

# Programación Orientada a Objetos

## Curso 2023/24

### Sesión 6

El objetivo de esta sesión de prácticas es utilizar herencia como mecanismo de reutilización de código donde se ha de cumplir la relación “es-un” entre las clases. Así mismo, también se podrá valorar la necesidad de la redefinición de métodos en la aplicación de la herencia.

#### Ejercicios

1. En la aplicación de las subastas queremos incorporar nuevos tipos de subastas. Estos tipos se definen del siguiente modo:

- **Subasta limitada:** es un tipo de subasta que fija el número máximo de pujas que se pueden hacer. Este número se establece en la construcción del objeto y no puede cambiar. Si se alcanza el máximo de pujas, la subasta se ejecuta automáticamente. Por último, se podrá consultar el número de pujas pendientes, esto es, el número de pujas que todavía puede admitir.
- **Subasta temporal:** es una subasta limitada que se caracteriza por permitir pujas dentro de un periodo fijo de tiempo expresado en horas. Esta propiedad se establece en la construcción y puede ser consultada. El tiempo empieza a contar en el momento de la creación de la subasta. Otra propiedad es el número de horas restantes que indica el tiempo restante para pujar en la subasta. Por tanto, este tipo de subasta añade una restricción a las condiciones generales de aceptación de pujas: sólo se puede pujar si no ha finalizado el tiempo establecido. Así mismo, el número máximo de pujas, que es la característica de las subastas limitadas, depende de la duración inicial de la subasta y se calcula como el doble de las horas establecidas como periodo de tiempo de la subasta. Por ejemplo, si la subasta es de 24 horas, sólo se admiten 48 pujas. Por último, este tipo de subastas sólo se pueden ejecutar si ha finalizado el tiempo.

**Gestión del tiempo:** Añade un **atributo de implementación**, llamado `inicio`, de la clase `java.time.LocalDateTime`. En la construcción este atributo será inicializado con el instante actual. El atributo de implementación se utilizará como soporte para calcular el tiempo transcurrido, por tanto, no es una propiedad de la clase y no se tiene que ofrecer un método de consulta.

Para el cálculo del tiempo transcurrido, se utilizará el tipo de datos `java.time.temporal.ChronoUnit`: `ChronoUnit.HOURS.between(inicio, fin);`, donde `inicio` corresponde con el atributo de implementación y `fin` con el instante actual.

- **Subasta mínima:** es un tipo de subasta que se caracteriza por tener una cantidad mínima para aceptar una puja como ganadora. Esta cantidad se establece en el constructor y puede modificarse en cualquier momento. La aceptación de las pujas tiene las mismas restricciones que se establecieron en la sesión 4. En cambio, para la ejecución de la subasta se añade como restricción que la puja mayor sea mayor o igual que la cantidad mínima. Por último, es posible cerrar directamente la subasta, esto es, sin necesidad de ser ejecutada.

2. Escribe el siguiente programa:

- Declara dos variables de tipo String que represente a los usuarios de nombre "Juan" y "Enrique".
- Declara la variable local *limitada* de tipo `SubastaLimitada` y asigna una subasta limitada que sólo permita 1 puja. El producto que ofrece es "Disco duro multimedia" y el usuario propietario es *juan*.
- Declara la variable local *temporal* de tipo `SubastaTemporal` y asigna una subasta temporal con un tiempo de subasta de 3 horas. El producto que ofrece es "Teclado" y el usuario propietario es *juan*.
- Declara la variable local *minima* de tipo `SubastaMinima` y asigna una subasta mínima con un precio mínimo de 100 euros. El producto que ofrece es "Impresora Láser" y el usuario propietario es *juan*.
- Crea una colección de subastas y añade todas las subastas anteriores.
- Recorre la colección de subastas:
  - o Muestra el nombre del producto que se está subastando
  - o Para cada una de las subastas aplica el método *pujar* con usuario *enrique* y cantidad 10 euros.
  - o De nuevo, el usuario *enrique* puja por 20 euros.
  - o Para cada una de las subastas: si la subasta no está cerrada, entonces se aplica el método *ejecutar*. Si no se ha podido ejecutar se muestra un mensaje por la consola informando.