

Examen de Prácticas de Introducción al Software (Ingeniería Informática)

Septiembre 2015

Desde el 1 de Julio de 2015 se ha puesto en marcha en España un sistema de facturación por horas del consumo eléctrico de los hogares. Con el nuevo sistema, las compañías facturarán los consumos por horas, teniendo en cuenta que cada hora tiene un precio distinto. Se desea hacer parte de un programa que permita al cliente adaptar los consumos de sus electrodomésticos y así reducir el importe de su recibo.

Cada electrodoméstico se almacena en el programa mediante un objeto de la clase Electrodomestico cuyo diagrama se muestra abajo y que ya está implementada. Sus atributos son:

- nombre: es un texto que describe el electrodoméstico
- horaInicio: hora de encendido (de 0 a 23); se enciende al principio de esta hora
- horaFinal: hora de apagado (de 0 a 23); se apaga al final de esta hora
- consumo: consumo promedio en kW

Si por ejemplo un electrodoméstico funciona solo durante una hora, pongamos las 20h, entonces horaInicio=20 y horaFinal=20.

Como puede observarse, se dispone de un constructor al que se le pasan los datos del electrodoméstico. También hay métodos observadores, uno para cada atributo:

Electrodomestico	Hogar
-String nombre -int horaInicio -int horaFinal -double consumo	-ArrayList<Electrodomestico> lista -double[] precioLuz -boolean precioRellenado
+Electrodomestico (String nombre, int horaInicio, int horaFinal, double consumo) +String nombre() +int horaInicio() +int horaFinal() +double consumo()	+Hogar(String[] electrodomésticos) +void obtenerPrecioLuz() +ArrayList<Electrodomestico> encendidos(int hora) +Electrodomestico masCaro() +double gastoEnergetico (String nombre)

Lo que se pide es implementar en Java parte de la clase Hogar cuyo diagrama de clases se muestra arriba. La clase dispone de los siguientes atributos:

- lista: guarda una lista de objetos de la clase Electrodomestico
- precioLuz: es un array de 24 casillas conteniendo los precios de la luz en cada hora del día, en Euros/kWh
- precioRellenado: booleano que indica si el array precioLuz tiene sus casillas rellenas con datos válidos o no.

Algunos de los métodos de la clase Hogar se dan ya hechos:

- *constructor*: recibe como parámetro un array de strings que contiene los datos de todos los electrodomésticos de la casa, crea un objeto de la clase Electrodoméstico con cada uno y añade estos objetos al atributo lista. El parámetro electrodomésticos contiene un electrodoméstico en cada casilla, con el formato que se muestra en este ejemplo, donde el orden de los datos es: Nombre, horaInicio, horaFinal, consumo(kW). Ejemplo:

```
"Lavaplatos    20        21        1.3"
```

Además, da valor al atributo precioLuz creando para él un array con 24 casillas, pero no rellena sus valores. Por último pone precioRellenado a False.

Se puede suponer que el array de strings es correcto y que horaFinal >= horaInicio para cada caso.

- obtenerPrecioLuz(): Se conecta por Internet a un servidor en el que se recogen los precios de la luz en cada hora y con ellos se rellena el array precioLuz. Además pone precioRellenado a true.

De este método se proporciona ya hecha una simulación que pone unos precios concretos.

Se pide implementar los siguientes métodos de la clase Hogar:

- encendidos(): retorna un ArrayList conteniendo los electrodomésticos que se encuentran encendidos a la hora indicada por el parámetro hora.
- masCaro(): Retorna el electrodoméstico más caro en coste diario de la luz. El coste diario de un electrodoméstico se calcula como

$$\text{coste} = \sum_{h = \text{horaInicio}}^{\text{horaFinal}} \text{consumo} \cdot \text{precioLuz}[h]$$

En este método si precioRellenado es False se retorna null.

- gastoEnergetico(): Busca el electrodoméstico cuyo nombre coincide con el parámetro nombre. Si lo encuentra retorna el gasto energético diario de ese electrodoméstico, que es el número de horas en que está encendido por su consumo: (horaFinal-horaInicio+1)*consumo. Si no lo encuentra o si precioRellenado es False retorna Double.NaN.

Finalmente, se pide hacer un *programa principal* en una clase aparte que haga lo siguiente:

- a. Crea un objeto de la clase Hogar a partir del siguiente array:

```
{"Lavaplatos    20        21        1.3",
 "Lavadora     21        23        1.8",
 "Aspiradora   10        11        2.3",
 "Cafetera     8         9         0.5"}
```

- b. Invoca el método masCaro() y comprueba que retorna null, ya que los precios aún no se han relleno. Así lo indica con un mensaje en pantalla.
- c. Invoca el método obtenerPrecioLuz() para obtener (la simulación de) los precios del día publicados en Internet.

- d. Muestra en pantalla los nombres de los electrodomésticos que se encuentran encendidos a las 21 horas.
- e. Muestra en pantalla el nombre del electrodoméstico más caro en coste diario de la luz.
- f. Muestra en pantalla el gasto energético de la "Lavadora".
- g. Comprueba que al obtener el gasto energético de un electrodoméstico inexistente se obtiene Double.NaN y así lo indica con un mensaje en pantalla.

Entregar: Un informe según la plantilla que se encuentra en moodle, y que contendrá el código de la clase Hogar, el código del programa principal y una captura de la ejecución del programa principal.

Valoración:

- encendidos: 2 puntos
- masCaro: 2 puntos
- gastoEnergetico: 2 puntos
- programa principal: 2 puntos
- captura de pantalla de la ejecución del programa principal: 2 puntos