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).

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

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?

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

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?

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

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

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

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

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?

Solución

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 al crear un objeto moto y que al llamar su método pitar entregue otro mensaje diferente al de su clase padre, podemos sobre cargar el método creando uno igual en la clase moto pero con otro mensaje ejemplo

```
public class Moto extends Auto {
   // ... (resto del código de Moto)

   // Sobrecarga del método pitar()
   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?

esta clase seria un hijo directo de la clase moto y tendría todos los atributos propios, de su madre moto y de su abuelo auto, no habría problema pero si por ejemplo existe un método pitar en moto y en auto esta clase solo podrá acceder al de su padre no al de su abuelo

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 sobre escribir este método a pesar de que la moto herede todos los métodos publico de la clase auto esta sigue teniendo la capacidad de modificar o especializarse redefiniendo un método ya existente en su padre simplemente reescribiéndolo con la misma firma.

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

Porque la clase moto es una clase hija de auto y en este mismo se definió el método pitar, al ser una hija y no tener su propio método pitar utiliza el método de su padre y hereda también el valor por defecto definido en su padre.

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 crear un método getter para acceder a el numero de autos creados y agregarlo a un string dentro del código ObjTaller5H llamanda al método getter para tomar el valor almacenado en la instacion num_autos, motos y bus al ser hijos de auto siguen aumentando el contador de este objeto siendo entonces que esto no seria un problema en si y al ser esta instancia publica el acceso por el método getter no seria mayor problema.

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

El método acelerar es un método de paquete no uno publico entones los hijos de auto que pertenezcan a otro paquete no podrán acceder a este método a pesar de ser hijas de auto

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

Porque este método en la clase auto es publico es decir si puede acceder al método de su padre.

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

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

En la línea de 10 el método imprime 30 que seria el atributo de velocidad por defecto de un objeto de clase moto al redefinirse al crearse la 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?

Solo sería necesario utilizar métodos getter con los cuales accedas a los datos guadados en las instancias de la clase moto y bus respectivamente y ejecutar estos métodos getter en el main