

## Ejercicios UD4 POO 2

## Boletín 1 Herencia-Polimorfismo

1. Crear un programa con una clase Ordenador que permita modelar uno con características como capacidad de disco duro, frecuencia del procesador, precio base y marca. Crear dos clases hijas, llamadas Sobremesa y Portátil. Añadir a Sobremesa el atributo precioMontaje y al portátil otra cantidad para el precio del seguro de la pantalla.

La clase madre debe tener un método para calcular el precio de venta al público aplicando un % de ganancia del vendedor al precio base. Este método se debe reescribir en las clases hijas sumando la cantidad adecuada en cada caso.

Crear un main sencillo (no es necesario crear menú) para comprobar todo, basta llamar a los métodos en orden adecuado y mostrar el resultado de los cálculos que hace cada uno, es decir, el precio de venta al público de un objeto de cada tipo (un ordenador, un sobremesa y un portátil).

2. Escribir un programa que tenga una clase **Documento** y dos clases hijas, **Tarjeta de visita** y **Carta**. Crear los métodos necesarios para que, la cabecera de cada tipo de documento se imprima en pantalla de una forma diferente, según sus características. Como es un ejemplo académico, podemos hacer los métodos de impresión dentro de cada clase.

Crear un main para hacer una prueba con cada método llamado con objetos de las distintas clases. Por ejemplo, el documento genérico solo tiene un pequeño encabezado con los datos de la empresa, la tarjeta puede llevar, además, los datos de contacto de una persona y la carta, una fecha leída por teclado.

3. Los vehículos a motor pagan un determinado impuesto de circulación. La cantidad a pagar depende de la cilindrada, potencia en caballos y categoría de emisiones contaminantes (cero, ECO, tipo B y tipo C). Crear un programa para calcular la cantidad que debe pagar un vehículo dependiendo de sus características. (Los vehículos a motor pueden ser motocicletas, coches y furgonetas).

El impuesto se calcula con una cantidad fija cuyo valor depende del tipo de emisiones por el simple hecho de ser un vehículo con motor más:

- El 60 % de la cilindrada en el caso de las motocicletas.
- Otra fija a las furgonetas por ser transporte de mercancías.
- El 25 % de la potencia en el caso de los coches.

La jerarquía de clases no tiene mucho sentido en este ejemplo, pero está puesto para reescribir métodos. Probar todo en un main sencillo.



## POLIMORFISMO

4. Hacer un programa que imprima un ticket de compra, basándonos en un supermercado. Debemos crear la clase Producto genérica y tres hijas con un atributo más cada una (Alimentación, Ropa y Electrónica). Cada Producto genérico deberá estar caracterizado por el precio unitario, nombre, código de identificación, etc.

El precio de los productos de electrónica lleva un impuesto especial por ser de lujo y la alimentación un descuento cuando le quedan menos de 2 días para caducar.

La clase Línea de venta, con un producto y una cantidad y la clase Venta con un array de líneas de venta.

El ticket debe mostrar una venta completa con varios productos y varias cantidades. Se pueden crear los productos directamente en el main antes de empezar.

Agregar un método solo en la clase alimentación, que avise si al producto le quedan menos de 2 días para caducar.

Crear un main, crearemos directamente los objetos necesarios. Solo habrá la opción imprimir ticket y listar todos los productos guardados, avisando cuando a un producto de alimentación le falten menos de dos días para caducar.

Ctd.	Producto	Precio	Total
1	PATATAS BACON	3,90	3,90
1	ENS PASTA FRESCA	3,90	3,90
1	RAVIOLI SALMON	3,90	3,90
1	BOT REFRESCO	1,60	1,60
1	AGUA 0.5	1,00	1,00
1	CAFE	1,10	1,10
<b>TOTAL IVA INCLUIDO:</b>			<b>15,40</b>

Dia: 22/11/2013 Hora: 16:08  
 tipo Base Total  
 IVA10 14,00 1,40  
 Gracias por su visita  
 MESA: 033

5. Crear una clase Empleado con los atributos nombre, apellidos, sueldo base y nº de Empleado. Introducir los métodos necesarios y oportunos.

Crear una clase derivada de ella, llamada Vendedor que tenga como atributos la cantidad de ventas y un incentivo (un porcentaje de las ventas realizadas en un mes sumado a su sueldo base) y los métodos apropiados para manejarla.

Otra clase hija puede ser Gerente, cuyo sueldo es el base, pero al que se le quita un % de impuestos (consideraremos que al vendedor no se le quita nada para que el ejercicio tenga algo más de sentido).

Crear una clase Oficina caracterizada por una lista de Empleados donde se incluya un método que calcule el sueldo de un empleado pasándole como parámetro un empleado y otro método que calcule lo gastado por la oficina en total por el pago de sueldos a todos los empleados.

Probar el programa en una clase principal, instanciando un array de empleados guardando varios tipos incluyendo un empleado genérico y mostrando su sueldo. Si el empleado es un vendedor se le debe felicitar con un mensaje por pantalla cuando haya vendido más de una cantidad pasada como parámetro que será su objetivo de ventas.

6. Implementar un programa para un banco, donde se gestione la parte de cuentas de un usuario. Este banco solo y exclusivamente tiene 3 tipos de cuentas, cuenta corriente, cuenta joven y cuenta empresa. Cada una de ellas se caracteriza por:

- La cuenta corriente paga mantenimiento fijo y acumula puntos cuando se usa en el cajero.
- La cuenta joven no tiene mantenimiento y además, se le regala 1€ cada vez que hace un ingreso.
- La cuenta de empresa, no tiene cuota de mantenimiento, pero el banco cobra un euro a la empresa cuando saca dinero ya que está pensada para hacer transferencias y no usar dinero en efectivo.

El cliente tendrá un saldo, podrá sacar o meter dinero (reintegro e ingreso) por el cajero y dependiendo del tipo de cuenta se le hará un determinado cálculo y se le actualizará el saldo.

Hacer una clase oficina con un array de cuentas y probar los métodos anteriores, calcular el total de dinero que tiene la oficina entre todas las cuentas, cuánto se ha ganado la oficina por las veces que las cuentas de empresa han sacado dinero, cuánto ha gastado la oficina del banco en regalar el euro cuando se ingresa dinero en una cuenta joven, etc.

Ampliación: Quien lo desee, puede plantearse alguna clase más, como la clase cliente que puede tener varias cuentas o que en la clase Oficina también haya un atributo que sea la lista de Clientes.

**DE REFUERZO:** Este ejercicio no es obligatorio y puede servir para hacer todo lo visto en el tema poco a poco.

Construiremos varias clases y modificaremos otras paso a paso. Esto puede servir para aclarar todos los pasos que, en general, se pueden necesitar en un problema cualquiera. OJO, será necesario crear varios proyectos, cada vez que queramos probar una clase con una principal, lo haremos copiando el código de las clases que se repitan en un nuevo proyecto para tener todo guardado y poder ver cada caso por separado.

- Crear una clase llamada punto, con dos atributos (las coordenadas x e y). Añadir el constructor sin parámetros y el constructor con los dos parámetros.
  - Hacer una clase de prueba donde se use, los dos constructores, los métodos y se impriman las coordenadas del punto entre paréntesis (por ejemplo (5,3)).
  - Crear una nueva clase llamada círculo. Los atributos serán las coordenadas x e y del centro y el radio. Debe tener todos los constructores y métodos get y set posibles para todos los atributos.
  - Añadir a la clase anterior métodos para calcular el diámetro, la longitud de la circunferencia y el área.
  - Hacer otra clase prueba para la clase círculo. Se deben dar valores en el constructor, cambiarlos por otros leídos desde teclado y calcular con esos últimos el diámetro, la longitud de la circunferencia y el área.
  - Vamos a realizar ahora la clase círculo2 que hereda de la clase Punto. En ella el único nuevo atributo será el radio, pues las coordenadas ya están en la clase punto. Debe tener el constructor sin parámetros y el constructor con los tres parámetros (uso de super ( ) ). El correspondiente get y set para el atributo radio y los métodos de cálculo.
  - Ahora probaremos la clase anterior con una clase llamada PruebaCirculo2. Hacer lo mismo que en las anteriores pruebas.
  - Por último, intentad reescribir y usar polimorfismo con los métodos de cálculo añadiendo lo que consideréis oportuno.
2. Definir una clase figura y dos clases derivadas para representar un rectángulo y un triángulo. Crear un método área ( ) en cada una de las clases para calcular el área de la figura.

### DE AMPLIACIÓN:

Intenta crear un programa completo. Piensa lo que quieres hacer, el diagrama o jerarquía de clases (incluyendo una clase gestión), los atributos y métodos de cada una. En el main, se pueden ir realizando todos los casos estudiados: rescritura de métodos, sobrecarga, arrays de objetos, polimorfismo, clases abstractas... Puedes empezar por lo más sencillo y posteriormente añadir más detalles.