

PROGRAMACIÓN II - EXAMEN DE EVALUACIÓN CONTINUA

1.1 DATOS DEL PROGRAMA

El departamento de informática de Cruces está realizando un nuevo registro de datos de todas las personas que tienen relación con esa institución, que son: los empleados, los médicos y los pacientes.

Los empleados son categorizados en función de si son contratados en plantilla o de forma eventual. En este último caso, se calcula su salario en función de las horas que ha trabajado y un coste hora.

CLASE PERSONA

Representa cada persona relacionada con el hospital.

- Paquete: examen.backend.datos
- Atributos:
 - (String) DNI;
 - (String) nombre;
 - (String) apellido;
 - (String) direccion;
 - (String) procedencia;

CLASE PACIENTE

Describe cada paciente del hospital, es subclase de la clase Persona.

- Paquete: examen.backend.datos
- Atributos:
 - (Int) historiaClinica;
 - (String) medicamentos[]: Medicamentos a los que es alérgico.

CLASE EMPLEADO

Representa a un empleado eventual.

- Paquete: examen.backend.datos
- Atributos:
 - (Float) salarioMensual;

CLASE EMPLEADOPLANTILLA

Representa a un empleado eventual.

- Paquete: examen.backend.datos
- Atributos:
 - (Float) salarioMensual;

CLASE EMPLEADOEVENTUAL

Representa a empleados eventuales.

- Paquete: examen.backend.datos
- Atributos:
 - (float) pagoHora;

- (Int) horasTotales: Es calculado.
- (Int) horasNormales;
- (Float) salarioTotal: Es calculado.

INTERFAZ ISALARIO

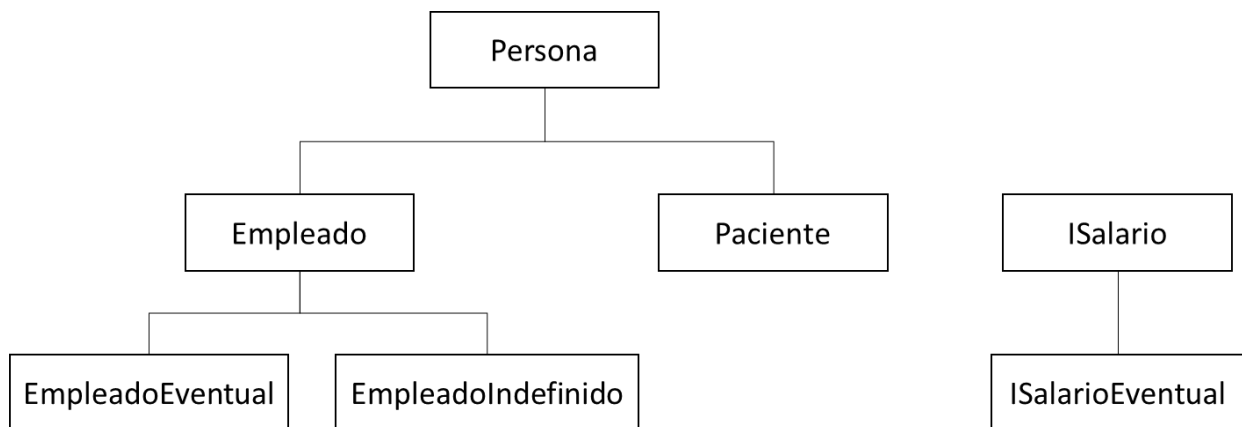
Interfaz para todos aquellos empleados que cobran un salario.

- Paquete: examen.interfaces
- Método:
 - Método que calcula el salario de un empleado
 - Getter del salario de un empleado.

INTERFAZ ISALARIOEVENTUAL

Interfaz para todos aquellos empleados eventuales que cobran un salario.

- Paquete: examen.interfaces
- Método:
 - Método `calcularHorasTotales` que calcula el total de las horas que ha trabajado un empleado eventual.



IMPORTANTE: implementar solo los métodos necesarios para resolver el problema planteado.

Sobre estos datos, se pide implementar los métodos para llevar a cabo las siguientes tareas:

1.2 REGISTRARPERSONA

Este método debe pedir al usuario el tipo de empleado, leerá de teclado los datos correspondientes a la persona elegida y los almacenará en un array de 5 posiciones llamado `aPersonas`.

Importante: cada clase implicada debe tener su propio método `registrar`, y desde el método `registrarPersona` se debe realizar una única llamada genérica de tipo, por ejemplo:

```
obj.registrar();
//Siendo obj el objeto creado de acuerdo al tipo especificado por el usuario
```

1.3 COSTESALARIOS

Haciendo un uso correcto de las interfaces, se deben mostrar el coste total en cuanto a salarios del hospital.

1.4 LISTARMEDICAMENTOS

El método recorrerá todos aquellos pacientes que estén almacenados en `aPersonas` y mostrará los nombres de todos los medicamentos a los que son alérgicos los pacientes.

El programa principal deberá estar alojado en el paquete `examen.frontend`. Deberá realizar las siguientes acciones:

- Crear el array para almacenar la estructura arriba definida
- Añadir un empleado indefinido.
- Añadir un paciente
- Añadir un empleado temporal.
- Una llamada a cada uno de los métodos expuestos arriba.

1.5 ORGANIZACIÓN DE CLASES Y PAQUETES

Paquete `examen.backend.datos` contiene las clases:

- `Empleado.java`
- `EmpleadoEventual.java`
- `EmpleadoPlantilla.java`
- `Paciente.java`
- `Persona.java`

Paquete `examen.interfaces` contiene la interfaz:

- `ISalario.java`
- `ISalarioEventual.java`

Paquete `examen.frontend` contiene la clase:

- `ProgramaPrincipal.java` (esta clase contiene los métodos *registrarPersona*, *costeSalarios* y *listarMedicamentosr*, además del programa principal)

1.6 ESTILO DEL CÓDIGO Y COMENTARIOS

Es importante, y se valorará, un adecuado estilo del código, siguiendo las normas de estilo comentadas en clase y documentando el programa con Javadoc.