

EJERICICIO DE STREAM

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.

Define la clase Usuario de la aplicación de subastas.

Un usuario se caracteriza por dos propiedades (consultables):

- nombre: es una cadena que almacena el nombre del usuario. El valor de esta propiedad no podrá variar una vez establecida en la construcción.
- crédito: número real que corresponde con el dinero disponible para poder pujar en las subastas.

La clase ofrece dos constructores. El primer constructor recibe como parámetros las dos propiedades del tipo de datos (nombre y crédito). El segundo constructor solo recibe el nombre, quedando el crédito inicialmente con valor 50 (constante).

La funcionalidad de la clase es:

- incrementar crédito: incrementa el crédito del usuario en una cantidad (parámetro).
- decrementar crédito: decrementa el crédito del usuario en una cantidad.

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 (todas consultables):

- pujador: referencia al objeto Usuario que la realiza la puja.
- cantidad: número real que representa la cantidad de dinero que se ofrece en la puja.
- subasta: referencia al objeto Subasta en la que se realiza la puja.

Las propiedades no pueden ser modificadas. Las pujas se construyen estableciendo las tres propiedades que caracterizan a la clase.

Por último, la clase no proporciona ninguna funcionalidad

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

Las propiedades que caracterizan a una subasta son (todas consultables):

- nombre del producto: cadena que describe el producto subastado.
- propietario: referencia al objeto Usuario que es propietario de la subasta.
- abierta: valor booleano que indica si la subasta está abierta o cerrada. El valor true significa que está abierta. En cambio, el valor false indica que está cerrada.
- pujas: lista de pujas realizadas por el producto que ofrece la subasta.
- puja mayor: referencia al objeto Puja con la cantidad mayor ofrecida en la subasta (propiedad calculada). En caso de que no existan pujas, el valor de esta propiedad es null.

No se ofrecen operaciones directas de modificación (métodos set) sobre las propiedades. Las propiedades cambiarán aplicando la funcionalidad que ofrece 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. Los parámetros de este método son: la referencia al objeto usuario que realiza la puja y la cantidad por la que puja.

- La puja es aceptada si: la subasta está abierta, el crédito del usuario que la realiza es suficiente para la cantidad por la que puja, el usuario no es propietario de la subasta y la cantidad es mayor que la cantidad de la puja mayor, si la hubiera. Si la puja es aceptada, entonces se construye una puja y se almacena en la lista de pujas. Por último, esta operación finaliza indicando si la puja ha sido aceptada (retorna un valor booleano).
- pugar: versión sobrecargada del método anterior que permite pugar sin indicar la cantidad. Es decir, sólo se tiene como parámetro 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. Esta operación no tiene parámetros. Una subasta se puede ejecutar si se cumplen las siguientes condiciones: existe alguna puja o la subasta está abierta. 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. Establecer a falso el valor de la propiedad abierta. El método finaliza informando si la subasta ha podido ejecutarse o no (retorna un valor booleano)

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 aceptada, ya que es menor que la primera.
- Ejecuta la subasta.
- Muestra por la consola los créditos de los tres usuarios.

Realiza las siguientes operaciones:

1. Cuenta los usuarios que tienen cuenta de correo en Gmail. Muestra el resultado en la consola.

2. Mostrar por la consola los nombres de usuarios que sean propietarios de subastas ordenados por orden alfabético inverso.
3. Mostrar por la consola los nombres de los productos cuyas subastas hayan recibido alguna puja ordenados alfabéticamente.
4. Mostrar por la consola el nombre de los productos de aquellas subastas que hayan recibido pujas superiores a 50 euros.
5. Consultar si hay usuarios que hayan ganado alguna subasta y que sean propietarios de subastas.
6. Crea un conjunto vacío de pujas, añade a ese conjunto todas las pujas que se hayan realizado en las subastas.