

Taller de Java # 5

By: Andrés Felipe Muñoz Ortiz

Documento: 1011395924

Solución:

1. Habría que sobrescribir el método, escribiéndolo nuevamente dentro de la clase moto para evitar que se ejecute el heredado por Auto, se escribiría: `public void pitar(){ System.out.println("Las motos no pitan");}`
2. Se podría crear, pero no heredaría los atributos placa ni modelo ya que estos son privados de Moto, por lo que sí necesita usarlos tendría que crearlos en la propia clase, y no habría problema cuando los llama el método constructor moto() ya que los métodos constructores no se heredan
3. Sí es posible, pues este al ser un método de instancia y no de clase dentro de la clase madre, puede ser sobrescrito escribiendo un método con el mismo nombre dentro de la clase hija para que así se ejecute este nuevo método y no el heredado
4. Porque aunque la clase no lo haya declarado, este es heredado por la clase madre Auto quien si lo tiene y le hereda todos sus métodos y atributos públicos a sus subclases
5. Se puede obtener de forma común y corriente, pues num_autos es un atributo público de la clase Auto, y se podría imprimir desde ObjTaller5H así: `System.out.println(Auto.num_autos);` y en este ejercicio imprimiría 2 ya que se crearán dos subclases, una moto y un bus
6. Porque este tiene modificador de acceso package por defecto ya que no se le especifica uno, y como ambas clases están en paquetes diferentes, entonces no podría ser llamada por la clase
7. Porque este es heredado de la clase madre Auto a su subclase Bus, haciendo que esta pueda ejecutar el método
8. Sí se puede ejecutar ya que este método de Auto es público y es heredado a todas sus subclases, incluida bus, y ya que este no la sobrescribe la puede ejecutar normal
9. Retorna 10, ya que el método getVelocidad() pertenece a la clase Auto y accede al atributo velocidad de Auto (no al de Moto)
10. Para obtener los valores de placa, modelo y capacidad, se deben agregar **métodos getter** en las clases Moto y Bus puesto que estos atributos son privados, y ya solo quedaría escribir el `System.out.println` que ejecute el método get de la clase necesaria y con el objeto deseado como por ejemplo:

`System.out.println("Moto Placa: " + moto.getPlaca());` para imprimir la placa siendo este atributo privado de Moto