

ISEN

ALL IS DIGITAL!

TOULON



yncréa

CAHIER DE RECETTES

Projet Java-Mathématiques 2017-2018

Version: CDR

Auteurs : Marion Bouteille, Thomas Artru, Yanis Meziane.

ISEN Toulon - Yncrea

Maison du Numérique et de l'Innovation

Place Georges Pompidou - 83000 Toulon

Description du document

<u>Type</u>	<u>Version</u>		<u>Confidentialité</u>	
Cahier des recettes	2.0		Usage Externe	
<u>Redacteurs</u>	<u>Nom</u>	<u>Fonction</u>	<u>Date</u>	<u>Visa</u>
	Meziane Yanis	Membre du projet	18/12/2017	
	Artru Thomas	Membre du projet		
	Bouteille Marion	Chef de projet	18/12/2017	
<u>Vérificateur</u>	Bouteille Marion	Chef de Projet	18/12/2017	
<u>Approbateur</u>	Bouteille Marion	Chef de projet	18/12/2017	
<u>Destinataire</u>		<u>Fonction</u>	<u>Organisme</u>	
Public			ISEN	

Révisions du document

<u>Version</u>	<u>Date</u>	<u>Redacteur</u>	<u>Modifications</u>
1.0	18/07/2016	FMC	Mise en forme
2.0	18/12/2017	Yanis Meziane Thomas Artru	Création Remplissage
2.1	20/12/2017	Bouteille Marion	Correction Verification

Sommaire

Introduction	4
Exigences Fonctionnelles :	5
Exigences non-fonctionnelles	17

Introduction

Ce document permet de vérifier la réponse du logiciel fourni aux exigences spécifiées dans le CDC¹. Chaque test est nommé en référence à l'exigence qu'il permet de valider, ainsi, par exemple, TEST_EF_0001_I permet de valider l'exigence EF_0001_I spécifiée dans le CDC.

Chaque test est muni d'une description permettant de savoir ce qui est attendu lors de sa validation et d'une démarche permettant de tester le comportement du logiciel et de vérifier ainsi que le comportement observé correspond au comportement attendu.

En cas de non-validation du test, il est alors nécessaire de remplir une fiche d'anomalie afin que le logiciel évolue pour corriger l'anomalie signalée.

¹ CDC : Cahier des charges

Exigences Fonctionnelles :

Fiche de Test Fonctionnel : TEST_EF_002_I	
Objectif :	Vérifier le bon fonctionnement de l'addition de nombres complexes
Exigence à tester :	EF_0002_I
Pré-requis :	Implémenter JUnit pour faire des tests unitaires
Initialisation :	Exécution du test unitaire

Etapas du test :				
Id	Démarche	Données	Comportement attendu	OK?
1	Lancer l'exécution du test avec JUnit	Deux nombres complexes	Additions des deux nombres complexes	ok
Rapport de test		Testé par :		Le :
Fonctionnalité		Conformité		Ergonomie
<div><input type="checkbox"/> Excellente</div> <div><input type="checkbox"/> Bonne</div> <div><input type="checkbox"/> Moyenne</div> <div><input type="checkbox"/> Faible</div>		<div><input type="checkbox"/> Excellente</div> <div><input type="checkbox"/> Bonne</div> <div><input type="checkbox"/> Moyenne</div> <div><input type="checkbox"/> Faible</div>		<div><input type="checkbox"/> Excellente</div> <div><input type="checkbox"/> Bonne</div> <div><input type="checkbox"/> Moyenne</div> <div><input type="checkbox"/> Faible</div>
Commentaire :				
Fiches d'anomalies émises :				
Approbation :				

Fiche de Test Fonctionnel : TEST_EF_003_I	
Objectif :	Vérifier le bon fonctionnement la soustraction de nombres complexes
Exigence à tester :	EF_0002_I
Pré-requis :	Implémenter JUnit pour faire des tests unitaires
Initialisation :	Exécution du test unitaire

Etapas du test :				
Id	Démarche	Données	Comportement attendu	OK?
1	Lancer l'exécution du test avec JUnit	Deux nombres complexes	Soustraction des deux nombres complexes	ok
Rapport de test		Testé par :	Le :	
Fonctionnalité	Conformité	Ergonomie		
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		
Commentaire : 				
Fiches d'anomalies émises : 				
Approbation : 				

Fiche de Test Fonctionnel : TEST_EF_004_I	
Objectif :	Vérifier le bon fonctionnement de la multiplication de nombres complexes
Exigence à tester :	EF_0003_I
Pré-requis :	Implémenter JUnit pour faire des tests unitaires
Initialisation :	Exécution du test unitaire

Etapas du test :				
Id	Démarche	Données	Comportement attendu	OK?
1	Lancer l'exécution du test avec JUnit	Deux nombres complexes	Multiplication des deux nombres complexes	ok
Rapport de test		Testé par :	Le :	
Fonctionnalité	Conformité	Ergonomie		
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		
Commentaire :				
Fiches d'anomalies émises :				
Approbation :				

Fiche de Test Fonctionnel : TEST_EF_005_I	
Objectif :	Vérifier que les nombres complexes se comportent bien comme des nombres
Exigence à tester :	EF_0005_I
Pré-requis :	Implémenter JUnit pour faire des tests unitaires
Initialisation :	Exécution du test unitaire

Etapas du test :				
Id	Démarche	Données	Comportement attendu	OK?
1	Lancer l'exécution du test avec JUnit	Tableau de nombre complexes	Le programme retourne bien un tableau de Nombres Complexes.	ok
Rapport de test		Testé par :	Le :	
Fonctionnalité		Conformité	Ergonomie	
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	
Commentaire : 				
Fiches d'anomalies émises : 				
Approbation : 				

Fiche de Test Fonctionnel : TEST_EF_006_I	
Objectif :	Vérifier que La FFT est correcte et cohérente avec des nombres réels
Exigence à tester :	EF_0006_I, EF_0007_I
Pré-requis :	Implémenter JUnit pour faire des tests unitaires
Initialisation :	Exécution du test unitaire

Etapas du test :				
Id	Démarche	Données	Comportement attendu	OK?
1	Lancer l'exécution du test avec JUnit	Signal constant	Le programme retourne bien un résultat cohérent mathématiquement.	ok
2	Lancer l'exécution du test avec JUnit	Pic de dirac	Le programme retourne bien un résultat cohérent mathématiquement	ok
Rapport de test		Testé par :		Le :
Fonctionnalité		Conformité		Ergonomie
<div><input type="checkbox"/> Excellente</div> <div><input type="checkbox"/> Bonne</div> <div><input type="checkbox"/> Moyenne</div> <div><input type="checkbox"/> Faible</div>		<div><input type="checkbox"/> Excellente</div> <div><input type="checkbox"/> Bonne</div> <div><input type="checkbox"/> Moyenne</div> <div><input type="checkbox"/> Faible</div>		<div><input type="checkbox"/> Excellente</div> <div><input type="checkbox"/> Bonne</div> <div><input type="checkbox"/> Moyenne</div> <div><input type="checkbox"/> Faible</div>
Commentaire :				
Fiches d'anomalies émises :				
Approbation :				

Fiche de Test Fonctionnel : TEST_EF_0010_I	
Objectif :	Vérifier que la FFT est correcte et cohérente avec des nombres complexes.
Exigence à tester :	EF_0010_I, EF_0011_I
Pré-requis :	Implémenter JUnit pour faire des tests unitaires
Initialisation :	Exécution du test unitaire

Etapas du test :				
Id	Démarche	Données	Comportement attendu	OK?
1	Lancer l'exécution du test avec JUnit	Signal constant	Le programme retourne bien un résultat cohérent mathématiquement.	ok
2	Lancer l'exécution du test avec JUnit	Pic de dirac	Le programme retourne bien un résultat cohérent mathématiquement.	ok
3	Lancer l'exécution du test avec JUnit	Sinus	Le programme retourne bien un résultat cohérent mathématiquement.	ok
4	Lancer l'exécution du test avec JUnit	Cosinus	Le programme retourne bien un résultat cohérent mathématiquement.	ok
5	Lancer l'exécution du test avec JUnit	Exponentielle imaginaire	Le programme retourne bien un résultat cohérent mathématiquement.	ok
Rapport de test		Testé par :	Le :	
Fonctionnalité		Conformité	Ergonomie	
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	

Commentaire :
Fiches d'anomalies émises :
Approbation :

Fiche de Test Fonctionnel : TEST_EF_0012_I	
Objectif :	Vérifier le fonctionnement de la fonction IFTT avec des nombres complexes , ainsi que sa cohérence mathématique.
Exigence à tester :	EF_00012_I, EF_00013_I
Pré-requis :	Implémenter JUnit pour faire des tests unitaires
Initialisation :	Exécution du test unitaire

Etapas du test :				
Id	Démarche	Données	Comportement attendu	OK?
1	Lancer l'exécution du test avec JUnit	Signal constant	Le programme retourne bien un résultat cohérent mathématiquement.	ok
2	Lancer l'exécution du test avec JUnit	Pic de dirac	Le programme retourne bien un résultat cohérent mathématiquement.	ok
3	Lancer l'exécution du test avec JUnit	Sinus	Le programme retourne bien un résultat cohérent mathématiquement.	ok
4	Lancer l'exécution du test avec JUnit	Cosinus	Le programme retourne bien un résultat cohérent mathématiquement.	ok
Rapport de test		Testé par :		Le :
Fonctionnalité		Conformité		Ergonomie
<div><input type="checkbox"/> Excellente</div> <div><input type="checkbox"/> Bonne</div> <div><input type="checkbox"/> Moyenne</div> <div><input type="checkbox"/> Faible</div>		<div><input type="checkbox"/> Excellente</div> <div><input type="checkbox"/> Bonne</div> <div><input type="checkbox"/> Moyenne</div> <div><input type="checkbox"/> Faible</div>		<div><input type="checkbox"/> Excellente</div> <div><input type="checkbox"/> Bonne</div> <div><input type="checkbox"/> Moyenne</div> <div><input type="checkbox"/> Faible</div>
Commentaire :				

Fiches d'anomalies émises :
Approbation :

Fiche de Test Fonctionnel : TEST_EF_0008_I	
Objectif :	Vérifier que La iFFT est correcte et cohérente avec des nombres réels
Exigence à tester :	EF_0007_I, EF_0008_I
Pré-requis :	Implémenter JUnit pour faire des tests unitaires
Initialisation :	Exécution du test unitaire

Etapas du test :				
Id	Démarche	Données	Comportement attendu	OK?
1	Lancer l'exécution du test avec JUnit	Signal constant	Le programme retourne bien un résultat cohérent mathématiquement.	ok
2	Lancer l'exécution du test avec JUnit	Pic de dirac	Le programme retourne bien un résultat cohérent mathématiquement	ok
Rapport de test		Testé par :	Le :	
Fonctionnalité		Conformité	Ergonomie	
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	
Commentaire :				
Fiches d'anomalies émises :				
Approbation :				

Fiche de Test Fonctionnel : TEST_EF_0016_I	
Objectif :	Vérifier que le système peut lire un fichier csv.
Exigence à tester :	EF_0016_I, EF_0017_I
Pré-requis :	Implémenter JUnit pour faire des tests unitaires
Initialisation :	Exécution du test unitaire

Etapas du test :				
Id	Démarche	Données	Comportement attendu	OK?
1	Lancer l'exécution du test avec JUnit	Adresse du fichier à lire	Le programme ouvre le fichier	ok
Rapport de test		Testé par :	Le :	
Fonctionnalité	Conformité		Ergonomie	
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	
Commentaire :				
Fiches d'anomalies émises :				
Approbation :				

Fiche de Test Fonctionnel : TEST_EF_0018_I	
Objectif :	Vérifier que le système peut écrire dans un fichier csv.
Exigence à tester :	EF_0016_I, EF_0017_I
Pré-requis :	Implémenter JUnit pour faire des tests unitaires
Initialisation :	Exécution du test unitaire

Etapas du test :				
Id	Démarche	Données	Comportement attendu	OK?
1	Lancer l'exécution du test avec JUnit	Nom du fichier	Le programme enregistre les données dans un fichier csv.	ok
Rapport de test		Testé par :	Le :	
Fonctionnalité		Conformité	Ergonomie	
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	
Commentaire :				
Fiches d'anomalies émises :				
Approbation :				

Exigences non-fonctionnelles

Identifiant	Exigence	Description	OK ?
Test_EI_0010_I	EI_0010_I	Le logiciel marche via JAVA	
Test_EI_0020_I	EI_0020_I	Le code est en JAVA.	
Test_EI_0030_I	EI_0030_I	Les noms de variables respectent les normes de nommage.	
Test_EP_0010_I	EP_0010_I	Le logiciel ne consomme pas plus de 200 Mo de mémoire.	
Test_EP_0020_I	EP_0020_I	L'ordinateur ne réfléchit pas plus de 20 secondes.	
Test_EM_0010_I	EM_0010_I	Le logiciel est maintenu jusqu'à la fin de l'année scolaire.	