

Ejercicio 1:

- A) El objeto queda elegible para que el Garbage Collector pueda eliminarlo en algún momento de la ejecución del código
- B) El nombre es “Alexander” aunque posteriormente se le reasigne “null” a persona[4].
- C) Utilizando el método setDueño de la clase vehículo
- D) Se debe utilizar el método getMotor() para obtener el tipo de motor del objeto creado para después llamar al método getVelocidadMaxima:

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

- E) El programa imprimirá matado a : Santiago, por la ejecución del método finalize()
- F) Persona[1] y persona[2] quedan apuntando al mismo objeto y por el método toString() que se ejecuta automáticamente, imprime en consola “Soy Daniel”
- G) Crear el metodo toString() en la clase vehículo para que imprima automáticamente la placa y el dueño, por ejemplo:

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

H) Se obtiene el dueño del auto con:

```
Persona dueñoAuto = auto.getDueno();
```

Después se asigna el tercer elemento de la lista personas como el mejor amigo del dueño del carro:

```
dueñoAuto.setMejorAmigo(personas[2]);
```

Ejercicio 2:

A)

Entra a int: 103

- porque el método `funcion(int a)`, tiene un parámetro superior más cercano (a comparación con el otro método) a `char` por sobrecarga de métodos, además al método solo recibir enteros, lo convierte en 103 por que es el numero entero al que pertenece la letra "g" en ASCII

Entra a int: 2

- porque el método `función(int a)`, tiene un parámetro superior más cercano (a comparación con el otro método) a `short` por sobrecarga de métodos.

Entra a int: 1

- porque el método `función(int a)`, tiene un parámetro superior más cercano (a comparación con el otro método) a `byte` por sobrecarga de métodos.

Entra a double: 9.99999999

- porque el método `funcion(double a)`, tiene un parámetro superior más cercano (a comparación con el otro método) a `long` por sobrecarga de métodos, además al método solo recibir valores tipo `double`, lo convierte automáticamente en uno añadiendo un punto decimal

Entra a int: 51232

- porque el método `función(int a)` concuerda con el tipo de dato ingresado por sobrecarga de métodos.

Entra double: 12.4

- por que el método función(double a) concuerda con el tipo de dato ingresado por sobrecarga de métodos.

Entra a double: 5.65

- porque el método funcion(double a), tiene un parámetro superior más cercano (a comparación con el otro método) a float por sobrecarga de métodos.

ACTIVAR LA funcion(short a): modifica las siguientes impresiones en consola (las demás siguen ejecutándose igual por que los demás métodos tienen parámetros superiores más cercanos)

Entra a short: 1

- porque el método funcion(short a), tiene un parámetro superior más cercano (a comparación con el otro método) a byte por sobrecarga de métodos.

Entra short a: 2

-porque el método función(short a) concuerda con el tipo de dato ingresado por sobrecarga de métodos

ACTIVAR LA funcion(float a): modifica las siguientes impresiones en consola (las demás siguen ejecutándose igual por que los demás métodos tienen parámetros superiores más cercanos)

Entra a float: 9.99999999

- porque el método funcion(float a), tiene un parámetro superior más cercano (a comparación con el otro método) a long por sobrecarga de métodos, además al método solo recibir valores tipo float, lo convierte automáticamente en uno añadiendo un punto decimal

Entra a float: 5.65

porque el método `funcion(float a)` concuerda con el tipo de dato ingresado por sobrecarga de métodos.

ACTIVAR LA `funcion(float a)` Y DESACTIVAR LA `funcion(double a)`: El código no compila debido a que el `double` no tiene ningún método al que incluirse implícitamente al ser un tipo de orden superior a los demás métodos.

DESACTIVAR TODO EXCEPTO `funcion(double a)`: el código compila con normalidad convirtiendo todo implícitamente en `double` (a excepción al `double` que ya lo era)