



El departamento de Informática de un hospital japonés llamado Aki Tematon está realizando un nuevo registro de datos de todas las personas que tienen relación con esa institución, que son:

- los empleados
- los médicos
- los pacientes

Los empleados son categorizados en función de si son contratados por plantilla o de forma eventual y son los encargados de los procesos administrativos.

Los médicos (que podrían considerarse un tipo especial de empleados contratados por plantilla) se encargan de las atenciones de las consultas médicas.

Los atributos y métodos de cada una de las clases a implementar se indican a continuación:

- **clase Persona (abstracta):**
 - **Atributos:** número de *DNI*, *nombre*, *apellido*
 - **Métodos:** *Constructores* (vacío y con atributos), *mostrar()* (método abstracto) que no recibe parámetros y no devuelve nada.
- **clase Paciente:**
 - **Atributos:** además de todos los atributos de la clase Persona, número de *historia* clínica, *alérgenos*: lista de medicamentos o alimentos a los que es alérgico (se almacenaran en un array y si el paciente no es alérgico a nada contendrá un único elemento que indique “Sin alergias”).
 - **Métodos:** *Constructores* (vacío y con atributos), *mostrar()* (mostrará todos sus atributos), *esAlergico(String)* recibe como parámetro el nombre de un alérgico y devuelve un valor booleano indicando si es alérgico o no (tener en cuenta que un alérgico puede estar escrito con mayúsculas o minúsculas y sigue siendo el mismo alérgico).
- **clase Empleado (abstracta):**
 - **Atributos:** además de todos los atributos de la clase Persona, *código* de Empleado, número de *horas extras*.
 - **Métodos:** *Constructores* (vacío y con atributos), *mostrar()* (método abstracto), *salario()* (método abstracto) que devuelve un valor de tipo double.
- **clase EmpleadoPlantilla:**
 - **Atributos:** además de todos los atributos de la clase Empleado, *salario base* mensual, *porcentaje* adicional por hora extra.
 - **Métodos:** *Constructores* (vacío y con atributos), *mostrar()*, *salario()* (será el resultado de sumar al salario base un porcentaje del mismo que se calculará multiplicando el “porcentaje adicional por hora extra” por el número de horas extras realizadas) que devuelve un valor de tipo double.
- **clase EmpleadoEventual:**

- **Atributos:** además de todos los atributos de la clase Empleado, *honorarios* por hora, número de *horas* (si tener en cuenta las horas extras) trabajadas.
- **Métodos:** *Constructores* (vacío y con atributos), *mostrar()*, *salario()* (será el resultado de multiplicar el número de horas trabajadas por los honorarios por hora, teniendo en cuenta que las horas extras se pagan al doble que las normales) que devuelve un valor de tipo *double*.

clase Médico:

- **Atributos:** además de todos los atributos de la clase EmpleadoPlantilla, *especialidad* (cirujano, oftalmólogo, etc).
- **Métodos:** Constructores (vacío y con atributos), *mostrar()*, *salario()* (heredado de la clase EmpleadoPlantilla)

SE PIDE:

1. Diseñar e implementar la jerarquía de clases descrita anteriormente.
2. Implementar un programa con una clase main que cree un array de elementos Persona al que llamaremos **listaPersonas** y al que se le añadirán los siguientes objetos:

```
Paciente p1 = new Paciente("12345", "Emilio", "Sánchez", 11111, 'H',  
                           "A+", new String[] {"Penicilina", "Huevo"});  
Paciente p2 = new Paciente("13335", "Pablo", "Corcho", 22222, 'H',  
                           "A-", new String[] {"Leche", "Huevo", "Nueces"});  
Paciente p3 = new Paciente("44445", "Daniel", "Esteban", 33333, 'H',  
                           "AB", new String[] {"Sin alergias"});  
EmpleadoEventual ee = new EmpleadoEventual("54321", "Pepe", "González",  
                                             2222, 10, 15.0, 150);  
EmpleadoPlantilla ep = new EmpleadoPlantilla("87545", "Laura", "Pérez",  
                                              65450, 15, 1500.0, 5);  
Medico m1 = new Medico("66665", "Ana", "Montero", 99950, 15, 2500.0, 10, "Cirujía");  
Medico m2 = new Medico("64445", "Luis", "Gómez", 88950, 8, 2000.0, 12, "Dermatología");
```

- En el programa principal, tras añadir los objetos, se recorrerá el array *listaPersonas* para mostrar todas las personas.
- Por último mostraremos sólo aquellos elementos de *listaPersonas* que correspondan a pacientes alérgicos a un medicamento o alimento introducido por el usuario. (Se le preguntará al usuario por teclado el alérgeno a buscar):

NOTA: Los datos de pacientes y empleados expuestos en el enunciado son un ejemplo de funcionamiento y serán los que utilice el alumno, pero deberá funcionar de la misma manera para cualquier conjunto de empleados y pacientes que introduzcamos en el array.

El resultado de mostrar todas las personas del array para el ejemplo anterior será tal y como se muestra a continuación. Tomar este listado como ejemplo para implementar los métodos ***mostrar()*** :

```
DATOS DEL PACIENTE
NOMBRE: Emilio
APELLIDO: Sánchez
HISTORIA CLÍNICA: 11111
ALÉRGICO A: Penicilina, Huevo

DATOS DEL PACIENTE
NOMBRE: Pablo
APELLIDO: Corcho
HISTORIA CLÍNICA: 22222
ALÉRGICO A: Leche, Huevo, Nueces

DATOS DEL PACIENTE
NOMBRE: Daniel
APELLIDO: Esteban
HISTORIA CLÍNICA: 33333
ALÉRGICO A: Sin alergias

DATOS DEL EMPLEADO
NOMBRE: Pepe
APELLIDO: González
CÓDIGO: 2222
SALARIO TOTAL: 2550.0

DATOS DEL EMPLEADO
NOMBRE: Laura
APELLIDO: Pérez
CÓDIGO: 65450
SALARIO TOTAL: 2625.0

DATOS DEL EMPLEADO
NOMBRE: Ana
APELLIDO: Montero
CÓDIGO: 99950
ESPECIALIDAD: Cirujía
SALARIO TOTAL: 6250.0

DATOS DEL EMPLEADO
NOMBRE: Luis
APELLIDO: Gómez
CÓDIGO: 88950
ESPECIALIDAD: Dermatología
SALARIO TOTAL: 3920.0
```