

Taller 5 java ejercicio 2

Jhoneyker Delgado Urbina

Preguntas de 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).

Si se desea modificar el mensaje del método pitar al crear un objeto moto sin alterar la clase Auto se debe agregar a la clase Moto un método llamado pitar() de la siguiente manera:

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

De esta manera java sobrescribe el método pitar() para esta clase sin alterar al padre, esto se permite ya que el método tiene la misma firma y un retorno igual al del padre.

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 se evidencia un problema, ya que Motoneta al intentar heredar de Moto se observa que moto no tiene ningún constructor que no reciba parámetros por lo que el constructor de Motoneta deberá llamar de manera explícita al constructor 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?

Si, si es posible sobrescribir el método ya que arrancar() se encuentra declarado en la clase Auto y al escribirlo en la clase Moto con la misma firma y el mismo retorno java tomara por jerarquía el de la clase del objeto que se creo.

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

Se puede utilizar el método pitar() ya que este se encuentra definido en la clase padre Auto por lo que la clase Moto al ser hijo de Auto hereda todos

los métodos y los atributos (que sean accesibles desde la clase Moto) y los puede utilizar de forma directa y sin inconvenientes.

- 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 puede obtener la cantidad de autos creados si colocamos:

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

En el constructor que no recibe ningún parámetro después de la línea que le suma 1 a la variable num_autos para que lo muestre cada vez que se cree un objeto de la Clase Auto o de una clase hija.

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

Este método no se puede utilizar debido a que al definirse adelantar() en la clase Auto y tener accesibilidad por defecto (de paquete) la clase Moto no la podrá heredar por que pertenece a otro paquete, de esta manera al tratar de utilizarse adelantar() desde un objeto Moto mostrara error.

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

Es posible utilizar el método arrancar debido a que la clase Bus al ser hijo de la clase Auto heredara todos los atributos y métodos que la accesibilidad le permita.

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

De hecho si se puede utilizar el método pitar, ya que tiene accesibilidad publica por ende fue heredado y se puede invocar en cualquier parte del programa.

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

No imprime nada ya que el método getVelocidad() solo devuelve el valor de la velocidad y no se pide explícitamente que lo imprima(ni en el método ni en la línea 10).

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 podría obtener estos valores imprimiéndolos directamente desde el constructos colocando en la clase Moto:

```
System.out.println(this.placa + " " + this.modelo);
```

Este se colocaría después de la línea donde se invoca al método pitar. Para la clase Bus se coloca:

```
System.out.println(this.placa + " " + this.capacidad);
```

Este se colocaría después de la línea 12 de la clase Bus en donde se dan valores a los atributos de la instancia.

Otra opción seria crear un get para cada uno de esos valores y llamarlo desde el main para imprimirlos.