Ejercicio Objetos-1

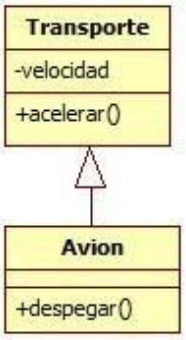
```
class Empleado{
    protected int dni;
    protected String nombre;
    protected int sueldobase;

    @Override
    public String toString() {
        return  dni.toString() + nombre + sueldobase.toString();
    }
}
class Ejecutivo extends Empleado{
    private Integer sueldoadicional;

    @Override
    public String toString() {
        return dni.toString() + nombre + sueldobase.toString() +  sueldoadicional.toString();
    }
}
```

a) Reescribir **Ejecutivo.toString()** para evitar repetir código.
b) Que propiedad de la herencia está utilizando? Explique.

Ejercicio Objetos-2



Transporte transporte= new Avion();

1) transporte.despegar();
2) transporte.acelerar();

¿Es posible la situación anterior? ¿Funcionaría, o hay que modificarla? ¿Cómo? Explicar para caso 1 y 2.

Soluciones

Ejercicio Objetos-1

a)

```
class Ejecutivo extends Empleado{
   private Integer sueldoadicional;

   @Override
   public String toString() {
      return super.toString() +  sueldoadicional.toString();
   }
}
```

b) Para referenciar a la clase de la cual heredamos, podemos hacerlo con la palabra reservada super, en este caso es necesario utilizarla para indicar que el método toString que queremos utilizar es el método de la clase padre o superclase, si no estaríamos generando un método recursivo. Si bien las clases derivadas heredan tanto atributos como métodos, es recomendable no acceder directamente a los atributos, para mantener el encapsulamiento, o sea que los atributos sean solo accedidos por la clase a la cual pertenecen. Para utilizarlos en las derivadas hacerlo a través de los métodos provistos por la superclase. De esta forma si se modificara algo en la superclase, no afectaría a las clases derivadas.

Ejercicio Objetos-2

1) transporte.despegar();

La clase transporte es la super clase solo conoce sus propios métodos. Si bien la instancia que se crea es de tipo Avion, cuando tratemos de usar el método despegar, al estar definido como transporte, no permite que se use un método que no le pertenece, no se podría resolver en tiempo de compilación, la instancia recién se crea en tiempo de ejecución.

Dado que despegar es un comportamiento propio de Avion, la mejor forma de resolverlo seria

Avion transporte = new Avion();

2) transporte.acelerar();

Esto funcionaria sin problemas el método esta en Transporte, y es heredado por Avion, por lo tanto lo puede resolver en tiempo de compilación, y en ejecución tampoco habría problema, cuando lo use una instancia de tipo Avion.