

Hoja 12

1. Crea un proyecto con dos paquetes: paquete1 y paquete2. En el paquete1 guarda dos clases: Alumno y Profesor. En el paquete2 guarda la clase Dirección.

La estructura de dichas clases sería esta:

```
class Alumno {  
  
    private String nombreAlum;  
  
    protected int edadAlum;  
  
    public void visNombreAlumno(){  
  
        //dar contenido  
  
    }  
  
    void visEdadAlumno(){  
  
        //dar contenido  
  
    }  
  
}  
  
class Profesor {  
  
    private String nombreProfe;  
  
    protected int edadProfe;  
  
    public void visNombreProfe(){  
  
    }  
  
    void visEdadProfe(){  
  
        //dar contenido  
  
    }  
  
}  
  
public class Dirección {  
  
    private String CargoDirec;  
  
    protected int AntigDirec;
```

```

public void visCargoDirec(){

    //dar contenido

}

void visAntigDirec(){

    //dar contenido

}

}

```

Observa que se han puesto diferentes modificadores de visibilidad a cada uno de los miembros de las distintas clases.

Una vez hecho esto, lleva a cabo las siguientes tareas:

- a) Prueba a poner a la clase Alumno el modificador `protected` o `private` ¿tiene sentido? [No tiene sentido ya que crear una clase private te quita el acceso a dicha clase](#)
- b) Añade a la clase Profesor este atributo: Alumno hijo, de tipo Alumno (Ya que del profesor nos interesa, además, la información de uno de sus hijos (que sea alumno).

Veremos que nos deja usar la clase Alumno, ya que las dos clases están dentro del mismo paquete. [Por el modificador de acceso es default](#)

- c) Instancia el objeto hijo que acabas de añadir en la clase Profesor.
Luego pon: "hijo." y observa cuáles son los atributos y métodos que te deja usar de la clase Alumno. [Los metodos de la super clase Object y los metodos de la clase Alumno](#)
- d) Añade a la clase Dirección este dato: Alumno tutelado.
Ya que cada persona del equipo directivo tiene un alumno tutelado, ¿te deja usar la clase Alumno? ¿Qué habría que hacer? [No se puede porque la clase no tiene el modificador de acceso Public](#)
- e) Haz que la clase Dirección herede de la clase Profesor, ya que toda persona que ocupa un puesto directivo es profesor.
- f) Dentro de cualquier método de la clase Dirección prueba a poner "this." Y observa cuáles son los miembros de la clase Profesor que te deja usar.

Obs: (Cuando se usa la palabra this seguido de un punto estamos haciendo referencia al propio objeto que ha llamado a dicho método y el compilador te visualizará todos los atributos y métodos de esa clase que podrá usar, dependiendo de los indicadores de visibilidad). [Al metodo visNombreProfe\(\)](#)

2. Consumo Vehículo

Se trata de crear una pequeña aplicación que permita al usuario calcular los tiempos y consumos invertidos en un trayecto realizado.

Organiza las clases en diferentes paquetes. Proporciona a las clases, métodos y atributos, el acceso más restringido y adecuado posible.

Después de estudiar el esquema del programa que se pide, piensa:

¿Tiene sentido la herencia? Si es así ¿Dónde se aplicaría? ¿Qué más subclases se podrían crear?

PROYECTO TRAYECTO

Crea una clase abstracta Vehiculo

Atributos:

kms: Kilómetros recorridos en el trayecto

tiempo: tiempo empleado en el trayecto en horas

Métodos:

calcularVelocidad: devuelve la velocidad media empleada en el viaje en km/hora

(recuerda: $\text{espacio} = \text{veloc} \times \text{tiempo}$)

Implementa una clase Automovil, tiene las siguientes características:

Atributos:

- litros: Litros de combustible consumido en el trayecto
- precGas: Precio del combustible (en Euros/litro)

Métodos:

- consumoMedio. Consumo medio del vehículo (en litros cada 100 kilómetros). Los cálculos se realizarán teniendo en cuenta los Km recorridos y los litros consumidos
- consumoEuros. Consumo medio del vehículo (en euros cada 100 Kilómetros). Los cálculos se realizarán teniendo en cuenta los Km recorridos, los litros consumidos y el precio del combustible

Crea un constructor para la clase que establezca el valor de los atributos necesarios para un objeto Automovil. Elige el tipo de datos más apropiado para cada atributo. Pensar bien los atributos asociados al objeto.

En una clase Principal

- Pon a prueba los métodos creando un objeto con los datos pasados por teclado.

Se pueden diseñar para ello los métodos pedirDatos(), imprimirDatos, validar,...

- Crea un menú a disposición del usuario. Se aplicarán los métodos implementados para resolver la petición del usuario.

INFORMACIÓN DEL TRAYECTO

1-Consultar datos del trayecto del vehículo

2-Consultar la velocidad media

3-Consultar consumo medio, en litros a los 100Km

4-Consultar consumo medio, en Euros a los 100Km

5-Salir

- Controla las excepciones y las repeticiones de menú,...

Implementa una clase Bicicleta, a la que añades atributos específicos y métodos relacionados.

En otra clase Ejecutable pon a prueba la creación de Automoviles y Bicicletas.