

### TPP: Trabajo en clase Práctica 3

El siguiente ejercicio no está pensado para completarse en su totalidad durante la clase de hoy. Es recomendable completarlo para entender el uso del polimorfismo y la herencia. Al menos debería implementarse una clase derivada de *Producto* y otra de *Evento*, junto con la interfaz y testear su funcionamiento.

Crear una clase *Producto* con las propiedades: **Nombre**, **Descripción**, **Stock** y **Disponibilidad**. Crear los métodos:

- Reponer: Añade n unidades al stock.
- Comprar: Elimina n unidades al stock.

Crear una clase derivada *Camiseta*, con las propiedades nuevas: **Talla (string)** y **Color (string)**.

Crear otra clase derivada *Bono*, con las propiedades: **Minutos** (int).

Crear una clase *Evento* (**NO** deriva de *Producto*) con las propiedades: **Nombre**, **Descripción**, **FechaInicio**, **FechaFin**.

Crear una clase *Exhibición* derivada de *Evento*, con la propiedad: **NumAtletas**.

Crear una clase *Estancia* derivada de *Evento*, con las propiedades: **NumParticipantes** e **IncluyeComida** (bool).

Crear la **interfaz** *IFacturable* que describa la siguiente funcionalidad:

- GetBase (double): Indicará el precio base.
- GetTipolva (double): Indicará el tipo de IVA → 0.21, 0.10...
- GetIva: Calculará el IVA en función de Base y TipoIVA .
- GetPrecio (double): Devolverá el precio (Base + aplicación de IVA al base).
- GetConcepto (string) Devolverá un texto descriptivo.
- ToTicket(): Devolverá una cadena con los valores anteriores.

Debe **implementarse la interfaz *IFacturable* en las clases anteriores**, teniendo en cuenta que:

- Las camisetas se facturan al 0.21 de IVA. Base: un número que cambie según el color.
- Los bonos se facturan al 0.21 de IVA. Base: un número que cambie según los minutos.
- Las exhibiciones se facturan al 0.21 de IVA. Base: un número según el número de atletas.
- Las Estancias se facturan al 0.10 de IVA si incluyen comida; 0.21 en caso contrario. Base: un número en función del número de participantes y el número de días entre FechaFin y FechaInicio (propiedad Days de DateTime).

Discurrir una forma de **coleccionar objetos de las clases anteriores en una colección (decidida por el alumno)** denominada facturación. Imprimir por pantalla los siguientes totales:

- Listado de Tickets.
- Sumatorios de: Base e Iva.