Clases y objetos
Relación de ejercicios

1. Identifique los datos que decidiría utilizar para almacenar el estado de los siguientes objetos en función del contexto en el que se vayan a utilizar:

    a. Un punto en el espacio.

    b. Un segmento de recta.

    c. Un polígono.

    d. Una manzana (de las que se venden en un mercado).

    e. Una carta (en Correos)

    f. Un libro (en una biblioteca)

    g. Un libro (en una librería)

    h. Una canción (en una aplicación para un reproductor MP3).

    i. Una canción (en una emisora de radio)

    j. Un disco de música (en una tienda de música).

    k. Un disco de música (en una discoteca).

    l. Un teléfono móvil (en una tienda de telefonía)

    m. Un teléfono móvil (en el sistema de una empresa de telecomunicaciones)

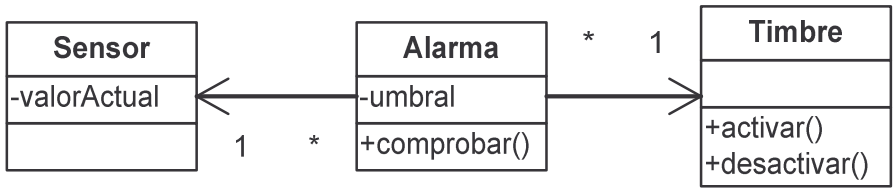
    n. Un ordenador (en una tienda de Informática)

    o. Un ordenador (en una red de ordenadores)

    p. Un ordenador (en el inventario de una organización)

    Declare las correspondientes clases en Java, defina los constructores que considere adecuados e implemente los correspondientes métodos para el acceso y la modificación del estado de los objetos (esto es, los métodos `get` y `set`).

2. Cree una clase denominada `Alarma` cuyos objetos activen un objeto de tipo `Timbre` cuando el valor medido por un `Sensor` supere un umbral preestablecido:



Implemente en Java todo el código necesario para el funcionamiento de la alarma, suponiendo que la alarma comprueba si debe activar o desactivar el timbre cuando se invoca el método `comprobar()`.

3. Cree una subclase de `Alarma` denominada `AlarmaLuminosa` que, además de activar el timbre, encienda una luz (que representaremos con un objeto de tipo `Bombilla`).

   NOTA: Procure eliminar la aparición de código duplicado al crear la subclase de `Alarma` y asegúrese de que, cuando se activa la alarma luminosa se enciende la luz de alarma y también suena la señal sonora asociada al timbre.

4. Diseñe jerarquías de clases para representar los siguientes conjuntos de objetos:

   a. Una colección de CDs, entre los cuales hay discos de música (CDs de audio), discos de música en MP3 (CD-ROMs con música), discos de aplicaciones (CD-ROMs con software) y discos de datos (CD-ROMs con datos y documentos).

   b. Los diferentes productos que se pueden encontrar en una tienda de electrónica, que tienen un conjunto de características comunes (precio, código de barras…) y una serie de características específicas de cada producto.

   c. Los objetos de una colección de monedas/billetes/sellos.

   Implemente en Java las jerarquías de clases que haya diseñado (incluyendo sus variables de instancia, sus constructores y sus métodos `get`/`set`). A continuación, escriba sendos programas que realicen las siguientes tareas:

   a. Buscar y mostrar todos los datos de un CD concreto (se recomienda definir el método toString en cada una de las subclases de CD).

   b. Crear un carrito de la compra en el que se pueden incluir productos y emitir un ticket en el que figuren los datos de cada producto del carrito, incluyendo su precio y el importe total de la compra.

   c. Un listado de todos los objetos coleccionables cuya descripción incluya una cadena de caracteres que el programa reciba como parámetro.

5. Implemente un programa que cree un objeto de la clase `Random` del paquete `java.util`, genere un número entero aleatoriamente y lo muestre en pantalla.

6. Cree un paquete denominado `documentos`…

    a. Incluya en él dos clases, `Factura` y `Pedido`, para representar facturas y pedidos, respectivamente.

    b. A continuación, ya fuera del paquete, cree un pequeño programa que cree objetos de ambos tipos y los muestre por pantalla.

    c. Añada un tercer tipo de documento, `PedidoUrgente`, que herede directamente de `Pedido`. Compruebe que el programa anterior sigue funcionando correctamente si reemplazamos un `Pedido` por un `PedidoUrgente`.

    d. Cree un nuevo tipo de documento, denominado `Contrato`, e inclúyalo en el subpaquete `documentos.RRHH`. En este último paquete, incluya también un tipo de documento `CV` para representar el currículum vitae de una persona.

    e. Si no lo ha hecho ya, cree una clase genérica `Documento` de la que hereden (directa o indirectamente) todas las demás clases que hemos definido para representar distintos tipos de documentos.

    f. Implemente un pequeño programa que cree un documento de un tipo seleccionado por el usuario. Muestre por pantalla el documento independientemente del tipo concreto de documento que se haya creado en el paso anterior.

OBSERVACIONES:

- Para cada clase que defina, determine qué miembros de la clase han de ser públicos (`public`), cuáles han de mantenerse privados (`private`) y, si lo considera oportuno, cuáles serían miembros protegidos (`protected`).

- Tenga en cuenta que no siempre se debe permitir la modificación desde el exterior de una variable de instancia (esto es, habrá variables de instancia a las que asociemos un método `get` pero no un método `set` y, de hacerlo, éste puede que sea privado o protegido).

- Analice también qué métodos de una clase deben declararse con la palabra reservada final para que no se puedan redefinir en subclases y qué clases han de ser “finales” (esto es, aquellas clases de las que no queramos permitir que se creen subclases).

- En los distintos programas de esta relación de ejercicios puede resultar necesaria la creación de colecciones de objetos de distintos tipos (p.ej. arrays de CDs, productos, objetos coleccionables o documentos).