



Preguntas de análisis

Andrea Merino Mesa 1034991011

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 pitán).

Podríamos sobrescribir el método pitar() para la clase moto, ya sea reemplazando lo que hay dentro del print por un string o sobrescribiendo también el atributo pitar:

Sln1:

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

Sln2:

```
public String pitar = "Las motos no pitán";  
public void pitar() {  
    System.out.println(this.pitar);  
}
```

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

En términos generales se puede crear una clase Motoneta que hereda de la clase Moto, pero hay que tener en cuenta que el **constructor de Moto es parametrizado**, y como cuando **la clase padre no tiene constructor por defecto**, es obligatorio llamar explícitamente al constructor de la clase padre desde el constructor de la clase hija, debemos de **definirle un constructor a la clase Motoneta que coincida con los parámetros y se los pase al constructor de la clase Moto**.

```
public Motoneta(String placa,String modelo){  
    super(placa,modelo);  
}
```

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, porque la **clase Moto es una clase hija de Auto** y por tanto puede sobrescribir los métodos de instancia que vienen de su clase padre, cambiándolos para sí misma.



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

Porque es un método de instancia que hereda de la clase padre (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?

Como queremos obtener el número de autos creados desde ObjTaller5H pero solo podemos hacer una pequeña modificación en la clase auto y los objetos no heredan el constructor, **debemos crear un método estático que retorne ese valor y llamarlo desde ObjTaller5H cuando queramos obtener el número de autos creados.**

```
public static void getNum_Autos() {  
    System.out.println(num_autos);}
```

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

Porque este método **tiene visibilidad de paquete**, y la clase Auto está en un paquete diferente a las demás, por lo que no se puede acceder a el método, ni siquiera importando la clase.

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

Porque Bus **hereda ese método de la clase Auto**, lo que significa que, aunque no lo tenga definido en su propia clase, como si se define en su clase padre, lo puede ejecutar.

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

No hay ningún tipo de error, **el método se puede utilizar sin problemas** porque es público, y bus hereda dicho método de su padre (Auto).

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

El método **getVelocidad() imprime 10** ya que cuando llama al atributo this.velocidad, no entra a la clase moto porque no hay sobreescritura sino al de la clase padre (Auto). Una vez allí, al hacer el this, **él método ejecuta en su propio**



contexto (clase Auto) y se va al atributo de dicha clase.

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?

Podemos añadir para cada una de las clases un **toString()** que, cada que se imprima un objeto de uno de esos tipos muestre dichos valores respectivamente.

Para moto:

```
public String toString() {return "Placa: "+this.placa+" modelo:"+this.modelo;}
```

Para Bus:

```
public String toString() {  
return "Placa: " + this.placa + "capacidad:" +    this.capacidad;}
```

En ObjTaller5H:

```
System.out.println(moto);  
System.out.println(bus);
```