



NOME _____ DATA _____

Se desexa implantar un sistema de información do control e facturación do consumo eléctrico para unha nova compañía. En un principio, se vai a facilitar un menú coas seguintes opcións:

* Alta de Cliente * Acceso ós Datos de Consumo

- No 'Alta de Cliente', se creará un novo Suministro protexido por unha password para un cliente que quedará almacenado na base de datos da empresa. Os datos relevantes, ademais dos datos persoais do cliente, serán:

Tipo de Tarifa: (TUR,2TR,LIB) **Potencia Contratada:** (Nº máx en KW)

Código de suministro: O Código de Suministro será un String formado do seguinte xeito: *AnoMesDiaAltaSuministroDNIClienteTipoTarifaPotencia*. Potencia será A,B,C ou D, segundo a táboa indicada máis abaixo.

- No 'Acceso ós Datos de Consumo', unha vez facilitada a password correspondente, se visualizarán os datos persoais do cliente, os datos de suministro (código de suministro, potencia contratada e tipo de tarifa) o consumo en Watios do último mes e o importe a pagar. As tarifas aplicables serán:

Término fixo:

Hasta 2,3Kw (A):	3,17 €/mes
Hasta 3,5Kw (B):	4,23 €/mes
Hasta 4,6Kw (C):	5,12 €/mes
A partir de 4,6Kw (D):	7,06 €/mes

Tarifa:

TUR:	0,007343 €/Kw
2TR:	0,008123 €/Kw
LIB:	0,011210 €/Kw

Se pide implementar esta aplicación desenvolvendo a clase *Suministro*:

Atributos:

cliente - Obxecto Cliente únicamente accesible polo propio obxecto, contén o Cliente titular do suministro
codigoSuministro - De tipo String e accesible únicamente polo propio obxecto.
tarifa - De tipo String e accesible polo propio obxecto ou obxectos herdados.
potencia - De tipo float e accesible polo propio obxecto ou obxectos herdados.
ultimaLectura - De tipo float e accesible polos obxectos do mesmo paquete. Contén o consumo total en Kw no momento da última facturación
lecturaActual - De tipo float e accesible por todo o mundo, contén o consumo actual total en Kw.
password - De tipo String e de acceso únicamente dende o propio obxecto. Contén a password de acceso aos datos do subministro.

Métodos:

Suministro(Cliente cl,String tarifa,float potencia,String passwd) throws SuministroException; Constructor que pode utilizarse dende calqueira sitio, crea un novo obxecto Suministro a partir dun Cliente inicializando ao valor correcto o valor de *ultimalectura* e almacenándoo na BBDD coa password indicada.

static Suministro getInstance(String codSuministro,String passwd) throws SuministroException; Se pode chamar sen instanciar obxectos Suministro e devolve o obxecto Suministro identificado por codSuministro da base de datos si se suministra a password correcta.

void procesaFactura() throws SuministroException; Se pode chamar únicamente dende obxectos do mesmo paquete e actualiza a información do Suministro na BBDD, actualizando o valor de *ultimalectura* co valor de *lecturaActual*

float getImporteFactura(); Se pode chamar dende calqueira sitio. Devolve o importe da factura á data actual.

float getUltimoConsumo(); Se pode chamar dende calqueira sitio e devolve o consumo total na última facturación en Kw.

float getConsumo(); Se pode chamar dende calqueira sitio. Devolve o consumo dende a última factura á actualidade en Kw.

Criterios de Avaliación:

- | | |
|---|---------|
| • Uso correcto dos modificadores de acceso - | 0.5 pto |
| • Uso correcto do control de erros mediante excepcións - | 1 pto |
| • A aplicación está correctamente implementada - | 1.5 pto |
| • O constructor de Suministro está correctamente implementado - | 2 pto |
| • O método getInstance está correctamente implementado - | 1 pto |
| • O método procesaFactura está correctamente implementado - | 1 pto |
| • O método getImporteFactura está correctamente implementado - | 2 pto |
| • O método getUltimoConsumo está correctamente implementado - | 0.5 pto |
| • O método getConsumo está correctamente implementado - | 0.5 pto |



Supoñemos que dispoñemos das seguintes clases ***xa programadas***:

```
class CCCException extends Exception {}  
class DNIException extends Exception {}  
class DBException extends Exception {}  
class SuministroException extends Exception {}
```

```
class Cliente {  
    private String dni;  
    private String nome;  
    private String direccion;  
    private String telefono;  
    private String email;  
    private String ccc;  
  
    public String getDNI();  
    public String getNome();  
    public String getDireccion();  
    public String getTelefono();  
    public String getEmail();  
    public String getCCC();  
    public String toString();  
  
    /**  
     * Solicita os datos do novo cliente por consola e devolve o obxecto Cliente correspondente.  
     */  
    public static Cliente newClienteConsole() throws CCCException, DNIException;  
}
```

```
class SuministroDB {  
    /**  
     * Carga o suministro identificado por codSuministro da BBDD, devolvendo o obxecto Suministro  
     * correspondente, co atributo lecturaActual a 0, e co atributo ultimaLectura co seu valor correcto.  
     */  
    Suministro loadSuministro(String codSuministro, String password) throws DBException;  
  
    /**  
     * Garda na BBDD o obxecto suministro protexido pola password password. Si o suministro xa  
     * existe e a password é a correcta, o actualiza.  
     */  
    void saveSuministro(Suministro suministro, String password) throws DBException;  
}
```

```
class Contador {  
    /**  
     * Crea o obxecto Contador asociado con codSuministro  
     */  
    public Contador(String codSuministro);  
  
    /**  
     * Devolve o consumo total en Kw do contador  
     */  
    public float getLectura();  
}
```

NOTAS:

- **Se poden agregar os métodos que se consideren oportunos á clase Suministro.**
- **As clases pertencen ao package *com.iesrodeira.consumo*, e a clase da aplicación ao package por defecto.**
- Para obter o ano, mes e día actual en Java pode facerse mediante a clase Calendar do paquete java.util:
Calendar cal=Calendar.getInstance();
int mes=cal.get(Calendar.MONTH);
int ano=cal.get(Calendar.YEAR);
int dia=cal.get(Calendar.DAY_OF_MONTH);