

Respuestas Ejercicio 1

A. El Garbage Collection se encargará de destruir el objeto, pero primero ejecutará el `finalize()` que en este caso es: **`System.out.println("Matando a: " + nombre);`**

B. Usando el método `auto.getDueno()`

C. Podemos usar la lista de objetos `Persona` `personas[]`, escribiendo `auto2.setDueno(personas[un número entre 0 y 4])`

D.

```
Vehiculo auto2 = new Vehiculo("TXT-452", Motor.GASOLINA);  
System.out.println(auto2.getMotor().getVelocidadMaxima());
```

E. Se imprimirá: **Matando a: Alejandro**, **Matando a: Alexander** y por supuesto también **Matando a: Santiago** porque se ejecutará el `finalize()` cada vez que se deja de apuntar a los objetos.

F. Imprimirá **Soy Jaime** porque ese es el `toString()` que tiene definida la clase `Persona`

G. Se deberá añadir el método `toString()` a la clase `Vehículo` así:

```
public String toString() {  
    return placa+" "+dueno;  
}
```

H.

```
Vehiculo auto = new Vehiculo("ABC-306", Motor.ELECTRICO, personas[4]);  
auto.getDueno().setMejorAmigo(personas[2]);
```

Respuestas Ejercicio 2.

A.

System.out.println("char : " + *funcion(c)*); Entra a int: 103, porque int es el tipo de dato inmediatamente superior a char, siendo 103 el código asignado a la letra g minúscula en el código ASCII

System.out.println("short : " + *funcion(s)*); Entra a int: 2 porque int es inmediatamente superior al tipo de dato short

System.out.println("byte : " + *funcion(b)*); Entra a int: 1 por la misma razón que el punto anterior

System.out.println("long : " + *funcion(l)*); Entra a double: 9.99999999E8, porque double es mayor que long, y lo expresa en notación científica.

System.out.println("integer : " + *funcion(i)*); Entra a int: 51232, porque el dato ingresado es un int

System.out.println("double : " + *funcion(d)*); Entra a double: 12.4, porque el dato ingresado es un double.

System.out.println("float : " + *funcion(f)*); Entra a double: 5.650000095367432, porque double es inmediatamente mayor a float, y añade esos decimales en un intento de aproximar en binario.

B.

Primer cambio, ahora short y byte entran por short y no por int, ya que short es el tipo de dato inmediatamente mayor a byte.

Segundo cambio, ahora long y float entran por float, ya que double es mayor que float y por lo tanto long que es menor que float y double, entra por float que es el inmediatamente mayor a él

Tercer cambio, lanza un error ya que double no puede entrar por ningún método, al ser el mayor tipo de dato primitivo.

Cuarto cambio, todos entran por double al ser el mayor tipo de dato.