

Nom du test	Scénario	Résultat attendu	Résultat obtenu	verdict
testHorlogeImpl	vérification de l'initialisation de la référence horlogeImpl	horlogeImpl non null	horlogeImpl non null	OK
testAddModuleObserver	Précondition: horlogeImpl est correctement initialisé moduleStub1 et moduleStub2 de type ModuleStub sont créés Donnée en entrée: l'appel opération tick Postcondition: moduleFunctionCall de moduleStub1 est true moduleFunctionCall de moduleStub2 est true	moduleFunction de moduleStub1 est true moduleFunctionCall de moduleStub2 est true	moduleFunction de moduleStub1 est true moduleFunctionCall de moduleStub2 est true	OK
testRemoveModuleObserver	Précondition: horlogeImpl est correctement initialisé moduleStub1 et moduleStub2 de type ModuleStub sont créés Donnée en entrée: l'appel function removeModule Observer pour supprimer moduleStub1, l'appel opération tick Postcondition: moduleFunctionCall de moduleStub1 est false moduleFunctionCall de moduleStub2 est true	moduleFunctionCall de moduleStub1 est false moduleFunctionCall de moduleStub2 est true	moduleFunctionCall de moduleStub1 est false	OK
testTick	Précondition: horlogeImpl est correctement initialisé moduleStub1 et moduleStub2 de type ModuleStub sont créés moduleFunctionCall de moduleStub1 et moduleStub2 sont false Donnée en entrée: l'appel function tick de horlogeImpl Postcondition: moduleFunctionCall de moduleStub1 et moduleStub2 sont true	moduleFunctionCall de moduleStub1 et moduleStub2 sont true	moduleFunctionCall de moduleStub1 et moduleStub2 sont true	OK
testStart	Précondition: horlogeImpl est correctement initialisé moduleStub1 et moduleStub2 de type ModuleStub sont créés Donnée en entrée: l'appel opération start Postcondition: moduleFunctionCall de moduleStub1 et moduleStub2 sont true	moduleFunctionCall de moduleStub1 et moduleStub2 sont true	moduleFunctionCall de moduleStub1 et moduleStub2 sont true	OK
testStop	Précondition: horlogeImpl est correctement initialisé l'appel opération start moduleFunctionCall de moduleStub1 et moduleStub2 sont true Donnée en entrée: l'appel opération stop défini moduleFunctionCall de moduleStub1 et moduleStub2 sont false Postcondition: moduleFunctionCall de moduleStub1 et moduleStub2 sont false	moduleFunctionCall de moduleStub1 et moduleStub2 sont false	moduleFunctionCall de moduleStub1 et moduleStub2 sont false	OK
testGetSampleRate	Précondition: horlogeImpl est correctement initialisé l'appel opération setSampleRate et définir la valeur 44100 Donnée en entrée: Aucun Postcondition: la valeur de sample rate est 44100	la valeur de sample rate est 44100	la valeur de sample rate est 44100	OK

testSetSampleRate	Précondition: horlogeImpl est correctement initialisé Donnée en entrée: l'appel opération setSampleRate et définir la valeur 44100 Postcondition: la valeur de sample rate est 44100	la valeur de sample rate est 44100	la valeur de sample rate est 44100	OK
précondition	Initialisation HorlogeImpl avec une instance horlogeImpl Initialisation ModuleStub avec deux instances moduleStub1 et moduleStub2 l'appel opération addModuleObserver avec la référence moduleStub1 pour horlogeImpl l'appel opération addModuleObserver avec la référence moduleStub2 pour horlogeImpl			
postcondition	l'appel opération removeModuleObserver avec la référence moduleStub1 pour horlogeImpl l'appel opération removeModuleObserver avec la référence moduleStub1 pour horlogeImpl libération de la référence horlogeImpl libération de la référence moduleStub1 libération de la référence moduleStub2			

Nom du test	Scénario	Résultat attendu	Résultat obtenu	Verdict
testInPortImpl	vérification de l'initialisation de la référence inPort	inPort not null	inPort not null	OK
testGetValue	Précondition: inPort est correctement initialisé moduleStub est correctement initialisé définir la valeur de inPort à 10 Donnée en entrée: Aucun Postcondition: la valeur de inPort est 10	la valeur de inPort est 10	la valeur de inPort est 10	OK
testSetValue	Précondition: inPort est correctement initialisé Donnée en entrée: l'appel opération setValue avec la valeur 5 Postcondition: la valeur de inPort est 5	la valeur de inPort est 5	la valeur de inPort est 5	OK
testIsEmpty	Précondition: inPort est correctement initialisé fonction isEmpty retourne true Donnée en entrée: l'appel opération setValue avec la valeur 5 Postcondition: fonction isEmpty retourne false	fonction isEmpty retourne false	fonction isEmpty retourne false	OK
testUpdate	Précondition: moduleStub est correctement initialisé la valeur de inPort est 5 Donnée en entrée: moduleStub appel l'opération moduleFunction Postcondition: la valeur de inPort est 10	la valeur de inPort est 10	la valeur de inPort est 10	OK
Pré conditions	Initialisation ModuleStub avec une instance moduleStub Initialisation InPortImpl avec une instance inPort l'appel opération setModule avec la référence moduleStub pour inPort l'appel opération addObserver avec la référence inPort pour moduleStub			OK
post conditions	libération de la référence inPort libération de la référence moduleStub			OK

Nom du test	Scénario	Résultat attendu	Résultat obtenu	Verdict
testOutPortImpl	vérification de l'initialisation de la référence outPort	outPort not null	outPort not null	OK
testGetValue	Précondition: outPort est correctement initialisé moduleStub est correctement initialisé définir la valeur de outPort à 10 Donnée en entrée: Aucun Postcondition: la valeur de outPort est 10	la valeur de outPort est 10	la valeur de outPort est 10	OK OK
testSetValue	Précondition: outPort est correctement initialisé Donnée en entrée: l'appel opération setValue pour outPort et définir la valeur 5 Postcondition: la valeur de outPort est 5	la valeur de outPort est 5	la valeur de outPort est 5	OK
testRemoveHead	Précondition: outPort est correctement initialisé Donnée en entrée: l'appel opération setValue pour outPort et définir la valeur 6 l'appel opération setValue pour outPort et définir la valeur 7 la valeur de outPort est 6 Donnée en entrée: l'appel opération removeHead pour outPort Postcondition: la valeur de outPort est 7	la valeur de outPort est 7	la valeur de outPort est 7	OK
testClear	Précondition: outPort est correctement initialisé l'appel opération setValue pour outPort et définir la valeur 6 l'appel opération setValue pour outPort et définir la valeur 7 confirmer la valeur de outPort est 6 Donnée en entrée: l'appel opération clear pour supprimer les valeurs de outPort l'appel opération setValue pour outPort et définir la valeur 8 Postcondition: la valeur de outPort est 8	la valeur de outPort est 8	la valeur de outPort est 8	OK
precondition	Initialisation ModuleStub avec une instance moduleStub Initialisation OutPortImpl avec une instance outPort l'appel opération setOutPort avec la référence outPort pour moduleStub			
postcondition	libération de la référence outPort libération de la référence moduleStub			

Nom du test	Scénario	Résultat attendu	Résultat obtenu	Verdict
testOut	vérification de l'initialisation de la référence out	message est non null	out est non null	OK
testGetInPorts	Précondition: out est correctement initialisé Donnée en entrée: Aucun Postcondition: l'appel opération getInPorts pour la référence out, la valeur retourne n'est pas null			OK
precondition	Initialisation Out avec une instance out			
postcondition	libération de la référence out			

Nom du test	Scénario	Résultat attendu	Résultat obtenu	Verdict
testWaveForm	Précondition: vco est correctement initialisé Donnée en entrée: l'appel opération setWaveForm avec le type de waveform SAW l'appel opération setBase pour définir la valeur de base à 8 l'appel opération setPitch pour définir la valeur de pitch à 1 l'appel opération setAtt pour défini la valeur de atténuateur à 1 l'appel opération moduleFunction pour la référence vco Postcondition: la valeur de inPortImpl est -32768 après l'appel opération moduleFunction pour la référence vco, la valeur de inPortImpl change à (327-32768)	la valeur de inPortImpl est -32768 après l'appel opération moduleFunction pour la référence vco, la valeur de inPortImpl change à (327-32768)	la valeur de inPortImpl est -32768 après l'appel opération moduleFunction pour la référence vco, la valeur de inPortImpl change à (327-32768)	OK
précondition	l'appel opération setSampleRate avec la valeur 44100 Initialisation VCO avec une instance vco Initialisation InPortImpl avec une instance inPortImpl l'appel opération addObserver avec la référence inPortImpl pour la référence vco			
postcondition	l'appel opération removeObserver avec la référence inPortImpl pour supprimer l'observateur inPortImpl pour vco libération la référence vco libération la référence inPortImpl			

Nom du test	Scénario	Résultat attendu	Résultat obtenu	Verdict
testWaveForm	Précondition: vco est correctement initialisé Donnée en entrée: l'appel opération setWaveForm avec le type de waveform SQUARE l'appel opération setBase pour définir la valeur de base à 8 l'appel opération setPitch pour définir la valeur de pitch à 1 l'appel opération setAtt pour défini la valeur de atténuateur à 1 l'appel opération moduleFunction pour la référence vco Postcondition: la valeur de inPortImpl est -32768 après l'appel opération moduleFunction pour la référence vco, la valeur de inPortImpl est (-32768)	la valeur de inPortImpl est -32768 après l'appel opération moduleFunction pour la référence vco, la valeur de inPortImpl est (-32768)	la valeur de inPortImpl est -32768 après l'appel opération moduleFunction pour la référence vco, la valeur de inPortImpl est (-32768)	OK
precondition	l'appel opération setSampleRate avec la valeur 44100 Initialisation VCO avec une instance vco Initialisation InPortImpl avec une instance inPortImpl l'appel opération addObserver avec la référence inPortImpl pour la référence vco			
postcondition	l'appel opération removeObserver avec la référence inPortImpl pour supprimer l'observateur inPortImpl pour vco libération la référence vco libération la référence inPortImpl			

Nom du test	Scénario	Résultat attendu	Résultat obtenu	Verdict
testWaveForm	Précondition: vco est correctement initialisé Donnée en entrée: l'appel opération setWaveForm avec le type de waveform TRIANGLE l'appel opération setBase pour définir la valeur de base à 8 l'appel opération setPitch pour définir la valeur de pitch à 1 l'appel opération setAtt pour défini la valeur de atténuateur à 1 l'appel opération moduleFunction pour la référence vco Postcondition: la valeur de inPortImpl est (32768-654) après l'appel opération moduleFunction pour la référence vco, la valeur de inPortImpl change à (32768-654*2)	la valeur de inPortImpl est (32768-654) après l'appel opération moduleFunction pour la référence vco, la valeur de inPortImpl change à (32768-654*2)	la valeur de inPortImpl est (32768-654) après l'appel opération moduleFunction pour la référence vco, la valeur de inPortImpl change à (32768-654*2)	OK
precondition	l'appel opération setSampleRate avec la valeur 44100 Initialisation VCO avec une instance vco Initialisation InPortImpl avec une instance inPortImpl l'appel opération addObserver avec la référence inPortImpl pour la référence vco			
postcondition	l'appel opération removeObserver avec la référence inPortImpl pour supprimer l'observateur inPortImpl pour vco libération la référence vco libération la référence inPortImpl			

Nom du test	Scénario	Résultat attendu	Résultat obtenu	Verdict
testVCO	vérification de l'initialisation de la référence vco	vco not null	vco not null	OK
testSetPitch	Précondition : vco est correctement initialisé Donnée en entrée: l'appel opération setPitch avec la valeur 2 Postcondition: la valeur de pitch est 2	la valeur de pitch est 2	la valeur de pitch est 2	OK
testGetPitch	Précondition : vco est correctement initialisé l'appel opération setPitch avec la valeur 2 Donnée en entrée: Aucun Postcondition: la valeur de pitch est 2	la valeur de pitch est 2	la valeur de pitch est 2	OK
testSetBase	Précondition : vco est correctement initialisé Donnée en entrée: l'appel opération setBase avec la valeur 16 Postcondition: la valeur de base est 16	la valeur de base est 16	la valeur de base est 16	OK
testGetBase	Précondition : vco est correctement initialisé l'appel opération setBase avec la valeur 16 Donnée en entrée: Aucun Postcondition: la valeur de base est 16	la valeur de base est 16	la valeur de base est 16	OK
testSetAtt	Précondition : vco est correctement initialisé Donnée en entrée: l'appel opération setAtt avec la valeur 50 Postcondition: la valeur de atténuateur est 50	la valeur de atténuateur est 50	la valeur de atténuateur est 50	OK
testGetAtt	Précondition : vco est correctement initialisé l'appel opération setAtt avec la valeur 50 Donnée en entrée: Aucun Postcondition: la valeur de atténuateur est 50	la valeur de atténuateur est 50	la valeur de atténuateur est 50	OK
testSetWaveForm	Précondition : vco est correctement initialisé Donnée en entrée: l'appel opération setWaveForm avec le type de WaveForm SAW Postcondition: l'appel opération getWaveForm pour vco et le résultat est WaveForm SAW	l'appel opération getWaveForm pour vco et le résultat est WaveForm SAW	l'appel opération getWaveForm pour vco et le résultat est WaveForm SAW	OK
testGetWaveForm	Précondition : vco est correctement initialisé l'appel opération setWaveForm avec le type de WaveForm SAW Donnée en entrée: Aucun Postcondition: l'appel opération getWaveForm pour vco et le résultat est WaveForm SAW	l'appel opération getWaveForm pour vco et le résultat est WaveForm SAW	l'appel opération getWaveForm pour vco et le résultat est WaveForm SAW	OK

testGetInPorts	Précondition : vco est correctement initialisé Donnée en entrée: Aucun Postcondition: l'appel opération getInPorts pour vco et le résultat n'est pas null	l'appel opération getInPorts pour vco et le résultat n'est pas null	l'appel opération getInPorts pour vco et le résultat n'est pas null	OK
testGetOutPorts	Précondition : vco est correctement initialisé Donnée en entrée: Aucun Postcondition: l'appel opération getOutPorts pour vco et le résultat n'est pas null	l'appel opération getOutPorts pour vco et le résultat n'est pas null	l'appel opération getOutPorts pour vco et le résultat n'est pas null	OK
testAddObserver	Précondition : vco est correctement initialisé Donnée en entrée: l'appel opération setWaveForm avec le type de WaveForm SAW l'appel opération setBase de vco avec la valeur 8 l'appel opération setPitch de vco avec la valeur 1 l'appel opération setAtt de vco avec la valeur 1 l'appel opération moduleFunction pour la référence vco Postcondition: l'appel opération isEmpty pour inPortImpl et le résultat est false	l'appel opération isEmpty pour inPortImpl et le résultat est false	l'appel opération isEmpty pour inPortImpl et le résultat est false	OK
testRemoveObserver	Précondition : vco est correctement initialisé Donnée en entrée: l'appel opération removeObserver avec la référence inPortImpl l'appel opération setWaveForm avec le type de WaveForm SAW l'appel opération setBase de vco avec la valeur 8 l'appel opération setPitch de vco avec la valeur 1 l'appel opération setAtt de vco avec la valeur 1 l'appel opération moduleFunction pour la référence vco Postcondition: l'appel opération isEmpty pour inPortImpl et le résultat est true	l'appel opération isEmpty pour inPortImpl et le résultat est true	l'appel opération isEmpty pour inPortImpl et le résultat est true	OK

testModuleFunction	<p>Précondition: vco est correctement initialisé</p> <p>Donnée en entrée: 1. l'appel opération setWaveForm avec le type de WaveForm SAW</p> <p>l'appel opération setBase de vco avec la valeur 8</p> <p>l'appel opération setPitch de vco avec la valeur 1</p> <p>l'appel opération setAtt de vco avec la valeur 1</p> <p>l'appel opération moduleFunction pour la référence vco</p> <p>2. l'appel opération setWaveForm avec le type de WaveForm TRIANGLE</p> <p>l'appel opération getValue pour la référence inPortImpl si inPortImpl n'est pas vide</p> <p>l'appel opération moduleFunction</p> <p>3. l'appel opération setWaveForm avec le type de WaveForm SQUARE</p> <p>l'appel opération getValue pour la référence inPortImpl si inPortImpl n'est pas vide</p> <p>l'appel opération moduleFunction</p> <p>Postcondition: 1. l'appel opération isEmpty pour inPortImpl et le résultat est false</p> <p>2. l'appel opération isEmpty pour inPortImpl et le résultat est false</p> <p>3. l'appel opération isEmpty pour inPortImpl et le résultat est false</p>	<p>1. l'appel opération isEmpty pour inPortImpl et le résultat est false</p> <p>2. l'appel opération isEmpty pour inPortImpl et le résultat est false</p> <p>3. l'appel opération isEmpty pour inPortImpl et le résultat est false</p>	<p>1. l'appel opération isEmpty pour inPortImpl et le résultat est false</p> <p>2. l'appel opération isEmpty pour inPortImpl et le résultat est false</p> <p>3. l'appel opération isEmpty pour inPortImpl et le résultat est false</p>	OK
testMasterNotify	<p>Précondition: vco est correctement initialisé</p> <p>Donnée en entrée: l'appel opération setWaveForm avec le type de WaveForm SAW</p> <p>l'appel opération setBase de vco avec la valeur 8</p> <p>l'appel opération setPitch de vco avec la valeur 1</p> <p>l'appel opération setAtt de vco avec la valeur 1</p> <p>l'appel opération moduleFunction pour la référence vco</p> <p>Postcondition: l'appel opération isEmpty et le résultat est false</p>	l'appel opération isEmpty et le résultat est false	l'appel opération isEmpty et le résultat est false	OK

testComputeFrequency	Précondition: vco est correctement initialisé l'appel opération setBase de vco avec la valeur 8 l'appel opération setPitch de vco avec la valeur 1 Donnée en entrée: Aucun Postcondition: l'appel opération computeFrequency pour vco et le résultat est 220	l'appel opération computeFrequency pour vco et le résultat est 220	l'appel opération computeFrequency pour vco et le résultat est 220	OK
précondition	Définir sampleRate de HorlogeImpl est 44100 Initialisation VCO avec une instance vco Initialisation InPortImpl avec une instance inPortImpl l'appel opération addObserver avec la référence inPortImpl pour la référence vco			
postcondition	l'appel opération removeObserver avec la référence inPortImpl pour vco Libération de la référence vco Libération de la référence inPortImpl			

Nom du test	Scénario	Résultat attendu	Résultat Obtenu	Verdict
testVCA	vérification de l'initialisation de la référence vca	vca not null	vca not null	OK
testMasterNotify	Précondition: vca est correctement initialisé Donnée en entrée: l'appel opération setAttVCA avec la valeur 1 pour vca l'appel opération moduleFunction pour la référence vca Postcondition: l'appel opération isEmpty et le résultat est false	l'appel opération isEmpty et le résultat est false	l'appel opération isEmpty et le résultat est false	OK
testAddObserver	Précondition : vca est correctement initialisé Donnée en entrée: l'appel opération setAttVCA avec la valeur 1 pour vca l'appel opération getInPorts de vca et définir la valeur 10 pour inPort "in" l'appel opération moduleFunction pour la référence vca Postcondition: l'appel opération isEmpty pour inPortImpl et le résultat est false	Postcondition : l'appel a l'opération getStartDate renvoie bien l'entier 310.	vérification que l'appel à l'opération getState puis à getName sur la référence renvoyé retourne bien la chaîne "CLOSED".	OK
testRemoveObserver	Précondition : vca est correctement initialisé Donnée en entrée: l'appel opération setAttVCA avec la valeur 1 pour vca l'appel opération moduleFunction pour vca Postcondition: l'appel opération isEmpty pour inPortImpl et le résultat est true	l'appel opération isEmpty pour inPortImpl et le résultat est true	l'appel opération isEmpty pour inPortImpl et le résultat est true	OK
testModuleFunction	Précondition: vca est correctement initialisé Donnée en entrée: 1. l'appel opération setAttVCA avec la valeur 1 pour vca l'appel opération getInPorts de vca et définir la valeur 10 pout inPort "in" l'appel opération moduleFunction 2. l'appel opération setAttVCA avec la valeur 2 pour vca l'appel opération getInPorts de vca et définir la valeur 10 pout inPort "in" l'appel opération moduleFunction 3. l'appel opération setAttVCA avec la valeur -2 pour vca l'appel opération getInPorts de vca et définir la valeur 10 pout inPort "in" l'appel opération moduleFunction Postcondition: 1. la valeur de inPortImpl est 10 2. la valeur de inPortImpl est 20 3. la valeur de inPortImpl est 5	1. la valeur de inPortImpl est 10 2. la valeur de inPortImpl est 20 3. la valeur de inPortImpl est 5	1. la valeur de inPortImpl est 10 2. la valeur de inPortImpl est 20 3. la valeur de inPortImpl est 5	OK
testSetAttVCA	Précondition : vca est correctement initialisé Donnée en entrée: l'appel opération setAttVCA avec la valeur 1 pour vca Postcondition: l'appel opération getAttVCA et le résultat est 1	l'appel opération getAttVCA et le résultat est 1	l'appel opération getAttVCA et le résultat est 1	OK
testGetAttVCA	Précondition : vca est correctement initialisé l'appel opération setAttVCA avec la valeur 1 Donnée en entrée: Aucun Postcondition: l'appel opération getAttVCA et le résultat est 1	l'appel opération getAttVCA et le résultat est 1	l'appel opération getAttVCA et le résultat est 1	OK
testGetInPorts	Précondition : vca est correctement initialisé Donnée en entrée: Aucun Postcondition: l'appel opération getInPorts pour vca et le résultat n'est pas null	l'appel opération getInPorts pour vca et le résultat n'est pas null	l'appel opération getInPorts pour vca et le résultat n'est pas null	OK
testGetOutPorts	Précondition : vca est correctement initialisé Donnée en entrée: Aucun Postcondition: l'appel opération getOutPorts pour vca et le résultat n'est pas null	l'appel opération getOutPorts pour vca et le résultat n'est pas null	l'appel opération getOutPorts pour vca et le résultat n'est pas null	OK

précondition	Initialisation VCA avec une instance vca Initialisation InPortImpl avec une instance inPortImpl l'appel opération addObserver avec la référence inPortImpl pour la référence vca			
postcondition	l'appel opération removeObserver avec la référence inPortImpl pour vca Libération de la référence vca Libération de la référence inPortImpl			