

Taller #5, Ejercicio 2

- 1) **Sí 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)**

Basta con añadir las líneas:

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

En la clase moto.

- 2) **Suponga que se agrega una nueva clase al código, class Motoneta, y esta hereda de la clase Moto, ¿Evidencia algún problema?**

No hay ningún problema siempre y cuando se establezca como una clase pública en un archivo Motoneta.java, dentro del paquete2 (Para evitar errores o que partes de la clase no se hereden por ser private-package).

También, es necesario que el constructor de la clase motoneta tenga definido (String, String).

- 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?

Sí, es posible.

Esto es debido a que es un método público de la clase Auto, además se está empleando la misma firma del método, por tanto sobrescribiendo el método en lugar de sobrecargarlo.

La notación @override es solo por convención, y no afecta en si el método se sobrescribe correctamente o no.

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

En el caso particular en que en la clase Auto se ha definido el método pitar() como "public", es posible debido a que los métodos públicos pueden ser heredados por cualquier clase hija, independientemente de si se encuentran en otro paquete o no.

- 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?**

Desde que es un atributo público, puede ser accedido sin necesidad de modificar la clase Auto, siempre que se acceda desde una de sus clases hijas, como Bus o Moto, de la forma: Moto.num_autos o Bus.num_autos.

Si se desea realizar una modificación, siempre se puede crear un método getter en la

clase Auto para acceder al archivo, este método tendrá que ser estáticos para acceder a los atributos, y podrá ser invocado (desde que sea publico) desde las clases Bus o Moto.

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

No se puede utilizar porque de hecho, no fue heredado.

En el caso particular en que solo se ha declarado el método como “void adelantar()”, entonces no es posible utilizar el método desde moto, esto debido a que el alcance del método en Auto se vuelve “private-package”, donde solo las clases hijas dentro del mismo paquete pueden heredar el método, y debido a que la clase Moto está definida en un paquete diferente, no se hereda.

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

Debido a que el método es heredado desde la clase Auto, ya que es un método público, y por tanto aunque no está definido en la clase Bus, aún existe.

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

El método si se puede utilizar, debido a que es definido como público y por tanto es heredado con normalidad para todas sus clases hijas, Dado que tanto Bus como Moto son clases hijas de Auto, ambas clases pueden utilizar el método.

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

Nada, debido a que el método no tiene ninguna línea en que se defina una instrucción para imprimir algo en consola. En su lugar, es un simple método que devuelve en su valor de retorno un int que corresponde con la variable velocidad del objeto desde el que se está llamando.

Si quisiéramos que esta información aparezca en pantalla, sin modificar el método, sería necesario llamar el método dentro de un System.out.println(). Es decir:
System.out.println(getVelocidad());

10) Si quisiera obtener el valor de la placa de las clases Moto y Bus, además de su modelo y capacidad respectivamente, ¿Qué debo agregar al código?

Debería añadir métodos getter para estos atributos, dentro de las respectivas clases. Para la placa, el método sería igual para ambas, y se añadiría un getModelo en la clase Moto, y un getCapacidad en la clase Bus.

Es necesario realizar esto desde un método dentro de la misma clase debido a que los atributos son privados, y por tanto no pueden accederse simplemente como bus.capacidad.

Así mismo, este método debe ser público.

O tendrá que existir un método público que al ser llamado, ejecute el método privado getter.

Por ejemplo, en Bus:

```
public int getCapacidad(){  
    System.out.println("La placa es:"+getPlaca());  
}
```

```
        return capacidad;
    }
    private String getPlaca(){
        return placa;
    }
```

La línea `System.out.println(bus.getCapacidad());` en la clase `ObjTaller5H` imprimirá la línea “La placa es: ABC345”, y luego la línea “20”.