Ejercicios con herencia y polimorfismo

En los primeros tres ejercicios solo se debe realizar el diagrama UML

1. Supongamos que una **empresa** desea hacer la liquidación de sueldos de su personal. El mismo está compuesto por dos tipos de empleados: los administrativos que trabajan en las oficinas y los vendedores que trabajan haciendo ventas desde sus casas. De cada empleado se conoce su nombre y su cuil. De cada vendedor también se conoce el importe acumulado de las ventas que hizo en el mes y el porcentaje de comisión que cobra por las mismas.

   Todos perciben un sueldo básico y un premio por presentismo que es un porcentaje sobre el sueldo básico. Estos valores varían según el tipo de empleado. También a todos ellos se les descuentan los aportes de jubilación y de obra social, que suman un 17% de su sueldo bruto (el sueldo bruto es lo que suma, en este caso incluye el básico más el presentismo si corresponde). El modo de cálculo de los sueldos es el siguiente:

   - administrativos: sueldo básico, más presentismo, menos aportes.
   - vendedores: sueldo básico, menos aportes, más un porcentaje de comisión sobre el importe de sus ventas.

   La funcionalidad que se pide es calcularSueldoACobrar.

2. El **puerto** de la ciudad de BuenPuerto dispone de varias amarras en alquiler para embarcaciones de distinto tipo. Por cada una de ellas se almacena el número de amarra, la ubicación y si está disponible.

   Para cada embarcación que alquila una amarra se almacena su matrícula, eslora en metros, año de fabricación y nombre del dueño. Si se trata de un velero también se guarda la información de la cantidad de mástiles que posee. Si es una embarcación deportiva a motor, se almacena la potencia en HP del motor y el factor aerodinámico (un número entero). Por último, en el caso de los yates se almacena la cantidad de camarotes.

   Para calcular el costo del alquiler que se debe pagar por una amarra se debe realizar el cálculo siguiente: PrecioBase + Adicional

- Precio Base: se obtiene multiplicando un valor definido por la administración del puerto para cada embarcación (valor base) por la eslora del barco.

- Adicional: la administración define un valor adicional para cada barco. Luego, según su tipo se obtiene:
  - Veleros: multiplicando el valor adicional por la cantidad de mástiles.
  - Deportivos: multiplicando el valor adicional por el 50% de la potencia.
  - Yates: multiplicando el valor adicional por la cantidad de camarotes.

En particular, las embarcaciones deportivas deben tener además un índice de cálculo de potencia de 0.35 y un cálculo de consumo con la siguiente fórmula:

potencia en HP / factor aerodinámico * índice cálculo de potencia.

Las funcionalidades deseadas son:

- barcosConAlqMayorA() que recibirá un importe y devuelve la cantidad de barcos que abonan un alquiler mayor al importe recibido.
- barcoConMayorConsumo() que debe retornar el barco deportivo de mayor consumo.

3. La empresa que administra la concesión de la **Autopista a Ningunaparte,** quiere automatizar sus estaciones de peaje. Cada estación tiene cabinas, un id y una descripción. También tiene un método llamado darHoraActual() que devuelve la hora del momento en el que se invoca. Cada cabina tiene un número identificatorio y un solo medio de pago. Los medios de pago pueden ser Sube, Pase o efectivo. De cada uno de ellos sabemos su descripción y de Sube y de Pase también los días de demora en el cobro. Cada medio de pago aplica un descuento distinto sobre el importe a abonar:
   - para Sube es un 10%.
   - para Pase es un 15% de descuento si la demora es menor a 5 días, sino es un 12%.
   - el medio de pago efectivo por ahora no tiene un descuento pero podría tenerlo más adelante.

   De cada vehículo que pasa se sabe su patente y su categoría (AUTO, MOTO, CAMION). Cada categoría tiene una tarifa base a abonar.

Cuando un vehículo pase por una cabina se le cobrará la tarifa según su categoría. Si pasa por la cabina entre las 6 y las 10hs. o entre las 17 y las 20hs. se considera horario pico y tendrá un incremento del 8% sobre la tarifa. Finalmente según el medio de pago se aplicará el descuento correspondiente, resultando el importe final a cobrar.

Nos piden implementar las funcionalidades siguientes en la clase que corresponda:

- cobrar, recibiendo un vehículo y retornando el importe final a cobrar al mismo.
- mostrar por pantalla la lista de cabinas que reciben efectivo.
- calcular el promedio de demora en el cobro, tomando para el cálculo los días de demora de las cabinas que no aceptan efectivo

Para los siguientes ejercicios se pide realizar el diagrama UML y también escribir el código Java. Se deben desarrollar los métodos indicados por las funcionalidades y también cualquier otro que fuera necesario para cumplir con las mismas, en las clases en que corresponda.

4. Una **tienda de electrodomésticos** nos encarga crear un programa para emitir los tickets de cada venta que realiza. Cuando llega un cliente se ingresa el nombre del mismo y luego debe aparecer en pantalla una lista de los productos a la venta. Se elige uno y el mismo se agrega al "carrito de compra". Esto se repetirá hasta que se indique que no se agregarán más artículos (ingresando cero como opción elegida). Entonces se emitirá el ticket por pantalla mostrando el nombre del cliente y los artículos que compró.

   De la tienda conocemos su nombre y la lista de productos que comercializa.

   Los productos que vende son televisores, heladeras, lavarropas y cocinas. De todos ellos se conoce la marca, el modelo, el número de serie (número único que identifica el producto) y el precio de venta. Además de los televisores se sabe el tamaño de la pantalla (en pulgadas); de las heladeras se sabe la capacidad (en litros), de los lavarropas su capacidad (en kg.) y de las cocinas sus medidas de ancho y profundidad (en cm.)

   Los precios de venta de los televisores pueden tener un porcentaje de descuento por estar en promoción. Las cocinas que admiten también funcionar con gas envasado tienen un recargo del 10% en su precio.

Entonces las funcionalidades que se requiere implementar son las siguientes:

- Ingresar el nombre del cliente (en clase PruebaTienda).
- Realizar una venta (recibe el nombre del cliente y permite elegir los productos que compre, agregándolos al carrito de compra).
- Emitir su ticket por pantalla, mostrando los artículos elegidos y el total a pagar por la compra.

Se deben probar las funcionalidades solicitadas desde la clase PruebaTienda.

5. El programa que realizamos en el punto anterior funcionó muy bien y nos piden una modificación del mismo. Los **tickets** de cada venta deberán quedar guardados para poder emitir al fin del día la planilla de caja, que muestra para cada ticket su número, el nombre del cliente y su importe total. Los tickets deben tener una **numeración única y correlativa** automática para que coincidan con lo que se registra en Afip. Además, para cada artículo elegido se deberá ingresar también la cantidad de unidades que se desea comprar.

   En resumen, se debe modificar el programa del ejercicio anterior para que cumpla lo siguiente:

   - los Tickets deben guardarse junto con los Items de los mismos.
   - cuando se elige un producto, también se debe ingresar la cantidad de unidades.
   - se debe poder emitir la planilla de caja.
   - se debe modificar la clase PruebaTienda para que tenga un ciclo en el que se vayan realizando ventas hasta que se ingrese un nombre de cliente vacío.

6. Nuestro programa de facturación está cada vez más completo. Ahora nos solicitan que tenga **dos funciones nuevas:**

   - Permitir consultar por pantalla los artículos que tiene un ticket conociendo su número. Pregunta: ¿podemos reutilizar algún método ya existente para cumplir este requerimiento?
   - Obtener informes de las unidades vendidas de cada tipo de artículo. Deberá retornar la descripción del tipo de artículo y la cantidad de unidades vendidas del mismo.

7. Ahora nuestro programa quedará full full! Agregaremos al ticket la **forma de pago** del mismo. Algunos medios de pago tienen descuento. Las opciones son las siguientes:

   - EFECTIVO: esta forma de pago no tiene descuento.
   - CREDITO: descuento del 5%
   - DEBITO: descuento del 10%
   - BILLETERA: descuento del 12%

   Se deberá modificar lo siguiente en el programa:

   - Para realizar una venta, se deberá ingresar también el medio de pago elegido (además del nombre del cliente).
   - El cálculo del total del ticket debe tener en cuenta el descuento del medio de pago (si corresponde).

8. El **taller mecánico** UtnBoxes desea realizar un programa para administrar los servicios que realiza. Estos servicios son cambio de aceite, alineación y encendido. De cada servicio se conoce su descripción, su porcentaje de ganancia y la patente del auto al que se le efectuó. Para cada tipo de servicio varía su precio de venta y su precio de costo. Además cada servicio es *Detallable*, lo que permite mostrar por pantalla qué tipo de servicio es, la patente del vehículo y el precio de venta del servicio.

   El modo de cálculo del costo de los servicios varía para cada uno de ellos:

   - Cambio de aceite: se sabe la cantidad de horas que va a llevar. Es Cotizable por mano de obra y Cotizable por materiales. Esto significa que el costo del mismo es la suma del costo de las horas más el costo de los materiales.
   - Alineación: de este servicio se sabe la cantidad de horas que va a llevar, si se incluye también balanceo o no (atributo boolean) y un importe extra que es el costo del balanceo. Es Cotizable por mano de obra, por lo que su costo es el costo de las horas más el costo extra del balanceo (si se incluyó).
   - Encendido: es Cotizable por materiales. Su costo es el costo de los materiales.

   Los costos tienen dos formas de calcularse, dependiendo del tipo de servicio (si para su cálculo se incluyen horas, materiales o ambas):

   - Si el costo incluye horas de mano de obra, se calcula multiplicando la cantidad de horas utilizadas para el servicio por el valor fijo de la hora.

- Si el costo incluye materiales, el mismos se calcula sumando el costo de los materiales (material1+material2).

Finalmente, el precio de venta para todos los servicios se calcula agregando al precio de costo el porcentaje de ganancia (este es general y único).

Se deben implementar las siguientes funcionalidades en donde corresponda:

- método detallar()
- método calcularCostoHoras()
- método calcularCostoMateriales()
- método agregarServicio(). Recibe un servicio para agregar en la lista de servicios (si no es nulo). Devuelve true o false según pudo agregarlo o no.
- método listarServicios(): muestra la cantidad de servicios efectuados de cada tipo y además muestra el total de las ventas de todos los servicios (es la sumatoria de sus precios de venta).

9. La **fábrica de muebles** LaMesaLoca se dedica a la fabricación y venta de muebles de distintos tipos. Cada mueble tendrá un modelo, un costo base y un porcentaje de ganancia. Además los muebles deberán ser "*Mostrables*", que significa que puedan mostrar por pantalla el tipo de mueble, el modelo y su precio de venta.

   La fábrica produce muebles de los siguientes tipos, indicando sus características:

   - Mesa: largo, ancho y tipo de mesa (Moderna, Antigua y De Cristal).
   - Silla: alto y material de fabricación (Madera, Metal o Plástico).
   - Sillón: cantidad de cuerpos y tipo de tela.

   Todos los muebles deberán poder calcular e informar su precio de venta. Este precio será el valor de costo más el porcentaje de ganancia de cada uno.

   El costo de los muebles depende del tipo:

   - Mesas: se multiplica el costo base por un multiplicador dependiente del tipo de mesa más el 20% de la superficie.
   - Sillas: al costo base se le suma la multiplicación entre el alto y el multiplicador dependiente del material
   - Sillones: se multiplica el costo base por la cantidad de cuerpos por un porcentaje definido para el tipo de tela pedido.

La fábrica debe tener un listado de muebles fabricados y un nombre. Será responsable de fabricar los muebles (registrar los muebles fabricados), poder indicar si un modelo de mueble ha sido fabricado y por último, al ser Mostrable, deberá mostrar por pantalla su nombre, la cantidad de muebles fabricados de cada tipo y el importe total de los muebles fabricados.

10. La **clínica veterinaria** del doctor Vetebicho nos pide desarrollar un sistema para gestionar los datos de las mascotas que quedan internadas en la misma. Se deberá contar con las siguientes funcionalidades:

    - admitir mascotas para su cuidado
    - buscar mascotas en base a su nombre, que se supone único. Se devolverá la mascota o nulo en caso de que no la encuentre.
    - vacunar a todas las mascotas que no estén vacunadas al momento de ejecutar la acción.
    - la veterinaria es “mostrable”, por lo tanto se deben mostrar sus datos y la cantidad de mascotas por tipo que tenga alojadas.

    Las mascotas son animales, de tres especies distintas, perros, gatos y conejos. De ellas se sabe su nombre, su peso, si están vacunadas o no y un estado que podrá ser Enfermo, Hambriento o Feliz. Todos los tipos de mascotas pueden comer, indicar su nombre, actualizar su peso en base a un porcentaje, indicar si está vacunada y ser vacunada. Además, al ser animales podrán comer una cantidad de comida. Al comer todas las mascotas proceden igual, actualizando su estado y emitiendo un mensaje en base a su estado actual y actualizando su peso en un porcentaje que es la multiplicación de una constante propia de cada tipo y la cantidad de comida que reciba.

    Es conveniente definir las interfases Mostrable y Animal con los métodos que sea necesario.