolution de sa production.
Scenario: Une série ajoute une saison # features/serie.feature:7
  Given une série intitulée "Astérix" avec 4 saison(s) # features/steps/serie_steps.py:29
  When j'ajoute une nouvelle saison # features/steps/serie_steps.py:42
  Then la série doit avoir 5 saison(s) # features/steps/serie_steps.py:50

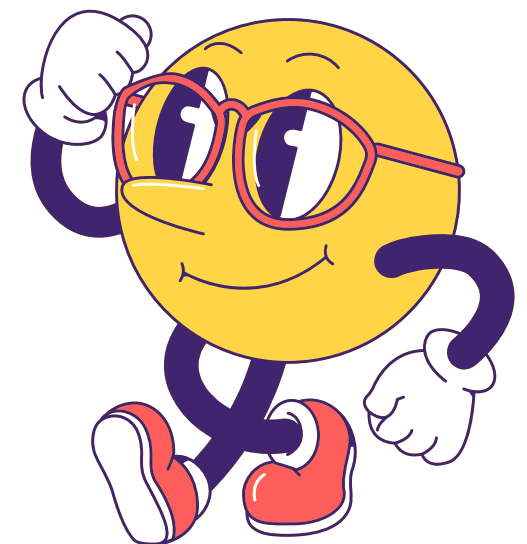
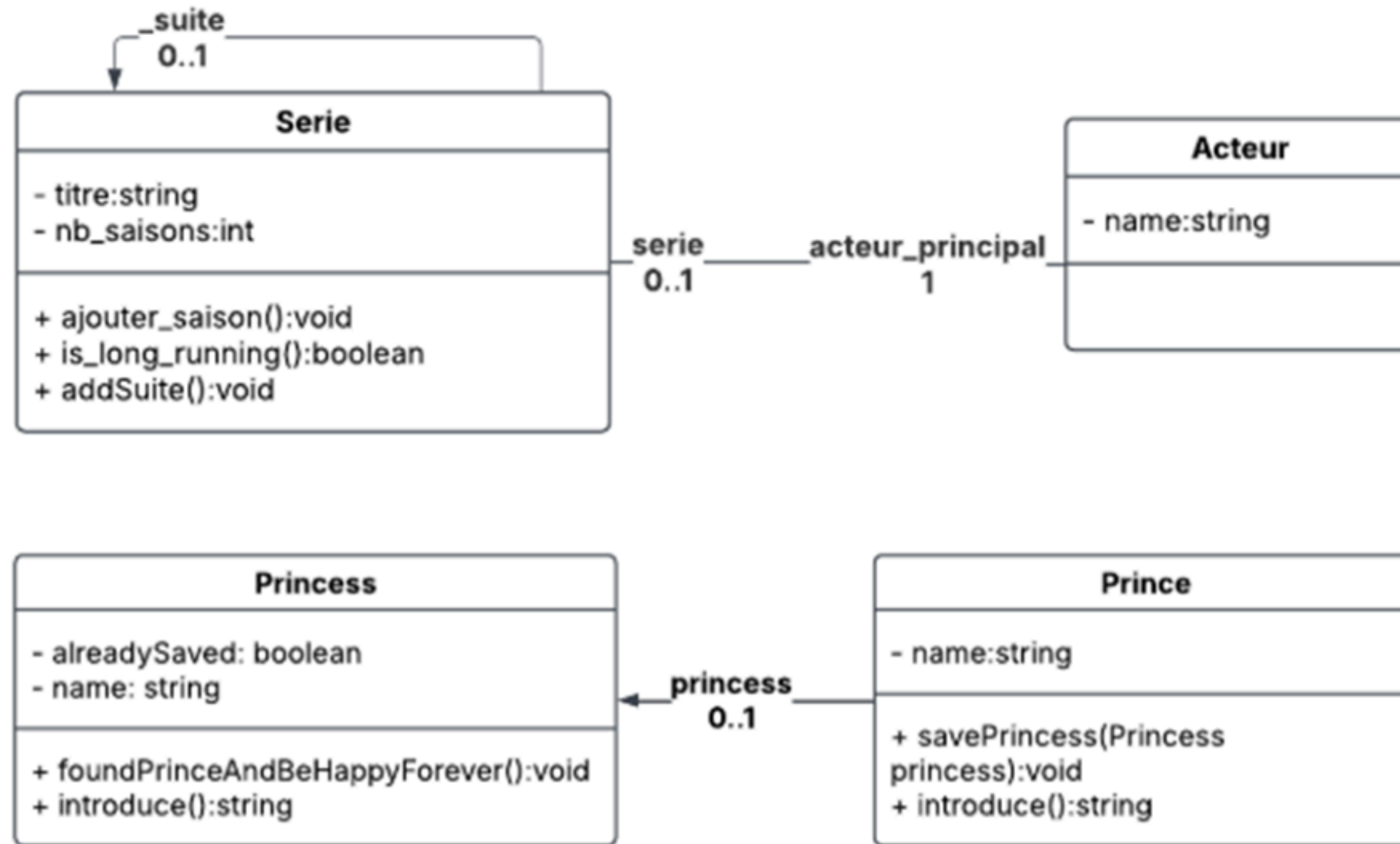
Scenario Outline: Ajouter une saison à une série existante -- @1.1 # features/serie.feature:19
  Given une série intitulée "Astérix" avec 4 saison(s) # features/steps/serie_steps.py:29
  When j'ajoute une nouvelle saison # features/steps/serie_steps.py:42
  Then la série doit avoir 5 saison(s) # features/steps/serie_steps.py:50

Scenario Outline: Ajouter une saison à une série existante -- @1.2 # features/serie.feature:20
  Given une série intitulée "Lupin" avec 5 saison(s) # features/steps/serie_steps.py:29
  When j'ajoute une nouvelle saison # features/steps/serie_steps.py:42
  Then la série doit avoir 6 saison(s) # features/steps/serie_steps.py:50

Scenario Outline: Ajouter une saison à une série existante -- @1.3 # features/serie.feature:21
  Given une série intitulée "The four seasons" avec 1 saison(s) # features/steps/serie_steps.py:29
  When j'ajoute une nouvelle saison # features/steps/serie_steps.py:42
  Then la série doit avoir 2 saison(s) # features/steps/serie_steps.py:50

2 features passed, 0 failed, 0 skipped
8 scenarios passed, 0 failed, 0 skipped
28 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m0.022s
```

Diagramme de classe - original



Nouveau user story

- En tant qu'un acteur
- Je souhaite qu'à chaque fois que je me présente, ce soit comme si c'était mon personnage qui le faisait
- Afin de pouvoir m'immerger pleinement dans le rôle et l'interpréter de la manière la plus authentique possible.

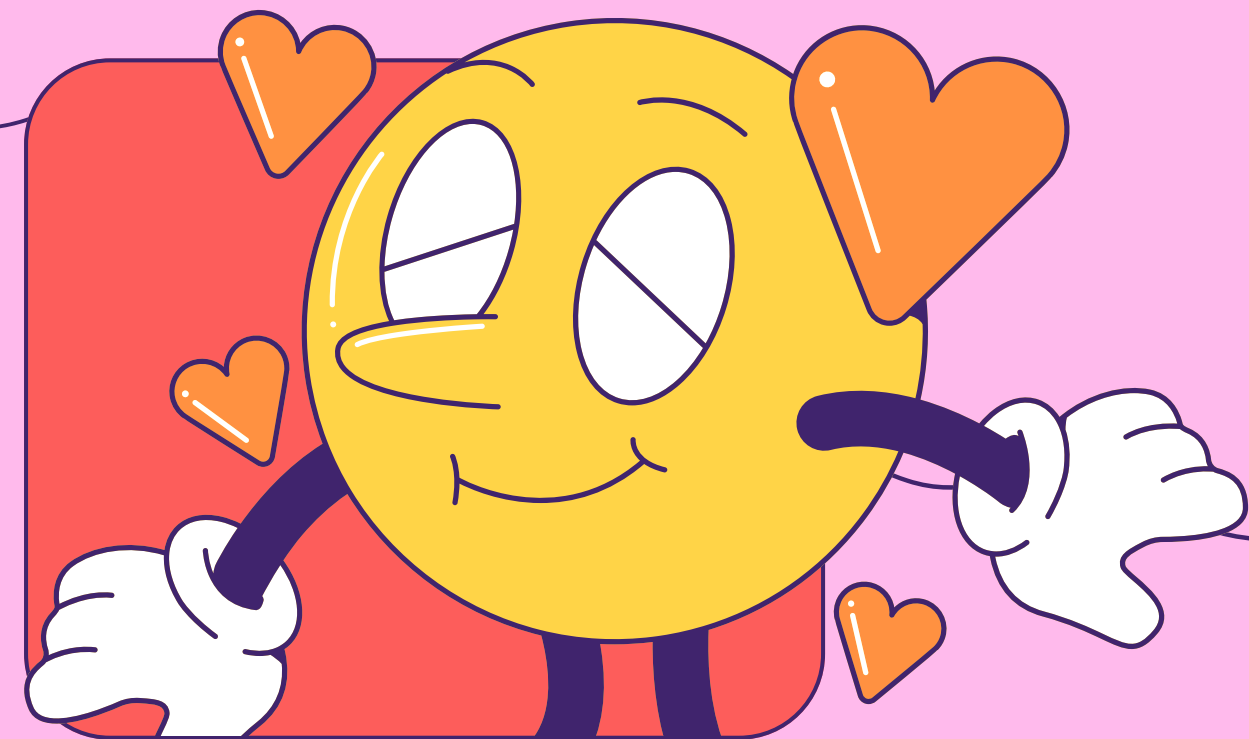
```
Scenario: Actrice Yang se présente en tant que Rapunzel quand il joue le rôle de Rapunzel
  Given une actrice nommé Yang
  And un personnage qui est une princesse nommée Rapunzel
  When Yang joue le rôle de Rapunzel
  Then Yang devrait se présenter en tant que Rapunzel
```

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

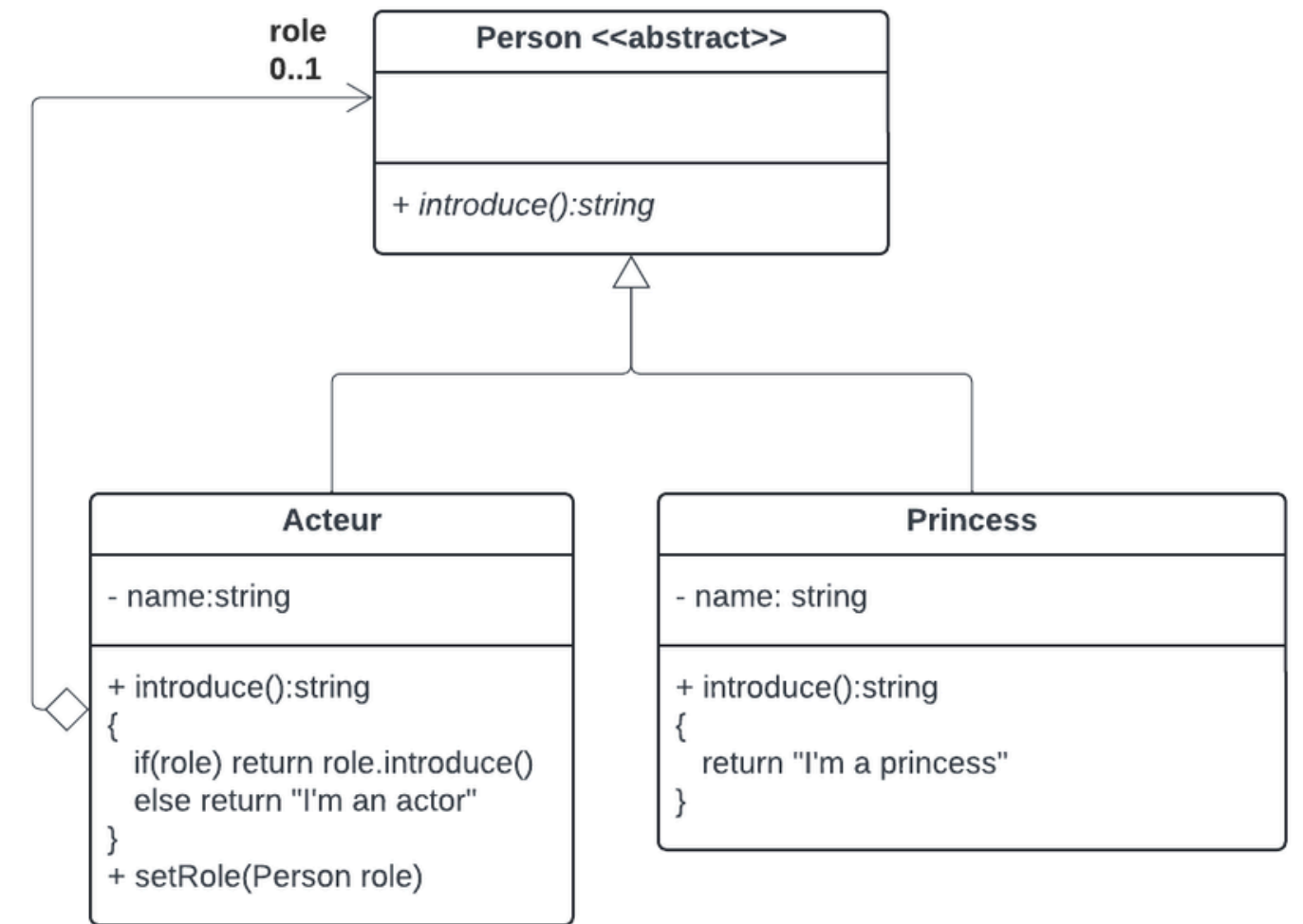
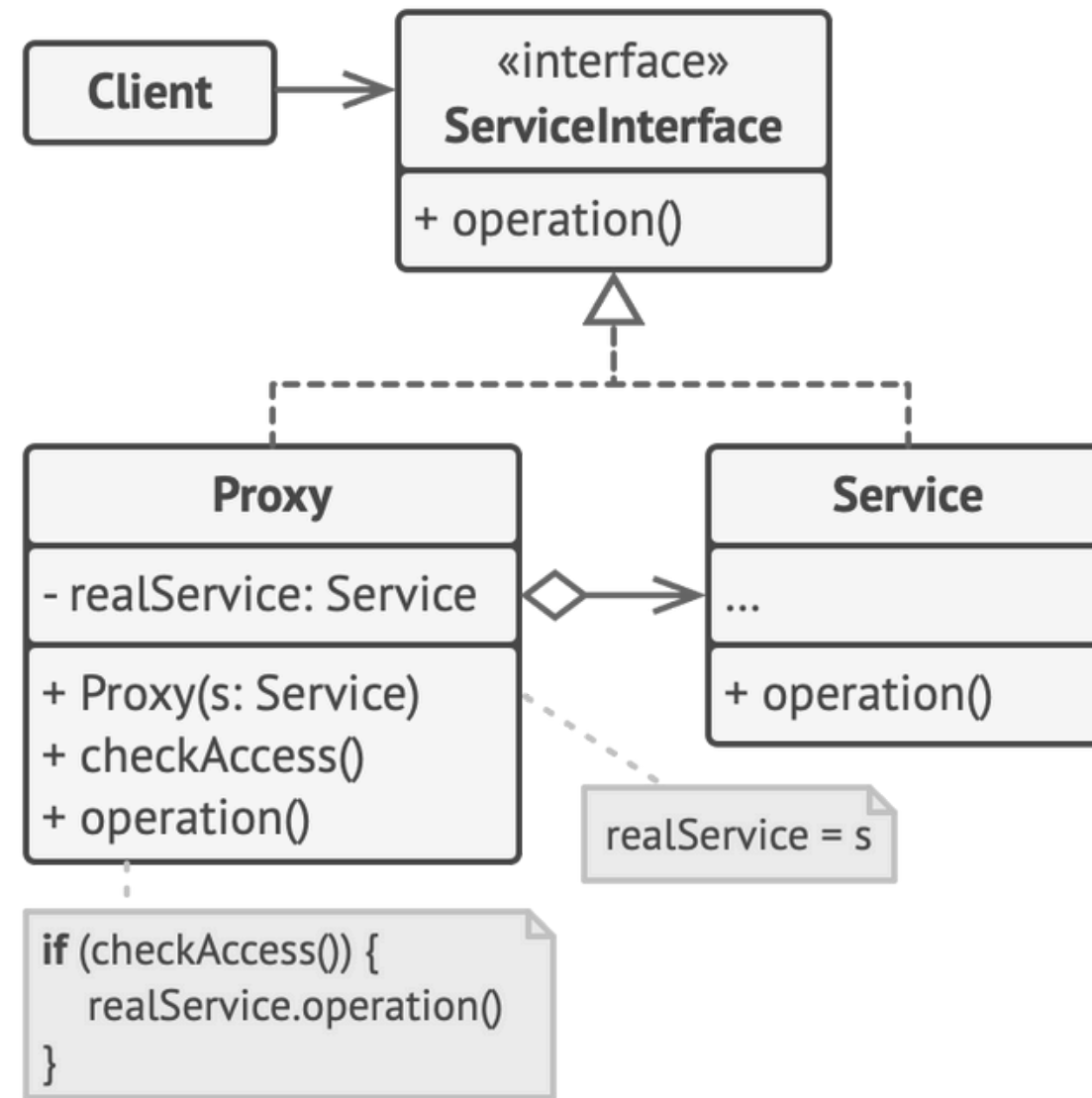
```
Scenario Outline: Un acteur se présente en tant que personnage qu'il joue
  Given un acteur/actrice nommé <actor>
  And un personnage qui est <role> nommé <roleName>
  When <actor> joue le rôle de <roleName>
  Then <actor> devrait se présenter en tant que <roleName>, sinon il/elle se présente en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

Examples:

actor	role	roleName	expectedIntroduction
Yang	Princess	Rapunzel	Je suis une pricesse, 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



Proxy



```
Person princess = new Princess()
```

```
Person actor1 = new Actor()
```

```
print actor1.introduce() => "I'm an actor"
```

```
actor1.setRole(princess)
```

```
print actor1.introduce() => "I'm a princess"
```

L'acteur qui joue le personnage est comme un « proxy » : c'est via lui que le personnage peut sortir de son contexte de film, interagir avec la vraie vie et « introduce himself » aux intervieweurs et aux auditeurs!


```
@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 nommé Yang # features/steps/introduce_steps.py:18
    And un personnage qui est une princesse nommée Rapunzel # features/steps/introduce_steps.py:22
    When Yang joue le rôle de Rapunzel # features/steps/introduce_steps.py:26
    Then Yang devrait se présenter en tant que Rapunzel # features/steps/introduce_steps.py:30

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

  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
    And un personnage qui est Princess nommé Rapunzel # features/steps/introduce_steps.py:48
    When Yang joue le rôle de Rapunzel # features/steps/introduce_steps.py:26
    Then Yang devrait se présenter en tant que Rapunzel, sinon il/elle se présente en tant que Yang : Je suis une pricesse, mon nom est Rapunzel # features/steps/introduce_steps.py:62

  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
    And un personnage qui est Prince nommé Flynn # features/steps/introduce_steps.py:48
    When Flynn joue le rôle de Flynn # features/steps/introduce_steps.py:57
    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

  Scenario Outline: Un acteur se présente en tant que personnage qu'il joue -- @1.3 # features/Introduce.feature:28
    Given un acteur/actrice nommé Yang # features/steps/introduce_steps.py:44
    And un personnage qui est None nommé None # features/steps/introduce_steps.py:48
    When Yang joue le rôle de None # features/steps/introduce_steps.py:57
    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
```

Tests

```
4 features passed, 0 failed, 0 skipped
17 scenarios passed, 0 failed, 0 skipped
63 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m0.024s
```

Nouveau user story

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.

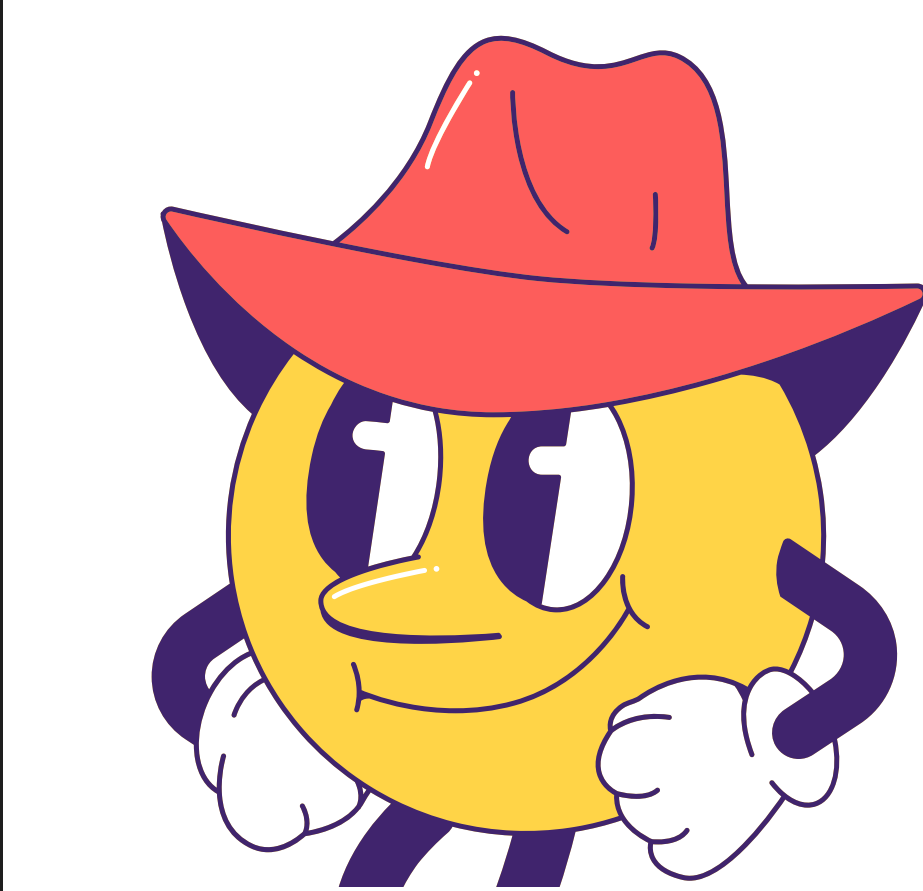
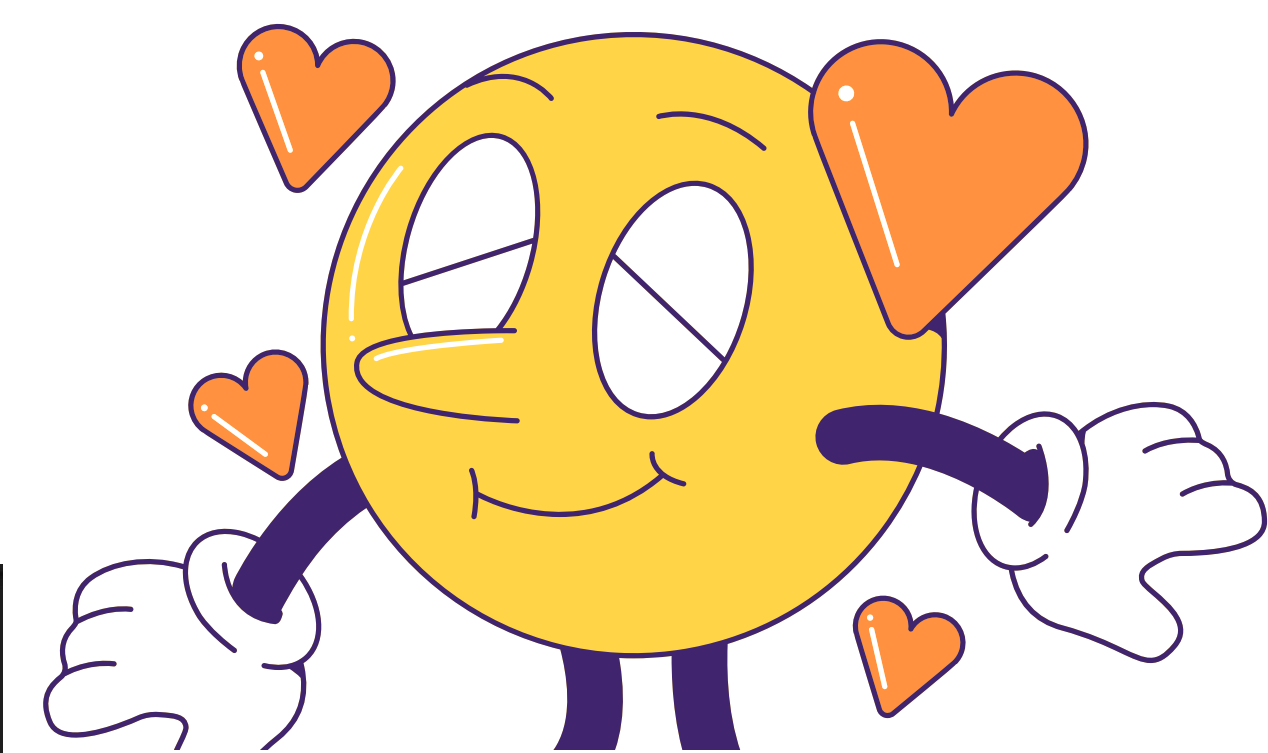
```
# Author: yang.yang2@dauphine.eu
@tag
Feature: Observer – Création de suite de série après sauvetage de la princesse
  En tant que princesse
  Je souhaite que mon histoire ne se termine pas après avoir été sauvée par le prince, et que les séries qui m’abonnent créent automatiquement une suite
  Afin 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 titrée "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"

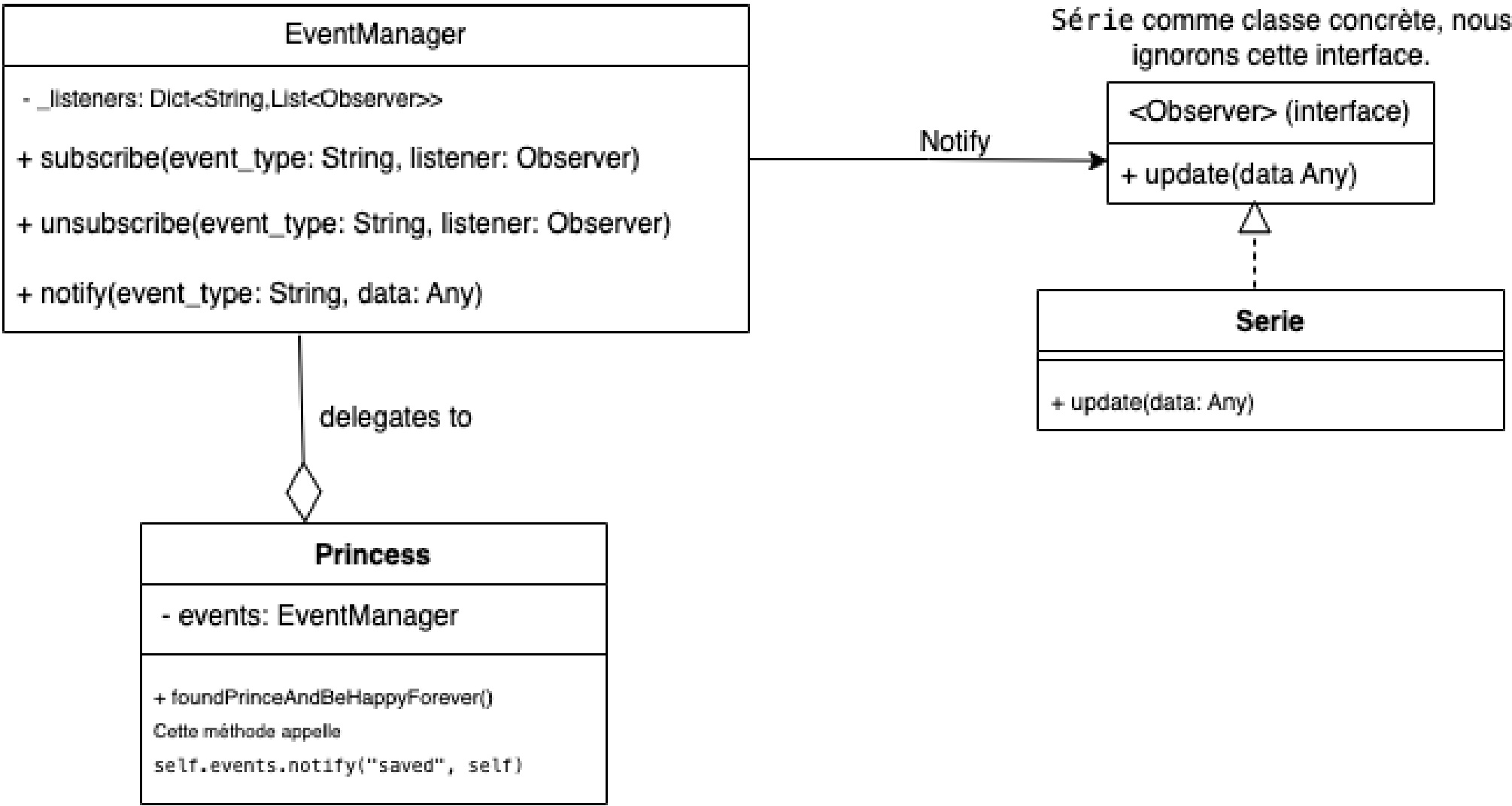
  Scenario: Une série qui a déjà une suite n'en crée pas de nouvelle
    Given une princesse nommée "Ariel" qui n'a pas encore été sauvée
    And un prince nommé "Eric" qui n'a pas de princesse
    And la série "La Petite Sirène" possède déjà une suite intitulée "La Petite Sirène – Suite"
    And la série "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 titrée "<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 |
```



Observateur



La princesse est sauvée, l'histoire se termine ?
Non ! Un nouveau chapitre(Serie) commence.

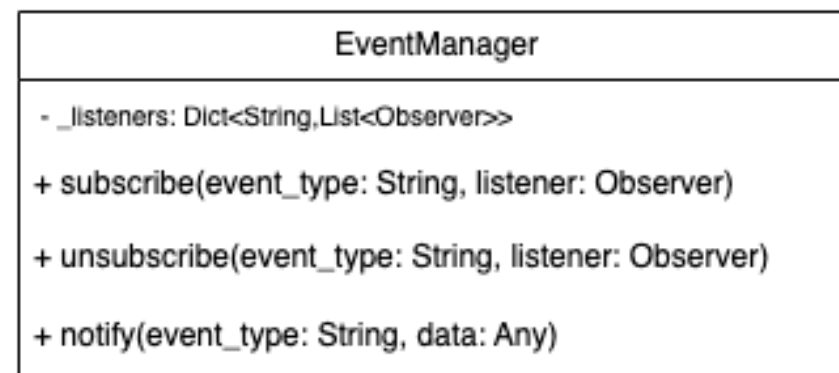
Observateur

```
class EventManager:
    def __init__(self):
        # Stockage : { type_d'événement : [ écouteur1, écouteur2, ... ] }
        self._listeners = {}

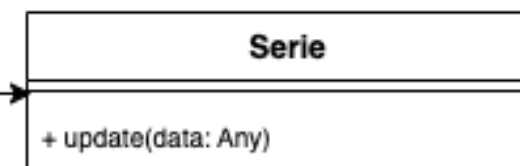
    def subscribe(self, event_type, listener):
        self._listeners.setdefault(event_type, []).append(listener)

    def unsubscribe(self, event_type, listener):
        if event_type in self._listeners:
            self._listeners[event_type].remove(listener)

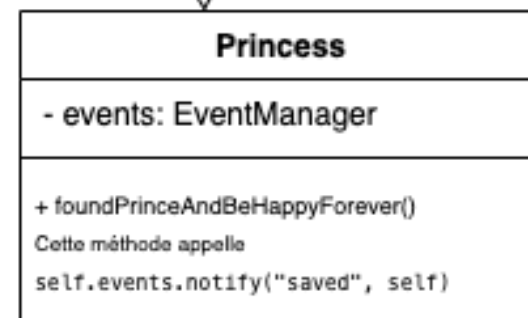
    def notify(self, event_type, data):
        for listener in self._listeners.get(event_type, []):
            listener.update(data)
```



Notify



delegates to



```
# Rappel de l'observateur
def update(self, subject):
    if isinstance(subject, Princess) and subject.isSaved():
        # **Ajouter la suite uniquement s'il n'existe pas déjà de suite**
        if self._suite is None:
            self.addSuite()
```

```
def foundPrinceAndBeHappyForever(self):
    self.alreadySaved = True
    self.events.notify("saved", self)
```


Tests

```
@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_creation_serie.py:6 0.000s
    And un prince nommé "Eric" qui n'a pas de princesse # features/steps/auto_creation_serie.py:10 0.000s
    And une série titrée "La Petite Sirène" sans suite enregistrée # features/steps/auto_creation_serie.py:14 0.000s
    And la série "La Petite Sirène" est abonnée à la princesse "Ariel" # features/steps/auto_creation_serie.py:19 0.000s
    When le prince "Eric" sauve la princesse "Ariel" # features/steps/auto_creation_serie.py:37 0.000s
    Then la série "La Petite Sirène" devrait avoir une suite intitulée "La Petite Sirène – Suite" # features/steps/auto_creation_serie.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_creation_serie.py:6 0.000s
    And un prince nommé "Eric" qui n'a pas de princesse # features/steps/auto_creation_serie.py:10 0.000s
    And la série "La Petite Sirène" possède déjà une suite intitulée "La Petite Sirène – Suite" # features/steps/auto_creation_serie.py:26 0.000s
    And la série "La Petite Sirène" est abonnée à la princesse "Ariel" # features/steps/auto_creation_serie.py:19 0.000s
    When le prince "Eric" sauve la princesse "Ariel" # features/steps/auto_creation_serie.py:37 0.000s
    Then la série "La Petite Sirène" ne devrait pas ajouter de suite supplémentaire # features/steps/auto_creation_serie.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_creation_serie.py:6 0.000s
    And un prince nommé "Adam" qui n'a pas de princesse # features/steps/auto_creation_serie.py:10 0.000s
    And une série titrée "La Belle et la Bête" sans suite enregistrée # features/steps/auto_creation_serie.py:14 0.000s
    And la série "La Belle et la Bête" est abonnée à la princesse "Belle" # features/steps/auto_creation_serie.py:19 0.000s
    When le prince "Adam" sauve la princesse "Belle" # features/steps/auto_creation_serie.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_creation_serie.py:46 0.000s

1 feature passed, 0 failed, 0 skipped
3 scenarios passed, 0 failed, 0 skipped
18 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m0.003s
```

Observateur - Tests unitaires

```
# tests/test_observateur.py

import pytest
from src.Kingdom import Prince, Princess
from src.Serie import Serie

@pytest.fixture
def setup_world():
    """Prépare la princesse, le prince et deux séries."""
    princess = Princess("Ariel")
    prince = Prince("Eric")
    serie1 = Serie(titre="La Belle et la Bête", nb_saisons=2)
    serie2 = Serie(titre="La Petite Sirène", nb_saisons=1)
    # On donne déjà une suite à serie2
    existing_suite = Serie(titre="La Petite Sirène 2", nb_saisons=1)
    serie2.suite = existing_suite

    # On abonne les séries à l'événement "saved" de la princesse
    princess.events.subscribe("saved", serie1)
    princess.events.subscribe("saved", serie2)

    return princess, prince, serie1, serie2, existing_suite

def test_series_without_suite_gets_one(setup_world):
    """Si une série n'a pas de suite, elle doit en créer une après sauvetage."""
    princess, prince, serie1, _, _ = setup_world

    prince.savePrincess(princess)

    assert serie1.suite is not None
    assert serie1.suite.titre == "La Belle et la Bête ☞ Suite"

def test_series_with_existing_suite_untouched(setup_world):
    """Une série ayant déjà une suite ne doit pas en recréer une."""
    princess, prince, _, serie2, existing_suite = setup_world

    prince.savePrincess(princess)

    assert serie2.suite is existing_suite
    assert serie2.suite.titre == "La Petite Sirène 2"
```

```
.venvyangyang@yangyangdeMacBook-Pro tests % pytest -q

..
2 passed in 0.02s
```

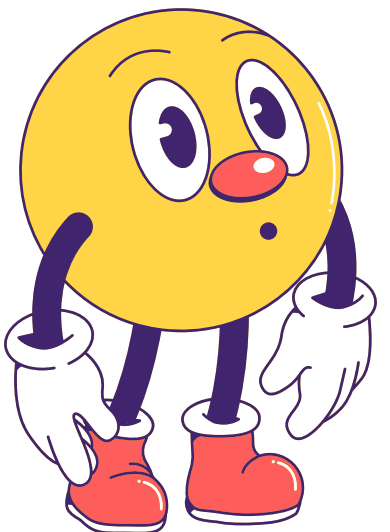
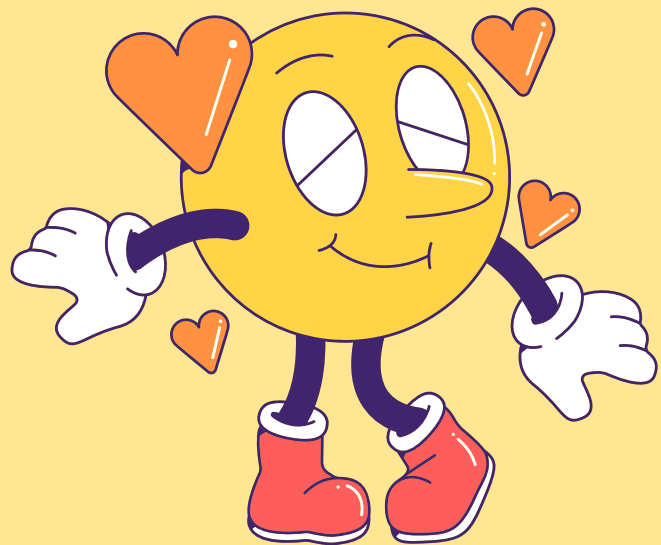
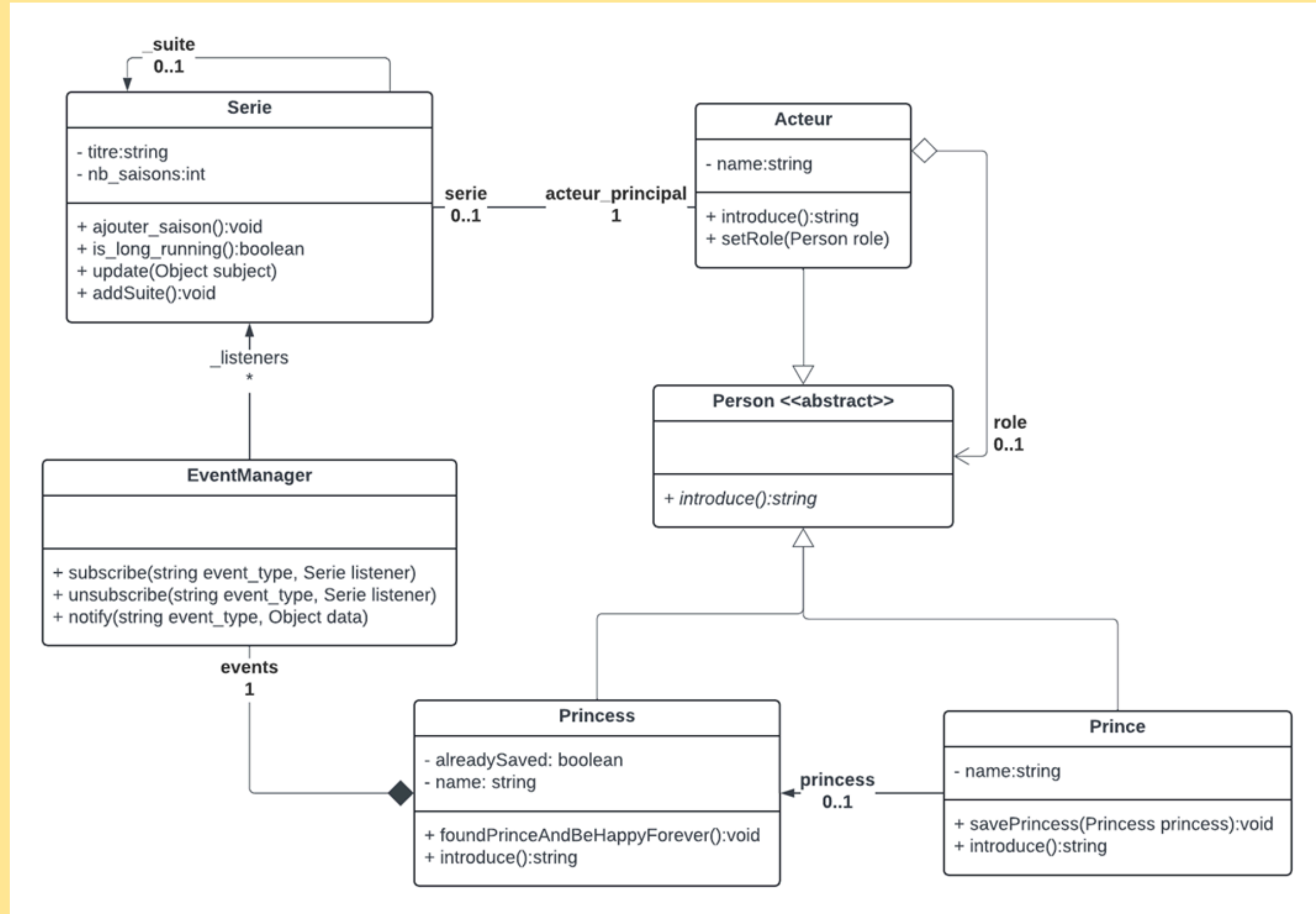


Diagramme de classe - complet





Conclusions

Merci beaucoup!

N'hésitez pas à poser des questions

