

Ejercicio 1

Se pide hacer una aplicación en Java que contenga los siguientes elementos, y lleve a cabo las tareas que se describen a continuación.

a) Crea una clase abstracta llamada **“Electrodomestico”** con los siguientes atributos y métodos:

- Atributo privado de tipo real de nombre “potencia”. Representa la potencia consumida por hora en Kw/h.
- Atributo privado de tipo LocalDate “fechaCompra”. Se guarda en el formato por defecto yyyy-MM-dd.
- getter y setter necesarios.
- Un constructor con parámetros para todos los atributos.
- Otro constructor con solo la potencia como parámetro, la fecha será la del día actual.
- Método abstracto **getConsumo**, que recibe como parámetro el número medio de horas que está en funcionamiento el electrodoméstico diariamente y devuelve el consumo diario en Kw. Este método se desarrolla en las clases hijas, ya que puede haber una variación en los Kw/h en determinadas condiciones.
- Método **getCosteConsumo** que nos permita calcular el precio a pagar por los Kw consumidos en función del número de horas de uso y el coste por hora, que son los parámetros que recibe.
- toString() con el valor de todos los atributos de un electrodoméstico (en una línea y separados por “**”) según se ve en el ejemplo de ejecución).

Crea la clase **“Lavadora”** que hereda de “Electrodomestico” y tiene las siguientes características:

- Se añade un booleano aguaCaliente para indicar si se usa con agua caliente o no.
- Un constructor con parámetros para todos los atributos.
- Otro constructor con solo la potencia como parámetro. Para el resto de los atributos:
 - La fecha de compra, que será la de hoy.
 - El agua caliente, que siempre es true en este constructor.
- getter y setter necesarios.
- Método **getConsumo** implementado de forma que si funciona con agua caliente, se ve incrementado el consumo por hora en un 20%.
- Método toString(), con el valor de todos los atributos (en una línea y separados por “**”) según se ve en el ejemplo).

Crea la clase **“Frigorifico”** que hereda de “Electrodomestico” y tiene las siguientes características:

- Se añade un booleano modoVacaciones para indicar si el aparato está funcionando en modo vacaciones o no.
- Un constructor con parámetros para todos los atributos.
- Otro constructor con solo la potencia como parámetro. Para el resto de los atributos:

- La fecha de compra, que será la de hoy.
- El modo vacaciones, que siempre es false en este constructor.
- getter y setter necesarios.
- Método **getConsumo** implementado de forma que si funciona en modo vacaciones, se ve decrementado el consumo por hora en un 10%.
- Método **toString()**, con el valor de todos los atributos (en una línea y separados por “**”) según se ve en el ejemplo).

b) Crea una clase de nombre “**Tienda**” que tenga como atributos el nombre de la tienda donde se van a vender los electrodomésticos) y el ArrayList (atributo privado) donde estos se van a guardar. Tendrá también los siguientes métodos:

- Método de nombre **ordenar** que ordene los electrodomésticos del ArrayList por la fecha descendentemente.
- Método de nombre **mostrarElectrodomesticos** que reciba el precio del Kw como parámetro. Para cada objeto creado se mostrará su gasto medio diario en euros.
- Método “**toString**” para mostrar cada electrodoméstico con todos sus valores. Uno por línea.

c) Crea la clase “**Main**” con su método main, que hace lo siguiente:

- Crea varios objetos de tipo Lavadora y Frigorífico, añádelos al ArrayList de la clase Tienda creada en el apartado b). Los objetos créalos con datos fijados en el programa (no se piden al usuario), de manera que se prueben todas las posibilidades que has programado para los distintos tipos de electrodoméstico (con fecha de compra y sin ella, agua caliente y fría, modo vacaciones y modo normal).
- Mostrar ordenado el arraylist de electrodomésticos usando el método **toString()**.
- Ejecuta el método **mostrarElectrodomesticos()** de la clase Tienda.

d) Crea una clase de nombres Constantes donde tendrás las constantes con el número medio de horas de funcionamiento de cada electrodoméstico, precio del Kw y nombre de la tienda. Será 24 horas para los frigoríficos, 3 horas para las lavadoras y 0,5 € el precio del Kw.

A continuación, se muestra un ejemplo de ejecución.

*****Objetos ordenados por fecha de compra descendentemente:*****

THE BEST

```
potencia=1.7 ** fecha de compra=2024-05-24 ** Funcionando con agua caliente
potencia=0.85 ** fecha de compra=2024-05-24 ** Funcionamiento en modo normal
potencia=0.75 ** fecha de compra=2024-02-02 ** Funcionamiento en modo normal
potencia=1.75 ** fecha de compra=2024-01-01 ** Funcionando con agua caliente
potencia=1.57 ** fecha de compra=2023-05-02 ** Funcionando con agua fria
potencia=0.7 ** fecha de compra=2020-04-03 ** Funcionamiento en modo vacaciones
```

*****Gasto medio diario segun características:*****

```
Lavadora -> Con agua caliente 3,06 euros es el coste medio diario.
Frigorifico -> En modo normal 10,20 euros es el coste medio diario.
Frigorifico -> En modo normal 9,00 euros es el coste medio diario.
Lavadora -> Con agua caliente 3,15 euros es el coste medio diario.
Lavadora -> Con agua fria 2,36 euros es el coste medio diario.
Frigorifico -> En modo vacaciones 7,56 euros es el coste medio diario.
```

Ejercicio 1.

Una escuela tiene tres tipos de alumnos: a tiempo total, a tiempo parcial y a distancia.

- De **todos los alumnos** se almacena su nombre (nombre y apellido en el mismo atributo) y su DNI.
- De los **alumnos a tiempo parcial**, además se almacena el porcentaje de tiempo total que estudian (solo **30%, 50% o 70%** son valores válidos, crea un **enumerado** para controlarlo).
- De los **alumnos a distancia** se guarda su usuario y clave de acceso a la plataforma.
- Los de **tiempo total** no necesitan ningún atributo extra.

En este ejercicio, se pide:

1.- Diseña las clases (con sus constructores, getters, setters y toString) de forma que se pueda ampliar el tipo de alumnos que tenemos inicialmente sin hacer cambios en los existentes.

2.- Enumerado con los 3 únicos porcentajes mencionados.

3.- Para los **alumnos a tiempo parcial**, si se introduce otro valor distinto a los permitidos en los porcentajes, se lanza en el constructor una excepción propia de nombre **ExcepcionParcial**. El tratamiento de la excepción tendrá lugar en la llamada al constructor si se ha producido la excepción.

4.- Crea una clase de nombre Centro en la que se cree un ArrayList para guardar los distintos alumnos que iremos creando. Esta lista **no será accesible desde fuera de la clase de forma directa**. Se puntúa en este apartado el constructor y el método para añadir alumno.

5.- La aplicación tiene también que tener la capacidad de **ordenar los alumnos por su dni, descendentemente**. Para ello, debes hacer un método en la clase Centro que ordene el ArrayList de alumnos de la forma pedida.

6.- Para poder mostrar todos los alumnos, debes hacer un **método en la clase Centro** que los muestre indicando de qué tipo de alumno se trata en cada caso.

7.- Crea una **interfaz de nombre Becable**, con un método:

public int valorBeca()

Debes usar ese método para calcular la cuantía de la beca de cada tipo de alumno de forma que:

- Los alumnos a **distancia** siempre cobrarán 2000€
- Los alumnos a **tiempo total** siempre cobrarán 3000€
- Los alumnos a **tiempo parcial**:
 - Porcentaje de tiempo 30% cobrarán 1000
 - Porcentaje de tiempo 50% cobrarán 1500
 - Porcentaje de tiempo 70% cobrarán 2000

8.- Realiza un programa principal en el que crees **9** alumnos, **3 de cada tipo**, que vas a guardar en el **ArrayList de alumnos**. No hace falta menú, **no pidas los datos por teclado**, escribe los datos que quieras **directamente** en el código (uno detrás de otro para no perder tiempo con introducción de tantos datos).

A continuación, muestra cada uno de los alumnos junto con la cuantía de la beca que recibirá.