

EJERCICIOS HERENCIA.

1º. Indicar la salida siguiente código

```
class TestMilpies{  
    public static void main(String[] args){  
        MilpiesEsquiador m = new MilpiesEsquiador();  
    }  
}  
class Milpies{  
    protected int numeroDePies;  
    public Milpies(){  
        numeroDePies = 1000;  
        escribePies();  
    }  
    public void escribePies(){  
        System.out.println("Un Milpiés o Cochinilla tiene"  
                             + numeroDePies+" pies");  
    }  
}  
class MilpiesEsquiador extends Milpies{  
    protected int PiesRotos;  
    public MilpiesEsquiador(){  
        PiesRotos = 100;  
    }  
    public void escribePies(){  
        System.out.println("A un Milpiés esquiador le  
                             quedan "+(numeroDePies-PiesRotos)+" pies");  
    }  
}
```

2º. Una empresa informática necesita llevar un registro de todos sus empleados que se encuentran en la oficina central, para eso quiere un programa que debe incluir lo siguiente:

a) Empleado.

Atributos:

- nombre: tipo cadena
- NIF: tipo cadena
- edad : entero
- casado: boolean
- salario: real

Métodos:

- Constructor con y sin parámetros de entrada
- Método que permita mostrar la clasificación según la edad de acuerdo al siguiente algoritmo:

Si edad es menor o igual a 21, Principiante

Si edad es ≥ 22 y ≤ 35 , Intermedio

Si edad es > 35 , Senior.

- Imprimir los datos del empleado por pantalla (utilizar salto de línea \n para separar los atributos).
- Un método que permita aumentar el salario en un porcentaje que será pasado como parámetro al método.

b) Programador (Especialización de Empleado). Clase que hereda de Empleado todos los atributos y métodos.

Atributos:

- lineasDeCodigoPorHora : tipo entero
- lenguajeDominante: tipo cadena

Metodos:

- Constructor con y sin parámetros de entrada.

3º. Crear un programa que gestione una compañía de alquiler de vehículos.

La empresa va a tener furgonetas, coches, y motos. Todos los vehículos tendrán matrícula, kilometraje acumulado durante un alquiler y un valor que indica su estado actual (alquilado o no). Además, cada uno se podrá alquilar, devolver consultar sus datos, y calcular el importe a pagar por el alquiler.

Las motos podrán ser alquiladas incluyendo un casco o sin incluirlo. Su precio será de 0.6€ por km. recorrido. Si lleva casco, el precio total se incrementa en 10€.

Las furgonetas tendrán una capacidad de carga determinada. Su precio será de 0.5€ por km. recorrido.

Los coches podrán tener extras. Su precio será de 0.4€ por km, recorrido, con una cantidad adicional de 20€ si se trata de un coche con extras.

El programa principal debe tener las siguientes opciones:

- Dar de alta cualquiera de los 3 vehículos
- Alquilar cualquiera de los 3 vehículos
- Devolver vehículo (indicando el importe a pagar por el kilometraje realizado y posibles gastos adicionales)
- Listar los vehículos dados de alta (indicando tipo, matrícula, kilometraje, extras/casco, estado libre/alquilado).

4º. Definir una interfaz ITFImpresion que consta de un solo método imprimir, diseñado para imprimir un texto.

Crear dos clases ITFPositivo e ITFNegativo que implementen la interfaz ITFImpresion. La primera clase imprimirá ¡Que buen tiempo hace! y la segunda clase imprimirá ¡Odio los lunes! Crear una clase ITFPrueba que:

- Declara una variable ITFPositivo de tipo ITFImpresion
- Declara una variable ITFNegativo de tipo ITFImpresion

Para probar el polimorfismo hacer lo siguiente:

- Crear un ArrayList parametrizado para ITFImpresion y añadir a la lista ambas instancias
- Recorrer el ArrayList llamando al método imprimir

5º. Se dispone de una orquesta que puede tener instrumentos de viento, cuerda o percusión. Diseñar las correspondientes clases sabiendo que todas tienen un método en común que es tocar, que visualizará el mensaje “Estoy soplando/rascando/golpeando el instrumento” según sea de viento, cuerda o percusión, seguido del nombre de instrumento.

Crear una clase Orquesta que contenga una lista (atributo de la clase de tipo ArrayList) con un instrumento de cada tipo. La lista rellena se le pasará al constructor de Orquesta como parámetro. Dicha aplicación dispondrá además, de un método llamado afinar (que deberá ser estático) que recibirá como parámetro el instrumento a afinar y que invocará al correspondiente método tocar del instrumento, de forma que aparecerá algo así como:

Afinando...Estoy soplando/rascando/golpeando el instrumento....

Como instrumentos será necesario desde el programa principal, crear una flauta, un violín y un bombo, necesarios para crear la lista y poder instanciar un objeto de la clase Orquesta.

Inicialmente se llamará al método estático afinar, pasándole cada uno de los instrumentos y posteriormente, se sacará un mensaje “EMPEZAMOS EL CONCIERTO” y a través del objeto orquesta, llamaremos a un método concierto, que haga que toquen todos los instrumentos.

6º. Se pretende realizar una aplicación para gestionar un instituto. La información sobre las personas vinculadas con el mismo, se pueden clasificar en tres tipos: estudiantes, profesores y personal de servicio.

A continuación, se detalla qué tipo de información debe gestionar esta aplicación:

- Por cada persona, se debe conocer, al menos, su nombre y apellidos, su nº de identificación y su estado civil.

- Con respecto a los empleados, sean del tipo que sean, hay que saber su año de ingreso en el centro y qué número de despacho tienen asignado.
- En cuanto a los estudiantes, se requiere almacenar el curso en el que están matriculados.
- Por lo que se refiere a los profesores, es necesario gestionar a qué departamento pertenecen (lenguajes, matemáticas, arquitectura, ...).
- Sobre el personal de servicio, hay que conocer a qué sección están asignados (biblioteca, decanato, secretaría, ...).

En primer lugar definir la jerarquía de clases de esta aplicación. Después, se deben programar las clases definidas en las que, además de los constructores, hay que desarrollar los métodos correspondientes a las siguientes acciones:

- Cambio del estado civil de una persona.
- Reasignación de despacho a un empleado.
- Matriculación de un estudiante en un nuevo curso.
- Cambio de departamento de un profesor.
- Traslado de sección de un empleado del personal de servicio.
- Imprimir toda la información de cada tipo de individuo.
- Calcular el salario de los profesores y del personal de servicio, teniendo en cuenta:
 - El personal de servicio cobra una cantidad fija más un 5% si están casados.
 - El personal docente gana un fijo más un 8% si su fecha de incorporación es anterior al 2000.

Añadir un método en la clase que representa el centro que imprima el salario de todos los empleados del centro (nombre y apellidos más el sueldo).
Incluir un programa de prueba que cree objetos de los distintos tipos y pruebe los métodos desarrollados.