

**Taller 5 Java**

**Pablo Bernal García**

**1025888826**

**Grupo 2**

**Taller 5 Java – 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).

R//: Se puede sobrescribir el método pitar() en la clase Moto o modificar su constructor para cambiar el mensaje:

```
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?

R//: La herencia funcionaría correctamente, heredando todos los métodos y atributos de la clase Moto.

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?

R//: Sí es posible. Se necesita definir un método con la misma firma en la clase hija, como por ejemplo con el método `acelerar()` de las clases `Bus` y `Moto`

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

R//: Se puede usar el método `pitar()` ya que este fue heredado de la clase `Auto`; además, el método es público y se puede acceder por medio de las demás clases.

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`?

R//: Se podría hacer por medio de la siguiente línea:

```
System.out.println(Auto.num_autos);
```

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

R//: No se puede usar el método `adelantar()` porque el método en la clase `Auto` es de acceso por defecto, es decir, solo es accesible dentro del mismo paquete. Como `ObjTaller5H` está en un paquete diferente, no puede acceder a este método.

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

R//: Se puede utilizar gracias a que es un método público que es heredado por la clase `Auto`, lo que hace que las hijas lo puedan heredar sin problemas.

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

R//: No se puede usar el método `pitar()` ya que el método en la clase `Auto` es público, pero la instancia `bus` es de la clase `Bus`, y esta no sobrescribe el método `pitar()`.

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

R//: Imprime 30; debido a que dentro de la clase Moto se define una nueva variable velocidad que sobrescribe la de la clase padre, y el método getVelocidad() devuelve el valor de esta variable.

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?

R//: Se tendrían que agregar algunos getter:

Moto.java:

```
public String getPlaca() {  
    return this.placa;  
}  
  
public String getModelo() {  
    return this.modelo;  
}
```

Bus.java:

```
public String getPlaca() {  
    return this.placa;  
}  
  
public int getCapacidad() {  
    return this.capacidad;  
}
```