

Projet CSC4102 : « Suivi d'activité de projet »

Guillemet Samuel, Safon Clément

Année 2022-2023 - 1er février 2023

Table des matières

1	Specification	4
1.1	Diagrammes de cas d'utilisation	4
	Diagramme des cas d'utilisations des développeurs	4
	Diagramme des cas d'utilisations des activités	4
	Diagramme des cas d'utilisations des tâches	4
	Diagramme des cas d'utilisations des périodes de travail	5
	Diagramme des cas d'utilisations des labels	5
	Diagramme des cas d'utilisations de la corbeille	6
	Diagramme des cas d'utilisations des calculs	6
1.2	Priorités, et préconditions et postconditions des cas d'utilisation	7
	Créer un développeur	7
	Créer une activité	7
	Créer une tâche	7
	Créer une période de travail	7
	Mettre un développeur à la corbeille	8
	Mettre une tâche à la corbeille	8
	Restaurer une tâche	8
2	Préparation des tests de validation	9
2.1	Tables de décision des tests de validation	9
	Créer un développeur	9
	Créer une activité	9
	Créer une tâche	9
	Créer une période de travail	10
	Mettre un développeur à la corbeille	11
	Mettre une activité à la corbeille	11
	Mettre une tâche à la corbeille	11
	Afficher les développeurs	12
	Afficher les activités	12
	Afficher les tâches d'une activité	12
	Afficher les périodes de travail	12
	Afficher les périodes de travail d'un développeur	12
	Afficher les développeurs à la corbeille	13
	Afficher les activités à la corbeille	13
	Afficher les tâches à la corbeille	13
	Afficher les périodes de travail à la corbeille	13
	Restaurer une tâche	14
3	Conception	15
3.1	Diagrammes de classes	15
3.2	Diagrammes de séquences	17
	Créer un développeur	17
	Créer une activité	18
	Créer une tâche	19
	Mettre un développeur à la corbeille	20
	Mettre une tâche à la corbeille	21
4	Diagrammes de machine à états et invariants	22
4.1	Diagramme des machines à états des développeurs	22
4.2	Diagramme des machines à états des tâches	23
5	Fiche des classes	24
5.1	Classe Développeur	24
5.2	Classe Tâche	25
6	Préparation des tests unitaires	26
6.1	Classe Développeur	26
6.2	Classe Activite	27

6.3 Classe Tâche	28
6.4 Classe PeriodeDeTravail	29
6.5 Classe Corbeille	30

1 Specification

1.1 Diagrammes de cas d'utilisation

Diagramme des cas d'utilisations des développeurs

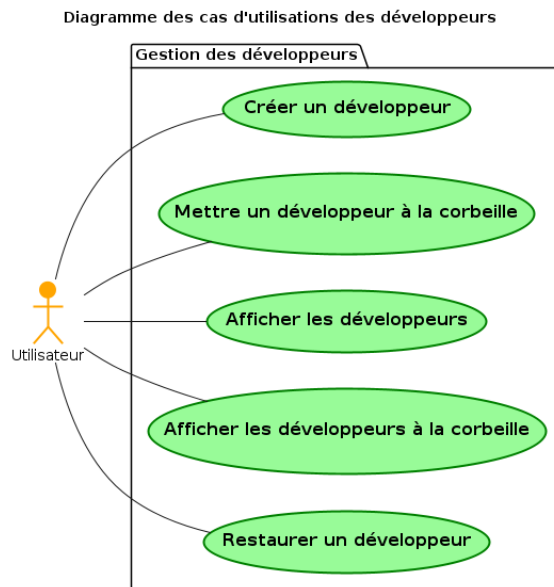


Fig.1 - Diagramme des cas d'utilisations des développeurs

Diagramme des cas d'utilisations des activités

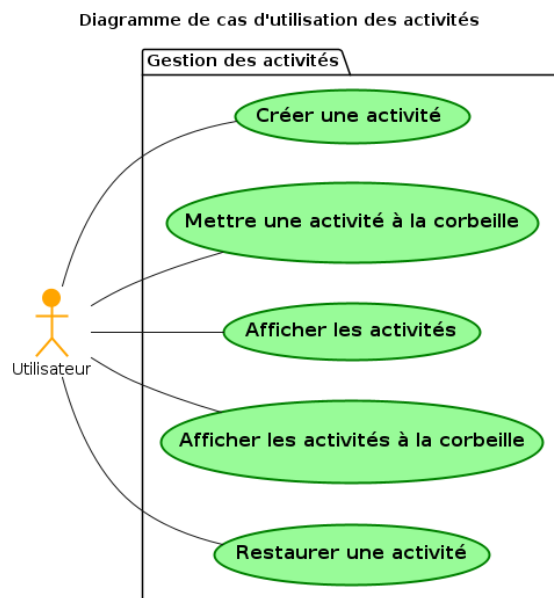


Fig.2 - Diagramme des cas d'utilisations des activités

Diagramme des cas d'utilisations des tâches

Diagramme de cas d'utilisation des tâches

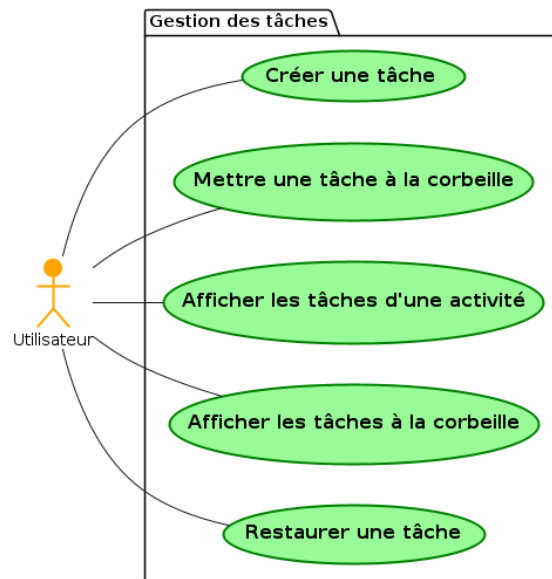


Fig.3 - Diagramme des cas d'utilisations des tâches

Diagramme des cas d'utilisations des périodes de travail

Diagramme de cas d'utilisation des périodes de travail

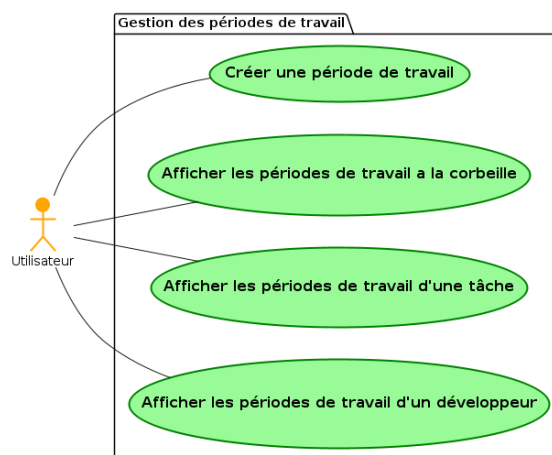


Fig.4 - Diagramme des cas d'utilisations des périodes de travail

Diagramme des cas d'utilisations des labels

Diagramme de cas d'utilisation des labels

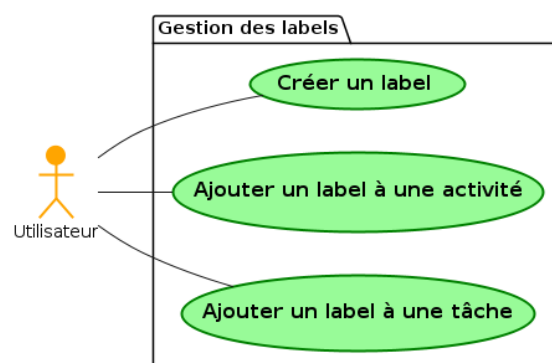


Fig.5 - Diagramme des cas d'utilisations des labels

Diagramme des cas d'utilisations de la corbeille

Diagramme de cas d'utilisation de la corbeille

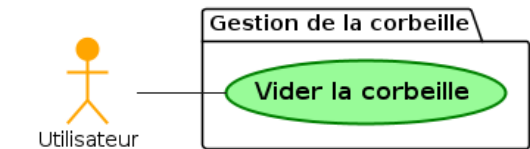


Fig.6 - Diagramme des cas d'utilisations de la corbeille

Diagramme des cas d'utilisations des calculs

Diagramme de cas d'utilisation des calculs

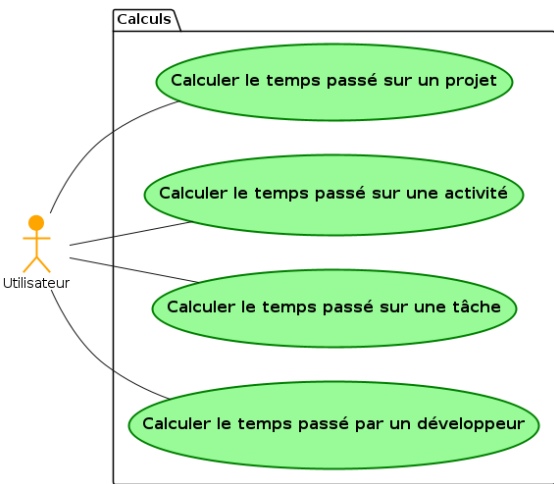


Fig.7 - Diagramme des cas d'utilisations des calculs

1.2 Priorités, et préconditions et postconditions des cas d'utilisation

Le champ enFonctionnement est un booléen qui indique si l'objet est dans la corbeille ou non.

Créer un développeur

Priorité : 1

Préconditions:

- nom fourni non null et non vide
- prenom fourni non null et non vide
- alias fourni non null et non vide
- alias unique

Postconditions:

- Création d'un développeur
 - Ajout du développeur à la liste des développeurs
-

Créer une activité

Priorité : 1

Préconditions:

- nom fourni non null et non vide
- id fourni non null et non vide
- id unique

Postconditions:

- Création d'une activité
 - Ajout de l'activité à la liste des activités
-

Créer une tâche

Priorité : 1

Préconditions:

- nom fourni non null et non vide
- id fourni non null et non vide
- activiteId fourni non null et non vide
- activite existante
- activite enFonctionnement
- id unique

Postconditions:

- Création d'une tâche
 - Tâche liée à l'activité
-

Créer une période de travail

Priorité : 1

Préconditions:

- date_de_debut fournie non null et non vide
- date_de_date_de_fin fournie non null et non vide
- date_de_debut < date_de_fin
- activiteId fournie non null et non vide
- activite existante
- activite enFonctionnement
- tacheId fournie non null et non vide
- tache existante
- tache enFonctionnement

- développeur(s) fourni(s) non null et non vide
- développeur(s) existant(s)
- développeur(s) enFonctionnement(s)
- Pas de chevauchement avec une autre période de travail

Postconditions:

- Création d'une période de travail
 - Période de travail liée à la tâche
 - Période de travail liée au développeur
-

Mettre un développeur à la corbeille

Priorité : 2

Préconditions:

- alias fourni non null et non vide
- Développeur existant

Postconditions:

- Développeur mis à la corbeille
 - Périodes de travail du développeur mises à la corbeille
-

Mettre une tâche à la corbeille

Priorité : 2

Préconditions:

- activiteld fourni non null et non vide
- activite existante
- tacheld fourni non null et non vide
- tache existante

Postconditions:

- Tâche mise à la corbeille
 - Périodes de travail de la tâche mises à la corbeille
-

Restaurer une tâche

Priorité : 1

Préconditions:

- activiteld fourni non null et non vide
- activite existante
- tacheld fourni non null et non vide
- tache existante

Postconditions:

- Tâche active
- Périodes de travail de la tâche en fonctionnement

2 Préparation des tests de validation

2.1 Tables de décision des tests de validation

Créer un développeur

Table de decision:

Préconditions	1	2	3	4	5
<i>nom fourni</i>	Non	Oui	Oui	Oui	Oui
<i>prenom fourni</i>		Non	Oui	Oui	Oui
<i>alias fourni</i>			Non	Oui	Oui
<i>alias unique</i>				Non	Oui
Postconditions					
<i>Création d'un développeur</i>	Non	Non	Non	Non	Oui
<i>Ajout du développeur</i>	Non	Non	Non	Non	Oui
Nombre de tests	2	2	2	1	1

Tests 8

Créer une activité

Table de decision:

Préconditions	1	2	3	4
<i>nom fourni</i>	Non	Oui	Oui	Oui
<i>id fourni</i>		Non	Oui	Oui
<i>id unique</i>			Non	Oui
Postconditions				
<i>Création d'une activité</i>	Non	Non	Non	Oui
<i>Ajout de l'activité</i>	Non	Non	Non	Oui
Nombre de tests	2	2	2	1

Tests 7:

Créer une tâche

Table de decision:

Préconditions	1	2	3	4	5	6	7
<i>nom fourni</i>	Non	Oui	Oui	Oui	Oui	Oui	Oui
<i>id fourni</i>		Non	Oui	Oui	Oui	Oui	Oui
<i>activiteId fournie</i>			Non	Oui	Oui	Oui	Oui
<i>activité existante</i>				Non	Oui	Oui	Oui
<i>activité enFonctionnement</i>					Non	Oui	Oui
<i>id unique</i>						Non	Oui
Postconditions							
<i>Création d'une tâche</i>	Non	Non	Non	Non	Non	Non	Oui
<i>Tâche liée à l'activité</i>	Non	Non	Non	Non	Non	Non	Oui
Nombre de tests	2	2	2	1	1	1	1

Tests 6:

Créer une période de travail

Table de decision:

Préconditions	1	2	3	4	5	6	7	8	9	10	11	12	13	14
<i>date_de_debut fourni</i>	Non	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
<i>date_de_fin fournie</i>		Non	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
<i>date_de_debut < date_de_fin</i>			Non	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
<i>activiteId fournie</i>				Non	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
<i>activite existante</i>					Non	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
<i>activite enFonctionnement</i>						Non	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
<i>tacheId fournie</i>							Non	Oui	Oui	Oui	Oui	Oui	Oui	Oui
<i>tache existante</i>								Non	Oui	Oui	Oui	Oui	Oui	Oui
<i>tache enFonctionnement</i>									Non	Oui	Oui	Oui	Oui	Oui
<i>développeur fourni</i>										Non	Oui	Oui	Oui	Oui
<i>développeur existant</i>											Non	Oui	Oui	Oui
<i>développeur enFonctionnement</i>												Non	Oui	Oui
<i>Pas de chevauchement avec une autre période de travail</i>													Non	Oui
Postconditions														
<i>Création d'une période de travail</i>	Non	Non	Non	Non	Non	Non	Non	Non	Non	Non	Non	Non	Non	Oui
<i>Période de travail liée à la tâche</i>	Non	Non	Non	Non	Non	Non	Non	Non	Non	Non	Non	Non	Non	Oui
<i>Période de travail liée au développeur</i>	Non	Non	Non	Non	Non	Non	Non	Non	Non	Non	Non	Non	Non	Oui
Nombre de tests	1	1	1	2	1	1	2	1	1	2	1	1	1	1

Tests 17:

Mettre un développeur à la corbeille

Table de decision:

Préconditions	1	2	3
<i>alias fourni</i>	Non	Oui	Oui
<i>Développeur existant</i>		Non	Oui
Postconditions			
<i>Développeur mis à la corbeille</i>	Non	Non	Oui
<i>Périodes de travail du développeur mises à la corbeille</i>	Non	Non	Oui
Nombre de tests	2	1	2

Tests 4:

Mettre une activité à la corbeille

Table de decision:

Préconditions	1	2	3
<i>id fourni</i>	Non	Oui	Oui
<i>Activité existante</i>		Non	Oui
Postconditions			
<i>Activité mise à la corbeille</i>	Non	Non	Oui
<i>Tâches de l'activité mises à la corbeille</i>	Non	Non	Oui
Nombre de tests	2	1	1

Tests 4:

Mettre une tâche à la corbeille

Table de decision:

Préconditions	1	2	3	4	5
<i>activiteId fourni</i>	Non	Oui	Oui	Oui	Oui
<i>tacheId fournie</i>		Non	Oui	Oui	Oui
<i>activite existante</i>			Non	Oui	Oui
<i>tache existante</i>				Non	Oui
Postconditions					
<i>Tâche mise à la corbeille</i>	Non	Non	Non	Non	Oui
<i>Périodes de travail de la tâche mises à la corbeille</i>	Non	Non	Non	Non	Oui
Nombre de tests	2	2	1	1	2

Tests 7:

Afficher les développeurs

Table de decision:

Préconditions	1	2
<i>printType correct</i>	Non	Oui
Postconditions		
<i>Affichage des développeurs</i>	Non	Oui
Nombre de tests	1	3

Tests 4:

Afficher les activités

Table de decision:

Préconditions	1	2
<i>printType correct</i>	Non	Oui
Postconditions		
<i>Affichage des activités</i>	Non	Oui
Nombre de tests	1	3

Tests 4:

Afficher les tâches d'une activité

Table de decision:

Préconditions	1	2	3	4
<i>activiteId fourni</i>	Non	Oui	Oui	Oui
<i>activite existante</i>		Non	Oui	Oui
<i>printType correct</i>			Non	Oui
Postconditions				
<i>Affichage des tâches</i>	Non	Non	Non	Oui
Nombre de tests	2	1	1	3

Tests 7:

Afficher les périodes de travail

Table de decision:

Préconditions	1	2	3	4	5	6
<i>activiteId fourni</i>	Non	Oui	Oui	Oui	Oui	Oui
<i>tacheId fournie</i>		Non	Oui	Oui	Oui	Oui
<i>activite existante</i>			Non	Oui	Oui	Oui
<i>tache existante</i>				Non	Oui	Oui
<i>printType correct</i>					Non	Oui
Postconditions						
<i>Affichage des périodes de travail</i>	Non	Non	Non	Non	Non	Oui
Nombre de tests	2	2	1	1	1	3

Tests 10:

Afficher les périodes de travail d'un développeur

Table de decision:

Préconditions	1	2	3	4
<i>alias fourni</i>	Non	Oui	Oui	Oui
<i>alias existante</i>		Non	Oui	Oui
<i>printType correct</i>			Non	Oui
Postconditions				
<i>Affichage des periodes de travail</i>	Non	Non	Non	Oui
Nombre de tests	2	1	1	3

Tests 7:

Afficher les développeurs à la corbeille

Table de decision:

Préconditions	1
Postconditions	
<i>Affichage des développeurs</i>	Oui
Nombre de tests	1

Tests 1:

Afficher les activités à la corbeille

Table de decision:

Préconditions	1
Postconditions	
<i>Affichage des activités</i>	Oui
Nombre de tests	1

Tests 1:

Afficher les tâches à la corbeille

Table de decision:

Préconditions	1
Postconditions	
<i>Affichage des tâches</i>	Oui
Nombre de tests	1

Tests 1:

Afficher les périodes de travail à la corbeille

Table de decision:

Préconditions	1
Postconditions	
<i>Affichage des périodes de travail</i>	Oui
Nombre de tests	1

Tests 1:

Restaurer une tâche

Table de decision:

Préconditions	1	2	3	4	5	6
<i>activiteId fourni</i>	Non	Oui	Oui	Oui	Oui	Oui
<i>tacheId fournie</i>		Non	Oui	Oui	Oui	Oui
<i>activite existante</i>			Non	Oui	Oui	Oui
<i>tache existante</i>				Non	Oui	Oui
Postconditions						
<i>Tache enFonctionnement</i>	Non	Non	Non	Non	Non	Oui
<i>Périodes de travail de la tâche enFonctionnement</i>	Non	Non	Non	Non	Non	Oui
Nombre de tests	2	2	1	1	1	1

Tests 9:

Par ailleurs la responsabilité de condition de restauration a été déporté dans les classes elles-mêmes. Par exemple, la classe `Tache` a une méthode `restaurer()` qui vérifie si son activité est en fonctionnement. Si c'est le cas, elle change son état et appelle sa méthode `specificRestaurer()`. La méthode `specificRestaurer()` dans les classes d'`ElementJetable` permet d'effectuer les opérations plus spécifiques à chaque classe. Par exemple, la méthode `specificRestaurer()` de la classe `Tache` appelle la méthode `restaurer()` de ses périodes de travail.

3 Conception

3.1 Diagrammes de classes

Nous définissons ici les diagrammes de classes de notre projet. Le champ `enFonctionnement` indique si un élément est dans la corbeille ou non.

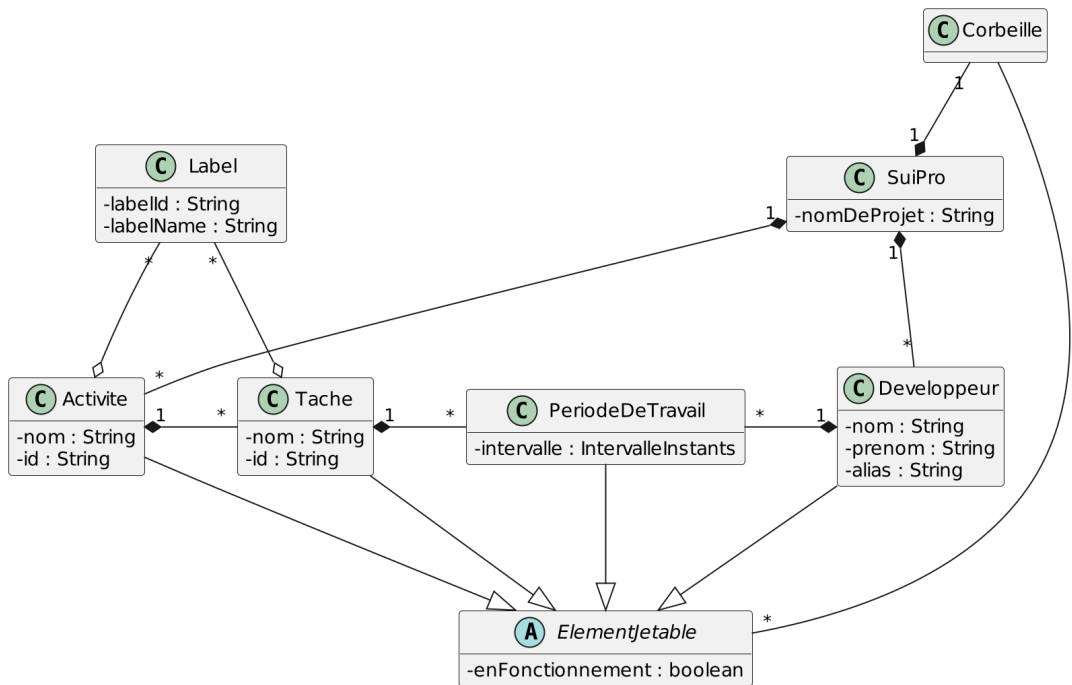


Fig.8 - Diagramme de classes

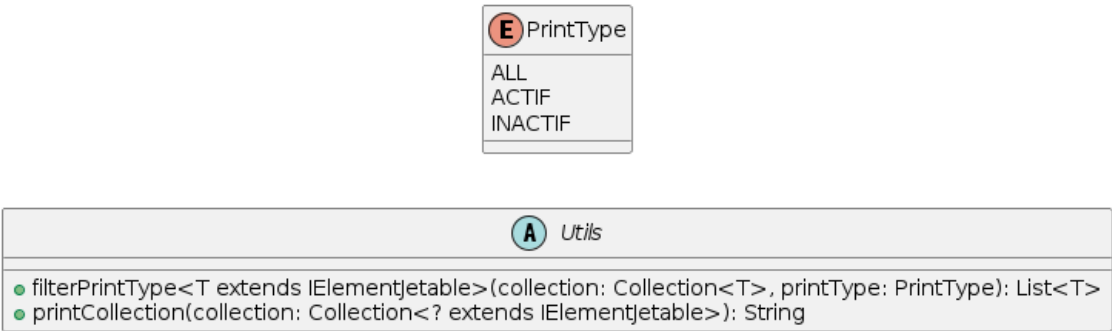


Fig.9 - Diagramme de classes supplémentaire

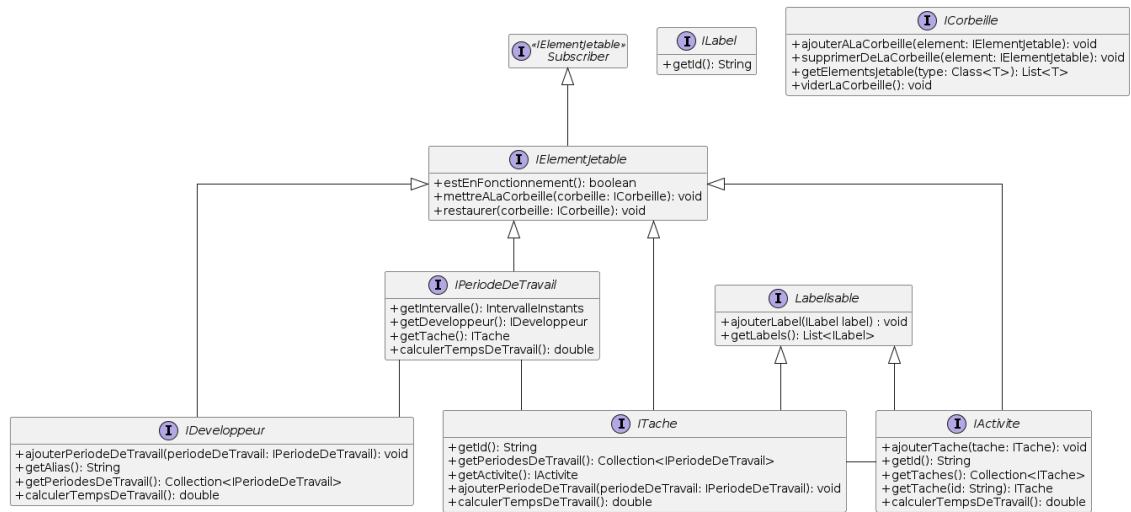


Fig.10 - Diagramme de classes des interfaces

3.2 Diagrammes de séquences

Créer un développeur

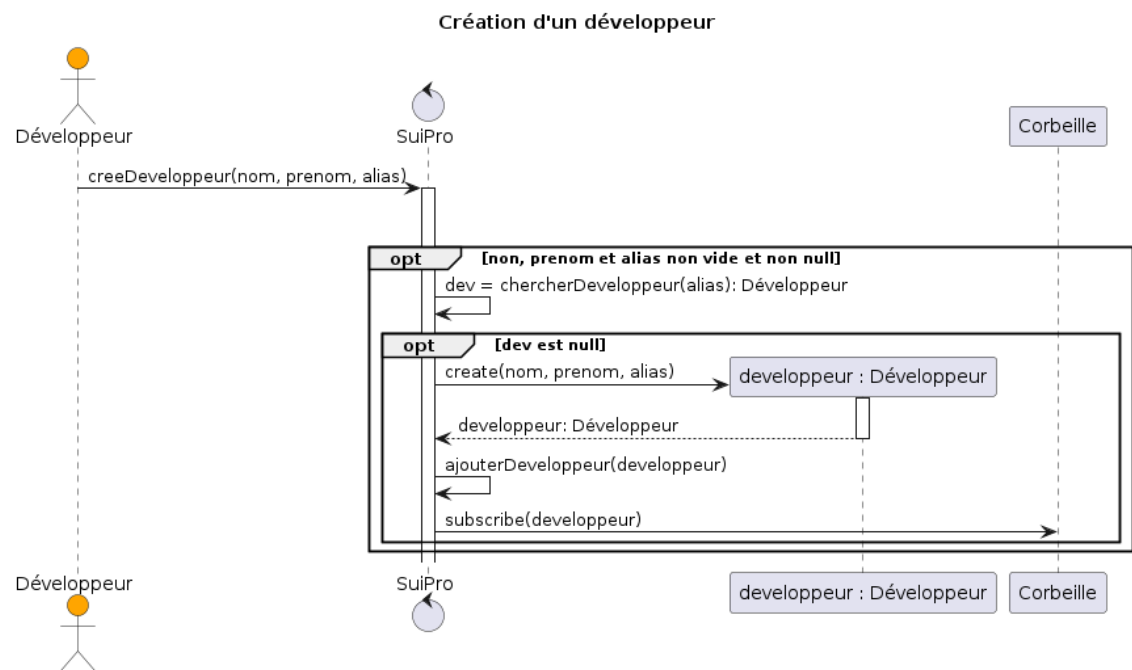


Fig.11 - Diagramme de séquence de création d'un développeur

Créer une activité

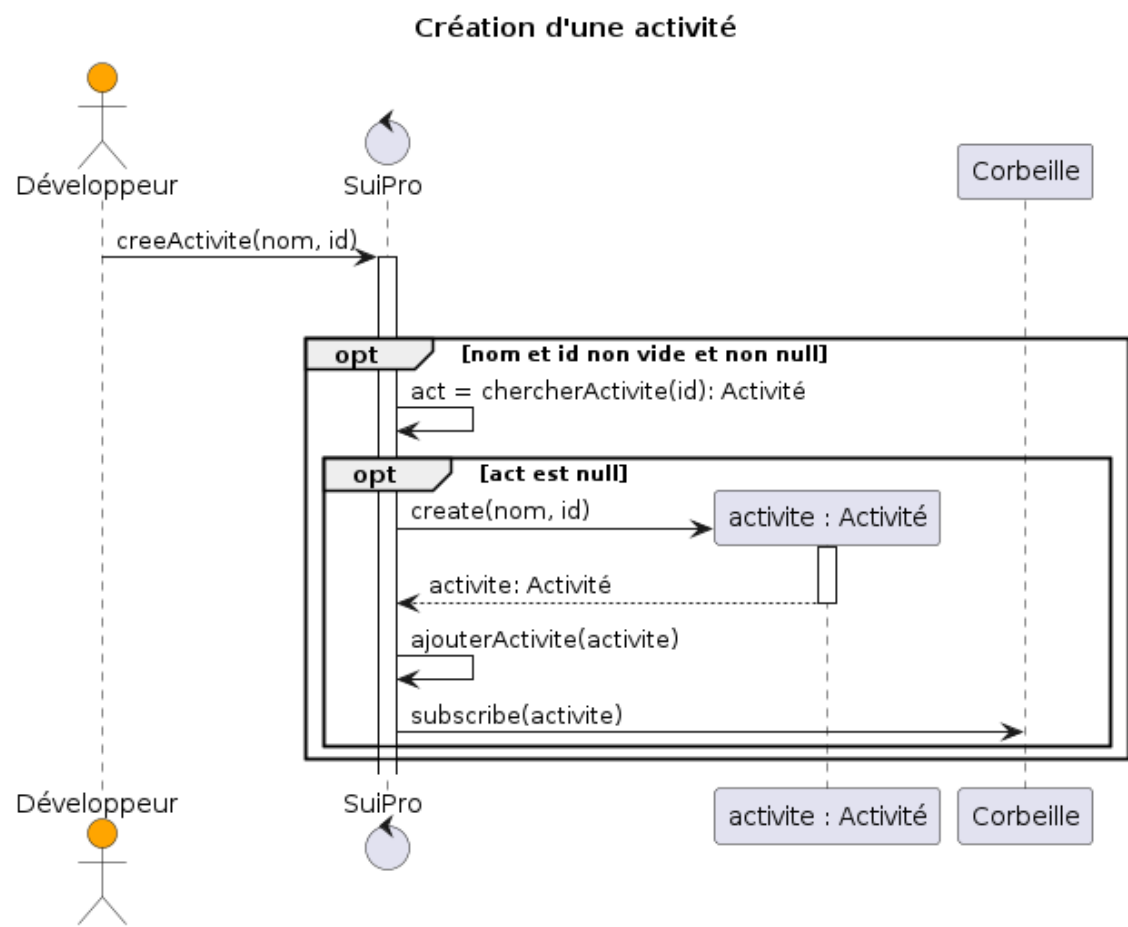


Fig.12 - Diagramme de séquence de création d'une activité

Créer une tâche

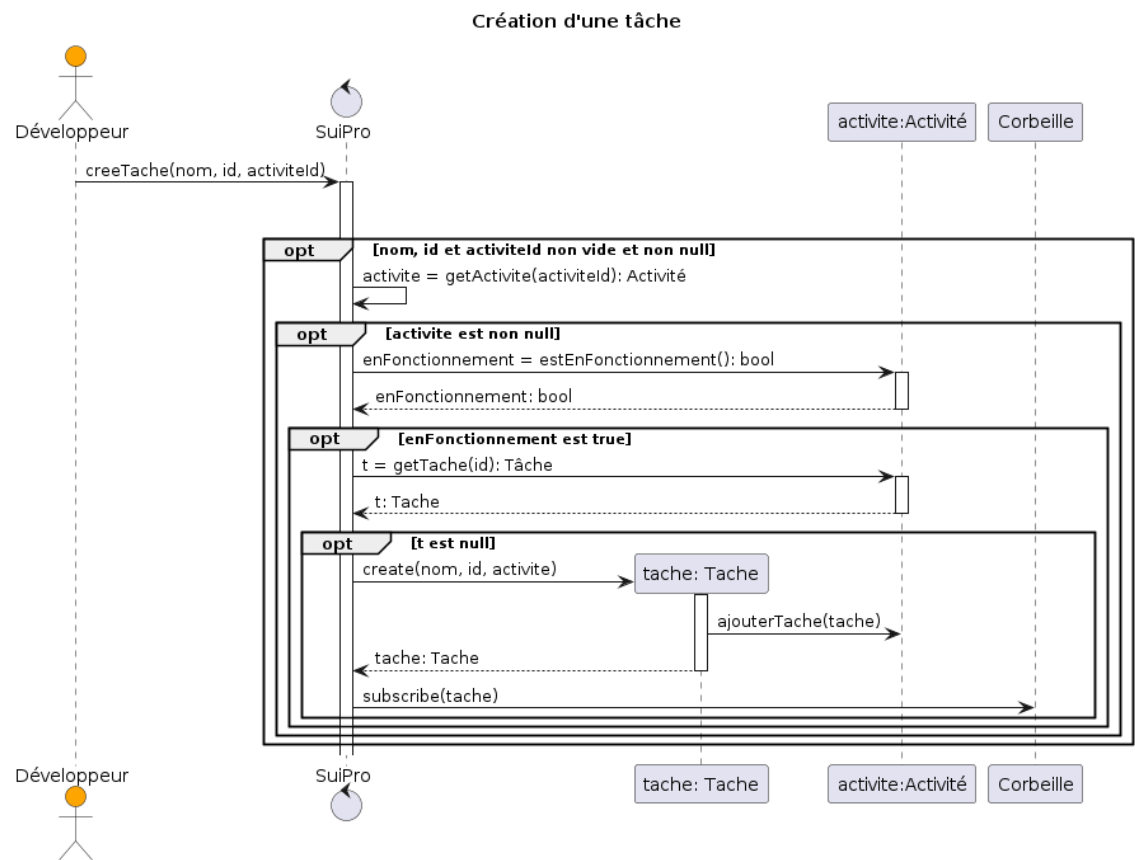


Fig.13 - Diagramme de séquence de création d'une tâche

Mettre un développeur à la corbeille

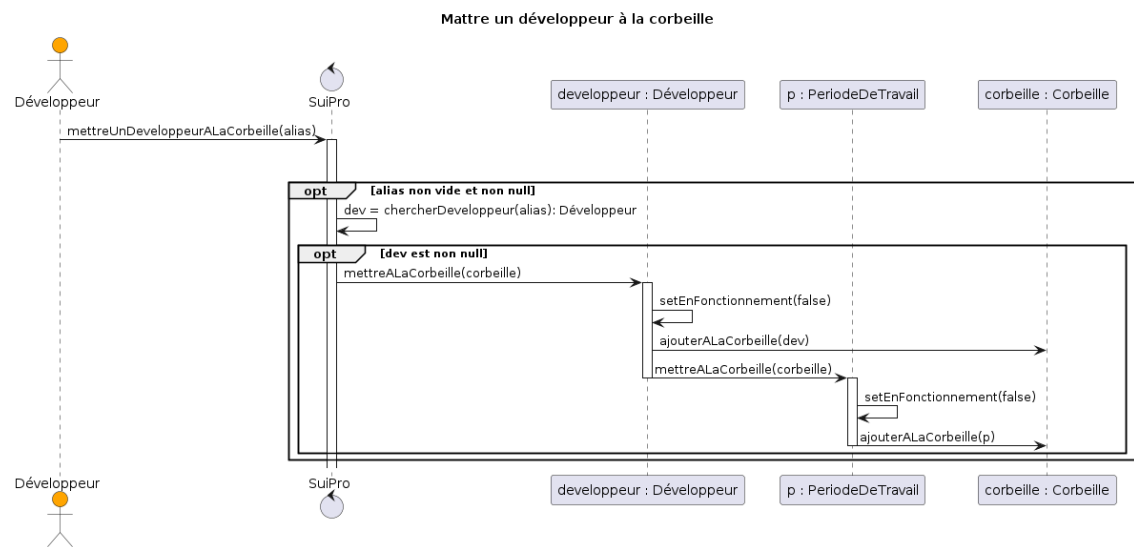


Fig.14 - Diagramme de séquence de mise à la corbeille d'un développeur

Mettre une tâche à la corbeille

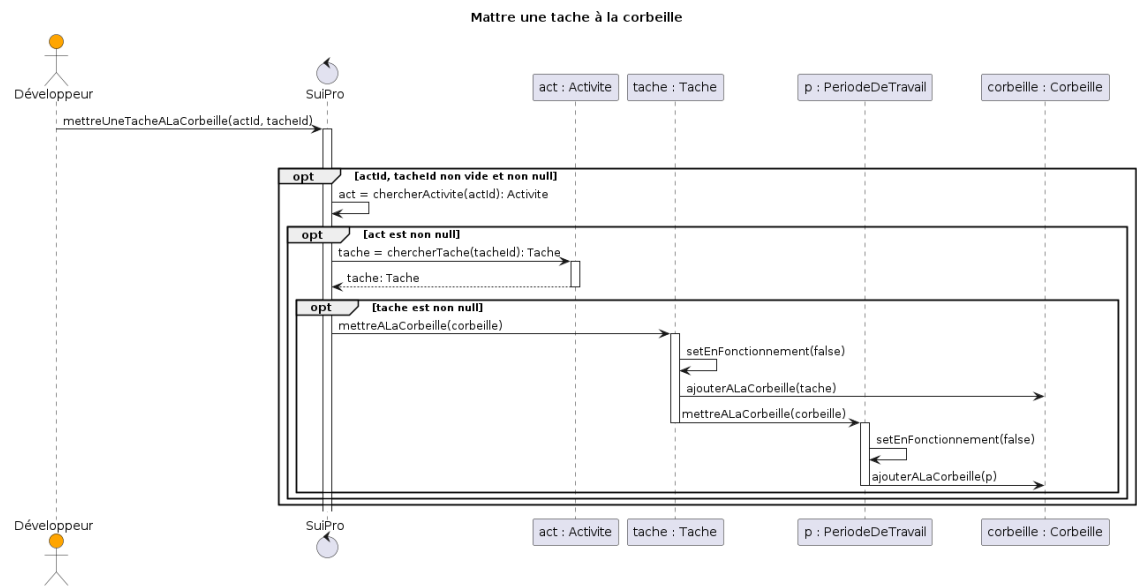


Fig.15 - Diagramme de séquence de mise à la corbeille d'une tâche

4 Diagrammes de machine à états et invariants

4.1 Diagramme des machines à états des développeurs

Diagramme d'état d'un développeur

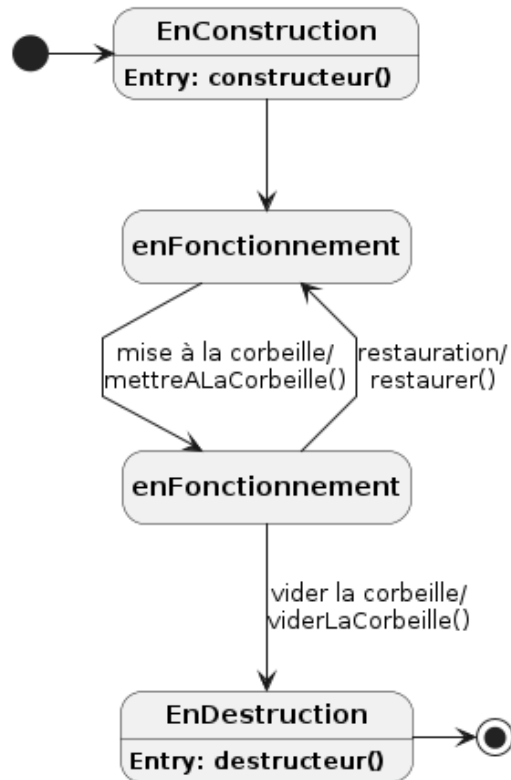


Fig.16 - Diagramme des machines à états des développeurs

Invariant de la classe `Développeur` :

- `alias` ne peut pas être `null` ou vide
- `nom` ne peut pas être `null` ou vide
- `prenom` ne peut pas être `null` ou vide
- `periodesDeTravail` ne peut pas être `null`

4.2 Diagramme des machines à états des tâches

Diagramme d'état d'une tâche

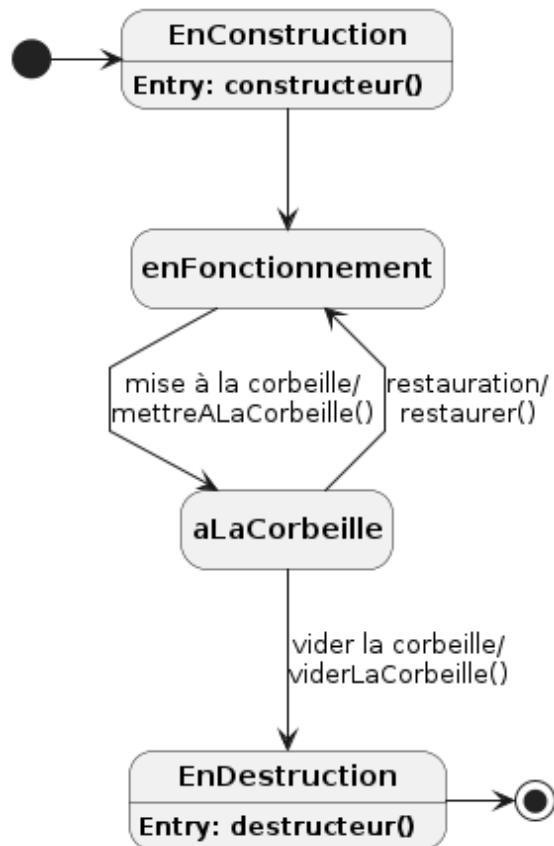


Fig.17 - Diagramme des machines à états des tâches

Invariant de la classe `Tâche` :

- `nom` ne peut pas être `null` ou vide
- `id` ne peut pas être `null` ou vide
- `activite` ne peut pas être `null`
- `periodeDeTravail` ne peut pas être `null`
- `labels` ne peut pas être `null`

5 Fiche des classes

5.1 Classe Développeur

Developpeur
Attributs par association
- periodesDeTravail: List<PeriodeDeTravail>
Attributs modifiables
- alias: String
- nom: String
- prenom: String
Attributs par héritage de ElementJetable
- enFonctionnement: boolean
Constructeur
+ Developpeur(alias: String, nom: String, prenom: String)
Méthodes
+ ajouterPeriodeDeTravail(periodeDeTravail: PeriodeDeTravail): void
+ invariant(): boolean
+ calculerTempsTravail(): double
Getters
+ getAlias(): String
+ getNom(): String
+ getPrenom(): String
+ getPeriodesDeTravail(): Collection<PeriodeDeTravail>
Redéfinitions de Object
+ hashCode(): int
+ equals(obj: Object): boolean
+ toString(): String
Spécification de ElementJetable
+ specificMettreALaCorbeille(corbeille: Corbeille): void
+ specificRestaurer(corbeille: Corbeille): void
Redéfinitions de Subscriber
+ onNext(elementJetable: ElementJetable): void

5.2 Classe Tâche

Tache
Attributs par association
- periodesDeTravail: List<PeriodeDeTravail>
- activite: Activite
- labels: List<Label>
Attributs modifiables
- nom: String
- id: String
Attributs par héritage de ElementJetable
- enFonctionnement: boolean
Constructeur
+ Tache(nom: String, id: String, activite: Activite)
Méthodes
+ ajouterPeriodeDeTravail(periodeDeTravail: PeriodeDeTravail): void
+ invariant(): boolean
+ calculerTempsTravail(): double
+ ajouterLabel(label: Label): void
Getters
+ getNom(): String
+ getId(): String
+ getPeriodesDeTravail(): List<PeriodeDeTravail>
+ getActivite(): Activite
+ getLabels(): List<Label>
Redéfinitions de Object
+ hashCode(): int
+ equals(obj: Object): boolean
+ toString(): String
Spécification de ElementJetable
+ specificMettreALaCorbeille(corbeille: Corbeille): void
+ specificRestaurer(corbeille: Corbeille): void
Redéfinitions de Subscriber
+ onNext(elementJetable: ElementJetable): void

6 Préparation des tests unitaires

6.1 Classe Développeur

Méthode `Developpeur(String alias, String nom, String prenom)` de la classe `Developpeur`

Numéro de test	1	2	3	4
alias non null et non vide	F	T	T	T
nom non null et non vide		F	T	T
prénom non null et non vide			F	T
alias' = alias				T
nom' = nom				T
prenom' = prenom				T
invariant()				T
Lève une exception	OUI	OUI	OUI	NON
Objet créé	NON	NON	NON	OUI
Nombre de jeu de tests	2	2	2	1

Méthode `mettreALaCorbeille(Corbeille corbeille)` de la classe `Developpeur`

Numéro de test	1	2
corbeille non null	F	T
Lève une exception	OUI	NON
Objet mis à la corbeille	NON	OUI
Période de travail mise à la corbeille	NON	OUI
Nombre de jeu de tests	1	1

Méthode `ajouterPeriodeDeTravail(PeriodeDeTravail periodeDeTravail)` de la classe `Developpeur`

Numéro de test	1	2	3	4	5	6
periodeDeTravail non null	F	T	T	T	T	T
periodeDeTravail enFonctionnement		F	T	T	T	T
periodeDeTravail associée au même développeur			F	T	T	T
chevauchement avec une autre période				F	T	T
developpeur enFonctionnement					F	T
invariant()						T
Lève une exception	OUI	OUI	OUI	OUI	OUI	NON
periodeDeTravail ajoutée	NON	NON	NON	NON	NON	OUI
Nombre de jeu de tests	1	1	1	1	1	1

Pour information, on ne vérifie pas la présence dans la liste des périodes de travail du développeur car cela sera vérifié avec le chevauchement.

6.2 Classe Activite

Méthode `Activite(String nom, String id)` de la classe `Activite`

Numéro de test	1	2	3
nom non null et non vide	F	T	T
id non null et non vide		F	T
nom' = nom			T
id' = id			T
invariant()			T
Lève une exception	OUI	OUI	NON
Objet créé	NON	NON	OUI
Nombre de jeu de tests	2	2	1

Méthode `mettreALaCorbeille(Corbeille corbeille)` de la classe `Activite`

Numéro de test	1	2
corbeille non null	F	T
Lève une exception	OUI	NON
Objet mis à la corbeille	NON	OUI
Taches mise à la corbeille	NON	OUI
Nombre de jeu de tests	1	1

Méthode `ajouterTache(Tache tache)` de la classe `Activite`

Numéro de test	1	2	3	4	5	6
tache non null	F	T	T	T	T	T
tache déjà présente		F	T	T	T	T
tache enFonctionnement			F	T	T	T
tache associée à une autre activité				F	T	T
activité enFonctionnement					F	T
invariant()						T
Lève une exception	OUI	OUI	OUI	OUI	OUI	NON
tache ajoutée	NON	NON	NON	NON	NON	OUI
Nombre de jeu de tests	1	1	1	1	1	1

6.3 Classe Tâche

Méthode `Tache(String nom, String id, Activite activite)` de la classe `Tache`

Numéro de test	1	2	3	4	6
nom non null et non vide	F	T	T	T	T
id non null et non vide		F	T	T	T
activite non null			F	T	T
activite enFonctionnement				F	T
nom' = nom					T
id' = id					T
activite' = activite					T
invariant()					T
Lève une exception	OUI	OUI	OUI	OUI	NON
Objet créé	NON	NON	NON	NON	OUI
Activité modifiée	NON	NON	NON	NON	OUI
Nombre de jeu de tests	2	2	1	1	1

Méthode `mettreALaCorbeille(Corbeille corbeille)` de la classe `Tache`

Numéro de test	1	2
corbeille non null	F	T
Lève une exception	OUI	NON
Objet mis à la corbeille	NON	OUI
Période de travail mise à la corbeille	NON	OUI
Nombre de jeu de tests	1	1

Méthode `ajouterPeriodeDeTravail(PeriodeDeTravail periodeDeTravail)` de la classe `Tache`

Numéro de test	1	2	3	4	5	6
periodeDeTravail non null	F	T	T	T	T	T
periodeDeTravail déjà présente		F	T	T	T	T
periodeDeTravail enFonctionnement			F	T	T	T
periodeDeTravail associée à la même tâche				F	T	T
tache enFonctionnement					F	T
invariant()						T
Lève une exception	OUI	OUI	OUI	OUI	OUI	NON
periodeDeTravail ajoutée	NON	NON	NON	NON	NON	OUI
Nombre de jeu de tests	1	1	1	1	1	1

Méthode `restaurer(Corbeille corbeille)` de la classe `Tache`

Numéro de test	1	2	3
corbeille non null	F	F	T
activité enFonctionnement		F	T
Lève une exception	OUI	NON	NON
Objet restauré de la corbeille	NON	NON	OUI
Période de travail restauré de la corbeille	NON	NON	OUI
Nombre de jeu de tests	1	1	1

6.4 Classe PeriodeDeTravail

Méthode PeriodeDeTravail(Instant debut, Instant fin, Tache tache, Developpeur developpeur)
de la classe PeriodeDeTravail

Numéro de test	1	2	3	4	5	6	7	8	9	10
debut non null	F	T	T	T	T	T	T	T	T	T
fin non null		F	T	T	T	T	T	T	T	T
tache non null			F	T	T	T	T	T	T	T
tache enFonctionnement				F	T	T	T	T	T	T
developpeur non null					F	T	T	T	T	T
developpeur enFonctionnement						F	T	T	T	T
intervalle correct							F	T	T	T
developpeur correct								F	T	T
tache correcte									F	T
intervalle' = intervalle										T
tache' = tache										T
developpeur' = developpeur										T
invariant()										T
Lève une exception	OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI	NON
Objet créé	NON	NON	NON	NON	NON	NON	NON	NON	NON	OUI
Méthode ajouterPeriodeDeTravail() de tache appelée	NON	NON	NON	NON	NON	NON	NON	NON	NON	OUI
Méthode ajouterPeriodeDeTravail() de developpeur appelée	NON	NON	NON	NON	NON	NON	NON	NON	NON	OUI
Nombre de jeu de tests	1	1	1	1	1	1	1	1	1	1

Méthode mettreALaCorbeille(Corbeille corbeille) de la classe PeriodeDeTravail

Numéro de test	1	2
corbeille non null	F	T
Lève une exception	OUI	NON
Objet mis à la corbeille	NON	OUI
Nombre de jeu de tests	1	1

6.5 Classe Corbeille

Méthode `ajouterALaCorbeille(ElementJetable elementJetable)` de la classe `Corbeille`

Numéro de test	1	2
elementJetable non null	F	T
Lève une exception	OUI	NON
Objet ajouté à la corbeille	NON	OUI
Nombre de jeu de tests	1	1

Méthode `supprimerDeLaCorbeille(ElementJetable elementJetable)` de la classe `Corbeille`

Numéro de test	1	2
elementJetable non null	F	T
Lève une exception	OUI	NON
Objet supprimé de la corbeille	NON	OUI
Nombre de jeu de tests	1	1

Méthode `getElementsJetable(Class<T> type)` de la classe `Corbeille`

Numéro de test	1	2
type non null	F	T
Lève une exception	OUI	NON
Retourne la liste	NON	OUI
Nombre de jeu de tests	1	4