

Programación- EXAMEN DE PRÁCTICAS. SEPTIEMBRE 2016

Se desea crear una aplicación que permita gestionar un sitio de subastas online, en el que los usuarios pueden ofrecer a la venta cualquier producto, indicando el precio de salida y la fecha de finalización de la subasta. A partir de ese instante, cualquier otro usuario puede realizar una puja para adquirir el producto. El usuario que haga la puja más alta al llegar el momento de finalización de la subasta, es quien compra el producto. La aplicación se organizará en las siguientes clases:

Clase Puja

Es una clase sencilla que contiene la información de cada puja que realiza un usuario para ganar una subasta. Se da ya implementada. Su diagrama de clase es el siguiente:

Puja
- String usuario - double precioOferta - Calendar fecha
+ Puja(String usuario, double precioOferta) + String usuario() + double precioOferta() + Calendar fecha()

- tres atributos: el nombre del usuario que realiza la puja, el precio que se oferta en la puja y el momento en que se hace la puja.
- constructor que recibe como parámetros el usuario y el valor de la puja
- un método observador para cada atributo

Clase Subasta

Contiene la información de cada subasta que se publica en el sitio, con la lista de las pujas que se han realizado sobre cada una de ellas y métodos para gestionarla. Se da ya implementada. Su diagrama de clase es:

Subasta
- int id - int static genId=1 - String usuario - String producto - double precioSalida - Calendar fechaFin - boolean cerrada - ArrayList<Puja> listaPujas
+ Subasta(String usuario, String producto, double precioSalida, int anyoFin, int mesFin, int diaFin, int horaFin, int minFin) +int getId() +String getUsuario() +String getProducto() +double getPrecioSalida() + void pujar(Puja puja) throws PujalInvalida + Puja pujaLider() + boolean estáCerrada() + void cerrar()

- Tiene 7 atributos de instancia: un código de identificación de la subasta, el nombre del usuario que publica la subasta, la descripción del producto a subastar, el precio de salida del mismo, la fecha de finalización de la subasta y un `ArrayList`, que llamamos `listaPujas`, para guardar todas las pujas realizadas por los distintos usuarios. Hay un octavo atributo, `genId`, que es estático y contiene el próximo identificador de subasta a asignar cuando se cree una nueva subasta.
- **Constructor**: Crea cada subasta a partir de los parámetros recibidos. Cada subasta creada tiene un identificador único que se asigna automáticamente.
- **Métodos observadores**: para los atributos necesarios.
- **pujar()**: Este método recibe una nueva puja como parámetro y la registra en la subasta, siempre que sea válida. Para que la puja sea válida debe hacerla un usuario distinto al que publicó la subasta, la puja debe estar abierta y el precio ofertado en la puja debe ser mayor que el precio de la puja más alta. En caso de que no haya aun pujas, el precio de la oferta debe ser mayor que el precio base de la puja. Si la puja no es válida, lanza la excepción.
- **pujaLider()**: método que devuelve la puja que va ganando la subasta. En caso de que no haya pujas, se devuelve null. Cuando la subasta se cierra, la puja líder pasa a ser inmediatamente la puja ganadora.
- **estaCerrada()**: método que devuelve verdadero si la subasta está cerrada. En caso contrario devuelve falso.
- **cerrar()**: método que cierra una puja. Para cerrarla, compara la fecha actual con la fecha de finalización de la subasta. Si es posterior, la marca como cerrada. Si no, no hace nada.

Clase CasaSubastas

Contiene una lista con el conjunto de subastas publicadas en el sitio de subastas. Se facilita un esqueleto de la clase, que hay que completar. El diagrama de clase es el siguiente:

CasaSubastas
- ArrayList<Subasta> subastasPublicadas
- ArrayList<Subasta> cargaDatos() + CasaSubastas() + int registrarSubasta(Subasta s) + void pujar(int idSubasta, Puja puja) throws SubastaIncorrecta, PujaInvalida + void cerrarSubastas() +void resumenActividad(String fich)

- Tiene un único atributo para almacenar la lista de subastas publicadas en el sitio.
- **cargaDatos**: Método privado que sirve para cargar los datos de las subastas ya publicadas en el sitio. Se da ya implementado.
- **Constructor**: Llama al método privado `cargaDatos` y asigna la lista que retorna a la lista de subastas publicadas.
- **registrarSubasta()**: Este método añade una nueva subasta a la lista de subastas publicadas. La subasta que se desea añadir se recibe como parámetro. Se retorna el identificador de la subasta.

- **pujar()**: método que busca una subasta dentro de la lista de subastas publicadas a partir del código de identificación que recibe en el primer parámetro. Si se encuentra y la puja que se recibe como segundo parámetro es válida, se registra esta puja. Si la puja no es válida, se lanzará la excepción PujalInvalida. Si no se encuentra la subasta, se lanzará la excepción SubastaIncorrecta.
- **cerrarSubastas()**: Este método invoca al método cerrar de todas las subastas, para que se cierren todas las que lo requieran.
- **resumenActividad()**: Este método escribe un fichero de texto, cuyo nombre se recibe como parámetro, con un resumen de la actividad de la casa de subastas. Tiene dos partes. En la primera parte se escribirá en el fichero la lista de las subastas abiertas en ese momento, con su correspondiente puja líder (si no hubiese pujas, se escribirá un guion). en la segunda parte se escribirá en el fichero un resumen económico con el número de subastas cerradas, el número de subastas desiertas (sin pujas) y el total recaudado. A modo de ejemplo, el contenido del fichero será similar al siguiente:

```

Resumen de actividad
-----
Listado de subastas abiertas
-----
usuario      producto  salida  pujaLider
adolfo       ordenador 100.00  125.00
antonio      camiseta  20.00   -
ricardo      cargador  15.00   35.20
adolfo       monitor   65,00   68.75
-----
Resumen subastas cerradas
-----
Número de subastas cerradas: 15
Número de subastas desiertas: 8
Total cifra de venta: 873.20

```

Valoración

Para la valoración, al tratarse de un examen de prácticas, es **imprescindible** que el código entregado compile correctamente y que los métodos funcionen correctamente. Para comprobar el funcionamiento se suministra un programa principal de prueba, cuya salida debe incluirse en el informe de la práctica.

- Constructor: 1 punto
- Método registraSubasta: 1 punto
- Método pugar: 2,5 puntos
- Método cerrarSubastas: 1,5 puntos
- Método resumenActividad, primera parte: 2 puntos
- Método resumenActividad, segunda parte: 2 puntos