



PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

Preguntas de análisis

1. Si se desea modificar el mensaje del método pitar al crear un objeto moto sin alterar la clase Auto, se debe agregar al código de la clase Moto lo siguiente:

```
public void pitar(){  
    System.out.println("Las motos no pitan");  
}
```

De esa manera se sobrescribe el método pitar de la clase Auto y sin cambiarla se imprime un mensaje diferente al hacer llamado de dicho método.

2. Suponiendo que se agrega una nueva clase al código, class Motoneta, y esta hereda de la clase Moto, se puede evidenciar algún problema al querer acceder a los atributos ya que estos, al ser privados y la clase al no tener métodos get o set, la clase Motoneta no podrá interactuar con ellos de ninguna forma, esto podría ser una limitación en la funcionalidad de Motoneta.

3. Suponiendo que se definió el método:

```
public void arrancar() {  
    System.out.println("Arrancando");  
}
```

en la clase Moto, es posible sobrescribir el método, dado que el método está definido en la clase Auto que es de la que hereda todas las características la clase Moto, y el método arrancar es público por lo que no habría inconveniente al sobrescribir dicho método.

4. En la línea 13 de la clase Moto, se puede hacer uso del método pitar, ya que es un método heredado de la clase Auto.
5. Haciendo una pequeña modificación a la clase Auto y utilizando la variable num_autos, sin modificar su modificador de acceso, se puede obtener el número de autos creados desde la clase ObjTaller5H importando la clase Auto del paquete 1 de la siguiente manera:
package paquete2;
import paquete1.Auto;
6. En la línea 7 de la clase ObjTaller5H, no se puede utilizar el método adelantar, dado que el método no está definido como público en la clase Auto y de hecho no tiene definida una visibilidad, por lo que se establece visibilidad por default, por lo que solo se puede acceder desde el mismo paquete.
7. En la línea 8 es posible utilizar el método arrancar a pesar de que la clase Bus no lo haya definido porque es un método público que se hereda de la clase Auto y que puede ejecutar su problema.
8. En la línea 9 de la clase ObjTaller5H, el método pitar se puede ejecutar sin problemas al ser heredado de la clase Auto.
9. En la línea 10 de la clase ObjTaller5H, el método getVelocidad() imprime 10 porque a pesar de que se haga el llamado del método desde el objeto moto, al el método ejecutarse desde la clase Auto, el this hace referencia a sí mismo, en ese caso se refiere al atributo velocidad de la clase.
10. Si quisiera obtener el valor de la placa de las clases Moto y Bus, además de su modelo y capacidad respectivamente, lo que se debe agregar al código es un get para cada uno de dichos atributos, es decir, un getPlaca() para cada uno, un getModelo() para la moto y un

`getCapacidad()` en el caso del bus, eso es necesario porque al ser los atributos de tipo privados por el principio del encapsulamiento, no hay manera de acceder a ellos de una manera diferente.