テスト設計

- 1. テスト計画
 - (ア) 概要 (範囲)
 - (イ) リスク分析
 - ・リスク分析結果 テスト計画は、この分析結果に基づいて策定する
 - (ウ) 実施するテストレベル
 - (エ) テスト対象機能・テスト非機能
 - ① テスト対象

テストの対象となるシステムを下記に示す

システム名		特記事項
	ScriptEnine を利用して、動	特になし
	的プログラム内の関数及び	
	メゾッドを実行	

② テスト対象外

テストの対象外となるシステムを下記に示す

(オ) アプローチ (テスト戦略)

全体レベル毎の戦略

・リスク分析に基づく効果的なテストの実施 事前に検討してリスク分析結果に基づき、重点的に確認すべき観点にリソースを投入する事で効果 的なテストを実施する。

(カ) 戦略に基づくテストレベルごとの確認観点・合否基準など 単体テスト

• 確認観点

実装した機能が仕様通りの動作を行っていることを、仕様書ベースで確認を行う

テストアーキテクチャ設計
 (ア) テストフレームモデリング



(イ) テストコンテナモデリング

テスト詳細

ID	大項目	中項目	小項目	入力値	期待結果
1	関数	パラメータテスト		無し	「test_paramX in」と表示される
2	関数	パラメータテスト	数値パラメータテスト	10	「test_paramN in
				20	iv: 10
				30.5	lv: 20
				40.5	fv: 30.5
					dv: 40.5」と表示される
3	関数	パラメータテスト	文字列パラメータテスト	id00006	「test_paramS in id00006 name00006 memo00006」
				name00006	と表示される
				memo00006	
4	関数	パラメータテスト	配列パラメータテスト	10 11 12 13	「test_paramAV in
				20 21 22 23	aiv
				30.5 31.5 32.5 33.5	index: 0
				40.5 41.5 42.5 43.5	data: 10
					index: 1
					data: 11
					index: 2
					data: 12
					index: 3
					data: 13
					alv
					index: 0
					data: 20
					index: 1
					data: 21
					index: 2
					data: 22

			T		
					index: 3
					data: 23
					afv
					index: 0
					data: 30.5
					index: 1
					data: 31.5
					index: 2
					data: 32.5
					index: 3
					data: 33.5
					adv
					index: 0
					data: 40.5
					index: 1
					data: 41.5
					index: 2
					data: 42.5
					index: 3
					data: 43.5」
					と表示される
5	関数	パラメータテスト	ハッシュパラメータテスト	{ key01: 10	「test_paramHV in
				,key02: 20	hv
				, key03: 30.5	index: key04
				, key04: 40.5	data: 40.5
				, key05:	index: key03
				"value05"	data: 30.5
				, key06: [10 11 12	index: key02
				13]	data: 20

1	1 05 [00 05	22	. 1 .1 .01
	, key07: [20 21	22	index: key01
	23]		data: 10
	, key08: [30.5 3	31.5	index: key08
	32.5 33.5]		
	, key09: [40.5 4	41.5	index: 0
	$42.5 \ 43.5$		data: 30.5
	,key10: {		index: 1
	keyX1: 10)	data: 31.5
	,keyX2: 20	C	index: 2
	,keyX3: 30	0.5	data: 32.5
	,keyX4: 40	0.5	index: 3
	}		data: 33.5
			index: key07
			index: 0
			data: 20
			index: 1
			data: 21
			index: 2
			data: 22
			index: 3
			data: 23
			index: key06
			index: 0
			data: 10
			index: 1
			IIIucv. 1

index: 2 data: 12 index: 3 data: 13 index: key05 data: value05 index: key09 index: 0 data: 40.5 index: 1 data: 41.5 index: 2 data: 42.5 index: 3 data: 42.5 index: 3 data: 43.5 index: 43.5 index: 40.5 index: key10 index: key10 index: key10 index: key10 index: key30 data: 40.5 index: key30 data: 30.5 index: key33 data: 30.5 index: key33 data: 30.5 index: key33 data: 30.5 index: key32 data: 20 index: key32	T T	
data: 12 index: 3 data: 13 index: key05 data: value05 index: key09 index: 0 data: 40.5 index: 1 data: 41.5 index: 2 data: 42.5 index: 3 data: 43.5 index: 8ey10 index: key10 index: keyX4 data: 40.5 index: keyX4 data: 40.5 index: keyX4 data: 40.5 index: keyX4 data: 30.5 index: keyX3 data: 30.5 index: keyX3 data: 20 index: keyX1		data: 11
index: 3 data: 13 index: key05 data: value05 index: key09 index: 0 data: 40.5 index: 1 data: 41.5 index: 2 data: 42.5 index: 3 data: 43.5 index: 3 data: 43.5 index: key10 index: keyX4 data: 40.5 index: keyX2 data: 20 index: keyX2		
data: 13 index: key05 data: value05 index: key09 index: 0 data: 40.5 index: 1 data: 41.5 index: 2 data: 42.5 index: 3 data: 43.5 index: key10 index: keyX4 data: 40.5 index: keyX4 data: 40.5 index: keyX3 data: 40.5 index: keyX2 data: 20 index: keyX2 data: 20 index: keyX1		data: 12
index: key05 data: value05 index: key09 index: 0 data: 40.5 index: 1 data: 41.5 index: 2 data: 42.5 index: 3 data: 43.5 index: key10 index: key10 index: keyX4 data: 40.5 index: keyX4 data: 40.5 index: keyX2 data: 30.5 index: keyX2 data: 20 index: keyX2		index: 3
data: value05 index: key09 index: 0 data: 40.5 index: 1 data: 41.5 index: 2 data: 42.5 index: 3 data: 43.5 index: key10 index: keyX4 data: 40.5 index: keyX3 data: 30.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		data: 13
data: value05 index: key09 index: 0 data: 40.5 index: 1 data: 41.5 index: 2 data: 42.5 index: 3 data: 43.5 index: key10 index: keyX4 data: 40.5 index: keyX3 data: 30.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		
index: key09 index: 0 data: 40.5 index: 1 data: 41.5 index: 2 data: 42.5 index: 3 data: 43.5 index: key10 index: key44 data: 40.5 index: keyX4 data: 40.5 index: keyX4 data: 30.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		index: key05
index: 0 data: 40.5 index: 1 data: 41.5 index: 2 data: 42.5 index: 3 data: 43.5 index: key10 index: keyX4 data: 40.5 index: keyX4 data: 40.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		data: value05
index: 0 data: 40.5 index: 1 data: 41.5 index: 2 data: 42.5 index: 3 data: 43.5 index: key10 index: keyX4 data: 40.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		index: key09
data: 40.5 index: 1 data: 41.5 index: 2 data: 42.5 index: 3 data: 43.5 index: key10 index: keyX4 data: 40.5 index: keyX4 data: 40.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		
index: 1 data: 41.5 index: 2 data: 42.5 index: 3 data: 43.5 index: key10 index: keyX4 data: 40.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		index: 0
data: 41.5 index: 2 data: 42.5 index: 3 data: 43.5 index: key10 index: keyX4 data: 40.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		data: 40.5
index: 2 data: 42.5 index: 3 data: 43.5 index: key10 index: keyX4 data: 40.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		index: 1
data: 42.5 index: 3 data: 43.5 index: key10 index: keyX4 data: 40.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		data: 41.5
index: 3 data: 43.5 index: key10 index: keyX4 data: 40.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		index: 2
data: 43.5 index: key10		data: 42.5
index: key10 index: keyX4 data: 40.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		index: 3
index: keyX4 data: 40.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		data: 43.5
index: keyX4 data: 40.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		
index: keyX4 data: 40.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		index: key10
data: 40.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		
data: 40.5 index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		index: keyX4
index: keyX3 data: 30.5 index: keyX2 data: 20 index: keyX1		
data: 30.5 index: keyX2 data: 20 index: keyX1		
index: keyX2 data: 20 index: keyX1		
data: 20 index: keyX1		
index: keyX1		
data: 10」と表示される		data: 10」と表示される

6	関数	戻り値テスト	Int 型戻り値テスト	10	「test_returnIV in: 10
					value:20」と表示される
7	関数	戻り値テスト	Long 型戻り値テスト	20	「test_returnLV in: 20
					value:40」と表示される
8	関数	戻り値テスト	Float 型戻り値テスト	30.5	「test_returnFV in: 30.5
					value:61.000000」と表示される
9	関数	戻り値テスト	Double 型戻り値テスト	40.5	「test_returnDV in: 40.5
					value:81.000000」と表示される
10	関数	戻り値テスト	文字列戻り値テスト		「test_returnPV in
					value:Hello javaScript world!!」と表示される
11	関数	戻り値テスト	Int 型の配列戻り値テスト		「test_returnAIV in
					100 110 120 130 140」と表示される
12	関数	戻り値テスト	Long 型の配列戻り値テスト		「test_returnALV in
					200 210 220 230 240」と表示される
13	関数	戻り値テスト	Float 型の配列戻り値テスト		「test_returnAFV in
					$300.500000\ 310.500000\ 320.500000\ 330.500000\ 340.500000$
					と表示される
14	関数	戻り値テスト	Double 型の配列戻り値テス		「test_returnADV in
			F		400.800000 410.800000 420.800000 430.800000 440.800000]
					と表示される
15	関数	戻り値テスト	オブジェクト型配列戻り値		「test_returnAOV in
			テスト		500 500.300000 <hello javascript="" world!!=""> 600 610 620 630</hello>
					640
					$700.300000\ 710.300000\ 720.300000\ 730.300000\ 740.300000 \rfloor$
					と表示される
16	関数	戻り値テスト	ハッシュ型の戻り値テスト		「test_returnHV in
					key:key01 1000
					key:key02 4000.600000

					key:key03 <hello javascript="" world!!=""></hello>
					key:key04 100 110 120 130 140
					key:key05 400.800000 410.800000 420.800000 430.800000
					440.800000
					key:key06 key:keyX01 1100
					key:keyX02 4100.700000
					key:keyX03
					key:keyX04 200 210
					key:keyX05 500.500000 510.500000」と表示される
17	関数	戻り値テスト	ハッシュ型の配列戻り値テ		\[\text_returnAHV in \]
- '	12,132		スト		idx:0
					key:key01 1100
					key:key02 <value01></value01>
					idx:1
					key:key01 1100
					key:key02 <value01></value01>
					idx:2
					key:key01 1100
					key:key02 <value01></value01>
					idx:3
					key:key01 1100
					key:key02 <value01>」と表示される</value01>
18	Static	パラメータテスト		無し	「staticTest_paramX in」と表示される
	Method				
19	Static	パラメータテスト	数値パラメータテスト	10	「staticTest_paramN in
	Method			20	iv: 10
				30.5	lv: 20
				40.5	fv: 30.5

					dv: 40.5」と表示される
20	Static	パラメータテスト	文字列パラメータテスト	id00006	「staticTest_paramS in id00006 name00006 memo00006」
	Method			name00006	と表示される
				memo00006	
21	Static	パラメータテスト	配列パラメータテスト	10 11 12 13	「staticTest_paramAV in
	Method			20 21 22 23	aiv
				30.5 31.5 32.5 33.5	index: 0
				40.5 41.5 42.5 43.5	data: 10
					index: 1
					data: 11
					index: 2
					data: 12
					index: 3
					data: 13
					alv
					index: 0
					data: 20
					index: 1
					data: 21
					index: 2
					data: 22
					index: 3
					data: 23
					afv
					index: 0
					data: 30.5
					index: 1
					data: 31.5
					index: 2

					data: 32.5
					index: 3
					data: 33.5
					adv
					index: 0
					data: 40.5
					index: 1
					data: 41.5
					index: 2
					data: 42.5
					index: 3
					data: 43.5]
					と表示される
22	Static	パラメータテスト	ハッシュパラメータテスト	{ key01: 10	「staticTest_paramHV in
	Method			,key02: 20	hv
				, key03: 30.5	index: key04
				, key04: 40.5	data: 40.5
				, key05:	index: key03
				"value05"	data: 30.5
				, key06: [10 11 12	index: key02
				13]	data: 20
				, key07: [20 21 22	index: key01
				23]	data: 10
				, key08: [30.5 31.5	index: key08
				32.5 33.5]	
				, key09: [40.5 41.5	index: 0
				42.5 43.5]	data: 30.5
				,key10: {	index: 1
				keyX1: 10	data: 31.5

,keyX2: 20	index: 2
,keyX3: 30.5	data: 32.5
,keyX4: 40.5	index: 3
	data: 33.5
	index: key07
	index: 0
	data: 20
	index: 1
	data: 21
	index: 2
	data: 22
	index: 3
	data: 23
	index: key06
	index: 0
	data: 10
	index: 1
	data: 11
	index: 2
	data: 12
	index: 3
	data: 13
	uata. 19
	index. Irox02
	index: key05
	data: value05

					index: key09
					index: 0
					data: 40.5
					index: 1
					data: 41.5
					index: 2
					data: 42.5
					index: 3
					data: 43.5
					index: key10
					index: keyX4
					data: 40.5
					index: keyX3
					data: 30.5
					index: keyX2
					data: 20
					index: keyX1
					data: 10」と表示される
23	Static	戻り値テスト	Int 型戻り値テスト	10	「staticTest_returnIV in: 10
	Method				value:30」と表示される
24	Static	戻り値テスト	Long 型戻り値テスト	20	「staticTest_returnLV in: 20
	Method				value:60」と表示される
25	Static	戻り値テスト	Float 型戻り値テスト	30.5	「staticTest_returnFV in: 30.5
	Method				value:91.500000」と表示される
26	Static	戻り値テスト	Double 型戻り値テスト	40.5	「staticTest_returnDV in: 40.5

	Method			value:121.500000」と表示される
27	Static	戻り値テスト	文字列戻り値テスト	「staticTest_returnPV in
	Method			value:Hello javaScript world!!」と表示される
28	Static	戻り値テスト	Int 型の配列戻り値テスト	「staticTest_returnAIV in
	Method			100 110 120 130 140」と表示される
29	Static	戻り値テスト	Long 型の配列戻り値テスト	「staticTest_returnALV in
	Method			200 210 220 230 240」と表示される
30	Static	戻り値テスト	Float 型の配列戻り値テスト	「staticTest_returnAFV in
	Method			$300.500000\ 310.500000\ 320.500000\ 330.500000\ 340.500000$
				と表示される
31	Static	戻り値テスト	Double 型の配列戻り値テス	「staticTest_returnADV in
	Method		F	400.800000 410.800000 420.800000 430.800000 440.800000]
				と表示される
32	Static	戻り値テスト	オブジェクト型の配列戻り	「staticTest_returnAOV in
	Method		値テスト	500 500.300000 <hello javascript="" world!!=""> 600 610 620 630</hello>
				640
				700.300000 710.300000 720.300000 730.300000 740.300000]
				と表示される
33	Static	戻り値テスト	ハッシュ型の戻り値テスト	「staticTest_returnHV in
	Method			key:key01 1000
				key:key02 4000.600000
				key:key03 <hello javascript="" world!!=""></hello>
				key:key04 100 110 120 130 140
				key:key05 400.800000 410.800000 420.800000 430.800000
				440.800000
				key:key06 key:keyX01 1100
				key:keyX02 4100.700000
				key:keyX03

					key:keyX04 200 210
					key:keyX05 500.500000 510.500000
					と表示される
34	Static	戻り値テスト	ハッシュ型の配列戻り値テ		「staticTest_returnAHV in
	Method		スト		idx:0
					key:key01 1100
					key:key02 <value01></value01>
					idx:1
					key:key01 1100
					key:key02 <value01></value01>
					idx:2
					key:key01 1100
					key:key02 <value01></value01>
					idx:3
					key:key01 1100
					key:key02 <value01>」</value01>
					と表示される
35	ObjectMethod	パラメータテスト		無し	「objectTest_paramX in」
					と表示される
36	ObjectMethod	パラメータテスト	数値パラメータテスト	10	「objectTest_paramN in
				20	iv: 10
				30.5	lv: 20
				40.5	fv: 30.5
					dv: 40.5
					と表示される
37	ObjectMethod	パラメータテスト	文字列パラメータテスト	id00006	「objectTest_paramS in id00006 name00006 memo00006」
				name00006	と表示される
				memo00006	

38	ObjectMethod	パラメータテスト	配列パラメータテスト	10 11 12 13	「objectTest_paramAV in
				20 21 22 23	aiv
				30.5 31.5 32.5 33.5	index: 0
				40.5 41.5 42.5 43.5	data: 10
					index: 1
					data: 11
					index: 2
					data: 12
					index: 3
					data: 13
					alv
					index: 0
					data: 20
					index: 1
					data: 21
					index: 2
					data: 22
					index: 3
					data: 23
					afv
					index: 0
					data: 30.5
					index: 1
					data: 31.5
					index: 2
					data: 32.5
					index: 3
					data: 33.5
					adv

	1		T		
					index: 0
					data: 40.5
					index: 1
					data: 41.5
					index: 2
					data: 42.5
					index: 3
					data: 43.5]
					と表示される
39	ObjectMethod	パラメータテスト	ハッシュパラメータテスト	{ key01: 10	「objectTest_paramHV in
				,key02: 20	hv
				, key03: 30.5	index: key04
				, key04: 40.5	data: 40.5
				, key05:	index: key03
				"value05"	data: 30.5
				, key06: [10 11 12	index: key02
				13]	data: 20
				, key07: [20 21 22	index: key01
				23]	data: 10
				, key08: [30.5 31.5	index: key08
				32.5 33.5]	
				, key09: [40.5 41.5	index: 0
				42.5 43.5]	data: 30.5
				,key10: {	index: 1
				keyX1: 10	data: 31.5
				,keyX2: 20	index: 2
				,keyX3: 30.5	data: 32.5
				,keyX4: 40.5	index: 3
				}	data: 33.5

index: key07 index: 0 data: 20 index: 1 data: 21 index: 2	
index: 0 data: 20 index: 1 data: 21	
index: 0 data: 20 index: 1 data: 21	
data: 20 index: 1 data: 21	
index: 1 data: 21	
data: 21	
index: 2	
data: 22	
index: 3	
data: 23	
index: key06	
index: 0	
data: 10	
index: 1	
data: 11	
index: 2	
data: 12	
index: 3	
data: 13	
index: key05	
data: value05	
index: key09	
index: 0	
data: 40.5	

					index: 1
					data: 41.5
					index: 2
					data: 42.5
					index: 3
					data: 43.5
					10.0
					index: key10
					index: keyX4
					data: 40.5
					index: keyX3
					data: 30.5
					index: keyX2
					data: 20
					index: keyX1
					data: 10]
- 10			T . TUI = 10 /+	10	と表示される
40	ObjectMethod	戻り値テスト	Int 型戻り値テスト	10	\[\text{logical} \] \[
					value:30
					と表示される
41	ObjectMethod	戻り値テスト	Long 型戻り値テスト	20	「objectTest_returnLV in: 20
					value:60
					と表示される
42	ObjectMethod	戻り値テスト	Float 型戻り値テスト	30.5	objectTest_returnFV in: 30.5
					value:91.500000]
					と表示される
43	ObjectMethod	戻り値テスト	Double 型戻り値テスト	40.5	「objectTest_returnDV in: 40.5

				value:121.500000]
				と表示される
44	ObjectMethod	戻り値テスト	文字列戻り値テスト	「objectTest_returnPV in
				value:Hello javaScript world!!」
				と表示される
45	ObjectMethod	戻り値テスト	Int 型の配列戻り値テスト	「objectTest_returnAIV in
				100 110 120 130 140」
				と表示される
46	ObjectMethod	戻り値テスト	Long 型の配列戻り値テスト	「objectTest_returnALV in
				200 210 220 230 240]
				と表示される
47	ObjectMethod	戻り値テスト	Float 型の配列戻り値テスト	「objectTest_returnAFV in
				300.500000 310.500000 320.500000 330.500000 340.500000]
				と表示される
48	ObjectMethod	戻り値テスト	Double 型の配列戻り値テス	「objectTest_returnADV in
			F	400.800000 410.800000 420.800000 430.800000 440.800000]
				と表示される
49	ObjectMethod	戻り値テスト	オブジェクト型の配列戻り	「objectTest_returnAOV in
			値テスト	500 500.300000 <hello javascript="" world!!=""> 600 610 620 630</hello>
				640
				700.300000 710.300000 720.300000 730.300000 740.300000]
				と表示される
50	ObjectMethod	戻り値テスト	ハッシュ型の戻り値テスト	「objectTest_returnHV in
				key:key01 1000
				key:key02 4000.600000
				key:key03 <hello javascript="" world!!=""></hello>
				key:key04 100 110 120 130 140
				key:key05 400.800000 410.800000 420.800000 430.800000

	1	T	T	
				440.800000
				key:key06 key:keyX01 1100
				key:keyX02 4100.700000
				key:keyX03
				key:keyX04 200 210
				key:keyX05 500.500000 510.500000
				と表示される
51	ObjectMethod	戻り値テスト	ハッシュ型の配列戻り値テ	「objectTest_returnAHV in
			スト	idx:0
				key:key01 1100
				key:key02 <value01></value01>
				idx:1
				key:key01 1100
				key:key02 <value01></value01>
				idx:2
				key:key01 1100
				key:key02 <value01></value01>
				idx:3
				key:key01 1100
				key:key02 <value01>」</value01>
				と表示される