

## Ejercicios Tema 4 para resolver.

Desarrollar un programa que haga un carga de clientes a partir de un texto. Un ejemplo de funcionamiento sería el siguiente:

Clientes

-----

Cliente 1

Nombre: Pepe

Apellidos: Lopez Perez

NIF: 12345678a (Formato VALIDO)

Direccion: C/Luna, 27, 30132, Murcia

Email: pepelopez@gmail.com

Fecha de alta: 12/11/2019 (Formato VALIDO)

Tipo cliente: normal

Descuento: 0

Cliente 2

Nombre: Maria

Apellidos: Alvarez Martinez

NIF: 87654321b (Formato VALIDO)

Direccion: C/ Falsa, 123, 30009, Murcia

Email: mariaalvarez@gmail.com

Fecha de alta: 8/10/2020 (Formato VALIDO)

Tipo cliente: premium

Descuento: 25

Cliente 3

Nombre: Alberto

Apellidos: Espinosa Carillo

NIF: X2654874e (Formato VALIDO)

Direccion: C/ Arce, 10, Murcia, 30005

Email: aespinosa@gmail.com

Fecha de alta: 02/09/2019 (Formato VALIDO)

Tipo cliente: normal

Descuento: 0

Cliente 4

Nombre: Miguel

Apellidos: Caballero Pinto

NIF: 1122344C (Formato NO VALIDO)

Direccion: C/ Olmo, 13, Murcia, 30015

Email: michaelknight@gmail.com

Fecha de alta: 45/03/2016 (Formato NO VALIDO)

Tipo cliente: normal

Descuento: 5

Cliente 5  
null

Cliente 6  
null

Cliente 7  
null

Cliente 8  
null

Cliente 9  
null

Cliente 10  
null

Cliente 11  
null

Cliente 12  
null

Consiste en la implementación de métodos, dado el código fuente siguiente:

```
public class Principal {  
    public static void main(String[] args) {  
  
        //Prepara texto para la prueba con una cantidad arbitraria de objetos Cliente  
        //Cada objeto Cliente va separado por: ;  
        //Cada atributo -campo de Cliente- va separado por: ,,  
  
        //...  
  
        //Llamada al método de importación de clientes y recoge el resultado  
        //en un array de objetos de tipo Cliente  
  
        //...  
  
    }  
  
    // Otros métodos...  
  
} //class
```

```

class Cliente {
    public String nombre;
    public String apellidos;
    public String nif;
    public String direccion;
    public String email;
    public String fechaAlta;
    public String tipo;
    public int descuento;
} //class

```

Se pide:

- En la misma clase `Principal`, crear un método llamado `importarClientes()` que importa nuevos clientes en un array de objetos de tipo `Cliente`. El método recibe como parámetro de entrada, un texto que contiene los valores para los atributos de objetos de la clase `Cliente` separados por punto y coma ( ; ). Los campos de cada cliente van separados por dos comas ( , , ).
- Se debe crear un método llamado `mostrarClientes()`, que recibe un array con los clientes que se han cargado y los muestra por pantalla.
- Se debe crear un método llamado `validarFormatoNIF()` que solo valide el formato de caracteres de un NIF o NIE.
- Se debe crear un método llamado `validarFormatoFecha()` que solo valide el formato de una fecha.
- Entregar el código fuente completo de la clase `Principal` con los métodos pedidos.

La clase debe cumplir los detalles y criterios que se especifican a continuación:

Métodos:

`importarClientes()`

Recibe el texto con los datos a importar que pueden corresponder a cualquier cantidad arbitraria de clientes separados por punto y coma. El valor de cada atributo de un mismo objeto va separado por doble coma.

- No se conoce a priori el tamaño del array.
- El tamaño inicial del array debe establecerse al número de clientes a importar más una cantidad extra de 8.
- Carga objetos de la clase `Cliente` en un array de objetos de la clase `Cliente`.
- El método devuelve el array de objetos `Cliente`.

- Ejemplo de texto que recibe desde el `main()`:

```
//Hay cuatro clientes en el texto de ejemplo

String texto = "Pepe,,Lopez Perez,,12345678a,,C/Luna, 27, 30132,
Murcia,,pepelopez@gmail.com,,12/11/2019,,normal,,0;Maria,,Alvarez
Martinez,,87654321b,,C/ Falsa, 123, 30009,
Murcia,,mariaalvarez@gmail.com,,12/10/2020,,premium,,25;Alberto,,Espinosa
Carrillo,,X2654874E,,C/ Arce, 10, Murcia,
30005,,aespinoza@gmail.com,,02/9/2019,,normal,,0;Miguel,,Caballero
Pinto,,1122344C,,C/ Olmo, 13, Murcia,
30015,,michaelknight@gmail.com,,45/03/2016,,normal,,5";
```

`mostrarClientes()`

- Recibe un array de objetos de tipo `Cliente` y muestra los datos de cada cliente según el ejemplo que aparece más arriba, incluidas las validaciones del formato de algunos campos.
- Algunos elementos del array estarán vacíos, por lo que hay que comprobarlo antes de mostrar los datos de cada elemento del array. **IMPORTANTE:** No se puede utilizar "try/catch" en ninguna parte del examen

`validarFormatoNIF()`

- Recibe un texto correspondiente a un NIF o un NIE y devuelve el valor `true` si corresponde con cualquiera de los dos tipos de documento.
- Formato de un NIF: Esta formato por 8 caracteres numéricos, seguido de un único carácter alfabético. Debe admitir mayúsculas y minúsculas.
- Formato de un NIE: Esta formato por un primer carácter que puede ser "X", "Y" o "Z", seguido de 7 caracteres numéricos, seguido de un único carácter alfabético. Debe admitir mayúsculas y minúsculas

`validarFormatoFecha()`

- Recibe un texto correspondiente a una fecha en formato `dd/mm/aaaa` y devuelve el valor `true` si tiene un formato de fecha válida. **NO ES NECESARIO VALIDAR LA FECHA EN SÍ MISMA**, solo el formato.
- La fecha debe separar cada una de sus partes por una barra ( / ).
- La parte de día puede tener 1 o 2 caracteres numéricos, siendo el primero entre 0 y 3, y el segundo entre 0 y 9.
- La parte de mes puede tener 1 o 2 caracteres numéricos, siendo el primero entre 0 y 1, y el segundo entre 0 y 9.
- La parte del año tiene cuatro caracteres numéricos.

Hay que tener en cuenta que:

- Se debe probar el método pedido desde `main()`.
- El método pedido debe tener una cabecera de comentarios con una descripción del método y el autor.
- En el código fuente se deben incluir comentarios aclaratorios en las partes relevantes de su imº

En la escritura del código fuente se deben seguir las recomendaciones y estilo estándar del lenguaje

## Ejercicio Resuelto 2. Gestión de Libros

Se pide crear un programa que pida secuencialmente los datos de una cantidad determinada de libros. Dicha cantidad será una propiedad constante y con valor positivo de la clase principal.

Una vez se hayan recogido los datos de todos los libros, deberá mostrar un listado ordenado alfabéticamente por título para seleccionar un libro y mostrar sus datos.

Crear una clase llamada Libro con atributos privados: título, autor y isbn. Proporcionar métodos públicos (getters y setters) para acceder a ellos.

En la clase principal:

1. Definir una constante `CANTIDAD_LIBROS` con un valor positivo.
2. Método `pedirLibros()`:
  - Pida los datos (título, autor, ISBN) para cada uno de los libros, a partir de la constante `CANTIDAD_LIBROS`.
  - Guarde cada libro en un array de tipo Libro.
  - Devuelva el array.
3. Implementar un método `ordenarLibros()` que reciba el array de libros y lo ordene alfabéticamente por el título. Debe devolver el array ya ordenado.
4. Después de obtener el array de libros mediante `pedirLibros()` y ordenarlo con `ordenarLibros()`, mostrar un menú con todos los títulos para que el usuario seleccione uno por su número.
5. Implementar un método `mostrarLibro()` que reciba el array y el número seleccionado, y muestre todos los datos del libro correspondiente. Si el número no se corresponde con un elemento existente, indicarlo adecuadamente.

Ejemplo: Supongamos que `CANTIDAD_LIBROS = 3`.

Se piden datos para 3 libros, como por ejemplo:

1. Título: "La sombra del viento" Autor: "Carlos Ruiz Zafón" ISBN: "978-84-08-05679-1"
2. Título: "1984" Autor: "George Orwell" ISBN: "978-0-452-28423-4"
3. Título: "El nombre de la rosa" Autor: "Umberto Eco" ISBN: "978-84-339-7135-6"

Tras ordenarlos por título, el menú podría mostrar:

4. 1984
5. El nombre de la rosa
6. La sombra del viento

Si el usuario elige la opción 2, se mostrarán todos los datos de "El nombre de la rosa".