

## PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

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).

**Rta/=** Para modificar el mensaje del método pitar para las motos sin alterar la clase Auto, se puede sobrescribir el método pitar en la clase Moto para así cambiar el mensaje que imprime. En este caso se vería así:

```
@Override
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?

**Rta/=** Al agregar la clase Motoneta que hereda de la clase Moto, no hay ningún problema, ya que, en este caso, Java permite que una clase herede de otra sin restricciones.

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?

**Rta/=** Es posible sobrescribir el método arrancar en la clase Moto, porque el método existe en la clase Auto y es accesible desde la clase Moto al ser público.

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

**Rta/=** En la línea 13, se puede utilizar el método pitar porque es un método público y la clase Moto hereda este método 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, ¿cómo puedo obtener el número de autos creados desde la clase ObjTaller5H?

**Rta/=** Ya que num\_autos es un atributo de clase de Auto, se puede acceder directamente a él usando Auto.num\_autos en la clase ObjTaller5H.

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

**Rta/=** No se puede utilizar el método adelantar en la clase ObjTaller5H, debido a que este método tiene un modificador de acceso por defecto, es decir, de paquete, lo que significa que solo es accesible dentro del mismo paquete. Como ObjTaller5H pertenece a paquete2 y el método adelantar está definido en paquete1, no se puede usar desde ObjTaller5H.

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

**Rta/=** Es posible utilizar el método arrancar desde la clase Bus, porque la clase Bus hereda de la clase Auto, y en la clase Auto el método arrancar está definido y está público, lo que lo hace accesible para cualquier clase que herede de él.

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

**Rta/=** El método pitar si se puede utilizar en la línea 9 de la clase ObjTaller5H, ya que este es público y esta definido en la clase Auto.

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

**Rta/=** El método getVelocidad no imprime nada, porque no se utiliza el System.out.println sobre el valor que retorna.

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?

**Rta/=** Para obtener el valor de la placa, su modelo y capacidad, se deben agregar métodos getter en las clases Moto y Bus respectivamente:

```
public String getPlaca() {  
    return placa;  
}  
  
public String getModelo() {  
    return modelo;  
}  
  
public int getCapacidad() {  
    return capacidad;  
}
```