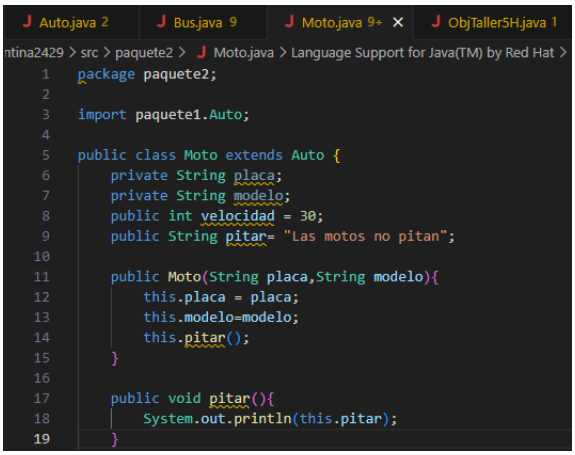
Preguntas de análisis

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 no tener que alterar nada de la clase Auto, se tendría que sobrescribir el método pitar en la clase Moto y agregarle un atributo de tipo String pitar que sea. “Las motos no pitan “, de esta forma cuando se cree un objeto de este tipo se ejecutara este método pitar con el atributo que agregamos ya que este se ejecuta junto con el constructor de los objetos tipo moto.



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

Evidenciaría un problema si intentan crear un objeto tipo Motoneta sin crear su propio constructor porque no heredaría el constructor de su clase padre (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 sobrescribir este método porque en la clase padre este es un método de instancia, además tiene un tipo de retorno compatible y la misma firma que el método heredado, asi que no hay problema

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

Porque este método originalmente se hereda de la clase padre Auto, así que puede llamarla aun dentro del constructor de la clase Moto

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?

Creando un método static que nos retorne el valor de num_autos ya que es un atributo de clase

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

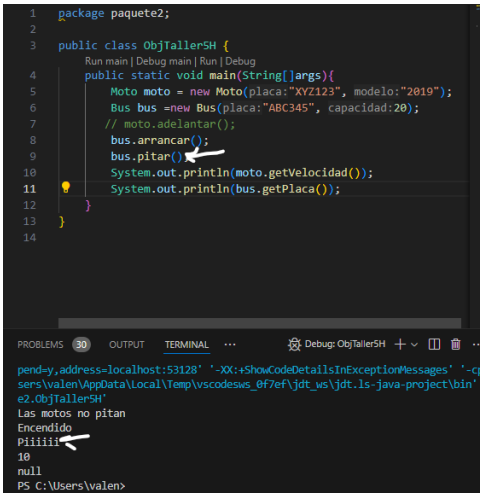
Porque el método adelantar de la clase Auto tiene visibilidad por default, es decir, tipo package y como auto es la única clase en paquete 1, entonces el resto de clases no pueden acceder a ella.

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

Porque es un método de la clase Auto así que lo hereda a la clase Bus, gracias a que tiene visibilidad pública se hace sin ningún problema

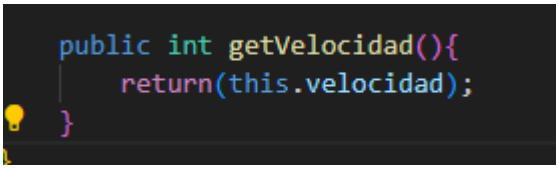
8. En la línea 9 de la clase ObjTaller5H, ¿por qué no puedo utilizar el método pitar, si este fue heredado?

A mi me parece que si puede utilizarse, ya que es un método con visibilidad publica de la clase Auto así que la clase Bus al ser su hijo lo hereda



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

Imprime 10 por ligadura estática, es decir, originalmente es un método de la clase Auto (padre), y como esta escrito en ese archivo, con el this, entonces este hace referencia a la velocidad de la clase Auto, aun cuando este método es heredado por sus hijos



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 el método get() para cada atributo requerido en su respectiva clase , es decir, getCapacidad() y getPlaca() para bus, además de getModelo() y getPlaca() para moto, ya que estos atributos están definidos en cada clase (son propios de ella).