

Laboratorio 2

1. Requerimientos funcionales

Nombre	R1.Generar el cuadrado mágico.
Resumen	El programa deberá generar el cuadrado mágico a partir del orden, dirección y fila o columna perimetral de inicio dada por el usuario
Entrada	Un entero impar que equivale al Orden, un String que equivale a la dirección y un String equivalente al lugar de inicio
Salida	

Nombre	R2.Cambiar el color de las casillas
Resumen	El programa deberá permitir cambiar el color de las columna y fila seleccionada
Entrada	
Salida	Fila y columna pintada

Nombre	R3.Desarrollar pruebas unitarias.
Resumen	El programa deberá disponer de las pruebas unitarias de los métodos del paquete modelo y excepciones personalizadas.
Entrada	
Salida	

2. Tabla de trazabilidad

Requerimiento	Métodos	Clase
R1	+createMagicSquare(ActionEvent event):void +initialize():void +changes():void +setting():void +develop():void +fillSquare(int size, String sence, String ubicacion):void	MagicSquareController MagicSquareController
R3	+setUpScenary1():void +testUpNo():void +testUpNe():void +testRightNe():void +testRightSe():void +testDownSo():void +testDownSe():void +testLeftNo():void +testLeftSo():void	MagicSquareTest

	+notNullTest():void +negativeArraySizeExceptionTest1():void	
--	--	--

3. Pruebas unitarias

Escenarios

Nombre	Clase	Escenario
setupScenary1	MagicSquareTest	magic Square=new Magic Square();

Casos de prueba

Objetivo: Verificar que el método createdSquare desarrolle su cuerpo sin problema				
Clase	Metodo	Escenario	Valores de entrada	Resultado
MagicSquare	createdSquare	setupScenary1	Size: 3 Direction:"NO" Way: "UP"	True, El metodo se desarrolla en perfectas condiciones, dando como verdadero la condición propuesta. int [][] matrisCompare={{6,1,8},{7,5,3},{2,9,4}};
MagicSquare	createdSquare	setupScenary1	Size: 3 Direction:"NE" Way: "UP"	True, El metodo se desarrolla en perfectas condiciones, dando como verdadero la condición propuesta. int [][] matrisCompare={{8,1,6},{3,5,7},{4,9,2}};

MagicSquare	createdSquare	setupScenary1	Size: 3 Direction:"NE" Way: "RIGHT"	True, El metodo se desarrolla en perfectas condiciones, dando como verdadero la condición propuesta. int [][] matrisCompare={{2,7,6},{9,5,1},{4,3,8}}; };
MagicSquare	createdSquare	setupScenary1	Size: 3 Direction:"SE" Way: "RIGHT"	True, El metodo se desarrolla en perfectas condiciones, dando como verdadero la condición propuesta. int [][] matrisCompare={{4,3,8},{9,5,1},{2,7,6}}; };
MagicSquare	createdSquare	setupScenary1	Size: 3 Direction:"SO" Way: "DOWN"	True, El metodo se desarrolla en perfectas condiciones, dando como verdadero la condición propuesta. int [][] matrisCompare={{2,9,4},{7,5,3},{6,1,8}}; };
MagicSquare	createdSquare	setupScenary1	Size: 3 Direction:"SE" Way: "DOWN"	True, El metodo se desarrolla en perfectas condiciones, dando como verdadero la condición propuesta. int [][] matrisCompare={{4,9,2},{3,5,7},{8,1,6}}; };

MagicSquare	createdSquare	setupScenary1	Size: 3 Direction:"NO" Way: "LEFT"	True, El metodo se desarrolla en perfectas condiciones, dando como verdadero la condición propuesta. int [][] matrisCompare={{4,9,2},{3,5,7},{8,1,6}}; };
MagicSquare	createdSquare	setupScenary1	Size: 3 Direction:"SO" Way: "LEFT"	True, El metodo se desarrolla en perfectas condiciones, dando como verdadero la condición propuesta. int [][] matrisCompare={{8,3,4},{1,5,9},{6,7,2}}; };

Objetivo: verificar que el metodo createdSquare retorne una matriz diferente de ceros				
Clase	Metodo	Escenario	Valores de entrada	Salida
MagicSquare	createdSquare	setupScenary1	Size: 3 Direction:"SO" Way: "LEFT"	False, El metodo se desarrolla en perfectas condiciones, dando como verdadero la condición de que la matriz creada es diferente es de la matriz de ceros

Objetivo: verificar que el metodo createdSquare lance la excepción NegativeArraySizeException				
Clase	Metodo	Escenario	Valores de entrada	Salida
MagicSquare	createdSquare	setupScenary1	Size: -3 Direction:"SO" Way: "LEFT"	True, El metodo se desarrolla en perfectas condiciones, dando como verdadero la lanzada de la excepción buscada