



PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

Respuestas

1. Sobrescribe el método pitar en Moto:

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

2. No hay problema directo al heredar, pero atributos private en Moto no serán accesibles en Motoneta.
3. Sí, porque arrancar es public en Auto.

```
@Override  
public void arrancar() {  
    System.out.println("La moto está arrancando");  
}
```

4. Porque pitar es public en Auto.
5. Usa la variable estática directamente:

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

6. No es accesible porque adelantar tiene acceso por defecto (package-private). Cámbialo a protected o public.
7. Porque Bus hereda arrancar de Auto, y es public.
8. Ocurre un conflicto entre el método y el atributo pitar en Auto.
Solución: renombra el atributo.
9. Imprime 10, porque getVelocidad() accede al atributo de la clase Auto, no al redefinido en Moto.
10. Agrega métodos getter:

```
public String getPlaca() { return this.placa; }  
public String getModelo() { return this.modelo; }  
public int getCapacidad() { return this.capacidad; }
```