Se necesita un sistema de gestión de nóminas para una empresa.

Existen tres tipos de empleados: empleados fijos, empleados por horas y empleados por comisión que tienen características distintas entre sí en cuanto a los procedimientos del cálculo de sus nóminas

Independientemente del tipo de empleado, todos ellos tienen un DNI, Nombre, Apellidos. El DNI es el que se utiliza como identificador unívoco de un empleado para cualquier búsqueda o acción

El procedimiento para calcular el sueldo mensual de los empleados es distinto dependiendo de su tipo:

* Los empleados fijos, reciben un sueldo fijo, sin importar el número de horas trabajadas. Dicho salario es 1.300€.
* Los empleados por horas, reciben un sueldo por hora normal y otro distinto por hora extra. Se considera hora extra, toda hora que pase de las primeras 160 trabajadas. El precio de la hora normal es de 8.5€ y de la hora extra 11.5€.
* Los empleados por comisión, reciben un salario base más un porcentaje de sus ventas. El salario base es 500€ y el porcentaje de las ventas un 25%.

Realizar:

1-Una clase abstracta denominada **Trabajador** que implemente las características comunes a todos los trabajadores.

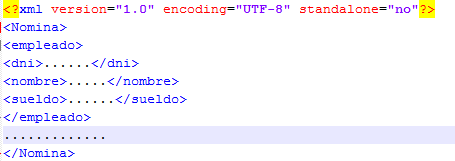
2-Las clases hijas de Trabajador que definirán las características distintas que tiene cada tipo de trabajador. Se denominaran: **Fijo, Comision, Horas**

Los atributos de las clases anteriores serán de tipo privado y los métodos que debe haber son los getter y setter (consulta y actualización de atributos) y el método **nomina** que calcula el sueldo de un tipo de empleado en particular.

3-Realizar una clase denominada **Empresa**. Esta clase tiene un atributo privado de tipo HashTable o HashMap (obligatorio) que almacena los distintos objeto Trabajador y cuya clave es el DNI del trabajador.

Esta clase tiene los siguientes métodos:

* El método **alta(....)** que permite guardar un trabajador en la estrutura anteriormente citada. Como hay diversos tipos de Trabajador, el mètodo deberá sobrecargarse para los diferentes tipos. Si ya existe en la estructura el trabajador, deberá lanzarse una excepción.
* El método **nomina(String dni)** que calcula la nómina de un trabajador determinado identificado por su dni. Si no encuentra el trabajador produce una excepción.
* El método **totalNomina()**, que devuelve el importe total del sueldo de toda la plantilla
* El método **crearXML()**, que crea un archivo denominado **“nomina.xml”.** Este archivo tendrá la siguiente estructura:



4-Modificar el programa Principal para probar las clases y métodos anteriorores:

NO ES NECESARIO NINGÚN ENTORNO GRÁFICO

NO SE PUEDE MODIFICAR EL CÓDIGO EXISTENTE.

import java.util.Scanner;

public class Principal {

public static void main(String[] args){

int opcion=0;

Scanner teclado=new Scanner(System.in);

Empresa miEmpresa=new Empresa();

Fijo t1=new Fijo("11111111A","Juan", "Nadie Nadie");

Comision t2=new Comision("22222222B", "Laura", "Alguien Alguien", 3970f);

Horas t3=new Horas("33333333C","Jaime","Cualquier Cualquier",150);

Horas t4=new Horas("44444444D","Nuria","G. M.", 161);

try{

miEmpresa.alta(t1);

miEmpresa.alta(t2);

miEmpresa.alta(t3);

miEmpresa.alta(t4);

}catch(Exception e){

System.out.println("Existe algún dni repetido");

}

do {

System.out.println("\n\n Mi Empresa");

System.out.println("Sueldo por empleado (dni)...........1");

System.out.println("Gasto total en sueldos..............2");

System.out.println("Generar archivo XML.................3");

System.out.println("Fin.................................0");

opcion=teclado.nextInt();

///////////////////////////////////////

//**INSERTA AQUÍ SI LO CREES CONVENIENTE EL CÓDIGO QUE NECESITES**

//////////////////////////////////////

}while(opcion!=0);

}

///////////////////////////////////////

//**INSERTA AQUÍ SI LO CREES CONVENIENTE EL CÓDIGO QUE NECESITES**

//////////////////////////////////////

}

La plantilla os la podéis descargar de: <http://www.v-espino.com:8383/epm2015>