

Relación de ejercicios entregables

1. Se desea crear una superclase llamada **Electrodomestico** con las siguientes características:

- Sus atributos son **precio**, **color**, **consumo energético** (letras entre A y F) y **peso**.
- Los métodos que implementara serán:
 - Métodos get de todos los atributos.
 - **comprobarConsumoEnergetico(char letra)**: comprueba que la letra es correcta, sino es correcta usara la letra por defecto. Se invocara al crear el objeto y no será visible.
 - **comprobarColor(String color)**: comprueba que el color es correcto, sino lo es usa el color por defecto. Se invocará al crear el objeto y no será visible.
 - **precioFinal()**: según el consumo energético, aumentara su precio, y según su tamaño, también. Esta es la lista de precios:

Crear una subclase llamada **Lavadora** con las siguientes características:

- Su atributo es **carga**, además de los atributos heredados.
- Por defecto, la carga es de 5 kg. Usa una constante para ello.
- Los constructores que se implementarán serán:
 - Un constructor por defecto.
 - Un constructor con el precio y peso. El resto por defecto.
 - Un constructor con la carga y el resto de atributos heredados. Recuerda que debes llamar al constructor de la clase padre.
- Los métodos que se implementara serán:
 - Método get de carga.
 - **precioFinal()**: si tiene una carga mayor de 30 kg, aumentara el precio \$50, sino es así no se incrementara el precio.

Crear una subclase llamada **Televisión** con las siguientes características:

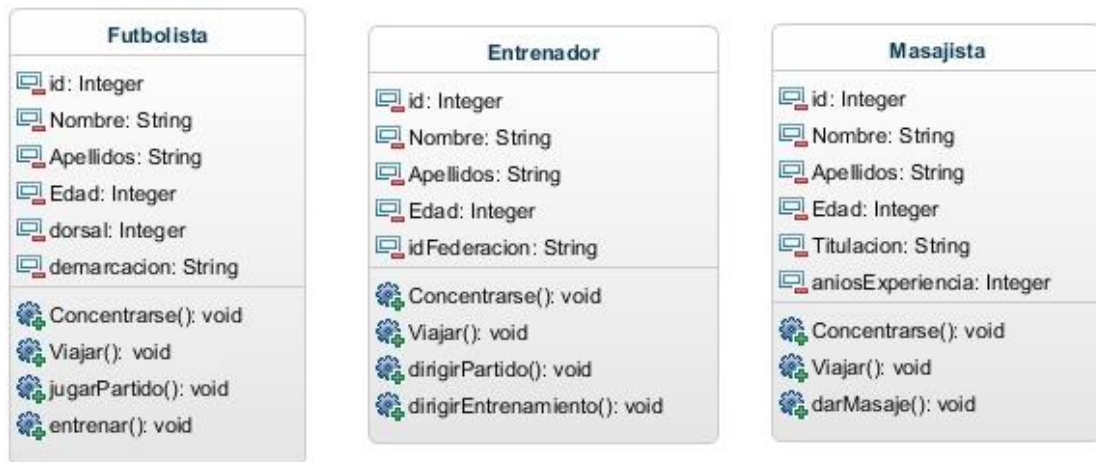
- Sus atributos son **resolución** (en pulgadas) y **sintonizador TDT** (booleano), además de los atributos heredados.
- Por defecto, la resolución será de 20 pulgadas y el sintonizador será false.
- Los constructores que se implementarán serán:
 - Un constructor por defecto.
 - Un constructor con el precio y peso. El resto por defecto.
 - Un constructor con la resolución, sintonizador TDT y el resto de atributos heredados.
- Los métodos que se implementara serán:

- Método **get** de resolución y sintonizador TDT.
 - **precioFinal()**: si tiene una resolución mayor de 40 pulgadas, se incrementara el precio un 30% y si tiene un sintonizador TDT incorporado, aumentara \$50.
2. Crear la super clase llamada **Persona** que siga las siguientes condiciones:
- Sus atributos son: **nombre**, **edad**, **CI**, **sexo** (H hombre, M mujer), **peso** y **altura**.
 - Se implantarán varios constructores:
 - Un constructor por defecto.
 - Un constructor con el nombre, edad y sexo, el resto por defecto.
 - Un constructor con todos los atributos como parámetro.
 - Los métodos que se implementaran son:
 - **calcularIMC()**: calculara si la persona está en su peso ideal (peso en kg/(altura² en m)), devuelve un -1 si está por debajo de su peso ideal, un 0 si está en su peso ideal y un 1 si tiene sobrepeso.
 - **esMayorDeEdad()**: indica si es mayor de edad, devuelve un booleano.
 - **comprobarSexo(char sexo)**: comprueba que el sexo introducido es correcto. Si no es correcto, será H.
 - **toString()**: devuelve toda la información del objeto.
 - **generaCI()**: genera un numero aleatorio de 10 cifras. Este método será invocado cuando se construya el objeto.
3. Se plantea desarrollar un programa Java que permita la gestión de una empresa agroalimentaria que trabaja con tres tipos de productos: productos frescos, productos refrigerados y productos congelados. Todos los productos llevan esta información común: fecha de caducidad y número de lote. A su vez, cada tipo de producto lleva alguna información específica. Los productos frescos deben llevar la fecha de envasado y el país de origen. Los productos refrigerados deben llevar el código del organismo de supervisión alimentaria, la fecha de envasado, la temperatura de mantenimiento recomendada y el país de origen. Los productos congelados deben llevar la fecha de envasado, el país de origen y la temperatura de mantenimiento recomendada.

Hay tres tipos de productos congelados: congelados por aire, congelados por agua y congelados por nitrógeno. Los productos congelados por aire deben llevar la información de la composición del aire con que fue congelado (% de nitrógeno, % de oxígeno, % de dióxido de carbono y % de vapor de agua). Los productos congelados por agua deben llevar la

información de la salinidad del agua con que se realizó la congelación en gramos de sal por litro de agua. Los productos congelados por nitrógeno deben llevar la información del método de congelación empleado y del tiempo de exposición al nitrógeno expresada en segundos.

4. Se desea simular el comportamiento que tendrían los diferentes integrantes de la selección ecuatoriana de futbol; tanto los Futbolistas como el cuerpo técnico (Entrenadores, Masajistas, etc...). Para simular este comportamiento se definen tres clases que van a representar a objetos Futbolista, Entrenador y Masajista. De cada uno se tienen los siguientes datos:



5. Se requiere realizar un programa que modele diferentes tipos de inmuebles. Cada inmueble tiene los siguientes atributos: identificador inmobiliario (tipo entero); área en metros cuadrados (tipo entero) y dirección (tipo String).

Los inmuebles para vivienda pueden ser casas o apartamentos. Los inmuebles para vivienda tienen los siguientes atributos: número de habitaciones y número de baños. Las casas pueden ser casas rurales o casas urbanas, su atributo es la cantidad de pisos que poseen. Los atributos de casas rurales son la distancia a la cabecera municipal y la altitud sobre el nivel del mar. Las casas urbanas pueden estar en un conjunto cerrado o ser independientes. A su vez, las casas en conjunto cerrado tienen como atributo el valor de la administración y si incluyen o no áreas comunes como piscinas y campos deportivos. De otro lado, los apartamentos pueden ser apartaestudios o apartamentos familiares. Los apartamentos pagan un valor de administración, mientras que los apartaestudios tienen una sola habitación. Los locales se clasifican en locales comerciales y oficinas. Los locales tienen como atributo su localización (si es interno o da a la calle). Los locales comerciales tienen un atributo para conocer el centro comercial donde están establecidos.

Las oficinas tienen como atributo un valor boolean para determinar si son del Gobierno. Cada inmueble tiene un valor de compra. Este depende del área de cada inmueble según la tabla:

Inmueble	Valor por metro cuadrado
Casa rural	1.500.000€
Casa en conjunto cerrado	2.500.000 €
Casa independiente	3.000.000 €
Apartaestudio	1.500.000 €
Apartamento familiar	2.000.000 €
Local comercial	3.000.000 €
Oficina	3.500.000 €

6. Se tiene una jerarquía taxonómica con los siguientes animales:

- Animal es la clase raíz con los atributos: sonidos, alimentos, hábitat y nombre científico (todos de tipo String). Esta clase tiene los siguientes métodos abstractos:
 - public abstract String getNombreCientífico()
 - public abstract String getSonido()
 - public abstract String getAlimentos()
 - public abstract String getHábitat()
- Los cánidos y los felinos son subclases de Animal.
- Los perros son cánidos, su sonido es el ladrido, su alimentación es carnívora, su hábitat es doméstico y su nombre científico es Canis lupus familiaris.
- Los lobos son cánidos, su sonido es el aullido, su alimentación es carnívora, su hábitat es el bosque y su nombre científico es Canis lupus.
- Los leones son felinos, su sonido es el rugido, su alimentación es carnívora, su hábitat es la pradera y su nombre científico es Panthera leo.
- Los gatos son felinos, su sonido es el maullido, su alimentación son los ratones, su hábitat es doméstico y su nombre científico es Felis silvestris catus.

Además, se requiere en una clase de prueba para desarrollar un método main que genere un array de animales y la pantalla debe mostrar los valores de sus atributos.

7. Definir una clase abstracta denominada Numérica que tenga los siguientes métodos abstractos:

- public String toString(): convierte el número a String.

- public boolean equals (Object ob): compara el objeto con el parámetro.
 - public Numérica sumar(Numérica número): retorna la suma de los dos números.
 - public Numérica restar(Numérica número): retorna la resta de los dos números.
 - public Numérica multiplicar(Numérica número): retorna la multiplicación de los dos números.
 - public Numérica dividir(Numérica número): retorna la división de los dos números.
8. Definir una clase Fracción que representa un número fraccionario, el cual hereda de la clase Numérica y tiene dos atributos (tipo int) que representan el numerador y denominador de la fracción. Se deben implementar todos los métodos heredados.
9. En una carrera ciclística, un equipo está conformado por un conjunto de ciclistas y se identifica por el nombre del equipo (tipo String), la suma de los tiempos de carrera de sus ciclistas en minutos (atributo estático) y país del equipo. Sus atributos deben ser privados.

Un ciclista es una clase abstracta que se describe con varios atributos: identificador (de tipo int), nombre del ciclista y tiempo acumulado de carrera (en minutos, con valor inicial cero). Los atributos deben ser privados. Un ciclista tiene un método abstracto imprimirTipo que devuelve un String.

Los ciclistas se clasifican de acuerdo con su especialidad (sus atributos deben ser privados y sus métodos protegidos). Estas especialidades no son clases abstractas y heredan los siguientes aspectos de la clase Ciclista:

- Velocista: tiene nuevos atributos como potencia promedio (en vatios) y velocidad promedio en sprint (Km/h) (ambos de tipo double).
- Escalador: tiene nuevos atributos como aceleración promedio en subida (m/s²) y grado de rampa soportada (grados) (ambos de tipo float).
- Contrarrelojista: tiene un nuevo atributo, velocidad máxima (km/h).

Definir clases y métodos para el ciclista y sus clases hijas para realizar las siguientes acciones:

- Constructores para cada clase (deben llamar a los constructores de la clase padre en las clases donde se requiera).
- Métodos get y set para cada atributo de cada clase.

- Imprimir los datos de un ciclista. Debe invocar el método de la clase padre e imprimir los valores de los atributos propios.
- Método imprimirTipo que devuelve un String con el texto “Es un xxx”. Donde xxx es la clase a la que pertenece.

La clase Equipo debe tener los siguientes métodos protegidos:

- Métodos get y set para cada atributo de la clase.
- Imprimir los datos del equipo en pantalla.
- Añadir un ciclista a un equipo.
- Calcular el total de tiempos de los ciclistas del equipo (suma de los tiempos de carrera de sus ciclistas, su atributo estático).
- Listar los nombres de todos los ciclistas que conforman el equipo.
- Dado un identificador de un ciclista por teclado, es necesario imprimir en pantalla los datos del ciclista. Si no existe, debe aparecer el mensaje correspondiente.

En una clase de prueba, en un método main se debe crear un equipo y agregar ciclistas de diferentes tipos.

10. Un médico posee como atributo su nombre y métodos get y set de dicho atributo. Los pediatras y ortopedistas son dos tipos de médicos. Estas dos subclases se modelan como subclases de médico.

Los pediatras pueden ser pediatras neurólogos o pediatras psicológicos. Esta tipología se modela como un atributo enumerado de pediatra. De igual manera, los ortopedistas cuentan con un atributo enumerado para describir su tipo que puede ser maxilofacial o pediátrica. Estas subclases cuentan con métodos get y set.

En una clase de prueba, en su método main, se solicita crear un vector con objetos tanto de tipo pediatra como ortopedistas. Luego, en pantalla se debe mostrar qué tipo de objeto está ubicado en cada posición del vector.

11. Se pretende desarrollar un conjunto de clases que representen, de forma simplificada, a una hipotética empresa dedicada a vender un producto. A continuación, se describen las características básicas de estas clases:
- Empleado. Clase básica que describe a un empleado. Incluye sus datos personales (nombre, apellidos, DNI, dirección) y algunos datos tales como los años de antigüedad, teléfono de contacto y su salario. Incluye también información de quién es el empleado que lo supervisa (Empleado *). Tendrá, al menos, las siguientes funciones miembro:
 - Constructores para definir correctamente un empleado, a partir de su nombre, apellidos, DNI, dirección, teléfono y salario.

- Imprimir (A través de los operadores de E/S redefinidos) •
 - Cambiar supervisor
 - Incrementar salario
- Secretario. Tiene despacho, número de fax e incrementa su salario un 5% anual. Tendrá, al menos, las siguientes funciones miembro:
 - Constructores (debe rellenar la información personal y los datos principales)
 - Imprimir (debe imprimir sus datos personales y su puesto en la empresa).
- Vendedor. Tiene coche de la empresa (identificado por la matrícula, marca y modelo), teléfono móvil, área de venta, lista de clientes y porcentaje que se lleva de las ventas en concepto de comisiones. Incrementa su salario un 10% anual. Tendrá, al menos, las siguientes funciones miembro: •
 - Constructores (debe rellenar la información personal y los datos principales)
 - Imprimir (debe imprimir sus datos personales y su puesto en la empresa).
 - Dar de alta un nuevo cliente.
 - Dar de baja un cliente.
 - Cambiar de coche.
- Jefe de zona. Tiene despacho, tiene un secretario a su cargo, una lista de vendedores a su cargo y tiene coche de la empresa (identificado por la matrícula, marca y modelo). Incrementa su salario un 20% anual. Tendrá, al menos, las siguientes funciones miembro:
 - Constructores (debe rellenar la información personal y los datos principales)
 - Imprimir (debe imprimir sus datos personales y su puesto en la empresa).
 - Cambiar de secretario.
 - Cambiar de coche.
 - Dar de alta y de baja un nuevo vendedor en su zona.

Todos los empleados son vendedores, jefes de zona o secretarios. Hacer un programa de prueba que muestre cómo funciona. Probar, especialmente, que el método incrementar salario se comparta bien, según el empleado que sea así es la subida.

12. Se te ha encargado desarrollar un sistema de gestión de empleados para una empresa. El sistema debe permitir la gestión de diferentes tipos de empleados, como empleados a tiempo completo y empleados a tiempo parcial. Además, se requiere calcular el salario mensual de cada empleado, considerando sus respectivas características.

Requisitos:

1. Crea una clase abstracta llamada **Empleado** que contenga los siguientes atributos protegidos:
 - **nombre** (String)
 - **codigoEmpleado** (int)
 - **salarioBase** (double)
2. Define un constructor en la clase **Empleado** que permita inicializar los atributos mencionados.
3. Crea dos clases que hereden de **Empleado**: **EmpleadoTiempoCompleto** y **EmpleadoTiempoParcial**.
4. La clase **EmpleadoTiempoCompleto** debe tener un atributo adicional llamado **horasExtras** (int) y un método para calcular el salario total que incluya las horas extras.
5. La clase **EmpleadoTiempoParcial** debe tener un atributo adicional llamado **horasTrabajadas** (int) y un método para calcular el salario total basado en las horas trabajadas.
6. Implementa un método abstracto llamado **calcularSalario** en la clase **Empleado**. Este método debe ser implementado de manera específica en cada una de las clases derivadas.
7. Crea una clase llamada **SistemaGestionEmpleados** que contenga un array de objetos de tipo **Empleado**. La clase debe tener un método para calcular y mostrar el salario total de todos los empleados almacenados en el array.

Instrucciones adicionales:

- Implementa un programa principal en la clase **Main** que demuestre la creación de instancias de empleados, la asignación de valores a sus atributos y el cálculo del salario total utilizando polimorfismo.
- Asegúrate de manejar adecuadamente la entrada de datos para los diferentes tipos de empleados.