

PROGRAMACIÓN JAVA - TEMA 6 POO. Clases compuestas.

Una clase compuesta es una clase que tiene entre atributos una referencia a otra clase.

Por ejemplo, tenemos una clase Fecha con tres atributos de tipo int: día, mes y año.

```
public class Fecha {  
    private int dia;  
    private int mes;  
    private int año;  
  
    //resto de código de la clase Fecha  
}
```

Vamos a crear ahora una clase Persona con los siguientes atributos: id de tipo int, nombre de tipo String fechaNacimiento de tipo Fecha y contadorId que es un atributo static de tipo int.

```
public class Persona {  
    private int id;  
    private String nombre;  
    private Fecha fechaNacimiento;  
    private static int contadorId = 1000;  
  
    //resto de código de la clase Persona  
}
```

La clase Persona es una clase compuesta porque entre sus atributos hay uno, fechaNacimiento, que es una referencia a un objeto de otra clase, en concreto, es un atributo de tipo Fecha.

En UML la composición de clases se representa mediante una línea con un rombo junto a la clase compuesta (la clase que contiene el atributo del tipo de la otra clase):



Por norma general, los objetos que contiene una clase compuesta se crean e inicializan en el constructor.

Como podemos ver en la clase Persona no se ha creado el objeto Fecha, solo se ha declarado un atributo de ese tipo pero no se crea el objeto (no se hace new Fecha). El objeto se creará en el constructor de la clase Persona.

```
//Constructor por defecto de la clase persona  
public Persona() {  
    //En el constructor se deben crear los objetos que  
    //forman parte de los atributos de la clase.  
    this.fechaNacimiento = new Fecha();  
  
    //resto de código del constructor  
}
```

Para ver como se trabaja con las clases compuestas resolveremos el siguiente ejercicio. Utilizaremos las clases Persona y Fecha anteriores. La solución la tenéis en el archivo EjemploComposicionConArrays.rar

Ejercicio

Programa que pida por teclado los datos de una serie de personas y los guarde en un array. El array deberá ser de tipo Persona. El número total de personas a leer se pide por teclado.

A continuación mostrará lo siguiente:

- Todas las personas en orden alfabético
- Todas las personas ordenadas por edad de menor a mayor. Si varias personas tienen la misma edad se mostrarán las de la misma edad en orden alfabético.

La clase principal del proyecto deberá ser similar a esta:

```
public class EjemploComposicionConArrays {
    static Persona[] personas; //array donde se almacenan los objetos Persona.
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args) {
        crearArrayPersonas();
        introducirPersonas();
        listadoAlfabeticoPersonas();
        listadoPersonasPorEdad();
    } //fin del método main
} //Resto de métodos de la clase principal
```

La clase principal, además del método *main*, contendrá los siguientes métodos:

- **crearArrayPersonas()**: pide por teclado el número de personas a leer y crea el array *personas*.
- **introducirPersonas()**: pide por teclado los datos de todas las personas y las añade al array. A cada persona creada se le asigna un código o identificador numérico consecutivo (atributo *id*) de como mínimo 4 cifras. El código a asignar a la primera persona será el 1000, el 1001 para la segunda, etc. Este *id* no se introduce por teclado. Para obtener el correspondiente a cada persona se utiliza el atributo static *contadorId*.
- **listadoAlfabeticoPersonas()**: ordena el array *personas* alfabéticamente por el nombre de la persona y muestra el resultado.
- **listadoPersonasPorEdad()**: ordena el array *personas* por edad de menor a mayor y muestra el resultado. Si varias personas tienen la misma edad éstas se mostrarán por orden alfabético.