Bogdan Pasterak
15 February 2019

Test Data and expected results (JUnit Test Suite)

Test				Entred data		Expected answer
ID	*Name of test*	*Object for testing*	*JUnit method*	*Preparation*	*Value to compare*	*Program answer*
CustomerCreateTest (Junit Test Case)						
1	testCreateNewCustomer	Constructor of Customer class	assertNotNull		new Customer("Bogdan Pasterak", "087-1234-567", 5)	customer is not null
2	testCreateNewCustomer	Constructor of Customer class	assertEquals	customer from test 1.	1, customer	equals
3	testCreateNewCustomer	Customer/ setStartId	assertFalse		setStartId(10)	false (available before any customer is added)
4	testCreateNewCustomer	Constructor of Customer class	assertNotNull		new Customer("Jean Doe", "087-1111-555", 10)	customer2 is not null
5	testCreateNewCustomer	Constructor of Customer class	assertEquals	customer from test 4.	2, customer	equals (next available number)
6	testCreateNewCustomerFrom10	Customer/ setStartId	assertTrue		setStartId(10)	set starting ID
7	testCreateNewCustomerFrom10	Customer/ setStartId	assertFalse		setStartId(20)	false (available ones)
8	testCreateNewCustomerFrom10	Constructor of Customer class	assertNotNull		new Customer("Bogdan Pasterak", "087-1234-567", 5)	customer is not null
9	testCreateNewCustomerFrom10	Constructor of Customer class	assertEquals	customer from test 8.	10, customer	equals
10	testCreateNewCustomerFrom10	Constructor of Customer class	assertNotNull		new Customer("Jean Doe", "087-1111-555", 10)	customer is not null
11	testCreateNewCustomerFrom10	Constructor of Customer class	assertEquals	customer from test 10.	11, customer	equals
CustomerGetterSetterTest (Junit Test Case)						
	setUp	**Set customer object before all test**	**@ Before**		**new Customer("Bogdan Pasterak", "087-1234-567", 5)**	**customer : Customer with data**
12	testGetName	Customer/ getName	assertEquals	setUp	"Bogdan Pasterak" , customer.getName	equals
13	testGetPhone	Customer/ getPhone	assertEquals	setUp	"087 123 4567" , customer.getPhone	equals
14	testGetPaintCans	Customer/ getPaintCans	assertEquals	setUp	5 , customer.getPaintCans	equals
15	testSetName	Customer/ setName	assertEquals	setUp, customer.setName("Alice Someone")	"Alice Someone" , customer.getName	equals
15	testSetNameWithTab	Customer/ setName	assertEquals	setUp, customer.setName("Bob\tD'\tClark")	"Bob D' Clark" , customer.getName	equals
16	testGetPhone	Customer/ setPhone	assertEquals	setUp, customer.setPhone("00353-87-2222-333")	"087 222 2333" , customer.getPhone	equals
17	testSetPhoneWithTab	Customer/ setPhone	assertEquals	setUp, customer.setPhone("(00353) 87 2222 333")	"087 222 2333" , customer.getPhone	equals
18	testSetPaintCans	Customer/ setPaintCans	assertEquals	setUp, customer.setPaintCans(7)	7, customer.getPaintCans()	equals
19	testGetNewCustomerWrongOrder	Customer/ getNewCustomer	assertEquals	getNewCustomer("0\tBogdan Pasterak\t1234567890\t4")	try/ catch (IncorrectObjectTypeExeption)	catch Exeption
20	testGetNewCustomerBadInt	Customer/ getNewCustomer	assertEquals	getNewCustomer("a\tBogdan Pasterak\t1234567890\t4")	try/ catch (IncorrectObjectTypeExeption)	catch Exeption
21	testGetNewCustomerBadInt2	Customer/ getNewCustomer	assertEquals	getNewCustomer("0\tBogdan Pasterak\t1234567890\t4.5")	try/ catch (IncorrectObjectTypeExeption)	catch Exeption
22	testGetNewCustomerNumberData	Customer/ getNewCustomer	assertEquals	getNewCustomer("0\tBogdan Pasterak\t1234567890")	try/ catch (IncorrectObjectTypeExeption)	catch Exeption
23	testGetNewCustomerNumberDataOver	Customer/ getNewCustomer	assertEquals	getNewCustomer("a\tBogdan Pasterak\t1234567890\t5\t6")	try/ catch (IncorrectObjectTypeExeption)	catch Exeption
24	testGetNewCustomerNumberDataTrim	Customer/ getNewCustomer	assertEquals	getNewCustomer(" \t50\tBogdan Pasterak\t0871234567\t5\t")	50, "Bogdan Pasterak", "087-1234-567", 5 try/ catch	all data equals , no catch
25	testToSave	Customer/ toSave	assertEquals	set data("Bogdan", "/+353/87 1234567", 12)	get.Id + "\tBogdan\t087 123 4567\t12"	equals
CustomerToStringTest (Junit Test Case)						
26	testToString	Customer/toString	assertEquals	Customer("Bogdan Pasterak", "087-1234-567", 5)	"Customer ID - " + customer.getId() + "\n" + " Name - Bogdan Pasterak\n" + " Phone Nr - 087 123 4567\n" + " Ordered paint cans - 5";	equals
PaintRequiredCalculatorTest (Junit Test Case)						
27	numberOfCansTestMinus74_2	PaintRequiredCalculator/ numberOfCans	assertEquals	numberOfCans(-74.2)	0, 0.0001	equals
28	numberOfCansTest0	PaintRequiredCalculator/ numberOfCans	assertEquals	numberOfCans(0.0)	0, 0.0001	equals
29	numberOfCansTestLess14	PaintRequiredCalculator/ numberOfCans	assertEquals	numberOfCans(13.99)	1, 0.0001	equals
30	numberOfCansTestEquals14	PaintRequiredCalculator/ numberOfCans	assertEquals	numberOfCans(14.0)	1, 0.0001	equals
31	numberOfCansTestMore14	PaintRequiredCalculator/ numberOfCans	assertEquals	numberOfCans(14.01)	2, 0.0001	equals
32	numberOfCansTestLess28	PaintRequiredCalculator/ numberOfCans	assertEquals	numberOfCans(27.99)	2, 0.0001	equals
33	numberOfCansTestEquals28	PaintRequiredCalculator/ numberOfCans	assertEquals	numberOfCans(28.0)	2, 0.0001	equals
34	numberOfCansTestMore28	PaintRequiredCalculator/ numberOfCans	assertEquals	numberOfCans(28.01)	3, 0.0001	equals
35	numberOfCansTest128_64	PaintRequiredCalculator/ numberOfCans	assertEquals	numberOfCans(128.64)	10, 0.0001	equals
SurfaceCalculatorTest (Junit Test Case)						
36	testSquareRoomH2S3	SurfaceCalculator/ squareRoom	assertEquals	squareRoom(2, 3)	24, 0.00001	equals
37	testSquareRoomH2_4S3_22	SurfaceCalculator/ squareRoom	assertEquals	squareRoom(2.4, 3.22)	30.912, 0.00001	equals
38	testSquareRoomH2S5_3	SurfaceCalculator/ squareRoom	assertEquals	squareRoom(2, 5.3)	42.4, 0.00001	equals
39	testSquareRoomH2_77S3	SurfaceCalculator/ squareRoom	assertEquals	squareRoom(2.77, 3)	33.24, 0.00001	equals
40	testRectangularRoomH2L3W4	SurfaceCalculator/ rectangularRoom	assertEquals	rectangularRoom(2, 3, 4)	28, 0.00001	equals
41	testRectangularRoomH2L3W4	SurfaceCalculator/ rectangularRoom	assertEquals	rectangularRoom(2.8, 3.5, 4.2)	43.12, 0.00001	equals
42	testCylindricaRoomH2D2	SurfaceCalculator/ cylindricalRoom	assertEquals	cylindricalRoom(2, 2)	4 * Math.PI, 0.00001	equals
43	testCylindricaRoomH2_8D3	SurfaceCalculator/ cylindricalRoom	assertEquals	cylindricalRoom(2.8, 3)	8.4 * Math.PI, 0.00001	equals
44	testCylindricaRoomH2_8D2_2	SurfaceCalculator/ cylindricalRoom	assertEquals	cylindricalRoom(2.8, 2.2)	6.16 * Math.PI, 0.00001	equals

FormatPhoneNoTest (Junit Test Case)

45	test1	MyScanner/ formatPhoneNo	assertEquals	formatPhoneNo("0871234567")	"087 123 4567"	equals
46	test2	MyScanner/ formatPhoneNo	assertEquals	formatPhoneNo("08 71\t234567")	"087 123 4567"	equals
47	test3	MyScanner/ formatPhoneNo	assertEquals	formatPhoneNo("011234567")	"01 123 4567"	equals
48	test4	MyScanner/ formatPhoneNo	assertEquals	formatPhoneNo("0402123456")	"0402 123 456"	equals
49	test5	MyScanner/ formatPhoneNo	assertEquals	formatPhoneNo("0411234567")	"041 123 4567"	equals
50	test6	MyScanner/ formatPhoneNo	assertEquals	formatPhoneNo("0831234567")	"083 123 4567"	equals
51	test7	MyScanner/ formatPhoneNo	assertEquals	formatPhoneNo("0821234567")	null	equals

ValidatePhoneNoTest (Junit Test Case)

52	testTrue01	MyScanner/ validatePhoneNo	assertTrue	validatePhoneNo("012345678")	true	equals
53	testTrue02	MyScanner/ validatePhoneNo	assertTrue	validatePhoneNo("0 1 2-3*4/5:6")	true	equals
54	testTrue03	MyScanner/ validatePhoneNo	assertTrue	validatePhoneNo("0;1,2.3 / 4/5:6")	true	equals
55	testTrue04	MyScanner/ validatePhoneNo	assertTrue	validatePhoneNo("021 12345")	true	equals
56	testTrue05	MyScanner/ validatePhoneNo	assertTrue	validatePhoneNo("029 1234567")	true	equals
57	testTrue06	MyScanner/ validatePhoneNo	assertTrue	validatePhoneNo("0402 12345")	true	equals
58	testTrue07	MyScanner/ validatePhoneNo	assertTrue	validatePhoneNo("0404 1234567")	true	equals
59	testTrue08	MyScanner/ validatePhoneNo	assertTrue	validatePhoneNo("041 12345")	true	equals
60	testTrue09	MyScanner/ validatePhoneNo	assertTrue	validatePhoneNo("047 1234567"")	true	equals
61	testTrue10	MyScanner/ validatePhoneNo	assertTrue	validatePhoneNo("071 12345")	true	equals
62	testTrue11	MyScanner/ validatePhoneNo	assertTrue	validatePhoneNo("074 1234567")	true	equals
63	testTrue12	MyScanner/ validatePhoneNo	assertTrue	validatePhoneNo("087 1234567")	true	equals
64	testTrue13	MyScanner/ validatePhoneNo	assertTrue	validatePhoneNo("+35387 1234567")	true	equals
65	testTrue14	MyScanner/ validatePhoneNo	assertTrue	validatePhoneNo("0035387 1234567")	true	equals
66	testFalse15	MyScanner/ validatePhoneNo	assertFalse	validatePhoneNo("00 1234567")	false	equals
67	testFalse16	MyScanner/ validatePhoneNo	assertFalse	validatePhoneNo("01 12345678")	false	equals
68	testFalse17	MyScanner/ validatePhoneNo	assertFalse	validatePhoneNo("01 1234")	false	equals
69	testFalse18	MyScanner/ validatePhoneNo	assertFalse	validatePhoneNo("0402 1234")	false	equals
70	testFalse19	MyScanner/ validatePhoneNo	assertFalse	validatePhoneNo("0402 123456789")	false	equals
71	testFalse20	MyScanner/ validatePhoneNo	assertFalse	validatePhoneNo("0 1 2-3*4/5:6")	false	equals
72	testFalse21	MyScanner/ validatePhoneNo	assertFalse	validatePhoneNo("0032374 1234567")	false	equals
73	testFalse21	MyScanner/ validatePhoneNo	assertFalse	validatePhoneNo("+32374 1234567")	false	equals
74	testFalse21	MyScanner/ validatePhoneNo	assertFalse	validatePhoneNo("087 123456")	false	equals
75	testFalse21	MyScanner/ validatePhoneNo	assertFalse	validatePhoneNo("087 12345678")	false	equals
76	testFalse21	MyScanner/ validatePhoneNo	assertFalse	validatePhoneNo("087 1234S67")	false	equals