

Taller Java #4

Ejercicio 1

a) Al no estar referenciado, será cuestión para que el GC elimine al objeto y libere el espacio de memoria y se imprimirá un mensaje que dice que se está matando a la persona debido a la función `finalize()`.

b) Habría que llamar al método `getDueno` en el contexto del objeto `auto`, y se debe imprimir esto, pues gracias al método `toString()`, imprimir el mero objeto y nos da el nombre del dueño.

Ejemplo:

```
String dueno = auto.getDueno().getNombre();
```

c) Habría que llamar al método `setDueno` en el contexto del `auto2`, y poner como argumento a la persona a la cual se le quiere atribuir este `auto`.

Ejemplo:

```
Auto2.setDueno(personas[4]);
```

d)

```
int velocidad = auto2.getMotor().getVelocidadMaxima();
```

e) Matando a: Santiago, pues si el GC es super eficiente no daría tiempo a imprimir el contenido de `personas[3]`.

f) Se imprimiría Hola, soy Daniel. Esto debido a que esto es lo que se imprime en el objeto con nombre Daniel, al cuál está apuntando el apuntador `personas[1]`.

g) Habría que añadir a la clase `Auto` un método `toString()` con las siguientes instrucciones:

```
public String toString() {  
    return "Dueño: " + this.getDueno().getNombre + ", Placa: " + getPlaca()  
}
```

h)

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

Ejercicio 2

a) Impresiones:

13: char; Entra a int: 103

La variable `c` es de tipo `char`, por lo que se ejecutará en el método con parámetro `int`, e imprime 103 porque `char` puede devolver tanto su valor como carácter como su valor numérico, y al ser `int` como argumento devuelve el valor numérico del carácter `g`.

14: short; Entra a int: 2

La variable `s` es de tipo `short`, por lo que se puede convertir a `int`, y entra en este método, e imprime su valor.

15: byte; Entra a int: 1

La variable b es tipo byte, por lo que entra al método de int, e imprime su valor.

16: long: Entra a double: 999999999

La variable l entra a double porque es más grande que int, e imprime su valor.

17: integer: Entra a int: 51232

La variable i entra a int porque su tipo coincide con el parámetro del método e imprime su valor.

18: double: Entra a double: 12.4

Al coincidir el tipo de d, este va al método con tipo double, e imprime su valor.

19: float: Entra a double: 5.65

Porque una variable tipo float no se puede convertir a una de tipo int, y la f al final representa su tipo float, sin embargo, no hace parte de su valor por lo que esta no es tenida en cuenta al imprimir su valor.

b)

- Si se habilita el método que recibe un argumento de tipo short, las líneas 14 y 15, cambian su impresión por:
short: Entra a short: 2
byte: Entra a short: 1
Respectivamente pues si ambos tipos de variables pueden ser tomadas como short, y los argumentos tipo short tienen prioridad respecto a los de tipo int.
- Si se habilita el método que recibe argumentos de tipo float, la línea 19 va a imprimir:
float: Entra a float: 5.65
Porque al coincidir con el tipo de parámetro este tendrá prioridad a uno diferente.
- Si se comenta el método de tipo double y se habilita el de tipo float saldrá un error, pues el método llamado en la línea 18 lo hace con un argumento de tipo double, que no puede transformarse a ningún tipo diferente.
- Todas imprimirán la Entra a double, pues todas las variables podrían convertir a double.