Programación Orientada a Objetos
Curso 2017/2018

Sesión 4

Previo

Para la implementación de los ejercicios es necesario hacer uso de listas. En Java existen varias clases que implementan listas. En esta práctica utilizaremos la clase `java.util.LinkedList`. Ten en cuenta que las colecciones de Java son clases genéricas. La genericidad se estudia más adelante en la asignatura. Para la implementación de la práctica es suficiente saber cómo se declaran y construyen las listas:

```
LinkedList<Subasta> subastas = new LinkedList<Subasta>();
```

Observa que se incluye el tipo de elementos que va a contener la lista entre < y >. En el ejemplo anterior se ha utilizado el constructor sin argumentos pero también está disponible el constructor de copia.

La documentación de la clase `java.util.LinkedList` está disponible en http://docs.oracle.com/javase/8/docs/api/java/util/LinkedList.html o línea en Eclipse.

A continuación se muestra un fragmento de código con las operaciones básicas para el manejo de listas:

```
LinkedList<String> lista = new LinkedList<String>();

// Añadir elementos por el final
lista.add("uno");       // Opción 1
lista.addLast("dos");   // Opción 2

// Añadir un elemento al principio
lista.addFirst("cero");

// Consultar el tamaño
System.out.println("Tamaño: " + lista.size());

// Consultar el primer elemento
String primero = lista.get(0);      // Opción 1
primero = lista.getFirst();         // Opción 2
System.out.println(primero);

// Consultar el último elemento
String ultimo = lista.get(lista.size() - 1);    // Opción 1
ultimo = lista.getLast();                       // Opción 2

System.out.println(ultimo);

// Borrado: remove, removeFirst y removeLast

// Recorrido for each
for (String elemento : lista) {
      System.out.println("Elemento: " + elemento);
}
// Copia
LinkedList<String> copia = new LinkedList<String>(lista);
```

Ejercicios

Crea el paquete `subastas` y sitúa en ese paquete todo el código de esta práctica. El objetivo de los ejercicios es implementar un sistema de **subastas** donde los usuarios puedan pujar por productos que ofrecen otros usuarios. A continuación se describen los requisitos de la aplicación:

1. Define la clase **`Usuario`** de la aplicación de subastas. Un usuario se caracteriza por dos propiedades: nombre y crédito. El nombre es una propiedad que no puede variar una vez establecida. El crédito corresponde al dinero disponible para poder pujar en las subastas.

   Un usuario se construye obligatoriamente estableciendo el nombre. Opcionalmente se puede establecer el crédito inicial.

   La funcionalidad de la clase es:

   - *incrementar crédito*: incrementa el crédito del usuario en una cantidad.
   - *decrementar crédito*: decrementar el crédito del usuario en una cantidad.

2. Define la clase **`Puja`** de la aplicación de subastas. Una puja representa la cantidad de dinero que ofrece un usuario por el producto de una subasta.

   Las propiedades que caracterizan a una Puja son el usuario que la realiza y la cantidad de dinero que ofrece. Estas propiedades no pueden ser modificadas.

   Las pujas se construyen estableciendo las dos propiedades que caracterizan a la clase.

   Por último, la clase no proporciona más funcionalidad que la consulta de las propiedades.

3. Define la clase **`Subasta`** que representa un producto por el que se pueden realizar pujas.

   Las propiedades que caracterizan a una subasta son: nombre del producto subastado, el usuario propietario de la subasta (objeto `Usuario`), si está abierta o cerrada, la lista de pujas realizada en la subasta y la puja mayor, que corresponde con la última puja de la lista de pujas o bien tendrá un valor `null` si aún no se han realizado pujas. Todas las propiedades se pueden consultar, pero ninguna podrá ser modificada *directamente*. Algunas propiedades serán modificadas tras la aplicación de los métodos de la clase.

   Una subasta se construye estableciendo el nombre del producto subastado y el usuario propietario. Una subasta recién construida está abierta y no tendrá pujas.

   La funcionalidad de esta clase es:

   - *pujar*: este método permite realizar una puja sobre la subasta. La información necesaria para pujar es el usuario que realiza la puja y la cantidad por la que puja. La puja es aceptada si: a) la subasta está abierta, b) el crédito del usuario que la realiza es suficiente para la cantidad por la que puja; c) el usuario no es propietario de la subasta y d) la cantidad es mayor que la cantidad de la *puja mayor*, si la hubiera. Por tanto, esta operación finaliza indicando si la puja ha sido aceptada (retorna un valor booleano). Si la puja es aceptada, entonces se construye una puja y se almacena en la lista de pujas.

- *pujar*: versión sobrecargada del método anterior que permite pujar sin indicar la cantidad, es decir, sólo se requiere indicar el usuario que la realiza. La cantidad será un euro más que la cantidad de la *puja mayor*. Si no hubiera *puja mayor*, la cantidad sería de un euro.
- *ejecutar*: este método cierra la subasta (la ejecuta) realizando las transferencias de crédito entre el usuario que ha ganado la subasta (puja mayor) y el usuario propietario. Es requisito para ejecutar una subasta que se haya realizado alguna puja y la subasta está abierta. En tal caso, la ejecución de una subasta consiste en decrementar el crédito del usuario que ha realizado la *puja mayor* e incrementar el crédito del propietario de la subasta por la cantidad de la *puja mayor*. Una vez ejecutada, la subasta quedará cerrada. El método finaliza informando si la subasta ha podido ejecutarse o no (retorna un valor booleano).

4. Realiza los cambios necesarios a la aplicación para que los usuarios tengan como propiedad las subastas de las que son propietarios. Esta propiedad debe ser actualizada en la operación de construcción de la subasta.

5. Escribe el siguiente programa:
   - Crea tres usuarios con nombre "Juan", "Pedro" y "Enrique" con un crédito inicial de 100, 150 y 300 euros, respectivamente.
   - Crea una subasta del producto "Teléfono Móvil" cuyo propietario sea el usuario *Juan*.
   - El usuario *Pedro* puja por esa subasta 100 euros.
   - Muestra en la consola la puja mayor de la subasta (nombre del usuario y cantidad).
   - El usuario *Enrique* puja por esa subasta 50 euros.
   - Muestra en la consola la puja mayor. Comprueba que esta segunda puja no ha sido acepta, ya que es menor que la primera.
   - Ejecuta la subasta.
   - El usuario *Enrique* puja de nuevo por esa subasta con 200 euros. Comprueba que no es aceptada, ya que la subasta ha sido cerrada.
   - Muestra por la consola los créditos de los tres usuarios. Observa que los créditos de *Juan* y *Pedro* han cambiado.
   - Muestra las subastas de las que son propietarios los tres usuarios.