


 IES Polígono Sur c/ Esclava del Señor,2 41013. Sevilla	PROGRAMACIÓN			
FP Parcial Presencial	Pág. 1 de 2	UNIDAD DIDÁCTICA 4 – R1 Clases Java	PRO	Curso 2011/12

DESARROLLO DE CLASES U UTILIZACIÓN DE OBJETOS

© Clase Bombilla

1. Crea la clase Bombilla de las transparencias, ahora retoca la clase Bombilla para que cumpla con los siguientes requisitos:
 - Si la bombilla estaba encendida y se vuelve a encender se debe mostrar el mensaje “La bombilla ya estaba encendida” y no aumentar el contador de número encendidos.
 - Si la bombilla estaba apagada y se vuelve a apagar se debe mostrar el mensaje “La bombilla ya estaba apagada”.
 - Cuando se produzca el encendido 1000 la bombilla debe fundirse y escribir el mensaje “La bombilla se ha fundido”. En este estado, si se intenta encender o apagar se debe mostrar el mensaje “La bombilla está fundida”.

Para probar la clase Bombilla crearemos otra clase *PruebaBombilla* y escribimos el siguiente código:


```
public class PruebaBombilla {

    public static void main(String[] args) {
        int i;
        Bombilla b = new Bombilla();
        // Creamos un objeto bombilla

        // La encendemos y apagamos 1000 veces
        for(i=1; i<=1000;i++)
        {
            b.encender();
            b.apagar();
        }
        // Y otra vez más, aunque ya debe estar fundida
        b.encender();
        b.apagar();
    }
}
```

Clase Fecha

2. Crear una clase fecha con los siguientes atributos:
 - a) int dia,mes,año: Todos ellos privados.
Los métodos públicos:
 - b) ModificarFecha
 - c) ModificarDia, ModificarMes, ModificarAño
 - d) DevoverDia,DevolverMes,DevolverAño

 IES Polígono Sur c/ Esclava del Señor,2 41013. Sevilla	PROGRAMACIÓN			 AENOR R Empresa Registrada ER-0934 / 2008 IONET CERTIFIED QUALITY SYSTEM
FP Parcial Presencial	Pág. 2 de 2	UNIDAD DIDÁCTICA 4 – R1 Clases Java	PRO	Curso 2011/12

e) toString

Un método privado, que se llama al modificar:

f) ComprobarFecha: Que hace una comprobación simple:

1<dia<31 ; 1<mes<12: Si la fecha no es correcta se da un error y no se cambia la fecha

Clase Temperatura

3. Crea una clase Temperatura, con los siguiente atributos:

- a) Una fecha (Composición)
- b) Una temperatura máxima y una temperatura mínima.

Los métodos:

- c) Los constructores
- d) Modificar temperaturas
- e) Devolver temp. max, temp. min. y temp_media
- f) toString: Que debe mostrar en pantalla la fecha, la temp. max., la temp. min y la temp. media.

Clase Factura

4. Diseñar una **clase Factura** que represente la venta de un producto en una tienda. La clase debe incluir información relativa al producto vendido (código, descripción y precio), datos acerca del cliente que compra el producto (nombre, apellidos, dirección, DNI) y el número de unidades compradas. Los métodos de la clase han de permitir obtener el importe total de la compra (suponiendo un porcentaje de IVA constante) y generar un informe con los datos de la factura (el “ticket” correspondiente a la venta), además de poder acceder y modificar los distintos datos recogidos en la factura.

- a) Representar gráficamente en UML la clase resultante.
- b) Implementar en Java la clase tal como esté representada en el diagrama.
- c) Crear un programa (**FacturaTest**) que compruebe el correcto funcionamiento de la implementación realizada.
- d) (Diagrama UML) Idear la forma de descomponer la clase Factura en varias clases de forma que la implementación resultante sea más cohesiva y las clases estén débilmente acopladas. Represente su diseño en UML e impleméntelo en Java teniendo en cuenta las relaciones existentes entre las distintas clases.

PISTA: La factura mezcla varios datos de productos con datos relativos a clientes