

Taller 5 Java Ejercicio 2

Emmanuel Valencia Lopera.

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).

Se debe agregar dentro de la clase Moto un nuevo método de instancia también llamado pitar() de la siguiente manera:

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

Así se sobre-escribe el método en la clase hija moto y se toma este método al usarlo en la clase principal.

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

Si, ya que al momento de crear la clase motoneta esta al ser heredera de la clase moto, la clase moto necesita un constructor por defecto para poder crear objetos de la clase motoneta, ya que se necesita un constructor en moto que acompañe a los objetos de la clase motoneta. Por lo que si se quiere que no haya problemas, se debe definir dentro del constructor de motoneta explícitamente con super() el constructor de 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?

Si es posible, porque se sobre-escribe el método arrancar(ligadura dinámica) que esta definida primeramente en la clase padre Auto, por lo que sí se puede sobre-escribir. Ahora el método arrancar en moto, se toma como un método de esta clase al llamarse una referencia de la clase Moto.

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

Si, ya que en el constructor de la clase Moto se usa el método pitar que se hereda de la clase padre Auto, por lo que la clase moto al heredar los métodos y atributos accesibles de Auto puede usarlos.

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?

Se debe agregar en el constructor por defecto lo siguiente:

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

Porque de esta forma en la clase principal cuando se crean los objetos con referencia bus y moto, estos al ser parte de las clases Bus y Moto respectivamente, son clases que heredan a la clase Auto por lo que por defecto entran al constructor Auto(), el cual es el constructor por defecto y por medio de este podemos obtener el número de autos creados.

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

No se puede utilizar ya que desde la clase moto no se puede heredar el método adelantar() ya que es de tipo default o paquete y no es accesible al heredar, y la clase Moto esta en un paquete diferente a la clase Auto.

7. En la línea 8, ¿por qué es posible utilizar el método arrancar si la clase Bus no lo define? Es posible usar el método arrancar(), ya que primero la clase Bus lo hereda de la clase carro y por consiguiente también se puede usar ya que este método es publico y se puede acceder a el.

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

Si se puede heredar ya que el método pitar es un método público, pero no se esta cumpliendo las buenas prácticas del buen encapsulamiento ya que el método pitar() debería ser encapsulado como protegido.

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

No imprime nada ya que ni en esta línea 10, ni en el método se pide imprimir el valor que retorna el método getVelocidad().

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?

Se debería agregar al código lo siguiente:

-En el constructor definido en la clase Bus se debe agregar lo siguiente para obtener la placa y la capacidad:

```
System.out.println(this.placa,this.capacidad);
```

- En el constructor definido en la clase Moto se debe agregar lo siguiente para obtener la placa y el modelo:

```
System.out.println(this.placa,this.modelo);
```

Otra opción puede ser crear métodos get de cada atributo en las clases Bus y Moto, y en la clase ObjTaller5H imprimir estos métodos.