

## Ejercicio 1

- A. ¿Qué pasaría si se pierde la referencia de un objeto de tipo Persona?

Si se pierde la referencia completamente, eventualmente el garbage collector liberará la memoria, cuando eso ocurra se imprime “matando a: “ y luego el nombre.

- B. ¿Cómo podríamos conocer el nombre del dueño de la variable auto de la línea 11 del método main?

Por `auto.getDueño()`

- C. ¿De qué manera podemos agregar un dueño al Vehículo de la línea 13 del método main?

Agregando un tercer argumento al llamar al constructor, de tipo Persona.

- D. Usando la variable `auto2` de la línea 13 del método main, obtenga el valor del atributo `velocidadMáxima` del motor del vehículo. Adjunte su propuesta.

usamos `auto2.getMotor().getVelocidadMaxima()`;

- E. Suponga que, al momento de perder la referencia al objeto, se borra del sistema, es decir el garbage collector es muy eficiente, ¿Qué imprimiría al ejecutar el método main por consola?

Matando a : Alejandro

Matando a: Jaime

Matando a: Alexander

Matando a: Santiago

“Soy Santiago”

- F. ¿Qué ocurre al momento de ejecutar la siguiente línea después de la línea 16 `System.out.println(personas[1])` ? Explique

Se imprime “Soy Daniel” Pues la línea 16 hace que el índice 1 apunte a Daniel, igual que el índice 2, por lo que el índice 1 pierde la referencia a Jaime.

- G. ¿Qué modificación al código debo hacer para que al momento de ejecutar `System.out.println(auto2)`, me aparezca la placa del vehículo y el dueño del vehículo?

Se debe sobrescribir `toString`, con algo así:

```

public String toString(){
    if (this.dueno==null){
        return "Placa: "+this.placa+ " |sin dueno";
    }
    else{
        return "Placa: "+this.placa+" dueno: "+ this.dueno.getNombre();
    }
}

```

- H. Usando la variable auto de la línea 11 del método main, y usando el atributo dueno, asigne de mejorAmigo al tercer elemento del listado personas. Adjunte su propuesta.

```
auto.getDueno().setMejorAmigo(personas[3]);
```

## Ejercicio 2

- A. Según el siguiente código, indique qué se imprime por consola y explique el porqué de cada línea donde se imprime.

Se imprime lo siguiente:

char : Entra a int: 103

short : Entra a int: 2

byte : Entra a int: 1

long : Entra a double: 9.99999999E8

integer : Entra a int: 51232

double : Entra a double: 12.4

float : Entra a double: 5.650000095367432

Esto ocurre pues java trata de meter el argumento en una función que acepte un tipo superior y lo mas cercano posible al valor dado, según la siguiente tabla:

Orden tipos de datos primitivos

- byte
- short
- int
- long
- float
- double

Esto significa que char, byte, short e int entran en int. Long no (De hecho utiliza mas memoria, 64 bits en vez de 32, como double) entonces entra a double, que es el tipo superior mas cercano disponible. Misma situación para double y float.

- B. Realice los siguientes cambios, teniendo siempre como referencia el código inicial. Explique cómo y por qué cambia lo que se imprime por pantalla.

- Active la función que recibe un short.

¡Listo! Entran, según la tabla, byte y short a la función de short, el resto sigue igual.

```
char : Entra a int: 103
short : Entra a short: 2
byte : Entra a short: 1
long : Entra a double: 9.99999999E8
integer : Entra a int: 51232
double : Entra a double: 12.4
float : Entra a double: 5.650000095367432
```

Ahora, con respecto a char, parece un tema técnico. Java usa 16 bits para los chars, igual que para los short. El problema con eso es que el short tiene signo, es decir, que en lo positivo, solo puede almacenar la mitad de números que char. Por tanto, no todos los caracteres caben en un short y toca usar un int, que, aún con signo, tiene espacio para todos los chars posibles.

- Active la función que recibe un float.

```
char : Entra a int: 103
short : Entra a short: 2
byte : Entra a short: 1
long : Entra a float: 1.0E9
integer : Entra a int: 51232
double : Entra a double: 12.4
float : Entra a float: 5.65
```

¡Listo! Según la tabla, entran long y float a float, double no cabe en float y entra en su propia función que recibe double.

- Comente la función que recibe un double y active la que recibe un float.

Según la tabla, double no cabe en ninguna de las funciones y no compila.

- Comente todas las funciones, excepto la que recibe un double.

```
char : Entra a double: 103.0
short : Entra a double: 2.0
byte : Entra a double: 1.0
long : Entra a double: 9.99999999E8
integer : Entra a double: 51232.0
double : Entra a double: 12.4
float : Entra a double: 5.650000095367432
```

Según la tabla, entra todo a double.