

PRÁCTICA METODOLOGÍA DE LA PROGRAMACIÓN

2022-23

TESTING

NOMBRE	APELLIDOS	DNI	GRADO
Lucía	Domínguez Rodrigo	49452469-P	GII+GIS
Ángel	Marqués García	34295108-S	GII+GIS
Marcos	Jiménez Pulido	04860135-M	GII+GIS
Sergio	De Oro Fernández	54712181-B	GII+GIS

LISTA DE DISTRIBUCIÓN DE ROLES

NOMBRE	ROL	ORGANIZACIÓN
Sergio	Jefe de Proyecto	URJC
Lucía	Analista Funcional	URJC
Marcos	Analista Programador	URJC
Sergio	Ingeniero de Desarrollo	URJC
Ángel	Quality Assurance Tester	URJC

ÍNDICE

Clases del juego.....	3
Test Suite usuario.....	3
Usuario	3
Jugador	3
Administrador	4
NumeroRegistro	4
Test suite esbirro.....	6
Demonio	6
Test Suite Equipo.....	7

Clases del juego

Test Suite usuario

Usuario

Método probado	condiciones de ejecución	Entrada	respuesta esperada	respuesta obtenida	Comentarios
obtenerNick	Creado un Usuario con this.nick == "Nick"		"Nick"	"Nick"	Respuesta esperada
compararContrasenia	Creado un Usuario con this.contrasenia == "Contrasenia"	"Contrasenia"	True	True	Respuesta esperada
		"otraString"	False	False	
esAdministrador	Creado un usuario sin NumReg. Luego se le ha puesto un NumReg.		True	True	Respuesta esperada
			False	False	
estaBaneado	Creado un usuario no baneado. Luego se le cambió this.baneado a true		False	False	Respuesta esperada
			True	True	
cambiarBaneo	Creado un usuario baneado. Usado este método para cambiarlo a true. Comprobado si cambia. Cambiado el atributo a false. Comprobado que cambia.	true	True	True	Respuesta esperada
		false	False	False	
ponerNumReg	Creado un usuario con un NumReg a 0. Cambiado el número de registro a uno concreto. Comprobado que haya cambiado	NumReg = "a12bc"	"a12bc"	"a12bc"	Respuesta esperada
compararNumReg	Creado un usuario con NumReg a 0. Comprobado que ese coincide con un NumReg a 0. Comprobado que no coincide con un NumReg distinto de 0.	"00000"	True	True	Respuesta esperada
		"a12bc"	False	False	

Jugador

Método probado	condiciones de ejecución	Entrada	respuesta esperada	respuesta obtenida	Comentarios
obtenerPersonaje	Explicado bajo la tabla*		null	null	Respuesta esperada
			Personaje1	Personaje1	
			Personaje2	Personaje2	

ponerPersonaje	Creado un jugador sin personaje. Se le ha puesto un personaje, y se ha comprobado que el cambio ocurre. Luego se le ha puesto el personaje a null, y se ha comprobado que el cambio ocurre	Personaje1	Personaje1	Personaje1	Respuesta esperada
		null	null	null	
incrementarVictorias	Se crea un jugador. Se obtiene su número de victorias. Se llama a incrementarVictorias. Se comprueba que lo que devuelve es igual al número de victorias más 1.		1	1	Respuesta esperada
obtenerVictorias	Se ha creado un Jugador con 0 victorias. Se ha comprobado que obtenerVictorias devuelve 0. Se le han añadido dos victorias. Se ha comprobado que obtenerVictorias devuelve 2.		0	0	Respuesta esperada
			2	2	

*Se ha creado un jugador sin personaje, y comprobado que obtenerPersonaje devuelve null. Luego se le ha asignado al jugador un personaje nuevo, y se ha comprobado que obtenerPersonaje devuelve ese personaje. Para poder crear un Personaje nuevo, se ha añadido un constructor vacío a Personaje, y se ha creado una clase privada en JugadorTest que implemente Personaje (ya que es abstracta).

Administrador

Método probado	condiciones de ejecución	Entrada	respuesta esperada	respuesta obtenida	Comentarios
estaBaneado	Se crea un Administrador. Se comprueba que devuelve false. Se intenta cambiar el baneo a true. Se comprueba que devuelve false		false	false	Respuesta esperada
			false	false	
cambiarBaneo	Se crea un Administrador. Se pone el baneo a true y se comprueba que no ha cambiado. Se pone el baneo a false y se comprueba que no ha cambiado	true	Sin cambios	Sin cambios	Respuesta esperada
		false	Sin cambios	Sin cambios	

NumeroRegistro

Nos referiremos a los objetos NumeroRegistro con una string "xnmyz", donde x,y,z corresponden al carácter 1, 2, y 3, y n,m corresponden al número 1, y 2, respectivamente.

Método probado	condiciones de ejecución	Entrada	respuesta esperada	respuesta obtenida	Comentarios
copiar	Creado un número de registro a 0 y otro con "a12bc". Copiado el segundo en el primero. Comprobado que son iguales.	"a12bc"	Ambos NumReg son iguales	Ambos NumReg son iguales	Respuesta esperada
incrementarNumReg	Las pruebas realizadas se detallan después de la tabla				
sonIguales	Se han creado dos NumReg vacíos, y se ha comprobado que son iguales. Se ha repetido esto con dos NumReg con el valor "a12bc". Luego se ha probado con uno vacío y uno con "a12bc". Luego se ha probado con uno a "a12bc" y con uno a null	NumReg="00000"	true	true	Detectado un error. Explicado después de la tabla
		NumReg="a12bc"	false	false	
		NumReg="00000"	false	false	
		NumReg=null	false	Error	

Error en sonIguales: No estaba considerado el caso en el cual se pasaba un NumeroRegistro null en el parámetro, por lo que daba error. Esto se ha solucionado añadiendo una condición que comprueba si el NumeroRegistro pasado como parámetro es o no null, y en caso de que lo devuelva devuelve false.

Pruebas de incrementarNumReg:

Se han llevado a cabo las siguientes pruebas:

Teniendo un NumReg con valor:	Al llamar incrementarNumReg toma valor:	Comentarios
"00000"	"00001"	Respuesta esperada
"00009"	"0000A"	Respuesta esperada
"0000Z"	"0000a"	Respuesta esperada
"0000z"	"00010"	Respuesta esperada
"000zz"	"00100"	Respuesta esperada
"001zz"	"00200"	Respuesta esperada
"009zz"	"01000"	Respuesta esperada
"099zz"	"10000"	Respuesta esperada
"z99zz"	NumRegOverflowException	Error detectado.

Realizando estas pruebas se encontró un error. No estaba considerado el comportamiento al hacer overflow. Se ha corregido haciendo no cambie el NumeroRegistro y que lance una excepción. El resto de las pruebas funcionaron correctamente.

Test suite esbirro

Demonio

obtenerVida:

obtenerVida devuelve la vida del demonio sobre el que se llame sumada a la vida de sus subordinados. Por tanto realizaremos pruebas modificando sus subordinados.

Prueba Nº	Valor de this.esbirros	Valor esperado	Valor Recibido	Comentarios
1	null	5	Error	Solucionado.
2	set con dos esbirros	10	10	
3	set con un Demonio, que a su vez tiene dos esbirros	15	15	

Al probar obtenerVida se ha encontrado un error en el constructor de Esbirro. Si se le pasa un set de esbirros vacío da error. Para solucionarlo, se ha añadido una comprobación. Si el set fuera null, se inicializarán los esbirros como un conjunto vacío.

tieneSubordinados:

Este método deberá devolver true si el set de esbirros no está vacío. Por tanto realizaremos pruebas modificando este set.

Prueba Nº	Valor de this.esbirros	Valor esperado	Valor Recibido	Comentarios
1	null	false	false	
2	set vacío	false	false	
3	set con dos esbirros	true	true	

tieneHumanos:

Este método deberá devolver true el set de esbirros contiene un esbirro de clase Humano. Por tanto realizaremos pruebas modificando este set de esbirros.

Prueba Nº	Valor de this.esbirros	Valor esperado	Valor Recibido	Comentarios
1	null	false	false	
2	set vacío	false	false	
3	set con dos esbirros	false	false	
4	set con un Demonio, cuyo set tiene dos esbirros	false	false	
5	Set con un Esbirro y un Humano	true	true	
6	Set con un Demonio, cuyo set tiene un Humano y un Esbirro	true	true	

Test Suite Equipo

Arma:

Método probado	condiciones de ejecución	Entrada	respuesta esperada	respuesta obtenida	Comentarios
fromListToString	Las pruebas realizadas para este método se detallan debajo de la tabla				Por falta de espacio, las pruebas se explican bajo la tabla
obtenerVariante	Se ha creado un arma a dos manos, y se ha comprobado que obtenerVariante devuelve dosManos. Luego se ha hecho lo mismo con un arma a una mano		dosManos	dosManos	Respuesta esperada
			unaMano	unaMano	
toString	Se ha creado un arma con un nombre, ataque, defensa, y variante conocidos.		Una String mostrando los atributos del arma con un formato concreto	Una String mostrando los atributos del arma con un formato concreto	Respuesta esperada

Para fromListToString se han realizado las siguientes pruebas:

Prueba Nº	Valor del parámetro	Valor esperado	Valor Recibido	Comentarios
1	New ArrayList<>()	""	""	
2	List con 3 objetos Arma	Una string mostrando los nombres y variantes de las armas con un formato concreto	Una string mostrando los nombres y variantes de las armas con un formato concreto	
3	null	RuntimeException	RuntimeException	

Armadura:

Las pruebas de fromListToString y toString son análogos a los realizados para la clase Arma.

Equipo:

Método probado	condiciones de ejecución	Entrada	respuesta esperada	respuesta obtenida	Comentarios
obtenerAtaque	Se ha creado un objeto Equipo con this.ataque == 3		3	3	Resultado esperado

obtenerDefensa	Se ha creado un objeto Equipo con this.defensa == 2		2	2	Respuesta esperada
obtenerNombre	Se ha creado un objeto Equipo con this.nombre == "Equipo1"		"Equipo1"	"Equipo1"	Respuesta esperada