

Antoine de ROQUEMAUREL  
Mathieu SOUM  
Geoffroy SUBIAS  
Marie-Ly TANG  
*Groupe B*

Pour Monsieur Max Chevalier (Resonsable projet)  
Madame Caroline Kross (Tutrice)  
Monsieur Thierry Millan (Client)

# Les plans et les cahiers de tests

---

Bibliothèque d'objets graphiques UML

# Table des matières

<b>1</b>	<b>Les plans de tests</b>	<b>3</b>
1.1	Les plans de non régression . . . . .	3
1.2	Les tests de validation . . . . .	3
1.2.1	Jeux d'essais . . . . .	3
1.2.1.1	Dessiner un diagramme complet . . . . .	3
1.2.1.2	Respecter les contraintes d'ajout d'éléments . . . . .	3
	Diagramme de cas d'utilisation . . . . .	3
	Diagramme de classe . . . . .	3
	Diagramme de séquence . . . . .	3
1.2.1.3	Respecter les contraintes de liaisons d'éléments . . . . .	4
	Diagramme de classe . . . . .	4
	Diagramme de cas d'utilisation . . . . .	4
	Diagramme de séquence . . . . .	4
1.2.2	Résultats obtenus . . . . .	4
1.3	Les tests d'intégration . . . . .	4
1.3.1	Les composants à tester . . . . .	4
1.3.2	Les jeux d'essais . . . . .	4
1.3.3	Les résultats attendus . . . . .	4
1.4	Les tests unitaires . . . . .	4
1.5	Le test de charges . . . . .	5
1.5.1	Le jeux d'essai . . . . .	5
1.5.2	Les résultats attendus . . . . .	5
<b>2</b>	<b>Les cahiers de tests</b>	<b>6</b>
2.1	Les tests de validation . . . . .	6
2.2	Les tests d'intégration . . . . .	6
2.3	Les tests unitaires . . . . .	6
.1	Le test de charge . . . . .	6
<b>A</b>	<b>Documentation des tests unitaires</b>	<b>7</b>

# Chapitre 1

## Les plans de tests

### 1.1 Les plans de non régression

Notre projet est de développer une bibliothèque UML<sup>1</sup>, aucun système n'existant au départ, il est donc inutile de faire un test de non régression étant donné que nous ne continuons pas un système, il n'y aura pas de régression.

### 1.2 Les tests de validation

Nous avons développé un démonstrateur qui utilise notre bibliothèque et qui fait office de test de validation. Il nous permet de vérifier que toutes les attentes du client sont respectées. Avec ce démonstrateur, nous pouvons montrer au client les différentes possibilités de la bibliothèque, ainsi qu'un exemple d'utilisation.

#### 1.2.1 Jeux d'essais

##### 1.2.1.1 Dessiner un diagramme complet

Nous avons effectués plusieurs jeux d'essais, le but est de pouvoir dessiner un pour chacun des types de diagrammes demandés par le client, un diagramme utilisant tous les composants de la norme UML 2.0 et respectant cette même norme, ceci pour chacun des diagrammes demandés par notre client.

- Dessiner un diagramme de classe
- Dessiner un diagramme de séquence
- Dessiner un diagramme de cas d'utilisation

##### 1.2.1.2 Respecter les contraintes d'ajout d'éléments

Pour chacun des diagrammes, les contraintes doivent être respectées, ainsi en fonction du type de diagramme, certains éléments ne doivent pas pouvoir être insérés. Voici la liste des éléments autorisés en fonction du type de diagramme.

#### Diagramme de cas d'utilisation

- Cas d'utilisation
- Acteur actif

#### Diagramme de classe

- Classe
- Interface

#### Diagramme de séquence

- Traitement
- Acteur actif
- Acteur passif
- Ligne de vie

---

1. Unified Modeling Language

### 1.2.1.3 Respecter les contraintes de liaisons d'éléments

Nous devons également respecter les contraintes de liaison entre les différents éléments. Ainsi voici la liste des liaisons possible.

#### Diagramme de classe

- Entre deux classes
  - Agrégation
  - Composition
  - Association
  - Spécialisation
- Entre une classe et une interface
  - Spécialisation
  - Dépendance fonctionnelle

#### Diagramme de cas d'utilisation

- Entre un acteur un cas d'utilisation
  - Association
- Entre deux cas d'utilisations
  - Dépendance fonctionnelle

#### Diagramme de séquence

- Entre deux traitements
  - Dépendance fonctionnelle
  - Déclenchements

## 1.2.2 Résultats obtenus

Nous vérifierons que les diagrammes sont bien réalisés, qu'il n'y a aucune erreur, et qu'ils correspondent aux besoins du client.

Également, nous vérifierons que nous ne pouvons pas relier ou ajouter des éléments non autorisés par la norme UML 2.0.

## 1.3 Les tests d'intégration

Nous avons développé un démonstrateur qui utilise notre bibliothèque et qui fait office de test d'intégration démontrant que notre système est utilisable en tant que composant par un logiciel extérieur. Il nous permet également de montrer au client les possibilités de notre bibliothèque.

### 1.3.1 Les composants à tester

Notre bibliothèque d'objet graphique en tant que composant de notre démonstrateur.

### 1.3.2 Les jeux d'essais

Dessiner un diagramme de classe, un diagramme de séquence ainsi qu'un diagramme de cas d'utilisation chacun utilisant tous les objets graphiques possibles.

Une fois ces diagrammes dessinés, nous testerons la suppression<sup>2</sup>, l'ajout de méthodes et d'attributs dans une classe, ainsi que la modification de données.

### 1.3.3 Les résultats attendus

Représentation graphique des différents diagrammes respectant les normes UML 2.0 avec possibilité d'ajout ou de suppression de données.

## 1.4 Les tests unitaires

Les tests unitaires servent à tester chacune des méthodes de notre bibliothèque. Nous avons utiliser JUnit pour tester toutes les méthodes de notre bibliothèque. Les tests sont validés

---

2. La suppression d'un élément devant supprimer les liens de l'élément supprimé

si la bibliothèque JUnit nous indique que tous les tests passent et ne posent pas de problème. La description des tests unitaires est disponible sous format Javadoc à l'annexe A page 7. Elle est également disponible sous format HTML à l'adresse suivante :

▷ <http://documentation.joohoo.fr/libUML/junitTests/index.html>

## 1.5 Le test de charges

### 1.5.1 Le jeux d'essai

Le test de charge permet de savoir si la création d'un grand nombre d'élément graphique ne posera pas de problème. Ainsi, nous allons créer les éléments suivants dans un diagramme sans contraintes.

- 20 Classes
- 20 Cas d'utilisation
- 20 Traitements
- 20 Acteurs actifs
- 20 Acteurs passifs
- 20 interface
- 60 liens

Une fois ces éléments suivants créer, nous supprimerons 40 éléments, qui devront supprimer les liens reliés aux éléments.

### 1.5.2 Les résultats attendus

Aucune exception, les éléments ont été bien créer et certains éléments ont été supprimés avec les liens.

# Chapitre 2

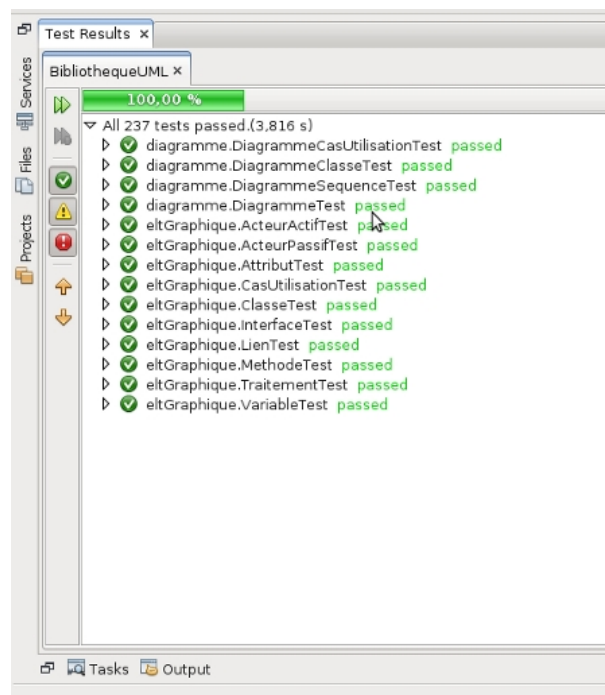
## Les cahiers de tests

### 2.1 Les tests de validation

### 2.2 Les tests d'intégration

### 2.3 Les tests unitaires

Après lancement des tests unitaires, ils sont tous passés. Nous pouvons donc en conclure que les différents modules de la bibliothèque UML fonctionnent.



### 2.4 Le test de charge

## **Annexe A**

### **Documentation des tests unitaires**

<b>diagramme</b>
<b>Class DiagrammeCasUtilisationTest</b>
java.lang.Object └─ diagramme.DiagrammeCasUtilisationTest

```
public class DiagrammeCasUtilisationTest
extends java.lang.Object
```

Cas de test JUnit regroupant les tests unitaires de chaque méthode de la classe DiagrammeCasUtilisationTest

**See Also:**  
DiagrammeCasUtilisation

<b>Constructor Summary</b>
<a href="#">DiagrammeCasUtilisationTest()</a>

<b>Method Summary</b>
<div>void <a href="#">setUp()</a> Initialisation des champ avant chaque test</div> <div>void <a href="#">tearDown()</a> Suppression de ces champs après chaque test</div> <div>void <a href="#">testEltAutoriseActeurActif()</a> Test unitaire qui vérifie que l'élément acteur actif est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode eltAutorise() retourne vrai</div> <div>void <a href="#">testEltAutoriseActeurPassif()</a> Test unitaire qui vérifie que l'élément acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode eltAutorise() retourne faux</div> <div>void <a href="#">testEltAutoriseCasUtilisation()</a> Test unitaire qui vérifie que l'élément cas d'utilisation est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode eltAutorise() retourne vrai</div>

void <a href="#">testEltAutoriseClasse()</a> Test unitaire qui vérifie que l'élément classe n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode eltAutorise() retourne faux
void <a href="#">testEltAutoriseInterface()</a> Test unitaire qui vérifie que l'élément Interface n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode eltAutorise() retourne faux
void <a href="#">testEltAutoriseLien()</a> Test unitaire qui vérifie que l'élément lien est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode eltAutorise() retourne vrai
void <a href="#">testEltAutoriseTraitement()</a> Test unitaire qui vérifie que l'élément traitement n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode eltAutorise() retourne faux
void <a href="#">testLienAutoriseAgregation()</a> Test unitaire qui vérifie que le lien d'agrégation entre n'importe quel élément graphique n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseAssociationActeurActifActeurActif()</a> Test unitaire qui vérifie que le lien d'association entre un acteur actif et un acteur actif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseAssociationActeurActifActeurPassif()</a> Test unitaire qui vérifie que le lien d'association entre un acteur actif et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseAssociationActeurActifCasUtilisation()</a> Test unitaire qui vérifie que le lien d'association entre un acteur actif et un cas d'utilisation est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne vrai
void <a href="#">testLienAutoriseAssociationActeurPassifActeurActif()</a> Test unitaire qui vérifie que le lien d'association entre un acteur passif et un acteur actif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseAssociationActeurPassifActeurPassif()</a> Test unitaire qui vérifie que le lien d'association entre un acteur passif et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux

void <a href="#">testLienAutoriseAssociationActeurPassifCasUtilisation()</a> Test unitaire qui vérifie que le lien d'association entre un acteur passif et un cas d'utilisation n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseAssociationCasUtilisationActeurActif()</a> Test unitaire qui vérifie que le lien d'association entre un cas d'utilisation et un acteur actif est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne vrai
void <a href="#">testLienAutoriseAssociationCasUtilisationActeurPassif()</a> Test unitaire qui vérifie que le lien d'association entre un cas d'utilisation et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseAssociationCasUtilisationCasUtilisation()</a> Test unitaire qui vérifie que le lien d'association entre un cas d'utilisation et un cas d'utilisation est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne vrai
void <a href="#">testLienAutoriseClasseAssociation()</a> Test unitaire qui vérifie que le lien de classe-association entre n'importe quel élément graphique n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseComposition()</a> Test unitaire qui vérifie que le lien de composition entre n'importe quel élément graphique n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseDependanceActeurActifActeurActif()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un acteur actif et un acteur actif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseDependanceActeurActifActeurPassif()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un acteur passif et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseDependanceActeurActifCasUtilisation()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un acteur actif et un cas d'utilisation n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux

void <a href="#">testLienAutoriseDependanceActeurPassifActeurActif()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un acteur passif et un acteur actif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseDependanceActeurPassifCasUtilisation()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un acteur passif et un cas d'utilisation n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseDependanceCasUtilisationActeurActif()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un cas d'utilisation et un acteur actif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseDependanceCasUtilisationActeurPassif()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un cas d'utilisation et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseDependanceCasUtilisationCasUtilisation()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un cas d'utilisation et un cas d'utilisation est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne vrai
void <a href="#">testLienAutoriseFleche()</a> Test unitaire qui vérifie que le lien de fleche entre n'importe quel élément graphique n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseGeneralisationActeurActifActeurActif()</a> Test unitaire qui vérifie que le lien de généralisation entre un acteur actif et un acteur actif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseGeneralisationActeurActifActeurPassif()</a> Test unitaire qui vérifie que le lien de généralisation entre un acteur actif et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void <a href="#">testLienAutoriseGeneralisationActeurActifCasUtilisation()</a> Test unitaire qui vérifie que le lien de généralisation entre un acteur actif et un cas d'utilisation n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux



	faux
void	<a href="#">testLienAutoriseGeneralisationActeurPassifActeurActif()</a> Test unitaire qui vérifie que le lien de généralisation entre un acteur passif et un acteur actif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseGeneralisationActeurPassifActeurPassif()</a> Test unitaire qui vérifie que le lien de généralisation entre un acteur passif et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseGeneralisationActeurPassifCasUtilisation()</a> Test unitaire qui vérifie que le lien de généralisation entre un acteur passif et un cas d'utilisation n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseGeneralisationCasUtilisationActeurActif()</a> Test unitaire qui vérifie que le lien de généralisation entre un cas d'utilisation et un acteur actif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseGeneralisationCasUtilisationActeurPassif()</a> Test unitaire qui vérifie que le lien de généralisation entre un cas d'utilisation et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseGeneralisationCasUtilisationCasUtilisation()</a> Test unitaire qui vérifie que le lien de généralisation entre un cas d'utilisation et un cas d'utilisation est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne vrai
void	<a href="#">testLienAutoriseSpecialisation()</a> Test unitaire qui vérifie que le lien de spécialisation entre n'importe quel élément graphique n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

DiagrammeCasUtilisationTest

testLienAutoriseAssociationActeurActifActeurActif

public void <a href="#">testLienAutoriseAssociationActeurActifActeurActif()</a>
Test unitaire qui vérifie que le lien d'association entre un acteur actif et un acteur actif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux

testLienAutoriseAssociationActeurPassifActeurPassif

public void <a href="#">testLienAutoriseAssociationActeurPassifActeurPassif()</a>
Test unitaire qui vérifie que le lien d'association entre un acteur passif et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux

testLienAutoriseAssociationActeurPassifCasUtilisation

public void <a href="#">testLienAutoriseAssociationActeurPassifCasUtilisation()</a>
Test unitaire qui vérifie que le lien d'association entre un acteur passif et un cas d'utilisation n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux

testLienAutoriseAssociationCasUtilisationActeurPassif

public void <a href="#">testLienAutoriseAssociationCasUtilisationActeurPassif()</a>
Test unitaire qui vérifie que le lien d'association entre un cas d'utilisation et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux

testLienAutoriseAssociationActeurActifActeurPassif

public void <a href="#">testLienAutoriseAssociationActeurActifActeurPassif()</a>
Test unitaire qui vérifie que le lien d'association entre un acteur actif et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux

public [DiagrammeCasUtilisationTest\(\)](#)

Method Detail

setUp

public void <a href="#">setUp()</a>
Initialisation des champ avant chaque test

tearDown

public void <a href="#">tearDown()</a>
Suppression de ces champs après chaque test

testLienAutoriseAssociationActeurActifCasUtilisation

public void <a href="#">testLienAutoriseAssociationActeurActifCasUtilisation()</a>
Test unitaire qui vérifie que le lien d'association entre un acteur actif et un cas d'utilisation est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne vrai

testLienAutoriseAssociationCasUtilisationActeurActif

public void <a href="#">testLienAutoriseAssociationCasUtilisationActeurActif()</a>
Test unitaire qui vérifie que le lien d'association entre un cas d'utilisation et un cas d'utilisation est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne vrai

testLienAutoriseAssociationCasUtilisationCasUtilisation

public void <a href="#">testLienAutoriseAssociationCasUtilisationCasUtilisation()</a>
Test unitaire qui vérifie que le lien d'association entre un cas d'utilisation et un cas d'utilisation est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne vrai

testLienAutoriseAssociationActeurPassifActeurActif

public void <a href="#">testLienAutoriseAssociationActeurPassifActeurActif()</a>
Test unitaire qui vérifie que le lien d'association entre un acteur passif et un acteur actif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux

testLienAutoriseGeneralisationCasUtilisationCasUtilisation

public void <a href="#">testLienAutoriseGeneralisationCasUtilisationCasUtilisation()</a>
Test unitaire qui vérifie que le lien de généralisation entre un cas d'utilisation et un cas d'utilisation est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne vrai

testLienAutoriseGeneralisationActeurActifCasUtilisation

public void <a href="#">testLienAutoriseGeneralisationActeurActifCasUtilisation()</a>
Test unitaire qui vérifie que le lien de généralisation entre un acteur actif et un cas d'utilisation n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux

testLienAutoriseGeneralisationCasUtilisationActeurActif

public void <a href="#">testLienAutoriseGeneralisationCasUtilisationActeurActif()</a>
Test unitaire qui vérifie que le lien de généralisation entre un cas d'utilisation et un acteur actif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux

testLienAutoriseGeneralisationActeurPassifCasUtilisation

public void <a href="#">testLienAutoriseGeneralisationActeurPassifCasUtilisation()</a>
Test unitaire qui vérifie que le lien de généralisation entre un acteur passif et un cas d'utilisation n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux

**testLienAutoriseGeneralisationCasUtilisationActeurPassif**

```
public void testLienAutoriseGeneralisationCasUtilisationActeurPassif()

    Test unitaire qui vérifie que le lien de généralisation entre un cas d'utilisation et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseGeneralisationActeurActifActeurActif**

```
public void testLienAutoriseGeneralisationActeurActifActeurActif()

    Test unitaire qui vérifie que le lien de généralisation entre un acteur actif et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseGeneralisationActeurPassifActeurPassif**

```
public void testLienAutoriseGeneralisationActeurPassifActeurPassif()

    Test unitaire qui vérifie que le lien de généralisation entre un acteur passif et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseGeneralisationActeurActifActeurPassif**

```
public void testLienAutoriseGeneralisationActeurActifActeurPassif()

    Test unitaire qui vérifie que le lien de généralisation entre un acteur actif et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseGeneralisationActeurPassifActeurActif**

```
public void testLienAutoriseGeneralisationActeurPassifActeurActif()

    Test unitaire qui vérifie que le lien de généralisation entre un acteur passif et un acteur actif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseDependanceCasUtilisationActeurPassif**

```
public void testLienAutoriseDependanceCasUtilisationActeurPassif()

    Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un cas d'utilisation et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseDependanceActeurActifActeurPassif**

```
public void testLienAutoriseDependanceActeurActifActeurPassif()

    Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un acteur passif et un acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseDependanceActeurPassifActeurActif**

```
public void testLienAutoriseDependanceActeurPassifActeurActif()

    Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un acteur passif et un acteur actif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseAgregation**

```
public void testLienAutoriseAgregation()

    Test unitaire qui vérifie que le lien d'agrégation entre n'importe quel élément graphique n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseClasseAssociation**

```
public void testLienAutoriseClasseAssociation()

    Test unitaire qui vérifie que le lien de classe-association entre n'importe quel élément graphique n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseDependanceCasUtilisationCasUtilisation**

```
public void testLienAutoriseDependanceCasUtilisationCasUtilisation()

    Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un cas d'utilisation et un cas d'utilisation est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne vrai
```

---

**testLienAutoriseDependanceActeurActifActeurActif**

```
public void testLienAutoriseDependanceActeurActifActeurActif()

    Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un acteur actif et un acteur actif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseDependanceActeurActifCasUtilisation**

```
public void testLienAutoriseDependanceActeurActifCasUtilisation()

    Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un acteur actif et un cas d'utilisation n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseDependanceCasUtilisationActeurActif**

```
public void testLienAutoriseDependanceCasUtilisationActeurActif()

    Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un cas d'utilisation et un acteur actif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseDependanceActeurPassifCasUtilisation**

```
public void testLienAutoriseDependanceActeurPassifCasUtilisation()

    Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un acteur passif et un cas d'utilisation n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseComposition**

```
public void testLienAutoriseComposition()

    Test unitaire qui vérifie que le lien de composition entre n'importe quel élément graphique n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseSpecialisation**

```
public void testLienAutoriseSpecialisation()

    Test unitaire qui vérifie que le lien de spécialisation entre n'importe quel élément graphique n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseFleche**

```
public void testLienAutoriseFleche()

    Test unitaire qui vérifie que le lien de fleche entre n'importe quel élément graphique n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testEltAutoriseCasUtilisation**

```
public void testEltAutoriseCasUtilisation()

    Test unitaire qui vérifie que l'élément cas d'utilisation est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode eltAutorise() retourne vrai
```

---

**testEltAutoriseLien**

```
public void testEltAutoriseLien()

    Test unitaire qui vérifie que l'élément lien est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode eltAutorise() retourne vrai
```

---

**testEltAutoriseActeurActif**

public void **testEltAutoriseActeurActif()**

Test unitaire qui vérifie que l'élément acteur actif est autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode eltAutorise() retourne vrai

**testEltAutoriseActeurPassif**

public void **testEltAutoriseActeurPassif()**

Test unitaire qui vérifie que l'élément acteur passif n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode eltAutorise() retourne faux

**testEltAutoriseTraitement**

public void **testEltAutoriseTraitement()**

Test unitaire qui vérifie que l'élément traitement n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode eltAutorise() retourne faux

**testEltAutoriseClasse**

public void **testEltAutoriseClasse()**

Test unitaire qui vérifie que l'élément classe n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode eltAutorise() retourne faux

**testEltAutoriseInterface**

public void **testEltAutoriseInterface()**

Test unitaire qui vérifie que l'élément Interface n'est pas autorisé dans un diagramme de cas d'utilisation Vérifie que : - la méthode eltAutorise() retourne faux

void	<a href="#">testEltAutoriseClasse()</a> Test unitaire qui vérifie que l'élément classe est autorisé dans un diagramme de classe Vérifie que : - la méthode eltAutorise() retourne vrai
void	<a href="#">testEltAutoriseInterface()</a> Test unitaire qui vérifie que l'élément interface n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode eltAutorise() retourne faux
void	<a href="#">testEltAutoriseLien()</a> Test unitaire qui vérifie que l'élément lien est autorisé dans un diagramme de classe Vérifie que : - la méthode eltAutorise() retourne vrai
void	<a href="#">testEltAutoriseTraitement()</a> Test unitaire qui vérifie que l'élément traitement n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode eltAutorise() retourne faux
void	<a href="#">testLienAutoriseAgregationClasseClasse()</a> Test unitaire qui vérifie que le lien d'agrégation entre une classe et une classe n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseAgregationClasseLien()</a> Test unitaire qui vérifie que le lien d'agrégation entre une classe et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseAgregationLienClasse()</a> Test unitaire qui vérifie que le lien d'agrégation entre un lien et une classe n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseAgregationLienLien()</a> Test unitaire qui vérifie que le lien d'agrégation entre un lien et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseAssociationClasseClasse()</a> Test unitaire qui vérifie que le lien d'association entre une classe et une classe est autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne vrai
void	<a href="#">testLienAutoriseAssociationClasseLien()</a> Test unitaire qui vérifie que le lien d'association entre une classe et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux

diagramme

**Class DiagrammeClasseTest**

java.lang.Object  
└─ **diagramme.DiagrammeClasseTest**

public class **DiagrammeClasseTest**  
extends java.lang.Object

Cas de test JUnit regroupant les tests unitaires de chaque méthode de la classe DiagrammeClasseTest

**See Also:**

DiagrammeClasse

**Constructor Summary**

[DiagrammeClasseTest\(\)](#)

**Method Summary**

void	<a href="#">setUp()</a> Initialisation des champ avant chaque test
void	<a href="#">tearDown()</a> Suppression de ces champs après chaque test
void	<a href="#">testEltAutoriseActeurActif()</a> Test unitaire qui vérifie que l'élément acteur actif n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode eltAutorise() retourne faux
void	<a href="#">testEltAutoriseActeurPassif()</a> Test unitaire qui vérifie que l'élément acteur passif n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode eltAutorise() retourne faux
void	<a href="#">testEltAutoriseCasUtilisation()</a> Test unitaire qui vérifie que l'élément cas d'utilisation n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode eltAutorise() retourne faux

void	<a href="#">testLienAutoriseAssociationLienClasse()</a> Test unitaire qui vérifie que le lien d'association entre un lien et une classe n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseAssociationLienLien()</a> Test unitaire qui vérifie que le lien d'association entre un lien et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseClasseAssociationLienLien()</a> Test unitaire qui vérifie que le lien de classe association entre un lien et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseClasseAssociationLienClasse()</a> Test unitaire qui vérifie que le lien de classe association entre une classe et un lien est autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne vrai
void	<a href="#">testLienAutoriseCompositionClasseClasse()</a> Test unitaire qui vérifie que le lien de composition entre une classe et une classe est autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne vrai
void	<a href="#">testLienAutoriseCompositionClasseLien()</a> Test unitaire qui vérifie que le lien de composition entre une classe et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseCompositionLienClasse()</a> Test unitaire qui vérifie que le lien de composition entre un lien et une classe n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseCompositionLienLien()</a> Test unitaire qui vérifie que le lien de composition entre un lien et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseDependanceClasseClasse()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre une classe et une classe est autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne vrai
void	<a href="#">testLienAutoriseDependanceClasseLien()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre une classe et un lien est autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne vrai

void	<a href="#">testLienAutoriseDependanceLienClasse()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un lien et une classe est autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne vrai
void	<a href="#">testLienAutoriseDependanceLienLien()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un lien et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseFleche()</a> Test unitaire qui vérifie que le lien de flèche entre n'importe quels éléments graphiques n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseGeneralisationClasseClasse()</a> Test unitaire qui vérifie que le lien de généralisation entre une classe et une classe est autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne vrai
void	<a href="#">testLienAutoriseGeneralisationClasseLien()</a> Test unitaire qui vérifie que le lien de généralisation entre une classe et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseGeneralisationLienClasse()</a> Test unitaire qui vérifie que le lien de généralisation entre un lien et une classe n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseGeneralisationLienLien()</a> Test unitaire qui vérifie que le lien de généralisation entre un lien et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux
void	<a href="#">testLienAutoriseSpecialisation()</a> Test unitaire qui vérifie que le lien de spécialisation entre n'importe quels éléments graphiques n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### testLienAutoriseCompositionLienLien

public void [testLienAutoriseCompositionLienLien\(\)](#)

Test unitaire qui vérifie que le lien de composition entre un lien et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseAssociationClasseClasse

public void [testLienAutoriseAssociationClasseClasse\(\)](#)

Test unitaire qui vérifie que le lien d'association entre une classe et une classe est autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne vrai

### testLienAutoriseClasseAssociationLienClasse

public void [testLienAutoriseClasseAssociationLienClasse\(\)](#)

Test unitaire qui vérifie que le lien de classe association entre une classe et un lien est autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne vrai

### testLienAutoriseClaseeAssociationLienLien

public void [testLienAutoriseClaseeAssociationLienLien\(\)](#)

Test unitaire qui vérifie que le lien de classe association entre un lien et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseAssociationClasseLien

public void [testLienAutoriseAssociationClasseLien\(\)](#)

Test unitaire qui vérifie que le lien d'association entre une classe et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux

## DiagrammeClasseTest

public [DiagrammeClasseTest\(\)](#)

## Method Detail

### setUp

public void [setUp\(\)](#)

Initialisation des champ avant chaque test

### tearDown

public void [tearDown\(\)](#)

Suppression de ces champs après chaque test

### testLienAutoriseCompositionClasseClasse

public void [testLienAutoriseCompositionClasseClasse\(\)](#)

Test unitaire qui vérifie que le lien de composition entre une classe et une classe est autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne vrai

### testLienAutoriseCompositionClasseLien

public void [testLienAutoriseCompositionClasseLien\(\)](#)

Test unitaire qui vérifie que le lien de composition entre une classe et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseCompositionLienClasse

public void [testLienAutoriseCompositionLienClasse\(\)](#)

Test unitaire qui vérifie que le lien de composition entre un lien et une classe n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseAssociationLienClasse

public void [testLienAutoriseAssociationLienClasse\(\)](#)

Test unitaire qui vérifie que le lien d'association entre un lien et une classe n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseAssociationLienLien

public void [testLienAutoriseAssociationLienLien\(\)](#)

Test unitaire qui vérifie que le lien d'association entre un lien et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseGeneralisationClasseClasse

public void [testLienAutoriseGeneralisationClasseClasse\(\)](#)

Test unitaire qui vérifie que le lien de généralisation entre une classe et une classe est autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne vrai

### testLienAutoriseGeneralisationClasseLien

public void [testLienAutoriseGeneralisationClasseLien\(\)](#)

Test unitaire qui vérifie que le lien de généralisation entre une classe et un lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseGeneralisationLienClasse

public void [testLienAutoriseGeneralisationLienClasse\(\)](#)

Test unitaire qui vérifie que le lien de généralisation entre un lien et une classe n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode lienAutorise() retourne faux

**testLienAutoriseGeneralisationLienLien**

```
public void testLienAutoriseGeneralisationLienLien()

    Test unitaire qui vérifie que le lien de généralisation entre un lien et un
    lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la
    méthode lienAutorise() retourne faux
```

---

**testLienAutoriseDependanceClasseClasse**

```
public void testLienAutoriseDependanceClasseClasse()

    Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre une
    classe et une classe est autorisé dans un diagramme de classe Vérifie que :
    - la méthode lienAutorise() retourne vrai
```

---

**testLienAutoriseDependanceClasseLien**

```
public void testLienAutoriseDependanceClasseLien()

    Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre une
    classe et un lien est autorisé dans un diagramme de classe Vérifie que : -
    la méthode lienAutorise() retourne vrai
```

---

**testLienAutoriseDependanceLienClasse**

```
public void testLienAutoriseDependanceLienClasse()

    Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un
    lien et une classe est autorisé dans un diagramme de classe Vérifie que : -
    la méthode lienAutorise() retourne vrai
```

---

**testLienAutoriseDependanceLienLien**

```
public void testLienAutoriseDependanceLienLien()

    Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un
    lien et un lien n'est pas autorisé dans un diagramme de classe Vérifie que
    : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseFleche**

```
public void testLienAutoriseFleche()

    Test unitaire qui vérifie que le lien de flèche entre n'importe quels
    éléments graphiques n'est pas autorisé dans un diagramme de classe
    Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testEltAutoriseClasse**

```
public void testEltAutoriseClasse()

    Test unitaire qui vérifie que l'élément classe est autorisé dans un
    diagramme de classe Vérifie que : - la méthode eltAutorise() retourne vrai
```

---

**testEltAutoriseLien**

```
public void testEltAutoriseLien()

    Test unitaire qui vérifie que l'élément lien est autorisé dans un diagramme
    de classe Vérifie que : - la méthode eltAutorise() retourne vrai
```

---

**testEltAutoriseCasUtilisation**

```
public void testEltAutoriseCasUtilisation()

    Test unitaire qui vérifie que l'élément cas d'utilisation n'est pas autorisé
    dans un diagramme de classe Vérifie que : - la méthode eltAutorise()
    retourne faux
```

---

**testEltAutoriseActeurActif**

```
public void testEltAutoriseActeurActif()

    Test unitaire qui vérifie que l'élément acteur actif n'est pas autorisé dans
    un diagramme de classe Vérifie que : - la méthode eltAutorise() retourne
    faux
```

---

**testEltAutoriseActeurPassif**

```
public void testEltAutoriseActeurPassif()
```

**testLienAutoriseAgregationClasseClasse**

```
public void testLienAutoriseAgregationClasseClasse()

    Test unitaire qui vérifie que le lien d'agrégation entre une classe et une
    classe n'est pas autorisé dans un diagramme de classe Vérifie que : - la
    méthode lienAutorise() retourne faux
```

---

**testLienAutoriseAgregationClasseLien**

```
public void testLienAutoriseAgregationClasseLien()

    Test unitaire qui vérifie que le lien d'agrégation entre une classe et un
    lien n'est pas autorisé dans un diagramme de classe Vérifie que : - la
    méthode lienAutorise() retourne faux
```

---

**testLienAutoriseAgregationLienClasse**

```
public void testLienAutoriseAgregationLienClasse()

    Test unitaire qui vérifie que le lien d'agrégation entre un lien et une
    classe n'est pas autorisé dans un diagramme de classe Vérifie que : - la
    méthode lienAutorise() retourne faux
```

---

**testLienAutoriseAgregationLienLien**

```
public void testLienAutoriseAgregationLienLien()

    Test unitaire qui vérifie que le lien d'agrégation entre un lien et un lien
    n'est pas autorisé dans un diagramme de classe Vérifie que : - la méthode
    lienAutorise() retourne faux
```

---

**testLienAutoriseSpecialisation**

```
public void testLienAutoriseSpecialisation()

    Test unitaire qui vérifie que le lien de spécialisation entre n'importe quels
    éléments graphiques n'est pas autorisé dans un diagramme de classe
    Vérifie que : - la méthode lienAutorise() retourne faux
```

---

```
Test unitaire qui vérifie que l'élément acteur passif n'est pas autorisé dans
un diagramme de classe Vérifie que : - la méthode eltAutorise() retourne
faux
```

---

**testEltAutoriseTraitement**

```
public void testEltAutoriseTraitement()

    Test unitaire qui vérifie que l'élément traitement n'est pas autorisé dans
    un diagramme de classe Vérifie que : - la méthode eltAutorise() retourne
    faux
```

---

**testEltAutoriseInterface**

```
public void testEltAutoriseInterface()

    Test unitaire qui vérifie que l'élément interface n'est pas autorisé dans un
    diagramme de classe Vérifie que : - la méthode eltAutorise() retourne faux
```

---

---

<b>diagramme</b>
<b>Class DiagrammeSequenceTest</b>
java.lang.Object └─ <b>diagramme.DiagrammeSequenceTest</b>

```
public class DiagrammeSequenceTest
extends java.lang.Object
```

Cas de test JUnit regroupant les tests unitaires de chaque méthode de la classe DiagrammeSequenceTest

**See Also:**  
DiagrammeSequence

<b>Constructor Summary</b>
<a href="#">DiagrammeSequenceTest()</a>

<b>Method Summary</b>
<div>void <a href="#">setUp()</a> Initialisation des champ avant chaque test</div> <div>void <a href="#">tearDown()</a> Suppression de ces champs après chaque test</div> <div>void <a href="#">testEltAutoriseActeurActif()</a> Test unitaire qui vérifie que l'élément acteur actif est autorisé dans un diagramme de séquence Vérifie que : - la méthode eltAutorise() retourne vrai</div> <div>void <a href="#">testEltAutoriseActeurPassif()</a> Test unitaire qui vérifie que l'élément acteur passif est autorisé dans un diagramme de séquence Vérifie que : - la méthode eltAutorise() retourne vrai</div> <div>void <a href="#">testEltAutoriseCasUtilisation()</a> Test unitaire qui vérifie que l'élément cas d'utilisation n'est pas autorisé dans un diagramme de séquence Vérifie que : - la méthode eltAutorise() retourne faux</div>

void <a href="#">testEltAutoriseClasse()</a> Test unitaire qui vérifie que l'élément classe n'est pas autorisé dans un diagramme de séquence Vérifie que : - la méthode eltAutorise() retourne faux
void <a href="#">testEltAutoriseInterface()</a> Test unitaire qui vérifie que l'élément interface n'est pas autorisé dans un diagramme de séquence Vérifie que : - la méthode eltAutorise() retourne faux
void <a href="#">testEltAutoriseLien()</a> Test unitaire qui vérifie que l'élément lien n'est pas autorisé dans un diagramme de séquence Vérifie que : - la méthode eltAutorise() retourne faux
void <a href="#">testEltAutoriseTraitement()</a> Test unitaire qui vérifie que l'élément traitement est autorisé dans un diagramme de séquence Vérifie que : - la méthode eltAutorise() retourne vrai
void <a href="#">testLienAutoriseAgregation()</a> Test unitaire qui vérifie que le lien d'agrégation entre n'importe quels éléments graphiques n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseAssociationActeurActifActeurActif()</a> Test unitaire qui vérifie que le lien d'association entre un acteur actif et un acteur actif n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseAssociationActeurActifActeurPassif()</a> Test unitaire qui vérifie que le lien d'association entre un acteur actif et un acteur passif n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseAssociationActeurActifTraitement()</a> Test unitaire qui vérifie que le lien d'association entre un acteur actif et un traitement n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseAssociationActeurPassifActeurActif()</a> Test unitaire qui vérifie que le lien d'association entre un acteur passif et un acteur actif n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseAssociationActeurPassifActeurPassif()</a> Test unitaire qui vérifie que le lien d'association entre un acteur passif et un acteur passif n'est pas autorisé dans un diagramme de séquence.

void <a href="#">testLienAutoriseAssociationActeurPassifTraitement()</a> Test unitaire qui vérifie que le lien d'association entre un acteur passif et un traitement n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseAssociationTraitementActeurActif()</a> Test unitaire qui vérifie que le lien d'association entre un traitement et un acteur actif n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseAssociationTraitementActeurPassif()</a> Test unitaire qui vérifie que le lien d'association entre un traitement et un acteur passif n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseAssociationTraitementTraitement()</a> Test unitaire qui vérifie que le lien d'association entre un traitement et un traitement est autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseClasseAssociation()</a> Test unitaire qui vérifie que le lien de classe-association entre n'importe quels éléments graphiques n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseComposition()</a> Test unitaire qui vérifie que le lien de composition entre n'importe quels éléments graphiques n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseDependanceActeurActifActeurActif()</a> Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un acteur actif et un acteur actif n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseDependanceActeurActifActeurPassif()</a> Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un acteur actif et un acteur passif n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseDependanceActeurActifTraitement()</a> Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un acteur actif et un traitement n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseDependanceActeurPassifActeurActif()</a> Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un acteur passif et un acteur actif n'est pas autorisé dans un diagramme de séquence.

void <a href="#">testLienAutoriseDependanceActeurPassifActeurPassif()</a> Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un acteur passif et un acteur passif n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseDependanceActeurPassifTraitement()</a> Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un acteur passif et un traitement n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseDependanceTraitementActeurActif()</a> Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un traitement et un acteur actif n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseDependanceTraitementActeurPassif()</a> Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un traitement et un acteur passif n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseDependanceTraitementTraitement()</a> Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un traitement et un traitement est autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseFlecheActeurActifActeurActif()</a> Test unitaire qui vérifie que le lien de flèche entre un acteur actif et un acteur actif n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseFlecheActeurActifActeurPassif()</a> Test unitaire qui vérifie que le lien de flèche entre un acteur actif et un acteur passif n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseFlecheActeurActifTraitement()</a> Test unitaire qui vérifie que le lien de flèche entre un acteur actif et un traitement n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseFlecheActeurPassifActeurActif()</a> Test unitaire qui vérifie que le lien de flèche entre un acteur passif et un acteur actif n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseFlecheActeurPassifActeurPassif()</a> Test unitaire qui vérifie que le lien de flèche entre un acteur passif et un acteur passif n'est pas autorisé dans un diagramme de séquence.
void <a href="#">testLienAutoriseFlecheActeurPassifTraitement()</a> Test unitaire qui vérifie que le lien de flèche entre un acteur passif et un traitement n'est pas autorisé dans un diagramme de séquence.

void	<a href="#">testLienAutoriseFlecheTraitementActeurActif()</a> Test unitaire qui vérifie que le lien de flèche entre un traitement et un acteur actif n'est pas autorisé dans un diagramme de séquence.
void	<a href="#">testLienAutoriseFlecheTraitementActeurPassif()</a> Test unitaire qui vérifie que le lien de flèche entre un traitement et un acteur passif n'est pas autorisé dans un diagramme de séquence.
void	<a href="#">testLienAutoriseFlecheTraitementTraitement()</a> Test unitaire qui vérifie que le lien de flèche entre un traitement et un traitement est autorisé dans un diagramme de séquence.
void	<a href="#">testLienAutoriseGeneralisation()</a> Test unitaire qui vérifie que le lien de généralisation entre n'importe quels éléments graphiques n'est pas autorisé dans un diagramme de séquence.
void	<a href="#">testLienAutoriseSpecialisation()</a> Test unitaire qui vérifie que le lien de spécialisation entre n'importe quels éléments graphiques n'est pas autorisé dans un diagramme de séquence.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### DiagrammeSequenceTest

```
public DiagrammeSequenceTest()
```

## Method Detail

### setUp

```
public void setUp()
```

Initialisation des champ avant chaque test

### tearDown

```
public void tearDown()
```

### testLienAutoriseAssociationActeurActifActeurActif

```
public void testLienAutoriseAssociationActeurActifActeurActif()
```

Test unitaire qui vérifie que le lien d'association entre un acteur actif et un acteur actif n'est pas autorisé dans un diagramme de séquence. Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseAssociationActeurPassifActeurPassif

```
public void testLienAutoriseAssociationActeurPassifActeurPassif()
```

Test unitaire qui vérifie que le lien d'association entre un acteur passif et un acteur passif n'est pas autorisé dans un diagramme de séquence. Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseAssociationActeurActifActeurPassif

```
public void testLienAutoriseAssociationActeurActifActeurPassif()
```

Test unitaire qui vérifie que le lien d'association entre un acteur actif et un traitement n'est pas autorisé dans un diagramme de séquence. Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseAssociationActeurPassifActeurActif

```
public void testLienAutoriseAssociationActeurPassifActeurActif()
```

Test unitaire qui vérifie que le lien d'association entre un acteur passif et un traitement n'est pas autorisé dans un diagramme de séquence. Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseDependanceTraitementTraitement

```
public void testLienAutoriseDependanceTraitementTraitement()
```

Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un traitement et un traitement est autorisé dans un diagramme de séquence. Vérifie que : - la méthode lienAutorise() retourne vrai

Suppression de ces champs après chaque test

### testLienAutoriseAssociationTraitementTraitement

```
public void testLienAutoriseAssociationTraitementTraitement()
```

Test unitaire qui vérifie que le lien d'association entre un traitement et un traitement est autorisé dans un diagramme de séquence. Vérifie que : - la méthode lienAutorise() retourne vrai

### testLienAutoriseAssociationActeurActifTraitement

```
public void testLienAutoriseAssociationActeurActifTraitement()
```

Test unitaire qui vérifie que le lien d'association entre un acteur actif et un traitement n'est pas autorisé dans un diagramme de séquence. Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseAssociationActeurPassifTraitement

```
public void testLienAutoriseAssociationActeurPassifTraitement()
```

Test unitaire qui vérifie que le lien d'association entre un acteur passif et un traitement n'est pas autorisé dans un diagramme de séquence. Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseAssociationTraitementActeurActif

```
public void testLienAutoriseAssociationTraitementActeurActif()
```

Test unitaire qui vérifie que le lien d'association entre un traitement et un acteur actif n'est pas autorisé dans un diagramme de séquence. Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseAssociationTraitementActeurPassif

```
public void testLienAutoriseAssociationTraitementActeurPassif()
```

Test unitaire qui vérifie que le lien d'association entre un traitement et un acteur passif n'est pas autorisé dans un diagramme de séquence. Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseDependanceActeurActifActeurActif

```
public void testLienAutoriseDependanceActeurActifActeurActif()
```

Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un acteur actif et un acteur actif n'est pas autorisé dans un diagramme de séquence. Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseDependanceActeurActifTraitement

```
public void testLienAutoriseDependanceActeurActifTraitement()
```

Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un acteur actif et un traitement n'est pas autorisé dans un diagramme de séquence. Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseDependanceActeurPassifTraitement

```
public void testLienAutoriseDependanceActeurPassifTraitement()
```

Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un acteur passif et un traitement n'est pas autorisé dans un diagramme de séquence. Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseDependanceTraitementActeurActif

```
public void testLienAutoriseDependanceTraitementActeurActif()
```

Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un traitement et un acteur actif n'est pas autorisé dans un diagramme de séquence. Vérifie que : - la méthode lienAutorise() retourne faux

### testLienAutoriseDependanceTraitementActeurPassif

```
public void testLienAutoriseDependanceTraitementActeurPassif()
```

Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un traitement et un acteur passif n'est pas autorisé dans un diagramme de séquence. Vérifie que : - la méthode lienAutorise() retourne faux

**testLienAutoriseDependanceActeurPassifActeurPassif**

```
public void testLienAutoriseDependanceActeurPassifActeurPassif()

    Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un
    acteur passif et un acteur passif n'est pas autorisé dans un diagramme de
    séquence. Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseDependanceActeurActifActeurPassif**

```
public void testLienAutoriseDependanceActeurActifActeurPassif()

    Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un
    acteur passif et un acteur actif n'est pas autorisé dans un diagramme de
    séquence. Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseDependanceActeurPassifActeurActif**

```
public void testLienAutoriseDependanceActeurPassifActeurActif()

    Test unitaire qui vérifie que le lien de dependance fonctionnelle entre un
    acteur passif et un acteur actif n'est pas autorisé dans un diagramme de
    séquence. Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseFlecheTraitementTraitement**

```
public void testLienAutoriseFlecheTraitementTraitement()

    Test unitaire qui vérifie que le lien de flèche entre un traitement et un
    traitement est autorisé dans un diagramme de séquence. Vérifie que : - la
    méthode lienAutorise() retourne vrai
```

---

**testLienAutoriseFlecheActeurActifTraitement**

```
public void testLienAutoriseFlecheActeurActifTraitement()

    Test unitaire qui vérifie que le lien de flèche entre un acteur actif et un
    traitement n'est pas autorisé dans un diagramme de séquence. Vérifie que
    : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseFlecheActeurActifActeurPassif**

```
public void testLienAutoriseFlecheActeurActifActeurPassif()

    Test unitaire qui vérifie que le lien de flèche entre un acteur actif et un
    acteur passif n'est pas autorisé dans un diagramme de séquence. Vérifie
    que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseFlecheActeurPassifActeurActif**

```
public void testLienAutoriseFlecheActeurPassifActeurActif()

    Test unitaire qui vérifie que le lien de flèche entre un acteur passif et un
    acteur passif n'est pas autorisé dans un diagramme de séquence. Vérifie
    que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseSpecialisation**

```
public void testLienAutoriseSpecialisation()

    Test unitaire qui vérifie que le lien de spécialisation entre n'importe quels
    éléments graphiques n'est pas autorisé dans un diagramme de séquence.
    Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseAgregation**

```
public void testLienAutoriseAgregation()

    Test unitaire qui vérifie que le lien d'agrégation entre n'importe quels
    éléments graphiques n'est pas autorisé dans un diagramme de séquence.
    Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseClasseAssociation**

```
public void testLienAutoriseClasseAssociation()

    Test unitaire qui vérifie que le lien de classe-association entre n'importe
    quels éléments graphiques n'est pas autorisé dans un diagramme de
    séquence. Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseFlecheActeurPassifTraitement**

```
public void testLienAutoriseFlecheActeurPassifTraitement()

    Test unitaire qui vérifie que le lien de flèche entre un acteur passif et un
    traitement n'est pas autorisé dans un diagramme de séquence. Vérifie que
    : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseFlecheTraitementActeurActif**

```
public void testLienAutoriseFlecheTraitementActeurActif()

    Test unitaire qui vérifie que le lien de flèche entre un traitement et un
    acteur actif n'est pas autorisé dans un diagramme de séquence. Vérifie
    que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseFlecheTraitementActeurPassif**

```
public void testLienAutoriseFlecheTraitementActeurPassif()

    Test unitaire qui vérifie que le lien de flèche entre un traitement et un
    acteur passif n'est pas autorisé dans un diagramme de séquence. Vérifie
    que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseFlecheActeurActifActeurActif**

```
public void testLienAutoriseFlecheActeurActifActeurActif()

    Test unitaire qui vérifie que le lien de flèche entre un acteur actif et un
    acteur actif n'est pas autorisé dans un diagramme de séquence. Vérifie
    que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseFlecheActeurPassifActeurPassif**

```
public void testLienAutoriseFlecheActeurPassifActeurPassif()

    Test unitaire qui vérifie que le lien de flèche entre un acteur passif et un
    acteur passif n'est pas autorisé dans un diagramme de séquence. Vérifie
    que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseGeneralisation**

```
public void testLienAutoriseGeneralisation()

    Test unitaire qui vérifie que le lien de généralisation entre n'importe
    quels éléments graphiques n'est pas autorisé dans un diagramme de
    séquence. Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testLienAutoriseComposition**

```
public void testLienAutoriseComposition()

    Test unitaire qui vérifie que le lien de composition entre n'importe quels
    éléments graphiques n'est pas autorisé dans un diagramme de séquence.
    Vérifie que : - la méthode lienAutorise() retourne faux
```

---

**testEltAutoriseTraitement**

```
public void testEltAutoriseTraitement()

    Test unitaire qui vérifie que l'élément traitement est autorisé dans un
    diagramme de séquence Vérifie que : - la méthode eltAutorise() retourne
    vrai
```

---

**testEltAutoriseClasse**

```
public void testEltAutoriseClasse()

    Test unitaire qui vérifie que l'élément classe n'est pas autorisé dans un
    diagramme de séquence Vérifie que : - la méthode eltAutorise() retourne
    faux
```

---

**testEltAutoriseLien**

```
public void testEltAutoriseLien()

    Test unitaire qui vérifie que l'élément lien n'est pas autorisé dans un
    diagramme de séquence Vérifie que : - la méthode eltAutorise() retourne
    faux
```

---



testEltAutoriseActeurActif

```
public void testEltAutoriseActeurActif()

    Test unitaire qui vérifie que l'élément acteur actif est autorisé dans un
    diagramme de séquence Vérifie que : - la méthode eltAutorise() retourne
    vrai
```

testEltAutoriseActeurPassif

```
public void testEltAutoriseActeurPassif()

    Test unitaire qui vérifie que l'élément acteur passif est autorisé dans un
    diagramme de séquence Vérifie que : - la méthode eltAutorise() retourne
    vrai
```

testEltAutoriseInterface

```
public void testEltAutoriseInterface()

    Test unitaire qui vérifie que l'élément interface n'est pas autorisé dans un
    diagramme de séquence Vérifie que : - la méthode eltAutorise() retourne
    faux
```

testEltAutoriseCasUtilisation

```
public void testEltAutoriseCasUtilisation()

    Test unitaire qui vérifie que l'élément cas d'utilisation n'est pas autorisé
    dans un diagramme de séquence Vérifie que : - la méthode eltAutorise()
    retourne faux
```

	est bien relié à sa cellule.
void	<a href="#">getElementGraphiqueViaCelluleInterface()</a> Test unitaire qui vérifie que qu'un élément graphique interface est bien relié à sa cellule.
void	<a href="#">setUp()</a> Suppression de ces champs après chaque test
void	<a href="#">tearDown()</a> Suppression de ces champs après chaque test
void	<a href="#">testConstructeurDiagramme()</a> Test unitaire qui vérifie que l'instanciation de la liste d'éléments graphiques.
void	<a href="#">testEltAutoriseActeurActif()</a> Test unitaire qui vérifie que l'élément acteur actif est autorisé dans un diagramme.
void	<a href="#">testEltAutoriseActeurPassif()</a> Test unitaire qui vérifie que l'élément acteur passif est autorisé dans un diagramme.
void	<a href="#">testEltAutoriseCasUtilisation()</a> Test unitaire qui vérifie que l'élément cas d'utilisation est autorisé dans un diagramme.
void	<a href="#">testEltAutoriseClasse()</a> Test unitaire qui vérifie que l'élément classe est autorisé dans un diagramme.
void	<a href="#">testEltAutoriseInterface()</a> Test unitaire qui vérifie que l'élément interface est autorisé dans un diagramme.
void	<a href="#">testEltAutoriseLien()</a> Test unitaire qui vérifie que l'élément lien est autorisé dans un diagramme.
void	<a href="#">testEltAutoriseTraitement()</a> Test unitaire qui vérifie que l'élément traitement est autorisé dans un diagramme.
void	<a href="#">testGetElementsGraphiques()</a> Test unitaire qui vérifie que l'instanciation de la liste d'éléments graphiques.
void	<a href="#">testLienAutoriseAgregationClasseClasse()</a> Test unitaire qui vérifie que le lien d'agrégation entre une classe et une classe est autorisé dans un diagramme.

diagramme

Class DiagrammeTest

```
java.lang.Object
└─diagramme.DiagrammeTest
```

```
public class DiagrammeTest
    extends java.lang.Object
```

Cas de test JUnit regroupant les tests unitaires de chaque méthode de la classe DiagrammeTest

See Also:

Diagramme

Constructor Summary

[DiagrammeTest\(\)](#)

Method Summary

void	<a href="#">getElementGraphiqueViaCelluleActeurActif()</a> Test unitaire qui vérifie que qu'un élément graphique acteur actif est bien relié à sa cellule.
void	<a href="#">getElementGraphiqueViaCelluleActeurPassif()</a> Test unitaire qui vérifie que qu'un élément graphique acteur passif est bien relié à sa cellule.
void	<a href="#">getElementGraphiqueViaCelluleCasUtilisation()</a> Test unitaire qui vérifie que qu'un élément graphique cas d'utilisation est bien relié à sa cellule.
void	<a href="#">getElementGraphiqueViaCelluleClasse()</a> Test unitaire qui vérifie que qu'un élément graphique classe est bien relié à sa cellule.
void	<a href="#">getElementGraphiqueViaCelluleLien()</a> Test unitaire qui vérifie que qu'un élément graphique lien est bien relié à sa cellule.
void	<a href="#">getElementGraphiqueViaCelluleTraitement()</a> Test unitaire qui vérifie que qu'un élément graphique traitement

void	<a href="#">testLienAutoriseAgregationClasseLien()</a> Test unitaire qui vérifie que le lien d'agrégation entre une classe et une classe est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseAgregationLienClasse()</a> Test unitaire qui vérifie que le lien d'agrégation entre un lien et une classe est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseAgregationLienLien()</a> Test unitaire qui vérifie que le lien d'agrégation entre un lien et un lien est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseAssociationActeurActifActeurActif()</a> Test unitaire qui vérifie que le lien d'association entre un acteur actif et un acteur actif est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseAssociationActeurActifCasUtilisation()</a> Test unitaire qui vérifie que le lien d'association entre un acteur actif et un cas d'utilisation est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseAssociationActeurPassifActeurPassif()</a> Test unitaire qui vérifie que le lien d'association entre un acteur passif et un acteur passif est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseAssociationCasUtilisationActeurActif()</a> Test unitaire qui vérifie que le lien d'association entre un cas d'utilisation et un acteur actif est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseAssociationCasUtilisationCasUtilisation()</a> Test unitaire qui vérifie que le lien d'association entre un cas d'utilisation et un cas d'utilisation est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseAssociationClasseClasse()</a> Test unitaire qui vérifie que le lien d'association entre une classe et une classe est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseAssociationClasseLien()</a> Test unitaire qui vérifie que le lien d'association entre une classe et un lien est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseAssociationLienClasse()</a> Test unitaire qui vérifie que le lien d'association entre un lien et une classe est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseAssociationLienLien()</a> Test unitaire qui vérifie que le lien d'association entre un lien et un lien est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseAssociationTraitementTraitement()</a> Test unitaire qui vérifie que le lien d'association entre un traitement et un traitement est autorisé dans un diagramme.

void	<a href="#">testLienAutoriseClasseAssociation()</a> Test unitaire qui vérifie que le lien de classe association entre n'importe quels éléments graphiques est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseCompositionClasseClasse()</a> Test unitaire qui vérifie que le lien de composition entre une classe et une classe est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseCompositionClasseLien()</a> Test unitaire qui vérifie que le lien de composition entre une classe et un lien est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseCompositionLienClasse()</a> Test unitaire qui vérifie que le lien de composition entre un lien et une classe est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseCompositionLienLien()</a> Test unitaire qui vérifie que le lien de composition entre un lien et un lien est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseDependanceActeurActifActeurActif()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un acteur actif et un acteur actif est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseDependanceActeurActifActeurActif()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un acteur actif et un acteur actif est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseDependanceCasUtilisationCasUtilisation()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un cas d'utilisation et un cas d'utilisation est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseDependanceClasseClasse()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre une classe et une classe est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseDependanceClasseLien()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre une classe et un lien est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseDependanceLienClasse()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un lien et une classe est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseDependanceLienLien()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un lien et un lien est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseDependanceTraitementTraitement()</a> Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un traitement et un traitement est autorisé dans un diagramme.

## tearDown

public void **tearDown**()

Suppression de ces champs après chaque test

## testConstructeurDiagramme

public void **testConstructeurDiagramme**()

Test unitaire qui vérifie que l'instanciation de la liste d'éléments graphiques. Vérifie que : - la liste d'éléments graphiques est une instance de liste

## testLienAutoriseAssociationActeurActifCasUtilisation

public void **testLienAutoriseAssociationActeurActifCasUtilisation**()

Test unitaire qui vérifie que le lien d'association entre un acteur actif et un cas d'utilisation est autorisé dans un diagramme. Vérifie que : - la méthode lienAutorise() retourne vrai

## testLienAutoriseAssociationCasUtilisationActeurActif

public void **testLienAutoriseAssociationCasUtilisationActeurActif**()

Test unitaire qui vérifie que le lien d'association entre un cas d'utilisation et un acteur actif est autorisé dans un diagramme. Vérifie que : - la méthode lienAutorise() retourne vrai

## testLienAutoriseAssociationCasUtilisationCasUtilisation

public void **testLienAutoriseAssociationCasUtilisationCasUtilisation**()

Test unitaire qui vérifie que le lien d'association entre un cas d'utilisation et un cas d'utilisation est autorisé dans un diagramme. Vérifie que : - la méthode lienAutorise() retourne vrai

## testLienAutoriseAssociationActeurActifActeurActif

public void **testLienAutoriseAssociationActeurActifActeurActif**()

void	<a href="#">testLienAutoriseFleche()</a> Test unitaire qui vérifie que le lien de flèche entre n'importe quels éléments graphiques est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseGeneralisationActeurActifActeurActif()</a> Test unitaire qui vérifie que le lien de généralisation entre un acteur actif et un acteur actif est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseGeneralisationCasUtilisationCasUtilisation()</a> Test unitaire qui vérifie que le lien de généralisation entre un cas d'utilisation et un cas d'utilisation est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseGeneralisationClasseClasse()</a> Test unitaire qui vérifie que le lien de généralisation entre une classe et une classe est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseGeneralisationClasseLien()</a> Test unitaire qui vérifie que le lien de généralisation entre une classe et un lien est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseGeneralisationLienClasse()</a> Test unitaire qui vérifie que le lien de généralisation entre un lien et une classe est autorisé dans un diagramme.
void	<a href="#">testLienAutoriseGeneralisationLienLien()</a> Test unitaire qui vérifie que le lien de généralisation entre un lien et un lien est autorisé dans un diagramme.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

# Constructor Detail

## DiagrammeTest

public **DiagrammeTest**()

## Method Detail

## setUp

public void **setUp**()

Suppression de ces champs après chaque test

Test unitaire qui vérifie que le lien d'association entre un acteur actif et un acteur actif est autorisé dans un diagramme. Vérifie que : - la méthode lienAutorise() retourne vrai

## testLienAutoriseAssociationActeurPassifActeurPassif

public void **testLienAutoriseAssociationActeurPassifActeurPassif**()

Test unitaire qui vérifie que le lien d'association entre un acteur passif et un acteur passif est autorisé dans un diagramme. Vérifie que : - la méthode lienAutorise() retourne vrai

## testLienAutoriseGeneralisationCasUtilisationCasUtilisation

public void **testLienAutoriseGeneralisationCasUtilisationCasUtilisation**()

Test unitaire qui vérifie que le lien de généralisation entre un cas d'utilisation et un cas d'utilisation est autorisé dans un diagramme. Vérifie que : - la méthode lienAutorise() retourne vrai

## testLienAutoriseGeneralisationActeurActifActeurActif

public void **testLienAutoriseGeneralisationActeurActifActeurActif**()

Test unitaire qui vérifie que le lien de généralisation entre un acteur actif et un acteur actif est autorisé dans un diagramme. Vérifie que : - la méthode lienAutorise() retourne vrai

## testLienAutoriseDependanceCasUtilisationCasUtilisation

public void **testLienAutoriseDependanceCasUtilisationCasUtilisation**()

Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un cas d'utilisation et un cas d'utilisation est autorisé dans un diagramme. Vérifie que : - la méthode lienAutorise() retourne vrai

## testLienAutoriseDependanceActeurActifActeurActif

public void **testLienAutoriseDependanceActeurActifActeurActif**()

Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un



**testLienAutoriseAssociationTraitementTraitement**

```
public void testLienAutoriseAssociationTraitementTraitement()

    Test unitaire qui vérifie que le lien d'association entre un traitement et un
    traitement est autorisé dans un diagramme. Vérifie que : - la méthode
    lienAutorise() retourne vrai
```

**testLienAutoriseDependanceTraitementTraitement**

```
public void testLienAutoriseDependanceTraitementTraitement()

    Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un
    acteur actif et un traitement est autorisé dans un diagramme. Vérifie que :
    - la méthode lienAutorise() retourne vrai
```

**testLienAutoriseDependanceActeurActifActeurActif**

```
public void testLienAutoriseDependanceActeurActifActeurActif()

    Test unitaire qui vérifie que le lien de dépendance fonctionnelle entre un
    acteur actif et un acteur actif est autorisé dans un diagramme. Vérifie que
    : - la méthode lienAutorise() retourne vrai
```

**testLienAutoriseClasseAssociation**

```
public void testLienAutoriseClasseAssociation()

    Test unitaire qui vérifie que le lien de classe association entre n'importe
    quels éléments graphiques est autorisé dans un diagramme. Vérifie que : -
    la méthode lienAutorise() retourne vrai
```

**testLienAutoriseFleche**

```
public void testLienAutoriseFleche()

    Test unitaire qui vérifie que le lien de flèche entre n'importe quels
    éléments graphiques est autorisé dans un diagramme. Vérifie que : - la
    méthode lienAutorise() retourne vrai
```

**testEltAutoriseCasUtilisation**

```
public void testEltAutoriseCasUtilisation()

    Test unitaire qui vérifie que l'élément cas d'utilisation est autorisé dans
    un diagramme. Vérifie que : - la méthode eltAutorise() retourne vrai
```

**testGetElementsGraphiques**

```
public void testGetElementsGraphiques()

    Test unitaire qui vérifie que l'instanciation de la liste d'éléments
    graphiques. Vérifie que : - la liste d'éléments graphiques est une liste
    d'éléments graphiques vide
```

**getElementGraphiqueViaCelluleTraitement**

```
public void getElementGraphiqueViaCelluleTraitement()

    Test unitaire qui vérifie que qu'un élément graphique traitement est bien
    relié à sa cellule. Vérifie que : - la méthode
    getElementGraphiqueViaCellule retourne bien l'élément graphique dont
    la cellule est mise en paramètre.
```

**getElementGraphiqueViaCelluleClasse**

```
public void getElementGraphiqueViaCelluleClasse()

    Test unitaire qui vérifie que qu'un élément graphique classe est bien relié
    à sa cellule. Vérifie que : - la méthode getElementGraphiqueViaCellule
    retourne bien l'élément graphique dont la cellule est mise en paramètre.
```

**getElementGraphiqueViaCelluleLien**

```
public void getElementGraphiqueViaCelluleLien()

    Test unitaire qui vérifie que qu'un élément graphique lien est bien relié à
    sa cellule. Vérifie que : - la méthode getElementGraphiqueViaCellule
    retourne bien l'élément graphique dont la cellule est mise en paramètre.
```

**testEltAutoriseTraitement**

```
public void testEltAutoriseTraitement()

    Test unitaire qui vérifie que l'élément traitement est autorisé dans un
    diagramme. Vérifie que : - la méthode eltAutorise() retourne vrai
```

**testEltAutoriseClasse**

```
public void testEltAutoriseClasse()

    Test unitaire qui vérifie que l'élément classe est autorisé dans un
    diagramme. Vérifie que : - la méthode eltAutorise() retourne vrai
```

**testEltAutoriseLien**

```
public void testEltAutoriseLien()

    Test unitaire qui vérifie que l'élément lien est autorisé dans un
    diagramme. Vérifie que : - la méthode eltAutorise() retourne vrai
```

**testEltAutoriseActeurActif**

```
public void testEltAutoriseActeurActif()

    Test unitaire qui vérifie que l'élément acteur actif est autorisé dans un
    diagramme. Vérifie que : - la méthode eltAutorise() retourne vrai
```

**testEltAutoriseActeurPassif**

```
public void testEltAutoriseActeurPassif()

    Test unitaire qui vérifie que l'élément acteur passif est autorisé dans un
    diagramme. Vérifie que : - la méthode eltAutorise() retourne vrai
```

**testEltAutoriseInterface**

```
public void testEltAutoriseInterface()

    Test unitaire qui vérifie que l'élément interface est autorisé dans un
    diagramme. Vérifie que : - la méthode eltAutorise() retourne vrai
```

**getElementGraphiqueViaCelluleActeurActif**

```
public void getElementGraphiqueViaCelluleActeurActif()

    Test unitaire qui vérifie que qu'un élément graphique acteur actif est bien
    relié à sa cellule. Vérifie que : - la méthode
    getElementGraphiqueViaCellule retourne bien l'élément graphique dont
    la cellule est mise en paramètre.
```

**getElementGraphiqueViaCelluleActeurPassif**

```
public void getElementGraphiqueViaCelluleActeurPassif()

    Test unitaire qui vérifie que qu'un élément graphique acteur passif est
    bien relié à sa cellule. Vérifie que : - la méthode
    getElementGraphiqueViaCellule retourne bien l'élément graphique dont
    la cellule est mise en paramètre.
```

**getElementGreaphiqueViaCelluleInterface**

```
public void getElementGreaphiqueViaCelluleInterface()

    Test unitaire qui vérifie que qu'un élément graphique interface est bien
    relié à sa cellule. Vérifie que : - la méthode
    getElementGraphiqueViaCellule retourne bien l'élément graphique dont
    la cellule est mise en paramètre.
```

**getElementGraphiqueViaCelluleCasUtilisation**

```
public void getElementGraphiqueViaCelluleCasUtilisation()

    Test unitaire qui vérifie que qu'un élément graphique cas d'utilisation est
    bien relié à sa cellule. Vérifie que : - la méthode
    getElementGraphiqueViaCellule retourne bien l'élément graphique dont
    la cellule est mise en paramètre.
```

<b>eltGraphique</b>
<b>Class ActeurActifTest</b>
java.lang.Object ↳eltGraphique.ActeurActifTest
public class <b>ActeurActifTest</b> extends java.lang.Object
Cas de test JUnit regroupant les tests unitaires de chaque méthode de la classe ActeurActif
<b>See Also:</b> ActeurActif

<b>Constructor Summary</b>
<a href="#">ActeurActifTest()</a>

<b>Method Summary</b>
void <a href="#">setUp()</a> Initialisation du champ avant chaque test
void <a href="#">tearDown()</a> Suppression de ce champ après chaque test
void <a href="#">testApplicationStyle()</a> Test unitaire qui vérifie l'application du style sur l'acteur actif en récupérant le style de la cellule liée à l'acteur actif Vérifie que : - le style de la cellule est ACTEUR_ACTIF
void <a href="#">testCelluleParent()</a> Test unitaire qui teste l'acteur actif n'a pas d'élément parent Vérifie que : - la cellule parente est la même que la cellule actuelle
void <a href="#">testCelluleTarget()</a> Test unitaire qui teste l'acteur actif n'a pas d'élément cible Vérifie que : - la cellule cible est la même que la cellule actuelle
void <a href="#">testCreerCelluleNotNull()</a> Test unitaire qui teste la création de la cellule en vérifiant que l'attribut n'est pas null Vérifie que : - L'attribut celle du champ n'est pas

<b>tearDown</b>
public void <b>tearDown()</b>  Suppression de ce champ après chaque test

<b>testCreerStyle</b>
public void <b>testCreerStyle()</b>  Test unitaire qui teste la création du style de l'acteur actif en vérifiant chaque champ du nouveau style Vérifie que : - le style est créé - la forme est celle d'un acteur actif - l'inclusion d'un autre élément soit impossible - l'opacite est correcte - la couleur du texte est correcte - la couleur de la bordure est correcte

<b>testApplicationStyle</b>
public void <b>testApplicationStyle()</b>  Test unitaire qui vérifie l'application du style sur l'acteur actif en récupérant le style de la cellule liée à l'acteur actif Vérifie que : - le style de la cellule est ACTEUR_ACTIF

<b>testCreerCelluleNotNull</b>
public void <b>testCreerCelluleNotNull()</b>  Test unitaire qui teste la création de la cellule en vérifiant que l'attribut n'est pas null Vérifie que : - L'attribut celle du champ n'est pas null

<b>testInstance</b>
public void <b>testInstance()</b>  Test unitaire que teste l'instanciation du champ Vérifie que : - le champ est une instance de la classe ActeurActif

<b>testCelluleParent</b>
--------------------------

	null
void <a href="#">testCreerStyle()</a>	Test unitaire qui teste la création du style de l'acteur actif en vérifiant chaque champ du nouveau style Vérifie que : - le style est créé - la forme est celle d'un acteur actif - l'inclusion d'un autre élément soit impossible - l'opacite est correcte - la couleur du texte est correcte - la couleur de la bordure est correcte
void <a href="#">testInstance()</a>	Test unitaire que teste l'instanciation du champ Vérifie que : - le champ est une instance de la classe ActeurActif
void <a href="#">testSetCellule()</a>	Test unitaire qui teste la modification de la cellule lié à l'acteur actif Vérifie que : - la cellule actuelle a bien été remplacée - la cellule cible est la même que la cellule courante - la cellule parent est la mee que la cellule courante
void <a href="#">testSetTexte()</a>	Test unitaire qui teste la modification du texte lié à l'acteur actif Vérifie que : - le texte est modifié correctement
void <a href="#">testSupprimer()</a>	Test unitaire qui teste al suppression de la cellule liée à l'acteur Vérifie que : - la cellule courante est null - la cellule cible est null - la cellule parent est null

<b>Methods inherited from class java.lang.Object</b>
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

<b>Constructor Detail</b>
---------------------------

ActeurActifTest

public ActeurActifTest()

<b>Method Detail</b>
----------------------

setUp

public void setUp()

Initialisation du champ avant chaque test

public void testCelluleParent()

Test unitaire qui teste l'acteur actif n'a pas d'élément parent Vérifie que : - la cellule parente est la même que la cellule actuelle

testCelluleTarget

public void testCelluleTarget()

Test unitaire qui teste l'acteur actif n'a pas d'élément cible Vérifie que : - la cellule cible est la même que la cellule actuelle

testSetTexte

public void testSetTexte()

Test unitaire qui teste la modification du texte lié à l'acteur actif Vérifie que : - le texte est modifié correctement

testSetCellule

public void testSetCellule()

Test unitaire qui teste la modification de la cellule lié à l'acteur actif Vérifie que : - la cellule actuelle a bien été remplacée - la cellule cible est la même que la cellule courante - la cellule parent est la mee que la cellule courante

testSupprimer

public void testSupprimer()

Test unitaire qui teste al suppression de la cellule liée à l'acteur Vérifie que : - la cellule courante est null - la cellule cible est null - la cellule parent est null

<b>eltGraphique</b> <b>Class ActeurPassifTest</b>
<pre>java.lang.Object └─eltGraphique.ActeurPassifTest</pre>
<pre>public class ActeurPassifTest extends java.lang.Object</pre>
Cas de test JUnit regroupant les tests unitaires de chaque méthode de la classe ActeurPassif
<b>See Also:</b> ActeurPassif

Constructor Summary
<a href="#">ActeurPassifTest()</a>

Method Summary
<div><div>void</div><div><a href="#">setUp()</a></div></div> <div>Initialisation du champ avant chaque test</div>
<div><div>void</div><div><a href="#">tearDown()</a></div></div> <div>Suppression de ce champ après chaque test</div>
<div><div>void</div><div><a href="#">testCellule()</a></div></div> <div>Test unitaire qui teste que la methode getCellule retourne bien la bonne cellule Vérifie que : - la cellule renvoyée par getCellule est correcte</div>
<div><div>void</div><div><a href="#">testCelluleParent()</a></div></div> <div>Test unitaire qui teste l'acteur passif n'a pas d'élément parent Vérifie que : - la cellule parente est la même que la cellule actuelle</div>
<div><div>void</div><div><a href="#">testCelluleTarget()</a></div></div> <div>Test unitaire qui teste que l'acteur passif n'a pas d'élément cible Vérifie que : - la cellule cible est la même que la cellule actuelle</div>
<div><div>void</div><div><a href="#">testCreationStyle()</a></div></div> <div>Test unitaire qui teste la création du style de l'acteur passif en vérifiant chaque champ du nouveau style Vérifie que : - le style est créé -</div>

<pre>public void setUp()</pre>
Initialisation du champ avant chaque test

### tearDown

<pre>public void tearDown()</pre>
Suppression de ce champ après chaque test

### testCelluleParent

<pre>public void testCelluleParent()</pre>
Test unitaire qui teste l'acteur passif n'a pas d'élément parent Vérifie que : - la cellule parente est la même que la cellule actuelle

### testCelluleTarget

<pre>public void testCelluleTarget()</pre>
Test unitaire qui teste que l'acteur passif n'a pas d'élément cible Vérifie que : - la cellule cible est la même que la cellule actuelle

### testCellule

<pre>public void testCellule()</pre>
Test unitaire qui teste que la methode getCellule retourne bien la bonne cellule Vérifie que : - la cellule renvoyée par getCellule est correcte

### testInstance

<pre>public void testInstance()</pre>
Test unitaire qui teste l'instanciation du champ Vérifie que : - le champ est une instance de la classe ActeurPassif

### testCreationStyle

	la forme est celle d'un acteur passif - l'inclusion d'un autre élément soit impossible - l'opacite est correcte - la couleur du texte est correcte - la couleur de la bordure est correcte
<div><div>void</div><div><a href="#">testInstance()</a></div></div>	Test unitaire qui teste l'instanciation du champ Vérifie que : - le champ est une instance de la classe ActeurPassif
<div><div>void</div><div><a href="#">testSetCelluleParent()</a></div></div>	Test unitaire qui teste la modification de la cellule lié à l'acteur passif Vérifie que : - la cellule actuelle a bien été remplacée - la cellule parent est la même que la cellule courante
<div><div>void</div><div><a href="#">testSetCelluleTarget()</a></div></div>	Test unitaire qui teste la modification de la cellule lié à l'acteur passif Vérifie que : - la cellule cible est la même que la cellule courante
<div><div>void</div><div><a href="#">testSetTexte()</a></div></div>	Test unitaire qui teste la modification du texte lié à l'acteur passif Vérifie que : - le texte est modifié correctement
<div><div>void</div><div><a href="#">testStyle()</a></div></div>	Test unitaire qui vérifie l'application du style sur l'acteur passif en récupérant le style de la cellule liée à l'acteur passif Vérifie que : - le style de la cellule est ACTEUR_PASSIF
<div><div>void</div><div><a href="#">testSupprimer()</a></div></div>	Test unitaire qui teste al suppression de la cellule liée à l'acteur Vérifie que : - la cellule courante est null - la cellule cible est null - la cellule parent est null

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### ActeurPassifTest

<pre>public ActeurPassifTest()</pre>
Method Detail

#### setUp

<pre>public void testCreationStyle()</pre>
Test unitaire qui teste la création du style de l'acteur passif en vérifiant chaque champ du nouveau style Vérifie que : - le style est créé - la forme est celle d'un acteur passif - l'inclusion d'un autre élément soit impossible - l'opacite est correcte - la couleur du texte est correcte - la couleur de la bordure est correcte

### testStyle

<pre>public void testStyle()</pre>
Test unitaire qui vérifie l'application du style sur l'acteur passif en récupérant le style de la cellule liée à l'acteur passif Vérifie que : - le style de la cellule est ACTEUR_PASSIF

### testSetTexte

<pre>public void testSetTexte()</pre>
Test unitaire qui teste la modification du texte lié à l'acteur passif Vérifie que : - le texte est modifié correctement

### testSetCelluleParent

<pre>public void testSetCelluleParent()</pre>
Test unitaire qui teste la modification de la cellule lié à l'acteur passif Vérifie que : - la cellule actuelle a bien été remplacée - la cellule parent est la même que la cellule courante

### testSetCelluleTarget

<pre>public void testSetCelluleTarget()</pre>
Test unitaire qui teste la modification de la cellule lié à l'acteur passif Vérifie que : - la cellule cible est la même que la cellule courante

### testSupprimer

<pre>public void testSupprimer()</pre>
--

Test unitaire qui teste al suppression de la cellule liée à l'acteur Vérifie que : - la cellule courante est null - la cellule cible est null - la cellule parent est null

eltGraphique

## Class AttributTest

java.lang.Object  
└─eltGraphique.AttributTest

public class **AttributTest**  
extends java.lang.Object

Cas de test JUnit regroupant les tests unitaires de chaque méthode de la classe Attribut

**See Also:**  
Attribut

### Constructor Summary

[AttributTest\(\)](#)

### Method Summary

void	<a href="#">setUp()</a>	Initialisation du champ avant chaque test
void	<a href="#">tearDown()</a>	Suppression de ce champ après chaque test
void	<a href="#">testDeClasse()</a>	Test unitaire qui teste la modification du fait que l'attribut est de classe Vérifie que : - la modification de l'attribut deClasse est effective
void	<a href="#">testToString()</a>	Test unitaire qui teste le retour de la méthode toString() Vérifie que : - la génération de la chaîne est correcte
void	<a href="#">testVisibilite()</a>	Test unitaire qui teste la modification de l'attribut visibilite Vérifie que : - la modification de l'attribut visibilite est effective

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### AttributTest

public **AttributTest**()

### Method Detail

#### setUp

public void **setUp**()

Initialisation du champ avant chaque test

#### tearDown

public void **tearDown**()

Suppression de ce champ après chaque test

#### testVisibilite

public void **testVisibilite**()

Test unitaire qui teste la modification de l'attribut visibilite Vérifie que : - la modification de l'attribut visibilite est effective

#### testDeClasse

public void **testDeClasse**()

Test unitaire qui teste la modification du fait que l'attribut est de classe Vérifie que : - la modification de l'attribut deClasse est effective

#### testToString

public void **testToString**()

Test unitaire qui teste le retour de la méthode toString() Vérifie que : - la génération de la chaîne est correcte

<b>eltGraphique</b>
<b>Class CasUtilisationTest</b>
<pre>java.lang.Object └─eltGraphique.CasUtilisationTest</pre>
<pre>public class CasUtilisationTest extends java.lang.Object</pre>
Cas de test JUnit regroupant les tests unitaires de chaque méthode de la classe CasUtilisation
<b>See Also:</b> <div>CasUtilisation</div>

Constructor Summary
<a href="#">CasUtilisationTest()</a>

Method Summary
<div><div>void</div><div><a href="#">setUp()</a></div></div> <div>Initialisation du champ avant chaque test</div>
<div><div>void</div><div><a href="#">tearDown()</a></div></div> <div>Suppression de ce champ après chaque test</div>
<div><div>void</div><div><a href="#">testCelluleParent()</a></div></div> <div>Test unitaire qui teste que le cas d'utilisation n'a pas d'élément parent Vérifie que : - la cellule parente est la même que la cellule actuelle</div>
<div><div>void</div><div><a href="#">testCelluleTarget()</a></div></div> <div>Test unitaire qui teste le cas d'utilisation n'a pas d'élément cible Vérifie que : - la cellule cible est la même que la cellule actuelle</div>
<div><div>void</div><div><a href="#">testChampsStyle()</a></div></div> <div>Test unitaire qui teste la création du style du cas d'utilisation en vérifiant chaque champ du nouveau style Vérifie que : - le style est créé - la forme est celle du cas d'utilisation - l'inclusion d'un autre élément soit impossible - l'opacite est correcte - la couleur du texte est correcte - la couleur de la bordure est correcte</div>

<pre>public void <b>tearDown()</b></pre>
Suppression de ce champ après chaque test

### testCelluleParent

<pre>public void <b>testCelluleParent()</b></pre>
Test unitaire qui teste que le cas d'utilisation n'a pas d'élément parent Vérifie que : - la cellule parente est la même que la cellule actuelle

### testCelluleTarget

<pre>public void <b>testCelluleTarget()</b></pre>
Test unitaire qui teste le cas d'utilisation n'a pas d'élément cible Vérifie que : - la cellule cible est la même que la cellule actuelle

### testInstance

<pre>public void <b>testInstance()</b></pre>
Test unitaire que teste l'instanciation du champ Vérifie que : - le champ est une instance de la classe CasUtilisation

### testStyle

<pre>public void <b>testStyle()</b></pre>
Test unitaire qui vérifie l'application du style sur le cas d'utilisation en récupérant le style de la cellule liée à le cas d'utilisation Vérifie que : - le style de la cellule est USECASE

### testChampsStyle

<pre>public void <b>testChampsStyle()</b></pre>
Test unitaire qui teste la création du style du cas d'utilisation en vérifiant chaque champ du nouveau style Vérifie que : - le style est créé - la forme est celle du cas d'utilisation - l'inclusion d'un autre élément soit impossible - l'opacite est correcte - la couleur du texte est correcte - la

<div>void</div>	<div><a href="#">testInstance()</a></div> <div>Test unitaire que teste l'instanciation du champ Vérifie que : - le champ est une instance de la classe CasUtilisation</div>
<div>void</div>	<div><a href="#">testSetCellule()</a></div> <div>Test unitaire qui teste que la modification d'une cellule est effective Vérifie que : - la cellule actuelle est modifiée correctement - la cellule parente est modifiée correctement - la cellule cible est modifiée correctement</div>
<div>void</div>	<div><a href="#">testSetTexte()</a></div> <div>Test unitaire qui teste que la modification du texte est effective Vérifie que : - le nouveau texte est attribué correctement</div>
<div>void</div>	<div><a href="#">testStyle()</a></div> <div>Test unitaire qui vérifie l'application du style sur le cas d'utilisation en récupérant le style de la cellule liée à le cas d'utilisation Vérifie que : - le style de la cellule est USECASE</div>
<div>void</div>	<div><a href="#">testSupprimer()</a></div> <div>Test unitaire qui vérifie que la méthode supprimer fonctionne correctement Vérifie que : - la cellule actuelle est à null - la cellule parente est à null - la cellule cible est à null</div>

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### CasUtilisationTest

public **CasUtilisationTest()**

## Method Detail

### setUp

public void **setUp()**

Initialisation du champ avant chaque test

### tearDown

couleur de la bordure est correcte

### testSetTexte

public void **testSetTexte()**

Test unitaire qui teste que la modification du texte est effective Vérifie que : - le nouveau texte est attribué correctement

### testSetCellule

public void **testSetCellule()**

Test unitaire qui teste que la modification d'une cellule est effective Vérifie que : - la cellule actuelle est modifiée correctement - la cellule parente est modifiée correctement - la cellule cible est modifiée correctement

### testSupprimer

public void **testSupprimer()**

Test unitaire qui vérifie que la méthode supprimer fonctionne correctement Vérifie que : - la cellule actuelle est à null - la cellule parente est à null - la cellule cible est à null



<b>eltGraphique</b>
<b>Class ClasseTest</b>
<div>java.lang.Object</div> <div>↳eltGraphique.ClasseTest</div>
<div>public class <b>ClasseTest</b></div> <div>extends java.lang.Object</div>
Cas de test JUnit regroupant les tests unitaires de chaque méthode de la classe CasUtilisation
<b>See Also:</b>
CasUtilisation

Constructor Summary
<a href="#">ClasseTest()</a>

Method Summary
<div>void <a href="#">setUp()</a></div> <div>Initialisation du champ avant chaque test</div>
<div>void <a href="#">tearDown()</a></div> <div>Suppression de ce champ après chaque test</div>
<div>void <a href="#">testAjouterAttribut()</a></div> <div>Test unitaire qui vérifie lors de l'ajout d'un attribut à la classe que ledit est bien référencée dans la liste des attributs de la classe Vérifie que : - le nouvel attribut est présent dans la liste des attributs de la classe</div>
<div>void <a href="#">testAjouterAttributTailleListe()</a></div> <div>Test unitaire qui teste la modification de la taille de la liste d'attributs lors de l'ajout d'un attribut à la classe Vérifie que : - la nouvelle taille est égale à l'ancienne + 1</div>
<div>void <a href="#">testAjouterMethode()</a></div> <div>Test unitaire qui vérifie lors de l'ajout d'une méthode à la classe que ladite est bien référencée dans la liste des méthodes de la classe Vérifie que : - la nouvelle méthode est présente dans la liste des méthodes de la classe</div>

<div>void <a href="#">testGetMethodesInstance()</a></div> <div>Test unitaire qui teste l'instanciation de la liste de attributs de la classe Vérifie que : - la liste des attributs est une instance de Liste</div>
<div>void <a href="#">testSetAttributs()</a></div> <div>Test unitaire qui vérifie la modification de la liste d'attributs de la classe via la méthode setAttributs Vérifie que : - la nouvelle liste d'attributs est assignée correctement</div>
<div>void <a href="#">testSetCelluleParent()</a></div> <div>Test unitaire qui teste la modification de la cellule lié à la classe Vérifie que : - la cellule actuelle a bien été remplacée - la cellule parent est la même que la cellule courante</div>
<div>void <a href="#">testSetCelluleTarget()</a></div> <div>Test unitaire qui teste la modification de la cellule lié à la classe Vérifie que : - la cellule cible est la même que la cellule courante</div>
<div>void <a href="#">testSetMethodes()</a></div> <div>Test unitaire qui vérifie la modification de la liste de méthodes de la classe via la méthode setMethodes Vérifie que : - la nouvelle liste de méthodes est assignée correctement</div>
<div>void <a href="#">testSetTexte()</a></div> <div>Test unitaire qui teste la modification du texte lié à la classe Vérifie que : - le texte est modifié correctement</div>
<div>void <a href="#">testSupprimer()</a></div> <div>Test unitaire qui vérifie la suppression de l'élément Vérifie que : - la cellule actuelle est à null - la cellule parente est à null - la cellule cible est à null</div>

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail
--------------------

<b>ClasseTest</b>
<div>public <b>ClasseTest</b>()</div>
Method Detail

setUp

<div>void</div>	<div><a href="#">testAjouterMethodeTailleListe()</a></div> <div>Test unitaire qui teste la modification de la taille de la liste de méthode lors de l'ajout d'une méthode à la classe Vérifie que : - la nouvelle taille est égale à l'ancienne + 1</div>
<div>void</div>	<div><a href="#">testCelluleParentTarget()</a></div> <div>Test unitaire qui teste que la classe n'a pas d'élément parent et cible Vérifie que : - la cellule parente est la même que la cellule actuelle - la cellule cible est la même que la cellule actuelle</div>
<div>void</div>	<div><a href="#">testCreerInstance()</a></div> <div>Test unitaire qui teste si l'instanciation du champ est correcte Vérifie que : - le champ est une instance de Classe</div>
<div>void</div>	<div><a href="#">testCreerStyle()</a></div> <div>Test unitaire qui teste l'application du style à la cellule Vérifie que : - Le style appliqué est bien "CLASSE"</div>
<div>void</div>	<div><a href="#">testCreerStyleClasse()</a></div> <div>Test unitaire qui teste la création du style de la classe en vérifiant chaque champ du nouveau style Vérifie que : - le style est créé - la forme est celle d'une classe - l'inclusion d'un autre élément soit impossible - l'opacite est correcte - la couleur du texte est correcte - la couleur de la bordure est correcte</div>
<div>void</div>	<div><a href="#">testCreerStyleContenuClasse()</a></div> <div>Test unitaire qui teste la création du style de l'acteur passif en vérifiant chaque champ du nouveau style Vérifie que : - le style est créé - la forme est celle du contenu d'une classe - l'inclusion d'un autre élément soit impossible - l'opacite est correcte - la couleur du texte est correcte - la couleur de la bordure est correcte - le redimensionnement du contenu soit impossible - le déplacement du contenu soit impossible - la suppression du contenu soit impossible - l'alignement du texte est à gauche</div>
<div>void</div>	<div><a href="#">testEstAbstraite()</a></div> <div>Test unitaire qui vérifie que l'on peut mettre une classe abstraite Vérifie que : - la modification de l'attribut abstraite de la classe est effective</div>
<div>void</div>	<div><a href="#">testEstConstante()</a></div> <div>Test unitaire qui vérifie que l'on peut mettre une classe constante Vérifie que : - la modification de l'attribut constante de la classe est effective</div>
<div>void</div>	<div><a href="#">testGetAttributsInstance()</a></div> <div>Test unitaire qui teste l'instanciation de la liste de méthodes de la classe Vérifie que : - la liste des méthode est une instance de Liste</div>

<div>public void <b>setUp()</b></div>
Initialisation du champ avant chaque test

<b>tearDown</b>
<div>public void <b>tearDown()</b></div>
Suppression de ce champ après chaque test

<b>testCelluleParentTarget</b>
<div>public void <b>testCelluleParentTarget()</b></div>
Test unitaire qui teste que la classe n'a pas d'élément parent et cible Vérifie que : - la cellule parente est la même que la cellule actuelle - la cellule cible est la même que la cellule actuelle

<b>testCreerInstance</b>
<div>public void <b>testCreerInstance()</b></div>
Test unitaire qui teste si l'instanciation du champ est correcte Vérifie que : - le champ est une instance de Classe

<b>testCreerStyle</b>
<div>public void <b>testCreerStyle()</b></div>
Test unitaire qui teste l'application du style à la cellule Vérifie que : - Le style appliqué est bien "CLASSE"

<b>testSetTexte</b>
<div>public void <b>testSetTexte()</b></div>
Test unitaire qui teste la modification du texte lié à la classe Vérifie que : - le texte est modifié correctement

testSetCelluleParent

public void **testSetCelluleParent()**

Test unitaire qui teste la modification de la cellule lié à la classe Vérifie que : - la cellule actuelle a bien été remplacée - la cellule parent est la même que la cellule courante

**testSetCelluleTarget**

public void **testSetCelluleTarget()**

Test unitaire qui teste la modification de la cellule lié à la classe Vérifie que : - la cellule cible est la même que la cellule courante

**testSupprimer**

public void **testSupprimer()**

Test unitaire qui vérifie la suppression de l'élément Vérifie que : - la cellule actuelle est à null - la cellule parente est à null - la cellule cible est à null

**testAjouterMethodeTailleListe**

public void **testAjouterMethodeTailleListe()**

Test unitaire qui teste la modification de la taille de la liste de méthode lors de l'ajout d'une méthode à la classe Vérifie que : - la nouvelle taille est égale à l'ancienne + 1

**testAjouterMethode**

public void **testAjouterMethode()**

Test unitaire qui vérifie lors de l'ajout d'une méthode à la classe que ladite est bien référencée dans la liste des méthodes de la classe Vérifie que : - la nouvelle méthode est présente dans la liste des méthodes de la classe

**testAjouterAttribut**

public void **testAjouterAttribut()**

Test unitaire qui vérifie lors de l'ajout d'un attribut à la classe que ledit est bien référencé dans la liste des attributs de la classe Vérifie que : - le nouvel attribut est présent dans la liste des attributs de la classe

Test unitaire qui vérifie lors de la suppression d'un attribut de la classe que ledit est bien référencé dans la liste des attributs de la classe Vérifie que : - l'attribut est bien supprimé de la liste des attributs de la classe

Test unitaire qui vérifie lors de la suppression d'une méthode de la classe que ladite est bien référencée dans la liste des méthodes de la classe Vérifie que : - la méthode est bien supprimée de la liste des méthodes de la classe

Vérifie que : - la liste des attributs est une instance de Liste

**See Also:**  
Liste

**testSetAttributs**

public void **testSetAttributs()**

Test unitaire qui vérifie la modification de la liste d'attributs de la classe via la méthode setAttributs Vérifie que : - la nouvelle liste d'attributs est assignée correctement

**testSetMethodes**

public void **testSetMethodes()**

Test unitaire qui vérifie la modification de la liste de méthodes de la classe via la méthode setMethodes Vérifie que : - la nouvelle liste de méthodes est assignée correctement

**testCreerStyleClasse**

public void **testCreerStyleClasse()**

Test unitaire qui teste la création du style de la classe en vérifiant chaque champ du nouveau style Vérifie que : - le style est créé - la forme est celle d'une classe - l'inclusion d'un autre élément soit impossible - l'opacite est correcte - la couleur du texte est correcte - la couleur de la bordure est correcte

**testCreerStyleContenuclasse**

public void **testCreerStyleContenuclasse()**

Test unitaire qui teste la création du style de l'acteur passif en vérifiant chaque champ du nouveau style Vérifie que : - le style est créé - la forme est celle du contenu d'une classe - l'inclusion d'un autre élément soit impossible - l'opacite est correcte - la couleur du texte est correcte - la couleur de la bordure est correcte - le redimensionnement du contenu soit impossible - le déplacement du contenu soit impossible - la suppression du

Test unitaire qui vérifie lors de l'ajout d'un attribut à la classe que ledit est bien référencé dans la liste des attributs de la classe Vérifie que : - le nouvel attribut est présent dans la liste des attributs de la classe

**testAjouterAttributTailleListe**

public void **testAjouterAttributTailleListe()**

Test unitaire qui teste la modification de la taille de la liste d'attributs lors de l'ajout d'un attribut à la classe Vérifie que : - la nouvelle taille est égale à l'ancienne + 1

**testEstConstante**

public void **testEstConstante()**

Test unitaire qui vérifie que l'on peut mettre une classe constante Vérifie que : - la modification de l'attribut constante de la classe est effective

**testEstAbstraite**

public void **testEstAbstraite()**

Test unitaire qui vérifie que l'on peut mettre une classe abstraite Vérifie que : - la modification de l'attribut abstraite de la classe est effective

**testGetAttributsInstance**

public void **testGetAttributsInstance()**

Test unitaire qui teste l'instanciation de la liste de méthodes de la classe Vérifie que : - la liste des méthode est une instance de Liste

**See Also:**  
Liste

**testGetMethodesInstance**

public void **testGetMethodesInstance()**

Test unitaire qui teste l'instanciation de la liste de attributs de la classe

Test unitaire qui teste l'instanciation de la liste de méthodes de la classe

Test unitaire qui teste l'instanciation de la liste de attributs de la classe

contenu soit impossible - l'alignement du texte est à gauche

eltGraphique

Class InterfaceTest

java.lang.Object

↳eltGraphique.InterfaceTest

public class InterfaceTest

extends java.lang.Object

Cas de test JUnit regroupant les tests unitaires de chaque méthode de la classe Interface

See Also:

Interface

Constructor Summary
<a href="#">InterfaceTest()</a>

Method Summary
<div>void</div> <div><a href="#">setUp()</a></div> <div>Initialisation du champ avant chaque test</div>
<div>void</div> <div><a href="#">tearDown()</a></div> <div>Suppression de ce champ après chaque test</div>
<div>void</div> <div><a href="#">testFail()</a></div> <div>Les tests unitaires ne sont pas encore implémentés pour cette classe</div>

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

InterfaceTest

public InterfaceTest()

eltGraphique

Class LienTest

java.lang.Object

↳eltGraphique.LienTest

public class LienTest

extends java.lang.Object

Cas de test JUnit regroupant les tests unitaires de chaque méthode de la classe Lien

See Also:

Lien

Constructor Summary
<a href="#">LienTest()</a>

Method Summary
<div>void</div> <div><a href="#">setUp()</a></div> <div>Initialisation du champ avant chaque test</div>
<div>void</div> <div><a href="#">tearDown()</a></div> <div>Suppression de ce champ après chaque test</div>
<div>void</div> <div><a href="#">testCreerStyleAgregation()</a></div> <div>Test unitaire qui vérifie la création du style des liens d'agréation Vérifie que : - le style est créé - la forme est celle d'un lien d'agréation - l'opacite est correcte - le déplacement du lien est impossible - la couleur du lien est correcte</div>
<div>void</div> <div><a href="#">testCreerStyleAssociation()</a></div> <div>Test unitaire qui vérifie la création du style des liens d'association Vérifie que : - le style est créé - la forme est celle d'un lien d'association - l'opacite est correcte - le déplacement du lien est impossible - la couleur du lien est correcte</div>
<div>void</div> <div><a href="#">testCreerStyleComposition()</a></div> <div>Test unitaire qui vérifie la création du style des liens de composition Vérifie que : - le style est créé - la forme est celle d'un lien de composition - l'opacite est correcte - le déplacement du lien est</div>

Method Detail

setUp

public void setUp()

Initialisation du champ avant chaque test

tearDown

public void tearDown()

Suppression de ce champ après chaque test

testFail

public void testFail()

Les tests unitaires ne sont pas encore implémentés pour cette classe

	impossible - la couleur du lien est correcte
void	<a href="#">testCreerStyleFleche()</a> <div>Test unitaire qui vérifie la création du style des flèches Vérifie que : - le style est créé - la forme est celle d'une flèche - l'opacite est correcte - le déplacement du lien est impossible - la couleur du lien est correcte</div>
void	<a href="#">testCreerStyleSpecialisation()</a> <div>Test unitaire qui vérifie la création du style des liens de spécialisation Vérifie que : - le style est créé - la forme est celle d'un lien de spécialisation - l'opacite est correcte - le déplacement du lien est impossible - la couleur du lien est correcte</div>

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

LienTest

public LienTest()

Method Detail

setUp

public void setUp()

Initialisation du champ avant chaque test

tearDown

public void tearDown()

Suppression de ce champ après chaque test

testCreerStyleComposition

public void testCreerStyleComposition()

Test unitaire qui vérifie la création du style des liens de composition  
Vérifie que : - le style est créé - la forme est celle d'un lien de composition  
- l'opacite est correcte - le déplacement du lien est impossible - la couleur du lien est correcte

testCreerStyleAgregation

public void testCreerStyleAgregation()

Test unitaire qui vérifie la création du style des liens d'agrégation Vérifie que : - le style est créé - la forme est celle d'un lien d'agrégation - l'opacite est correcte - le déplacement du lien est impossible - la couleur du lien est correcte

testCreerStyleAssociation

public void testCreerStyleAssociation()

Test unitaire qui vérifie la création du style des liens d'association Vérifie que : - le style est créé - la forme est celle d'un lien d'association - l'opacite est correcte - le déplacement du lien est impossible - la couleur du lien est correcte

testCreerStyleSpecialisation

public void testCreerStyleSpecialisation()

Test unitaire qui vérifie la création du style des liens de spécialisation Vérifie que : - le style est créé - la forme est celle d'un lien de spécialisation - l'opacite est correcte - le déplacement du lien est impossible - la couleur du lien est correcte

testCreerStyleFleche

public void testCreerStyleFleche()

Test unitaire qui vérifie la création du style des flèches Vérifie que : - le style est créé - la forme est celle d'une flèche - l'opacite est correcte - le déplacement du lien est impossible - la couleur du lien est correcte

	paramètres de la méthode sans passage de l'index Vérifie que : - le nouveau paramètre est dans la liste des paramètres de la méthode
void	<a href="#">testAjouterParametreSansIndexTailleListe()</a> Test unitaire qui teste que la taille est incrémentée de 1 après ajout d'un paramètre sans index Vérifie que : - la nouvelle taille est égale à l'ancienne + 1
void	<a href="#">testConstante()</a> Test unitaire qui vérifie la modification de l'état constant d'une méthode Vérifie que : - la classe est passée constante
void	<a href="#">testDeClasse()</a> Test unitaire qui vérifie la modification de l'état de classe d'une méthode Vérifie que : - la classe est passée de classe
void	<a href="#">testInstanceParametre()</a> Test unitaire qui vérifie l'instanciation de la liste des paramètres d'une méthode Vérifie que : - la liste des paramètres est une instance de Liste
void	<a href="#">testNom()</a> Test unitaire qui teste la modification du nom de la méthode Vérifie que : - le texte est assigné correctement
void	<a href="#">testParametres()</a> Test unitaire qui vérifie la modification de la liste des paramètres de la méthode Vérifie que : - la liste des méthodes est assignée correctement
void	<a href="#">testParametresTailleListe()</a> Test unitaire qui vérifie la modification de la taille de la liste des paramètres de la classe lors de l'ajout d'un paramètre à la méthode Vérifie que : - la nouvelle taille est égale à l'ancienne + 1
void	<a href="#">testToString()</a> Test unitaire qui teste le retour de la méthode toString Vérifie que : - la chaîne retournée est correcte
void	<a href="#">testTypeDeRetour()</a> Test unitaire qui vérifie la modification du type de retour de la méthode Vérifie que : - le nouveau type de retour est assigné correctement
void	<a href="#">testVisibilite()</a> Test unitaire qui vérifie la modification de la visibilite de la méthode Vérifie que : - la nouvelle visibilite est assignée correctement

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

eltGraphique

Class MethodeTest

java.lang.Object  
↳ eltGraphique.MethodeTest

public class MethodeTest  
extends java.lang.Object

Cas de test JUnit regroupant les tests unitaires de chaque méthode de la classe Methode

See Also:

Methode

Constructor Summary

[MethodeTest\(\)](#)

Method Summary

void	<a href="#">setUp()</a> Initialisation du champ avant chaque test
void	<a href="#">tearDown()</a> Suppression de ce champ après chaque test
void	<a href="#">testAbstraite()</a> Test unitaire qui vérifie la modification de l'état abstrait d'une méthode Vérifie que : - la classe est passée abstraite
void	<a href="#">testAjouterParametreAvecIndex()</a> Test unitaire qui vérifie l'ajout d'un paramètre à la liste des paramètres de la méthode avec passage de l'index Vérifie que : - le nouveau paramètre est dans la liste des paramètres de la méthode
void	<a href="#">testAjouterParametreAvecIndexTailleListe()</a> Test unitaire qui teste que la taille est incrémentée de 1 après ajout d'un paramètre avec index Vérifie que : - la nouvelle taille est égale à l'ancienne + 1
void	<a href="#">testAjouterParametreSansIndex()</a> Test unitaire qui vérifie l'ajout d'un paramètre à la liste des

Constructor Detail

MethodeTest

public MethodeTest()

Method Detail

setUp

public void setUp()

Initialisation du champ avant chaque test

tearDown

public void tearDown()

Suppression de ce champ après chaque test

testAjouterParametreSansIndex

public void testAjouterParametreSansIndex()

Test unitaire qui vérifie l'ajout d'un paramètre à la liste des paramètres de la méthode sans passage de l'index Vérifie que : - le nouveau paramètre est dans la liste des paramètres de la méthode

testAjouterParametreSansIndexTailleListe

public void testAjouterParametreSansIndexTailleListe()

Test unitaire qui teste que la taille est incrémentée de 1 après ajout d'un paramètre sans index Vérifie que : - la nouvelle taille est égale à l'ancienne + 1

testAjouterParametreAvecIndex

```
public void testAjouterParametreAvecIndex()

    Test unitaire qui vérifie l'ajout d'un paramètre à la liste des paramètres de la méthode avec passage de l'index Vérifie que : - le nouveau paramètre est dans la liste des paramètres de la méthode
```

testAjouterParametreAvecIndexTailleListe

```
public void testAjouterParametreAvecIndexTailleListe()

    Test unitaire qui teste que la taille est incrémentée de 1 après ajout d'un paramètre avec index Vérifie que : - la nouvelle taille est égale à l'ancienne + 1
```

testAbstraite

```
public void testAbstraite()

    Test unitaire qui vérifie la modification de l'état abstrait d'une méthode Vérifie que : - la classe est passée abstraite
```

testConstante

```
public void testConstante()

    Test unitaire qui vérifie la modification de l'état constant d'une méthode Vérifie que : - la classe est passée constante
```

testDeClasse

```
public void testDeClasse()

    Test unitaire qui vérifie la modification de l'état de classe d'une méthode Vérifie que : - la classe est passée de classe
```

testNom

```
public void testNom()

    Test unitaire qui teste la modification du nom de la méthode Vérifie que : - le texte est assigné correctement
```

```
Test unitaire qui teste le retour de la méthode toString Vérifie que : - la chaîne retournée est correcte
```

testInstanceParametre

```
public void testInstanceParametre()

    Test unitaire qui vérifie l'instanciation de la liste des paramètres d'une méthode Vérifie que : - la liste des paramètres est une instance de Liste
```

testParametres

```
public void testParametres()

    Test unitaire qui vérifie la modification de la liste des paramètres de la méthode Vérifie que : - la liste des méthodes est assignée correctement
```

testParametresTailleListe

```
public void testParametresTailleListe()

    Test unitaire qui vérifie la modification de la taille de la liste des paramètres de la classe lors de l'ajout d'un paramètre à la méthode Vérifie que : - la nouvelle taille est égale à l'ancienne + 1
```

testVisibilite

```
public void testVisibilite()

    Test unitaire qui vérifie la modification de la visibilite de la méthode Vérifie que : - la nouvelle visibilite est assignée correctement
```

testTypeDeRetour

```
public void testTypeDeRetour()

    Test unitaire qui vérifie la modification du type de retour de la méthode Vérifie que : - le nouveau type de retour est assigné correctement
```

testToString

```
public void testToString()
```

eltGraphique  
Class TraitementTest

```
java.lang.Object
└─eltGraphique.TraitementTest
```

```
public class TraitementTest
extends java.lang.Object
```

Cas de test JUnit regroupant les tests unitaires de chaque méthode de la classe Traitement

See Also:  
Traitement

Constructor Summary

[TraitementTest\(\)](#)

Method Summary

void	<a href="#">setUp()</a>	Initialisation du champ avant chaque test
void	<a href="#">tearDown()</a>	Suppression de ce champ après chaque test
void	<a href="#">testCreerStyle()</a>	Test unitaire qui teste la création du style du traitement en vérifiant chaque champ du nouveau style Vérifie que : - le style est créé - la forme est celle d'un traitement - l'inclusion d'un autre élément soit impossible - l'opacite est correcte - la couleur du texte est correcte - la couleur de la bordure est correcte
void	<a href="#">testCreerStyleFleche()</a>	Test unitaire qui teste la création du style de la flèche de début de séquence en vérifiant chaque champ du nouveau style Vérifie que : - le style est créé - la forme est celle d'un traitement - le déplacement de la flèche soit impossible - le redimensionnement de la flèche soit impossible - la suppression de la flèche soit impossible - l'opacite est correcte - la direction est correcte - la'alignement du texte est à gauche - la couleur du texte est correcte - la couleur de fond est correcte - la

	couleur de la bordure est correcte
void	<a href="#">testDebutSequence()</a> Test unitaire qui teste la modification du champ de début de séquence du traitement Vérifie que : - l'état initial du traitement est faux (pas début de séquence) - la modification de l'état à vrai est effective
void	<a href="#">testEvenementDeclencheur()</a> Test unitaire qui test la modification de l'élément déclencheur du traitement Vérifie que : - le nouveau traitement est assigné correctement

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### TraitementTest

public **TraitementTest**()

## Method Detail

### setUp

public void **setUp**()

Initialisation du champ avant chaque test

### tearDown

public void **tearDown**()

Suppression de ce champ après chaque test

### testCreerStyle

public void **testCreerStyle**()

Test unitaire qui teste la création du style du traitement en vérifiant chaque champ du nouveau style Vérifie que : - le style est créé - la forme

est celle d'un traitement - l'inclusion d'un autre élément soit impossible - l'opacite est correcte - la couleur du texte est correcte - la couleur de la bordure est correcte

### testCreerStyleFleche

public void **testCreerStyleFleche**()

Test unitaire qui teste la création du style de la flèche de début de séquence en vérifiant chaque champ du nouveau style Vérifie que : - le style est créé - la forme est celle d'un traitement - le déplacement de la flèche soit impossible - le redimensionnement de la flèche soit impossible - la suppression de la flèche soit impossible - l'opacite est correcte - la direction est correcte - la'aligement du texte est à gauche - la couleur du texte est correcte - la couleur de fond est correcte - la couleur de la bordure est correcte

### testEvenementDeclencheur

public void **testEvenementDeclencheur**()

Test unitaire qui test la modification de l'élément déclencheur du traitement Vérifie que : - le nouveau traitement est assigné correctement

### testDebutSequence

public void **testDebutSequence**()

Test unitaire qui teste la modification du champ de début de séquence du traitement Vérifie que : - l'état initial du traitement est faux (pas début de séquence) - la modification de l'état à vrai est effective