

Taller #4 Java – Punto 2

A: ¿Qué pasaría si se pierde la referencia de un objeto tipo persona?

Dado que el objeto no crea ningún otro apuntador, el recolector se encargará automáticamente de liberar el espacio en memoria y podrá ser reasignado a otro objeto.

Aunque para algunos objetos persona, tales como el persona[4], es posible aún acceder a ellos a través del atributo dueño del objeto de la clase vehículo en que se referencia. (auto)

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

Con el método auto.getDueno(). Dado que es un objeto instanciado de la clase Vehículo, puede acceder a los atributos privados.

C: ¿De que manera podemos agregar un dueño al vehículo de la línea 13 del método main?

Con el método auto.setDueno(). Este Método está definido para asignar un objeto de la clase Persona como el valor del atributo “dueño” del objeto.

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

Con la instrucción `System.out.println(auto2.getMotor().getVelocidadMaxima());`

Se puede obtener el valor de la velocidad máxima del motor, esto debido a que en lugar de intentar acceder desde el vehículo (con `auto2.motor`), estamos haciendo uso de un método que retorna directamente el objeto motor dentro del vehículo, como sí mismo, y no como atributo de la clase vehículo, haciendo que la instrucción anterior sea desde la iteración de la clase motor, consecuentemente permitiendo acceder a sus atributos privados.

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?

Debido a que los objetos personas se encuentran dentro de un arreglo (Array), los índices del array que estaban ocupando se volverán null, permitiendo inmediatamente que se defina un nuevo objeto persona que ocupe el espacio ahora liberado.

Debido a que el main no está intentando imprimir ningún objeto que ya halla sido eliminada además de personas[3], el único cambio sería que el programa tendría un error de compilación al intentar llamar al método toString() de un Null. (Debido a que en la línea anterior se ejecuta `personas[3].finalize()`).

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

El programa arroja la línea “soy Daniel”, antes de la línea “matando a: Santiago”.

Esto es debido a que previamente se cambió el apuntador de personas[1], por el mismo apuntador que personas[2], cambiando de “Jaime” a “Daniel”.

La razón por la que imprime el atributo nombre en lugar del objeto, es debido a la creación del método toString() de la clase, que devuelve “Soy:+nombre” al ser llamado automáticamente a través del método `system.out.println()`.

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?

Sería necesario añadir el método publico `toString` para la clase vehículo, de tal forma que este devuelva el nombre del dueño y la placa del vehículo.

```
public String toString(){  
    return "placa: "+placa+"\n"+"dueño:"+dueno;  
}
```

Dado que `placa` es un `String` no habrá problema, así mismo, “dueño” llamará a su propio método `toString` (El de la clase `persona`), y eso será lo que se verá en pantalla.

En el caso específico del objeto `auto2`, dado que no se definió el atributo `dueño`, devolverá `dueño:null`.

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

Dado que no es posible utilizar la instrucción “`auto.dueno`” debido al acceso privado del atributo `dueño`, es necesario hacerlo a través de los métodos empleados en las clases para obtener los valores, tal que:

```
auto.getDueno().setMejorAmigo(personas[3]);  
System.out.println(auto.getDueno().getMejorAmigo());
```

Añadiendo estas 2 líneas es posible asignar la `personas[3]` como mejor amigo desde el objeto `auto`, haciendo uso de los métodos de cada respectiva instancia en lugar de intentar una asignación directa, que no funciona.