

Ejercicio 1

- a) El objeto se convierte en elegible para la recolección de basura (garbage collection). Si el recolector de basura decide eliminarlo, se llamará al método `finalize()` antes de liberar la memoria asociada. En este caso, el método `finalize()` imprimirá el mensaje:

Matando a: [nombre de la persona].

- b) Podemos acceder al atributo `dueno` de `auto` y luego llamar al método `getNombre()` de la clase `Persona`.
- c) Para agregar un dueño, usamos el método `setDueno()` y asignamos un objeto de tipo `Persona`.
- d) Para acceder al atributo `velocidadMaxima` del motor de `auto2`, primero obtenemos el motor usando `getMotor()` y luego llamamos a `getVelocidadMaxima()`.

Código:

```
int velocidadMaxima = auto2.getMotor().getVelocidadMaxima();
System.out.println(velocidadMaxima);
```

- e) Matando a: Santiago
Soy Santiago
- f) En la línea 16, `personas[1]` apunta a `personas[2]`. El método `toString()` de `Persona` será llamado, por lo que imprimirá:

Soy Daniel

- g) Debemos sobrescribir el método `toString()` en la clase `Vehiculo`. Ejemplo:

```
@Override
public String toString() {
    String duenoInfo = (dueno != null) ? dueno.getNombre() : "Sin dueño";
    return "Placa: " + placa + ", Dueño: " + duenoInfo;
}
```

Con esta modificación, la llamada a `System.out.println(auto2)` imprimirá algo como:

Placa: TXT-452, Dueño: Sin dueño
(si no se asignó dueño) o:

Placa: TXT-452, Dueño: [nombre del dueño]

h) `auto.getDueno().setMejorAmigo(personas[2]);`

Explicación:

`auto.getDueno()`: Obtiene al dueño del vehículo auto.

`.setMejorAmigo(personas[2])`: Asigna al tercer elemento del arreglo `personas` como el mejor amigo del dueño.