

Taller 5 de java: Análisis.

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).

```
19  
20 @Override  
21 public void pitar(){  
22     System.out.println(x:"Las motos no pitan");  
23 }  
24
```

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 se evidencia ningún problema pues:

- Java lo permite: La clase motoneta puede heredar de la clase moto que a su vez hereda de la clase auto. A pesar de que java no tiene herencia múltiple, se puede heredar de una clase que hereda de otra a su vez.
3. Suponga que se definió el metodo

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

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

Si se puede sobrescribir pues:

- Tiene la visibilidad correcta para ser heredado en primer lugar
 - Es de instancia como el de la clase padre
 - Retornan lo mismo
 - Tienen la misma firma
4. En la línea 13 de la clase moto, ¿Por qué puedo utilizar el método pitar?

Porque el método es publico y heredado de la clase Auto, por ende hace parte de this.

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?

Podemos crear un método getter estático que nos retorne dicho valor.

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

Porque el método adelantar es de visibilidad por defecto, y ObjTaller5H está en otro paquete.

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

Se puede usar porque arrancar es heredado de Auto.

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

Si se puede usar. Su modificador de visibilidad lo permite.

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

Imprime 10, pues Moto no define getVelocidad, se usa el de Auto, y a Auto no le competen sus hijos. La velocidad impresa es la definida en Auto.

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?

Debemos añadir getters públicos de instancia para estos atributos. De esta manera:

```
18
19     public String getPlaca(){
20         return this.placa;
21     }
22
23     public int getCapacidad(){
24         return this.capacidad;
25     }
26
```

En Bus

```
19
20     public String getPlaca(){
21         return this.placa;
22     }
23
24     public String getModelo(){
25         return this.modelo;
26     }
27
```

En Moto