

Respuestas

1. Si deseo modificar el mensaje del método pitar al crear un objeto moto sin alterar la clase Auto, ¿qué debo agregarle al código? (Por ejemplo, al llamar el método pitar imprima: Las motos no pitan).

Para modificar el método pitar al crear un objeto moto se debe sobrescribir el método pitar en la clase moto, así al llamarlo toma el método más específico, para ello se agrega el siguiente código:

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

2. Suponga que se agrega una nueva clase al código, class Motoneta, y esta hereda de la clase Moto, ¿evidencia algún problema? ¿Por qué?

No evidencia ningún problema en la herencia, sin embargo esta recibe también los atributos y métodos de la clase auto, pues la clase moto hereda de este.

3. Suponga que se definió el método:

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

en la clase Moto, ¿es posible sobrescribir el método? ¿Por qué?

Si es posible, debido a que el método arrancar en la clase auto no está definido como método estático en la clase auto, por tanto se resuelve por ligadura dinámica, la cual permite la sobrescritura del método.

4. En la línea 13 de la clase moto, ¿Por qué puedo utilizar el método pitar?

Debido a que hereda el método pitar de la clase padre que es auto.

5. Haciendo una pequeña modificación a la clase Auto y utilizando la variable num_autos, sin modificar su modificador de acceso, ¿cómo puedo obtener el número de autos creados desde la clase ObjTaller5H?

Para acceder al total de autos creados desde la clase ObjTaller5H se llama directamente al atributo num_autos de la clase Auto, pues este es static, por lo tanto, se accede desde la misma clase usando **Auto.num_autos**.

6. En la línea 7 de la clase ObjTaller5H, ¿Por qué no puedo utilizar el método adelantar, si este fue heredado?

No se puede usar el método adelantar, por el nivel de privacidad que tiene, pues al no estar definido el nivel de privacidad java toma este como package, es decir que solo pueden usarlo las clases que estén en el mismo paquete que Auto.

7. En la línea 8, ¿por qué es posible utilizar el método arrancar si la clase Bus no lo define?

Porque es un método heredado directamente de la clase Auto, por tanto este lo puede usar

8. En la línea 9 de la clase ObjTaller5H, ¿por qué no puedo utilizar el método pitar, si este fue heredado?

El método pitar si se puede utilizar ya que este método lo heredo la clase bus de auto, y se ejecuta con normalidad.

9. En la línea 10 de la clase ObjTaller5H, ¿qué imprime el método getVelocidad()? ¿Por qué?

Al momento de ejecutar el código, el método getVelocidad no imprime nada por si solo, ya que solo retorna un valor, pero este no se esta imprimiendo, sin embargo en caso de imprimir un valor este seria 10, pues el método toma el valor del atributo de la clase padre que es Auto y no de las subclases.

10. Si quisiera obtener el valor de la placa de las clases Moto y Bus, además de su modelo y capacidad respectivamente, ¿Que debo agregar al código?

En cada clase se debe agregar el método get correspondiente a lo que se quiere obtener, en estos casos, en ambas clases se debe definir el método **getPlaca()**, para la clase moto también el método **getModelo()** y para la clase Bus el método **getCapacidad()**