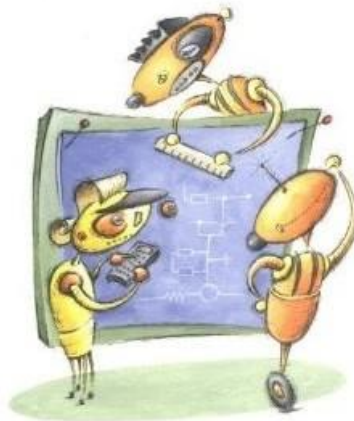




UNIVERSIDAD
DE BURGOS

PRACTICA DE JADE: PERSONAL LIBRARY



Eduardo Mora González

1. Introducción

Contemplando el funcionamiento del ejemplo de *bookTrading*, he observado una funcionalidad con la que se pueda usar este sistema, la funcionalidad no es si otra un sistema que ayude a un bibliófilo a saber si tiene un libro en su biblioteca o no.

2. Descripción de la estrategia

Para desarrollar la idea de la biblioteca de un bibliófilo, he creado la clase libro en java, en donde se muestra la descripción de cada libro.

Cuando ejecutamos el agente encargado de comprar los libros, añadimos una serie de argumentos:

- El primer argumento que añadimos es el libro que queremos comprar.
- El resto de los argumentos separados por comas, son todos los libros que este posee en su biblioteca.

Cuando el agente se ejecuta, lo que hace es comprobar que el libro que quiere comprar no exista en su biblioteca, en caso de existir no accede al comportamiento de comprar. En caso de que no exista en su biblioteca, se añade el comportamiento encargado de la compra del libro y sigue la ejecución de compra.

También para que el agente comprador pueda realizar una compra, debe existir al menos dos vendedores para poder comparar precios y comprar el de mejor, pero teniendo cuidado de que exista unidades de este.

Para no desistir en la primera consulta de un libro en el caso de no exista el mínimo de compradores o no exista unidades del mismo, el agente realizará 3 consultas, cada una a los 30 segundos, en el caso de que en la tercera vuelta no consiga comprar el libro, este se eliminará.

En el otro extremo, tenemos el agente vendedor, el cual, al añadir un nuevo libro, tendrá que añadir también el número de unidades que tiene y si no tiene unidades, debe decírselo al comprador.

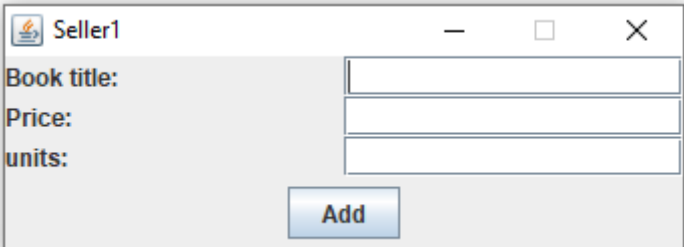
3. Descripción y motivación del comportamiento y otros cambios introducidos

El comportamiento implementado es un *OneShotBehaviour*, la particularidad de este comportamiento es que solo se ejecuta una vez, por lo que no necesita implementar el método *Done*.

La decisión de coger este comportamiento ha sido, que en circunstancias normales el agente solo se debería ejecutar una vez, no obstante y como he dicho antes, en el comportamiento se ha usado el método *reset()*, para volver a lanzar el comportamiento.

Si bien se puede pensar, de que con el método *Done* la repetición está implementada ya por defecto, he investigado otras formas de hacer lo mismo que este método con otros comportamientos.

Para implementar lo mencionado antes, lo primero que he hecho ha sido añadir un nuevo campo en el vendedor, a la hora de añadir un libro, también se añade la cantidad de unidades de ese libro que tenemos:



Y como debemos tener varios vendedores, cada vez que se añade un libro, se muestra el nombre del agente que lo ha añadido:

```
Seller1@172.18.125.42:1099/JADE inserted: El Quijote into catalogue. Price = 15 and I have: 2 units
```

Por el lado del comprador, cada vez que añadimos uno, debemos poner como argumentos el libro que queremos y los que tenemos:

Insert Start Parameters	
Agent Name	Eduardo
Class Name	original.BookBuyerAgent
Arguments	El Quijote, Harry Potter
Owner	
Container	Main-Container
<div>OK Cancel</div>	

El comportamiento ya comprobará si tiene el libro o no, en caso de que no lo tenga:

```
The buyer Eduardo@172.18.125.42:1099/JADE has the following books:
Harry Potter
```

```
=====
```

```
The buyer wants the book whose title is: El Quijote
```

```
=====
```

```
The buyer does not have the book, therefore he wants to buy it
```

En el caso de que lo tenga:

```
Hallo! Buyer-agent Masetto@172.18.125.42:1099/JADE is ready.
```

```
The buyer Masetto@172.18.125.42:1099/JADE has the following books:
El Quijote
```

```
=====
```

```
The buyer wants the book whose title is: El Quijote
```

```
=====
```

```
The buyer already has the book so he does not want to buy
```

El comportamiento comprobará que exista al menos dos compradores, y para dar un poco de más juego, se preguntará 3 veces, para cada pregunta he usado el método reset(30000) que reinicia el comportamiento a los 30 segundos y un contador para que solo se resetee 3 veces. La salida del agente sería:

```

*****

There is no minimum number of sellers... new check in 30 seconds

*****
*****

There is no minimum number of sellers... new check in 30 seconds

*****
*****

There is no minimum number of sellers... new check in 30 seconds

*****
*****
=====
Eduardo@172.18.125.42:1099/JADE I have tried 3 times and I have not been successful therefore I am leaving
=====

```

Si añadimos otro vendedor, ya si se podrá hacer la compra y se actualizará las unidades del libro que posee el vendedor:

```

Seller2@172.18.125.42:1099/JADE inserted: El Quijote into catalogue. Price = 1 and I have: 1 units
Found the following seller agents:
Seller1@172.18.125.42:1099/JADE
Seller2@172.18.125.42:1099/JADE
||Seller1@172.18.125.42:1099/JADE --> Now I have 1 units from the book El Quijote
||Seller2@172.18.125.42:1099/JADE --> Now I have 0 units from the book El Quijote
El Quijote sold to agent Eduardo@172.18.125.42:1099/JADE
El Quijote successfully purchased from agent Seller2@172.18.125.42:1099/JADE
Price = 1

```

En el caso de que exista el libro en los dos vendedores y aunque no exista unidades en uno de ellos, el comprador puede realizar la compra:

```

Found the following seller agents:
Seller1@172.18.125.42:1099/JADE
Seller2@172.18.125.42:1099/JADE
||Seller1@172.18.125.42:1099/JADE --> Now I have 0 units from the book El Quijote
=====
||Seller2@172.18.125.42:1099/JADE --> We do not have units of that book
=====
El Quijote sold to agent Edu@172.18.125.42:1099/JADE
El Quijote successfully purchased from agent Seller1@172.18.125.42:1099/JADE
Price = 15
Buyer-agent Edu@172.18.125.42:1099/JADE terminating.

```

Pero si ninguno tiene el libro, no se puede realizar la compra:

```

Found the following seller agents:
Seller1@172.18.125.42:1099/JADE
Seller2@172.18.125.42:1099/JADE
=====
||Seller1@172.18.125.42:1099/JADE --> We do not have units of that book
=====
||Seller2@172.18.125.42:1099/JADE --> We do not have units of that book
=====
Attempt failed: El Quijote not available for sale

```

4. Bibliografía

<https://saeedkhattak.wordpress.com/2012/03/07/jade-with-eclipse/>

<https://jade.tilab.com/doc/api/index.html>

<https://docs.oracle.com/javase/8/docs/api/java/util/HashMap.html>