

Revisión de herencia, clases abstractas y polimorfismo.

El director de una empresa de productos químicos desea gestionar el sueldo y las primas de sus empleados. Un empleado tiene un nombre, un apellido, una edad y una fecha de entrada en la empresa.

La clase empleado será una clase abstracta y tendrá :

- el método abstracto `calcularSueldo`, que será diferente en función del tipo de empleado
- un método `getNombreCompleto` que devolverá la concatenación de “El empleado :” y del nombre y del apellido
- un constructor

El método `calcularSueldo` depende del tipo de empleado:

- Los empleados que se dedican a la venta. Su sueldo mensual es el 20% de las ventas que han hecho en el mes más 400 euros
- Los representantes. Su sueldo mensual es el 20% de las ventas que han hecho en el mes más 800 euros.
- Los empleados de producción. Su sueldo mensual es el número de unidades producidas en el mes multiplicado por 5
- Los empleados de mantenimiento. Su sueldo mensual es el número de horas trabajadas en el mes multiplicado por 5

Desarrollar en las clases hijas el método `getNombreCompleto` cambiando la palabra “empleado” por lo que corresponda.

Cada clase hija tendrá un constructor.

Utilizar clases intermedias para evitar las redundancias de atributos y métodos.

Añadir una interfaz “Riesgo” que contenga la prima de riesgo que van a percibir los empleados de producción y los empleados de mantenimiento que manipulen productos de riesgo. Esta prima será de 1000€. No todos los empleados de producción y los de mantenimiento manipulan productos de riesgo.

Escribir un programa que gestione un array de tamaño el número de empleados de la plantilla, y que mediante un menú permita insertar empleados en el array, visualizar su sueldo y calcular la media de los sueldos por tipo de empleado.

Una de las opciones del array será su ordenación según diferentes criterios.

