

PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

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).

Para modificar el mensaje del método pitar en la clase moto, se debe sobrescribir el método.

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 este es:



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 sobrescribir ya que cumple con las condiciones y es que el método tiene la misma firma del que se encuentra en la clase padre, y el tipo de retorno es compatible.

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

Se puede utilizar porque es heredado de la clase auto, además su modificador de acceso es público, esto quiere decir que todas las subclases tienen acceso al método.

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?

La modificación propuesta es la siguiente:

```

package paquete1;

public class Auto {

    public static int num_autos = 0;
    public int velocidad = 10;
    public String pitar = "Piiiiii";

    public Auto(){
        num_autos++;
    }

    void adelantar(){
        System.out.println("Puedo adelantar autos");
    }

    public void pitar(){
        System.out.println(this.pitar);
    }

    public void arrancar(){
        System.out.println("Encendido");
    }

    public void acelerar(){
        System.out.println("Avanzo constantemente");
    }

    public void frenar(){
        System.out.println("Me detengo");
    }

    public int getVelocidad(){
        return(this.velocidad);
    }

    //Punto 5
    public static int getNumAutos() {
        return num_autos;
    }
}

```

```

package paquete2;

import paquete1.Auto;

public class ObjTaller5H {
    public static void main(String[] args){
        Moto moto = new Moto("XYZ123", "2019");
        Bus bus = new Bus("ABC345", 20);
        //moto.adelantar();
        bus.arrancar();
        bus.pitar();
        moto.getVelocidad();
        //Punto 5
        System.out.println(Auto.getNumAutos());
    }
}

```

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 una restricción y es que solo es accesible dentro del mismo paquete, es decir no se puede acceder desde subclases en otros paquetes.

```
void adelantar(){  
    System.out.println("Puedo adelantar autos");  
}
```

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

Se puede utilizar método `arrancar` ya que este método es **heredado de la clase `Auto`**, y no está sobrescrito en `Bus`. El comportamiento por defecto del método en `Auto` se ejecuta.

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

Si se puede utilizar, ya que es heredado y además de tipo público lo que lo hace accesible.

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

Devuelve el valor de velocidad, pero no lo imprime, porque falta el **`System.out.println()`**.

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?

Cambios propuestos:

Bus

```
package paquete2;  
  
import paquete1.Auto;  
  
public class Bus extends Auto {  
    private String placa;  
    private int capacidad;  
    public int velocidad=20;  
  
    public Bus(String placa, int capacidad){  
        this.placa = placa; this.capacidad = capacidad;  
    }  
  
    public void acelerar() {  
        System.out.println("Avanzo lento");  
    }  
  
    public String getPlaca() {  
        return this.placa;  
    }  
  
    public int getCapacidad() {  
        return this.capacidad;  
    }  
}
```

```

package paquete2;

public class ObjTaller5H {
    public static void main(String[] args){
        Moto moto = new Moto("XYZ123", "2019");
        Bus bus =new Bus("ABC345", 20);
        //moto.adelantar();
        bus.arrancar();
        bus.pitar();
        moto.getVelocidad();
        System.out.println("Placa del bus: " + bus.getPlaca());
        System.out.println("Capacidad del bus: " + bus.getCapacidad());
    }
}

```

Moto

```

package paquete2;

import paquete1.Auto;

public class Moto extends Auto {
    private String placa;
    private String modelo;
    public int velocidad = 30;

    public String pitar = "Las motos no pitan";

    public Moto(String placa,String modelo){
        this.placa = placa;
        this.modelo=modelo;
        this.pitar();
    }

    public void acelerar(){
        System.out.println("Avanzo muy rapido");
    }

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

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

```

```

package paquete2;

public class ObjTaller5H {
    public static void main(String[] args){
        Moto moto = new Moto("XYZ123", "2019");
        Bus bus =new Bus("ABC345", 20);
        //moto.adelantar();
        bus.arrancar();
        bus.pitar();
        moto.getVelocidad();
        System.out.println("Placa de la moto: " + moto.getPlaca());
        System.out.println("Modelo de la moto: " + moto.getModelo());
    }
}

```