# Code Source

/\*

\* TestOutilSauvegarde.java 24 mai 2021

\* IUT info1 2020-2021, pas de copyright pas de droit d'auteur

\*/

**package** othello.tests;

**import** **static** org.junit.jupiter.api.Assertions.\*;

**import** org.junit.jupiter.api.AfterEach;

**import** org.junit.jupiter.api.BeforeEach;

**import** org.junit.jupiter.api.Test;

**import** java.util.Arrays;

**import** othello.joueur.\*;

**import** othello.Plateau;

**import** othello.caracteristiques.Couleur;

**import** othello.Partie;

**import** othello.outils.OutilSauvegarde;

/\*\*

\* Tests unitaires de la classe OutilSauvegarde

\* **@author** David Simonin

\*

\*/

**class** TestOutilSauvegarde {

/\*\*

\* Jeu de données contenant des joueurs

\*/

**public** **final** Joueur[] JOUEURS = {

**new** Joueur("moska", Couleur.***NOIR***),

**new** Joueur("Mausaille", Couleur.***BLANC***)

};

/\*\*

\* Jeu de donnée contenant un plateau

\*/

**public** **final** Plateau DAMIER = **new** Plateau();

/\*\*

\* Jeu de donnée contenant des

\*/

**public** **final** Partie PARTIE = **new** Partie(JOUEURS[0], JOUEURS[1], DAMIER);

**final** String[] NOMS = **new** String[]{ "null", "Save1", "TestSave1",

"TestSave2" };

/\*\*

\* **@throws** java.lang.Exception

\*/

@BeforeEach

**void** setUp() **throws** Exception {

}

/\*\*

\* **@throws** java.lang.Exception

\*/

@AfterEach

**void** tearDown() **throws** Exception {

}

/\*\*

\* Test method for {@link othello.outils.OutilSauvegarde

\* #sauvegarder(othello.Partie, java.lang.String)}.

\*/

@Test

**void** testSauvegarder() {

/\* Test avec un nom de fichier valide \*/

*assertTrue*(OutilSauvegarde.*sauvegarder*(PARTIE, "TestSave2")

&& OutilSauvegarde.*charger*("TestSave2").equals(PARTIE));

/\* Test avec des noms de fichiers invalides \*/

*assertFalse*(OutilSauvegarde.*sauvegarder*(PARTIE, ""));

*assertFalse*(OutilSauvegarde.*sauvegarder*(PARTIE, **null**));

}

/\*\*

\* Test method for {@link othello.outils.OutilSauvegarde

\* #sauvegardeExiste(java.lang.String)}.

\*/

@Test

**void** testSauvegardeExiste() {

/\* Test avec un nom de fichier valide \*/

*assertTrue*(OutilSauvegarde.*sauvegardeExiste*("TestSave1"));

/\* Test avec des noms de fichiers invalides \*/

*assertFalse*(OutilSauvegarde.*sauvegardeExiste*("Save1484848"));

*assertFalse*(OutilSauvegarde.*sauvegardeExiste*(**null**));

}

/\*\*

\* Test method for {@link othello.outils.OutilSauvegarde

\* #charger(java.lang.String)}.

\*/

@Test

**void** testCharger() {

/\* Test avec un nom de fichier valide \*/

*assertTrue*(PARTIE.equals(OutilSauvegarde.*charger*("TestSave1")));

/\* Test avec des noms de fichiers invalides \*/

*assertTrue*(OutilSauvegarde.*charger*("Save1484848") == **null**);

*assertTrue*(OutilSauvegarde.*charger*(**null**) == **null**); }

/\*\*

\* Test method for {@link othello.outils.OutilSauvegarde

\* #getSauvegardes()}.

\*/

@Test

**void** testGetSauvegardes() {

OutilSauvegarde.*getSauvegardes*().forEach(fichier -> {

**if**(Arrays.*stream*(NOMS).anyMatch(

nom -> fichier.getName().equals(nom))){

}**else**{

*fail*();

}

});

}

}

# Résultats

Une image contenant texte

Description générée automatiquement

# Explication des tests

1. sauvegarder

Dans ce test, j’ai d’abord testé que la sauvegarde s’effectuée correctement avec un assertTrue, dans lequel se trouve la méthode pour sauvegarder. Puis comme les seuls cas d’échec de la méthode sont quand on ne renseigne pas de nom de fichier ou que le nom est vide, il m’a seulement fallut tester ces possibilités-là.

1. sauvegardeExiste

Dans ce test-là, pour le test de validité, j’ai préalablement créé une sauvegarde spéciale pour ce test-là. J’ai donc ensuite vérifié que la méthode retrouvait bien le fichier de sauvegarde. Ensuite come les deux seuls cas d’échec sont un fichier qui n’existe pas ou un String null, je n’ai eu qu’a vérifié que la méthode échouait bien.

1. charger

Pour le test de la méthode charger, j’ai d’abord vérifié que la méthode marchait correctement en comparant la partie retournait par la méthode et celle situait dans le jeu de test, qui sont identique. Pour les tests d’échec, encore une fois, les cas d’échec sont les mêmes que ceux d’avant et je n’ai donc eu qu’a vérifié que la partie retournait était null.

1. getSauvegardes

Pour ce dernier test, j’ai référencé tous les fichiers de sauvegarde qui se trouvait actuellement dans notre dossier de sauvegarde, et je n’ai eu qu’à comparer les valeurs récupéré par la méthode avec celle que moi j’avais noté.