

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

RÉSUMÉ

Ce document contient un modèle de plan de test.

Le plan de test permet de tester l'ensemble des méthodes du projet Simulateur.

Il se décompose en trois parties distinctes pour les 3 packages :

- Action
- discreteBehaviorSimulator
- Timer.

Le plan de test a été réalisé dans un objectif pédagogique.

HISTORIQUE DU DOCUMENT :

Version	Modification	Date	Auteur
0.1	Version initiale	28/01/2022	
0.2	Conception des jeux de test du package Action	02/02/2022	Célien & Dorian
0.3	Conception des jeux de test du package discreteBehaviorSimulator	02/02/2022	Frédéric & Clément
0.4	Conception des jeux de test du package Timer	02/02/2022	Maverick & Zied
0.5	Modification de la structure du document	31/03/2022	Dorian
1	Validation du document	14/04/2022	Dorian

DIFFUSION

Organisme	Nom	Mode de Diffusion
Polytech	VERNIER Flavien	Via l'architecture du projet GitHub

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

1. Introduction

1.1 BUT

Ce plan de test pour le projet Simulation de comportement a les objectifs suivants :

Le projet est un simulateur de comportement : il effectue des actions sur des objets selon une file d'attente avec un timer qui représente le laps de temps avant d'appeler l'action suivante.

1.2 DOCUMENTS

La table ci-dessous indique les documents disponibles pour l'activité de test.

Document	Version	Disponible	Notes
Rapport JUnit	v1.0	GitHub	
Rapport de couverture	v1.0	GitHub	
Rapport SonarLint	v1.0	GitHub	

Pour retrouver plus d'informations sur le projet, visitez notre GitHub :

https://github.com/M4verickFr/info832_unit-test

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

2. La stratégie de test

2.1 LES OUTILS

La table ci-dessous indique les outils utilisés pour l'activité de test.

Outil	Version	Informations
JUnit		
SonarLint		
Rapport de couverture IntelliJiDEA		

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

3. LES RESSOURCES

Nous allons voir dans cette partie les différentes ressources du projet

3.1 LES RESSOURCES HUMAINES

Ressources humaines		
Personne	Rôle(s)	Responsabilité(s)
CAULLIREAU Dorian	Concepteur / Testeur / Développeur	Rapport JUnit Package Action Cohésion entre les tâches
MASSIT Clément	Concepteur / Testeur / Développeur	Package Simulator
PERROLLAZ Maverick	Concepteur / Testeur / Développeur	Package Timer Rapport de couverture Rapport SonarLint
KOEBERLÉ Célien	Concepteur / Testeur / Développeur	Package Action
GOBJI Zied	Concepteur / Testeur / Développeur	Package Timer
GIGUERE-MARCIL Frederic	Concepteur / Testeur / Développeur	Package Simulator Refactoring

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

4. LES LIVRABLES DE L'ACTIVITÉ DE TEST

Document	Version	Disponible	Notes
Rapport JUnit	v1.0	GitHub	
Rapport de couverture	v1.0	GitHub	
Rapport SonarLint	v1.0	GitHub	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

Package Action :

Jeu de Test DiscreteAction				
Identifiant du test	Entrée	Description	Résultat Attendu	Résultat Obtenu
DA1	DiscreteAction(o, m, timer)	Test création d'une action discrète avec m correspondant à une méthode de l'objet o et timer existant	Objet créer avec l'objet o stocké dans la variable objet, la méthode m de l'objet o stocké dans la variable method et timer stocké dans la variable timmer	
DA2	DiscretAction(o, m, timer)	Test création d'une action discrète avec m correspondant à aucune méthode de l'objet o et timer existant	Exception	
DA3	spendTime(4)	Test de la méthode spendTime avec lapsTime = 10 donc lapsTime > 4 > 0	La valeur de lapsTime devient 6 et le log : "[DA] operate spendTime on objectName : objectHashCode: old time 10 new time 6" a été ajoutée	
DA4	spendTime(4)	Test de la méthode spendTime avec lapsTime = null	La log : "[DA] operate spendTime on objectName : objectHashCode: old time null new time null" a été ajoutée	
DA5	DiscreteAction(o, m, timer) getMethod()	Test du getter de method d'un objet initialisé	Retourne la méthode m	
DA6	DiscreteAction(o, m, timer) getCurrentLapsTime()	Test du getter de lapsTime d'un objet initialisé	Retourne null	
DA7	DiscreteAction(o, m, timer)	Test du getter de objet	Retourne l'objet o	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

	getObject()			
DA8	compareTo(c)	Test de comparaison avec lapsTime courant null	Retourne 1	
DA9	compareTo(c)	Test de comparaison avec lapsTime de c null	Retourne -1	
DA10	compareTo(c)	Test de comparaison avec lapsTime courant > lapsTime de c	Retourne 1	
DA11	compareTo(c)	Test de comparaison avec lapsTime courant < lapsTime de c	Retourne -1	
DA12	compareTo(c)	Test de comparaison avec lapsTime courant = lapsTime de c	Retourne 0	
DA13	DiscreteAction(o, m, timer) toString()	Test du toString d'un objet initialisé	Retourne le string: "Object : oName Method : mName Stat. : timer delay : lapsTime"	
DA14	DiscreteAction(o, m, timer) next()	Test méthode next() avec objet initialisé	La variable lapsTime prend la valeur suivante du timer, crée une log et retourne l'objet courant	
DA15	DiscreteAction(o, m, timer) hasNext()	Test méthode hasNext() avec timer != null et timer a une prochaine valeur	Retourne True	
DA16	DiscreteAction(o, m, timer) hasNext()	Test méthode hasNext() avec timer != null et timer n'a pas de prochaine valeur	Retourne False	
DA17	DiscreteAction(o, m, timer)	Test méthode hasNext() avec	Retourne False	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

	hasNext()	timer == null		
DA18	spendTime(4)	Test de la méthode spendTime avec lapsTime = 3 donc 4 > lapsTime > 0	Exception : "t > lapsTime"	
DA19	spendTime(-4)	Test de la méthode spendTime avec temps < 0	Exception "t < 0"	

Jeu de Test DiscreteActionDependent				
Identifiant du test	Entrée	Description	Résultat Attendu	Résultat Obtenu
DAD1	DiscreteActionDependent(obj, mtd, timer)	Test de la création de l'objet DiscreteActionDependent	Objet créer avec l'objet DiscreteAction(obj, mtd, timer) stocké dans la variable baseAction,, un TreeSet de DiscreteAction créé dans dependentActions, l'itérateur du TreeSet dans it et l'action courante (baseAction) dans currentAction	
DAD2	addDependence(obj, mtd, timer)	Test ajout d'une action	ajout de la discreteAction(obj, mtd, timer) dans le TreeSet	
DAD3	nextMethod()	Test de la méthode nextMethod avec currentAction = baseAction	la variable it prend la valeur de l'itérateur du TreeSet et la currentAction prend la valeur de la prochaine iteration	
DAD4	nextMethod()	Test de la méthode nextMethod avec currentAction égale a la dernière des	la variable currentAction prend la valeur de baseAction	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

		actions de la liste		
DAD5	nextMethod()	Test de la méthode nextMethod avec tout autre cas	la variable currentAction prend la valeur de l'itération suivante	
DAD6	spendTime(4)	Test de la méthode spendTime : chaque DiscreteAction du TreeSet se voit utiliser la méthode elt.spendTime(4)	Retourne void	
DAD7	getMethod()	Test du getter de method d'un objet initialisé	Retourne la méthode de currentAction	
DAD8	getCurrentLapTime()	Test du getter de lapTime d'un objet initialisé	Retourne le lapTime de currentAction	
DAD9	getObject()	Test du getter de objet	Retourne l'objet de currentAction	
DAD10	compareTo(c)	Test de compareTo	Retourne la valeur du compareTo de currentAction avec c	
DAD11	isEmpty()	Test de isEmpty avec TreeSet vide	retourne True	
DAD12	hasNext()	Test de hasNext avec baseAction a une action suivante et dependentsActions non vide	Retourne True	
DAD13	hasNext()	Test de hasNext avec baseAction a une action suivante et dependentsActions vide	Retourne True	
DAD14	hasNext()	Test de hasNext avec baseAction n'a pas d'action suivante et	Retourne True	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

		depedentsActions non vide		
DAD15	hasNext()	Test de hasNext avec baseAction n'a pas de d'action suivante et depedentsActions vide	Retourne False	

DiscreteActionOnOffDependent.java

Jeu de Test DiscreteActionOnOffDependent				
Identifiant du test	Entrée	Description	Résultat Attendu	Résultat Obtenu
DAOOD1	DiscreteActionOnOffDependent(obj,on,timerOn,off,timerOff)	Test de déclaration de l'objet DiscreteActionOnOffDependent	Objet créé avec un objet DiscreteAction(obj,on,timerOn) stocké dans la variable onAction, un second objet similaire DiscreteAction(obj,off,timerOff) stocké dans la variable offAction. L'instanciation de offAction est également stocké dans current action et le currentLapsTime est instancié à 0.	
DAOOD2	DiscreteActionOnOffDependent(obj, on, datesOn, off, datesOff)	Test du deuxième constructeur avec la première date de datesOn antérieur à la première date de datesOff	Objet créé avec un objet DiscreteAction(obj,on,new DateTimer(datesOn)) stocké dans la variable onAction, un second objet similaire DiscreteAction(obj,on,new DateTimer(datesOn)) stocké dans la variable offAction	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

			currentAction prend la valeur de onAction	
DAOOD3	DiscreteActionOnOffDependent(obj, on, datesOn, off, datesOff)	Test du deuxième constructeur avec la première date de datesOn postérieur à la première date de datesOff	Objet créé avec un objet DiscreteAction(obj,on, new DateTimer(datesOn)) stocké dans la variable onAction, un second objet similaire DiscreteAction(obj,on, new DateTimer(datesOn)) stocké dans la variable offAction currentAction prend la valeur de offAction	
DAOOD4	DiscreteActionOnOffDependent(obj, on, datesOn, off, datesOff)	Test du deuxième constructeur avec la première date de datesOn identique à la première date de datesOff	Objet créé avec un objet DiscreteAction(obj,on, new DateTimer(datesOn)) stocké dans la variable onAction, un second objet similaire DiscreteAction(obj,on, new DateTimer(datesOn)) stocké dans la variable offAction currentAction prend la valeur de onAction	
DAOOD5	nextAction()	Test de la méthode avec un currentAction différent de onAction	Remplacement de la currentAction par l'action suivante de l'action on	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

DAOOD6	nextAction()	Test de la méthode avec un currentAction égale à onAction	Remplacement de la currentAction par l'action suivante de l'off Action, et on attribue lastOffDelay par le LapsTime de la nouvelle current action	
DAOOD7	spendTime(4)	Test de la méthode spendTime : l'action courante se voit tester sur l'action spendTime()	Retourne void	
DAOOD8	getMethod()	Test du Getter pour la récupération de la méthode de l'objet courant	Récupération de la méthode de l'objet courant	
DAOOD9	getCurrentLapsTime()	Test du Getter pour la récupération du laps time de l'objet courant	Récupération du laps time de l'objet courant	
DAOOD10	getObject()	Test du Getter pour récupérer l'objet depuis l'action courante	Retour l'objet de l'action courante	
DAOOD11	compareTo(c)	Test de compareTo	Retourne la valeur du compareTo de currentAction avec c	
DAOOD12	next()	Test de next lorsqu'il n'y a pas d'action future	Retourne l'action courante	
DAOOD13	next()	Test de next lorsqu'il y a une action future	Retourne l'action courante	
DAOOD14	hasNext()	Test de hasNext avec onAction et offAction ont une action suivante	Retourne True	
DAOOD15	hasNext()	Test de hasNext avec onAction n'a pas d'action	Retourne True	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

		suivante et offAction a une action suivante		
DAOOD16	hasNext()	Test de hasNext avec onAction a une action suivante et offAction n'a pas d'action suivante	Retourne True	
DAOOD17	hasNext()	Test de hasNext avec onAction et offAction n'ont pas d'action suivante	Retourne False	
DAOOD18	spendTime()	Test de spendTime avec t < lapstime	Lapstime est diminué de t	

Package discreteBehaviorSimulator :

Clock.java

ID	Jeu de Test Clock	
C2	Entrée	getInstanceWithoutAttributeNull()
	Description	Test de getInstance() avec un attribut instance différent de null
	Résultat	Retourne la valeur de l'instance de la clock

Jeu de Test Clock				
Identifiant du test	Entrée	Description	Résultat Attendu	Résultat Obtenu
C2	getInstanceWithoutAttributeNull()	Test de getInstance() avec un attribut instance différent de null	Retourne la valeur de l'instance de la clock	
C3	addObserver(observ	Test de	Ajoute un observer à la	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

	er)	addObserver avec l'ajout d'un observer à une liste vide	liste d'observers	
C4	removeObserver(observer)	Test de removeObserver avec la suppression d'un observer présent dans une liste	Supprime un observer à la liste d'observers	
C5	setVirtualWithParam True(true)	Test de setVirtual avec le paramètre à true	Assigne la valeur de virtuel à true	
C6	setVirtualWithParam False(false)	Test de setVirtual avec le paramètre à false	Assigne la valeur de virtuel à false	
C7	isVirtual()	Test de isVirtual	Retourne la valeur de virtual	
C8	setNextJump(currentTime + 10)	Test de setNextJump sur une liste contenant un observer	Assigne la valeur nextJump à currentTime + 10 secondes	
C9	increaseTimeEqualNextJump(4)	Test de increase où le paramètre time est égal a nextJump	Augmente le temps de chaque observers de 4	
C10	increaseTimeDifferentNextJump(5)	Test de increase où time est différent de nextJump	Exception	
C11	getTimeWithVirtualTrue()	Test de getTime lorsque la valeur de virtual est true	Retourne la valeur de l'attribut time	
C12	getTimeWithVirtualFalse()	Test de getTime lorsque la valeur de virtual est	Retourne la valeur du temps courant (Date().getTime())	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

		false		
C13	lockReadAccess()	Test lockReadAccess	Bloque la lecture	
C14	unlockReadAccess()	Test unlockReadAccess	Débloque la lecture	
C15	lockWriteAccess()	Test lockWriteAccess	Bloque l'écriture	
C16	unlockWriteAccess()	Test unlockWriteAccess	Débloque l'écriture	
C17	toString()	Test de toString	Retourne la valeur de l'attribut time en chaîne de caractères	
C18	ClockForTimeAttribute()	Test du constructeur (assignation de time)	La valeur 0 sera assignée à l'attribut time	
C19	ClockForNextJumpAttribute()	Test du constructeur (assignation de nextJump)	La valeur 0 sera assignée à l'attribut nextJump	
C20	ClockForLockAttribute()	Test du constructeur (assignation de lock)	Un ReentrantReadWriteLock sera instancié et assigné à l'attribut lock	
C21	ClockForVirtualAttribute()	Test du constructeur (assignation de virtual)	La valeur True sera attribuée à virtual	
C22	ClockForObserversAttribute()	Test du constructeur (assignation de observers)	Un HashSet<ClockObserver> vide sera instancié et assigné à l'attribut observers	

DiscreteActionSimulator.java

Jeu de Test DiscreteActionSimulator

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

Identifiant du test	Entrée	Description	Résultat Attendu	Résultat Obtenu
DAS1	DiscreteActionSimulatorForLoggerAttribute()	Test du constructeur (assignation du nom du logger)	"DAS"	
DAS2	DiscreteActionSimulatorForLevelAttribute()	Test du constructeur (assignation du "level" du logger)	"Level.ALL"	
DAS3	DiscreteActionSimulatorForSetUseParentHandlersAttribute()	Test du constructeur (assignation du "setUseParentHandlers" du logger)	"true"	
DAS4	DiscreteActionSimulatorForFileNameAttribute()	Test du constructeur (nom du fichier)	"DiscreteActionSimulator.log"	
DAS5	DiscreteActionSimulator()	Test du constructeur (assignation du "formatter" du logFile)	Le formatter est un LogFormatter	
DAS6	DiscreteActionSimulator()	Test du constructeur (assignation de logConsole)	Le logConsole est un ConsoleHandler	
DAS7	new DiscreteActionSimulator()	Test du constructeur (ajout du "logFile" dans la liste de handlers du logger)	Le logger contient le logFile dans ses handlers	
DAS8	DiscreteActionSimulator()	Test du constructeur (ajout du "logConsole" dans la liste de handlers du logger)	Le logger contient le logConsole dans ses handlers	
DAS9	DiscreteActionSimulator()	Test du constructeur (assignation de globalTime)	Le globalTime est un Clock	
DAS10	DiscreteActionSimulator()	Test du constructeur (assignation de t)	t est un Thread	
DAS11	setNbLoop(0)	Test de setNbLoop avec un nombre nul de nbLoop	nbLoop = 0	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

DAS12	setNbLoop(0)	Test de setNbLoop avec un nombre nul de nbLoop	step = -1	
DAS13	setNbLoop(1)	Test de setNbLoop avec un nombre positif de nbLoop	nbLoop = 1	
DAS14	setNbLoop(1)	Test de setNbLoop avec un nombre positif de nbLoop	step = 1	
DAS16	addAction(c)	Test de addAction avec c qui ne possède des actions	La liste d'action est vide	
DAS17	addAction(c)	Test de addAction (collection triée)	la collection est triée	
DAS18	nextLapsTime()	Test de nextLapsTime	Retourne le laps de temps avant la prochaine action	
DAS19	runAction()	test de runAction avec globalTime qui est différent de null	Retourne la valeur de currentAction.get CurrentLapsTime()	
DAS20	runAction()	test de runAction avec globalTime qui est null	Retourne la valeur de currentAction.get CurrentLapsTime()	
DAS21	updateTimes(0)	Test de updateTimes avec aucun temps à passer	La liste d'actions demeure identique	
DAS22	updateTimes(4)	Test de updateTimes lorsque la première action n'a pas de hasNext	La première action est enlevée de la liste d'actions	
DAS23	updateTimes(4)	Test de updateTimes lorsque la première action a un hasNext	La première action est déplacée à la fin de la liste d'actions	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

DAS24	run()	Test de run lorsque la liste d'action est vide	le thread est stoppé	
DAS25	run()	Test de run lorsque la liste d'action n'est pas vide	les actions se lancent	
DAS26	start()	Test de start	passse la variable running à true	
DAS27	start()	Test de start (lancement du thread)	le thread est bien lancé	
DAS28	stop()	Test de stop	la variable running passe à false	
DAS21	toString()	Test de toString avec une liste vide de actionsList	----- TestAuto: 0 -----	
DAS22	toString()	Test de toString avec une liste actionsList contenant 2 objets	----- TestAuto:2 ETC ETC ETC -----	
DAS23	getRunning()	Test de getRunning si this.running = true	true	
DAS24	getRunning()	Test de getRunning si this.running = false	false	

LogFormatter.java

Jeu de Test LogFormatter				
Identifiant du test	Entrée	Description	Résultat Attendu	Résultat Obtenu
LF1	format(rec)	Test de format	Retourne le message du buffer	
LF2	calcDate(0)	Test de calcDate avec 0 pour valeur	1970.01.01 00:00:00.00	
LF3	getHead(h)	Test de getHead	""	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

LF4	getTail(h)	Test de getTail	“”	
-----	------------	-----------------	----	--

—

Package timer :

DateTimer.java

Jeu de Test DateTimer				
Identifiant du test	Entrée	Description	Résultat Attendu	Résultat Obtenu
DT0	DateTimer(new TreeSet<>())	Test hasNext() d'un DateTimer avec un set d'entiers nuls.	False	
DT1	DateTimer(new ArrayList<>())	Test hasNext() d'un DateTimer avec une ArrayList vide	False	
DT2	DateTimer(List<Integer>)	Test de l'exception pointeur nul avec la liste d'entiers vide	NullPointerException	
DT3	DateTimer(Set<Integer>)	Test de l'exception pointeur nul avec le set d'entiers vide	NullPointerException	
DT4	dateTimerWithSet.hasNext()	Test hasNext() avec set d'entiers	True	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

		non vide		
DT5	dateTimerWithSet. hasNext()	Test hasNext() lorsqu'il n'y a plus d'entiers dans le DateTimer	False	
DT6	assertTrue(dateTi merWithList.hasN ext());	Test hasNext() avec liste non vide	True	
DT7	dateTimerWithList .hasNext()	Test hasNext() lorsqu'il n'y a plus d'entiers dans la liste	False	
DT8	emptyDates.next()	Test si on a un TreeSet vide dans notre DateTimer	NoSuchElementExcepti on	
DT9	emptyLapsTime.n ext()	Test avec une ArrayList vide dans notre DateTimer	NoSuchElementExcepti on	
DT10	dateTimerWithList .next()	Test de la méthode next() avec une liste non vide	True	
DT11	dateTimerWithList .next()	Test lorsqu'il n'y a plus d'élément dans la liste	NoSuchElementExcepti on	
DT12	assertEquals(1, dateTimerWithSet. next());	Test de la méthode next() avec un DateTimer construit avec un set non vide	True	
DT13	dateTimerWithSet. next()	Test de l'exception lorsqu'il n'y a plus d'élément dans le set	NoSuchElementExcepti on	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

MergedTimer.java

Jeu de Test MergedTimer				
Identifiant du test	Entrée	Description	Résultat Attendu	Résultat Obtenu
MT1	MergedTimer(null, null)	Test d'une Merge avec 2 timers null	NullPointerException	
MT2	MergedTimer(oneShotTimer1, null)	Test d'une Merge avec 2eme timer null	NullPointerException	
MT3	MergedTimer(null, oneShotTimer2)	Test d'une Merge avec 1er timer null	NullPointerException	
MT4	MergedTimer(oneShotTimer1, oneShotTimer2)	Test d'une Merge avec 2 timers OneShotTimer	MergedTimer(oneShotTimer1, oneShotTimer2)	
MT5	mergedTimer1.hasNext()	Test de hasNext() sur mergedTimer1	True	
MT6	assertEquals(3, mergedTimer1.next())	Test de la valeur du mergedTimer1	True	
MT7	oneShotAndInfinite.hasNext()	Test de notre timer "infini"	True	
MT8	assertEquals(13, oneShotAndInfinite.next())	Test de la valeur du Merger	True	
MT9	oneShotAndInfinite.hasNext()	Test de hasNext	False	
MT10	oneShotAndInfinite.next()	Test de la valeur suivante	True	

OneShotTimer.java

Jeu de Test OneShotTimer

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

Identifiant du test	Entrée	Description	Résultat Attendu	Résultat Obtenu
OST1	oneShotTimer1.hasNext()	Test de hasNext sur un timer qui n'a jamais été utilisé	True	
OST2	assertEquals(value, oneShotTimer1.next())	Test de next sur un timer qui n'a jamais été utilisé	True	
OST3	oneShotTimer1.hasNext()	Test de hasNext sur un timer qui a été utilisé une première fois	False	
OST4	oneShotTimer1.next()	Test de next sur un timer qui a été utilisé une première fois	Null	

PeriodicTimer.java

Jeu de Test PeriodicTimer				
Identifiant du test	Entrée	Description	Résultat Attendu	Résultat Obtenu
PT1	PeriodicTimer(1)	Test d'avoir la période avec un Periodic Timer de base avec 1 int	True	
PT2	PeriodicTimer(1,1)	Test d'avoir la période avec un Periodic Timer de base avec 2 int	True	
PT3	PeriodicTimer(1, randomTimer)	Test d'avoir la période avec 2 int 1 random timer	True	
PT4	PeriodicTimer(1, 1, randomTimer)	Test d'avoir la période avec 2 int 1 random timer	True	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

PT5	PeriodicTimer(nextValue)	Test de next()	True	
PT6	PeriodicTimer(periodValue, nextValue)	Test de next()	True	
PT7	PeriodicTimer(periodValue, nextValue, randomTimer)	Test de next avec un RandomTimer et une loi Uniforme	True	
PT8	PeriodicTimer(nextValue, randomTimer)	Test avec un RandomTimer et une loi Uniforme	True, True	
PT9	PeriodicTimer(1)	Test d'avoir une période à partir d'un Periodic Timer	True	
PT10	PeriodicTimer(1, 1)	Test d'avoir une période à partir d'un Periodic Timer	True	
PT11	PeriodicTimer(1, randomTimer)	Test d'avoir la periode avec un Periodic Timer	True	
PT12	PeriodicTimer(1, 1, randomTimer)	Test d'avoir la periode avec un Periodic Timer	True	

RandomTimer.java

Jeu de Test RandomTimer				
Identifiant du test	Entrée	Description	Résultat Attendu	Résultat Obtenu
RT1	RandomTimer(RandomDistribution.EXP, param)	Test du rate	"Not yet implemented"	
RT2	RandomTimer(RandomDistribution.POISSON, param)	Create a RandomTimer of a poisson law	True	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

RT3	RandomTimer(RandomDistribution.GAUSSIAN, limitInferior, limitSuperior)	Teste d'un Random Timer avec une loi Gaussienne	"Not yet implemented"	
RT4	RandomTimer(RandomDistribution.UNIFORM, limitInferior, limitSuperior)	Teste d'un Random Timer avec une loi Uniforme	"Not yet implemented"	
RT5	RandomTimer(RandomDistribution.EXP, param)	Test d'une moyenne exponentielle	True	
RT6	RandomTimer(RandomDistribution.POISSON, param)	Test d'une moyenne de Poisson	True	
RT7	RandomTimer(RandomDistribution.GAUSSIAN, limitInferior, limitSuperior)	Test d'une moyenne normale	True	
RT8	RandomTimer(RandomDistribution.UNIFORM, limitInferior, limitSuperior)	Test d'une moyenne Uniforme	True	
RT9	RandomTimer(RandomDistribution.UNIFORM, 1)	Test Exception avec distribution uniforme avec mauvais arguments	IncorrectDistributionException	
RT10	RandomTimer(RandomDistribution.EXP, -1, 1)	Test Exception avec distribution exponentielle avec mauvais arguments	IncorrectDistributionException	
RT11	RandomTimer(RandomDistribution.UNIFORM, 1, -1)	Test limite inférieure > limite supérieure loi Uniforme	"Exception to be thrown, but nothing was thrown."	
RT12	RandomTimer(RandomDistribution.EXP, 1, 1)	Test de EXP, on	"Not yet implemented"	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

	ndomDistribution. EXP, param)	simule 100000 expériences et on les compare à la moyenne (mean) avec un intervalle d'erreur de 1% On teste en même temps si la nextValue est entre inferiorLimit et superiorLimit		
RT13	RandomTimer(Ra ndomDistribution. POISSON, param)	Idem avec une distribution de Poisson	"Not yet implemented"	
RT14	RandomTimer(Ra ndomDistribution. GAUSSIAN, limitInferior, limitSuperior)	Idem avec une distribution normale	"Not yet implemented"	
RT15	RandomTimer(Ra ndomDistribution. UNIFORM, limitInferior,limitSu perior)	Idem avec une loi Uniforme	"Not yet implemented"	
RT16	RandomTimer(Ra ndomDistribution. EXP, param)	Test de la méthode ToString sur une loi exponentielle	"Not yet implemented"	
RT17	RandomTimer(Ra ndomDistribution. POISSON, param)	Test de la méthode ToString sur une loi de Poisson	True	
RT18	RandomTimer(Ra ndomDistribution. GAUSSIAN, limitInferior,limitSu perior)	Test de la méthode ToString sur une loi normale	"Not yet implemented"	
RT19	RandomTimer(Ra ndomDistribution. UNIFORM, limitInferior,limitSu perior)	Test de la méthode ToString sur une loi uniforme	"Not yet implemented"	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

TimeBoundedTimer.java

Jeu de Test TimeBoundedTimer				
Identifiant du test	Entrée	Description	Résultat Attendu	Résultat Obtenu
TBT1	OneShotTimer(1)	Test du OneShotTimer avec une next value plus petite que le temps de départ	True	
TBT2	TimeBoundedTimer(oneShotTimer1, 5)	Test de la next value avec un TimeBoundedTimer	True	
TBT3	TimeBoundedTimer(oneShotTimer1, 5)	Test de la méthode next	True	
TBT4	TimeBoundedTimer(oneShotTimer1, 5)	Test de hasNext	True	
TBT5	TimeBoundedTimer(oneShotTimer1, 5)	Test de l'exception si on utilise un OneShotTimer	NullPointerException	
TBT6	TimeBoundedTimer(dummyTimer, startTime)	Test de hasNext	True	
TBT7	TimeBoundedTimer(dummyTimer, startTime)	Test de hasNext une seconde fois, comme la valeur de temps est inférieure à Integer.MAX_VALUE, on doit avoir True	True	
TBT8	dummyTimer.next(), timeBoundedTime	Test obtention de la next value du petit timer	False	

Polytech	Projet Simulateur Plan de Test	Projet Simulateur
IDU		V 1.0

	r2.next()			
TBT9	TimeBoundedTimer(dummyTimer, startTime, stopTime)	Test de hasNext et de next	False, False	