

Bolsa y empresas:

La bolsa es un HashMap debido a que se realizan muchas operaciones de obtención de valor (get, containsKey) y el orden no es relevante.

También se realiza bastante iteración sobre el mapa en muchos métodos. Por lo tanto, se evita la repetición mediante la clave del mapa (Cuidando la diferencia de mayúsculas).

Se permite elegir libremente la variación en los precios de la bolsa debido a que esto es solo una simulación, aunque en un caso real esto no sería posible.

El aumento y las disminuciones de valor a la hora de comprar y vender acciones siguen la fórmula:

x siendo valor de la acción actual
y siendo el número de acciones compradas o vendidas
La suma o resta depende de si se compra o vende respectivamente.

$$\text{precio} = \text{precioInicial} \pm (x \cdot 1/100) \cdot (5/y)$$

El aumento o disminución no puede ser menor de 1 €.

Agente de Inversiones:

La colección usada para ambas listas de mensajes es un ArrayList, debido a que las operaciones se realizan y guardan en orden. resultadoDeOperaciones es una lista con la que no se realiza ninguna operación, pero se ha implementado para guardar los resultados de las operaciones.

Esta decisión es debida a que se deseaba respetar el UML aportado en la práctica y que se puede usar en un futuro por si se quisiera usar para algo.

Al querer respetar la estructura de mensajes en los ejemplos aportados, para comprobar el tipo de mensaje en el método intentaOperacion de la bolsa, y debido a que recibe un String y no un Mensaje, al parsear el String se utiliza un cast para comprobar el tipo de mensaje:

Si es un mensaje de venta, el cast no lanza ninguna excepción debido a que el valor guardado en datos[3] (numero de acciones a vender) se puede hacer cast en un int.

Si es un mensaje de compra, el cast lanza la excepción NumberFormatException, ya que en datos[3] (cantidad máxima de saldo a invertir) se guarda un valor double.

Si es un mensaje de actualización, el cast lanza la excepción IndexOutOfBoundsException, ya que no existe datos[3] (solo se guarda la fecha)

Este método devuelve el string con el mensaje de respuesta codificado.

Al llegar de vuelta a EjecutarOperaciones, según el tipo de mensaje inicial (comprobado con instanceof) se realiza la operación pertinente. Al realizarse se guarda el resultado.

Se planteó que la clase Mensaje y sus hijas tuvieran método para crear un mensaje a partir del mismo parseado, pero se decidió que, debido a la insistencia en enunciado y explicaciones

de que fuera el bróker el que realizara la decodificación de mensajes, se le impone la tarea de la construcción de estos (Lo mismo es aplicable a la creación y codificación de mensajes de respuesta en la bolsa).

Banco y cliente:

El banco almacena los clientes en un HashMap de objetos de tipo cliente, utilizamos esta collection al no ser necesario un orden en los clientes y a las muchas búsquedas a realizar de dichos clientes. La clave de este HashMap es el propio DNI del cliente, que son debidamente tratados para que tengan el formato correcto y son unicos e irrepetibles. Tambien se tiene cuidado con los DNIs de los gestores y del agente.

Los metodos de carga y guardado de una copia de seguridad en fichero tan solo guardan y cargan dicho HashMap, ya que el agente de bolsa y el nombre del banco tienen una carga predefinida en la creacion del simulador.

El metodo Recomendacion es un metodo por el cual se llama al gestor propio del cliente (debe ser premium) y se le pide una recomendacion, este tan solo mira las valoraciones de la bolsa de valores, y devuelve el nombre de la empresa con mayor variacion.

La clase cliente es la clase utilizada por el HashMap, pero a veces es necesario guardar clientes premium, que ademas de los atributos anteriores tambien tienen a un gestor. Como cliente premium es hijo de cliente es posible guardarlos en la misma collecion.

A traves de instanceof podemos identificar si el objeto guardado es de la clase cliente premium.

Paquete de acciones:

Es una simple clase que contine una serie de atributos basicos, y unos extra derivados. Su unico metodo es la actualizacion de 1 accion.

Mensajes:

Los mensajes básicamente se diferencian en tres tipos: de compra, venta y actualización. Para llevar serializados hemos hecho un identificador aleatorio, el cual genera entre 1 y 10000 para llevar todas las operacines ordenadas.

En este caso la serialización de los mensajes está hecho en Bolsa, donde llama a mensajes.

En este apartado, únicamente vamos a codificar los mensajes para que lleguen a bolsa y se serialicen.

Empezamos por mensajes de compra. Para ello vamos a necesitar cliente, empresa y la cantidad a comprar. En caso de que sea posible la compra, habiendo invertido igual o más del valor de las acciones, el resto se almacenarña de nuevo en el saldo del cliente.

La venta de acciones no dará excepción si existen tantas acciones en posesión como las requeridas a vender

En cuanto a la actualización, nos ofrecerá la posibilidad de limitar la cota superior e inferior de los nuevos precios, los cuales se hallarán aleatoriamente entre ambos límites. Estos precios se guardarán de nuevo

en la copia bolsa, sustituyendo los anteriores y relegandolos a precio anterior.

Esta actualización como hemos dicho anteriormente, se realiza en Paquete de acciones.