

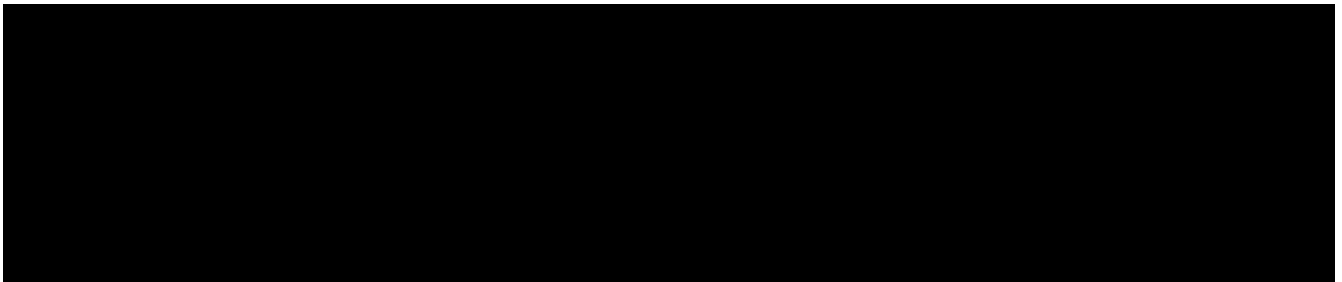
SAE 2.01 - 2.02 - Rapport de Dřveloppement Orientř Objet

Table of Contents

1. Semaine 1-2	1
1.1. UML Version 1-2	1
2. Semaine 3-4	1
2.1. UML Version 3-4	1
2.2. Principales řvolutions (Version 1-2 vers Version 3-4)	1
2.2.1. Fichier <code>src/main/Adolescent.java</code>	2
2.2.2. Fichier <code>src/main/Criteres.java</code>	2

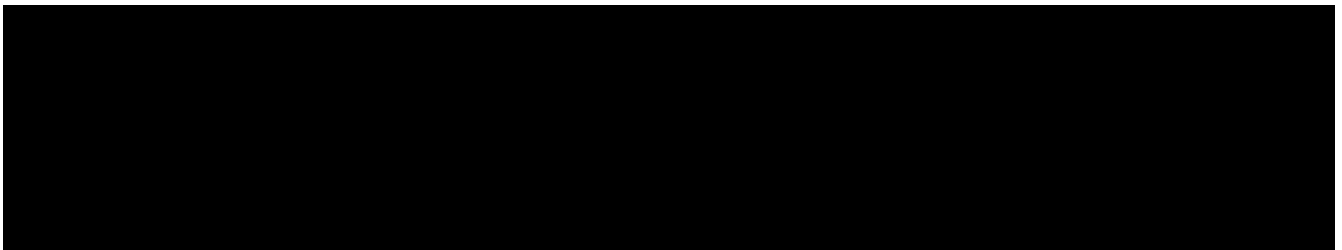
1. Semaine 1-2

1.1. UML Version 1-2



2. Semaine 3-4

2.1. UML Version 3-4



2.2. Principales řvolutions (Version 1-2 vers Version 3-4)

Voici le řsumř des modifications apportřes dans le řpertoire `src` depuis le commit `P00-v1`

(correspondant à la Version 1-2) jusqu'à la version actuelle (correspondant à la Version 3-4) :

Les modifications se concentrent sur les deux principaux aspects correspondant aux objectifs des semaines 3 et 4 : 1. Gestion de la validité des critères par un mécanisme d'exception 2. Développement des règles spécifiques de compatibilité pour certains pays

2.2.1. Fichier `src/main/Adolescent.java`

¥ Méthode `addCriterion(Criteres criterion, String value, boolean isHost)`:

- ! La signature a changé pour inclure la déclaration `throws IllegalArgumentException`.
- ! Validation améliorée : La méthode `criterion.isValid(value)` est appelée en premier et peut lever une `IllegalArgumentException` si la valeur n'est pas valide.
- ! Filtre HOST/GUEST : Empêche d'ajouter un critère de type `GUEST_` à un adolescent hôte et vice-versa pour maintenir la cohérence des données.

¥ Évolution des méthodes de compatibilité :

- ! `isAnimalCompatible(Adolescent other)` renommé en `animalScore(Adolescent other)` : retourne maintenant un score numérique (0 si compatible, -25 si incompatible).
- ! `isDietCompatible(Adolescent other)` renommé en `dietScore(Adolescent other)` : retourne un score numérique avec gestion robuste des `NullPointerException` lors du traitement des régimes alimentaires.

¥ Nouvelle méthode `isFrenchCompatible(Adolescent other)`:

- ! Vérifie la compatibilité si l'un des adolescents est français.
- ! Si aucun n'est français, ils sont compatibles.
- ! Si l'un est français, ils doivent avoir au moins un hobby en commun.
- ! Gère les `NullPointerException` lors du traitement des hobbies.

¥ Méthode `isCompatible(Adolescent other)`:

- ! La logique de compatibilité a été modifiée pour inclure la vérification `isFrenchCompatible(other)`.
- ! Utilise maintenant les nouvelles méthodes `animalScore()` et `dietScore()` pour déterminer la compatibilité.

2.2.2. Fichier `src/main/Criteres.java`

¥ Méthode `isValid(String valeur)`:

- ! Déclare maintenant `throws IllegalArgumentException`.
- ! Logique de validation grandement améliorée et plus stricte :
 - " Gère `null` différemment selon le critère (permis pour `PAIR_GENDER`, `HISTORY`, `GUEST_FOOD`, `HOST_FOOD`, `HOBBIES` mais lève une exception pour `GUEST_ANIMAL_ALLERGY`, `HOST_HAS_ANIMAL_GENDER`).
 - " Retourne `true` si toutes les validations passent.
 - " Lève une `IllegalArgumentException` avec un message d'erreur spécifique si la validation

Žhoue.