



PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

Preguntas de análisis Ejercicio 1:

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

Si un objeto de tipo persona deja de tener algún apuntador que lo referencie, este “muere” porque **ya no se puede acceder a él** y en el caso de Java, **en algún momento el Garbage Collector elimina el objeto** de la memoria del computador para liberarla y que pueda ser reutilizada. **Pero antes**, se fija si hay un método **finalize()** y ejecuta dicho código antes de eliminar por completo el objeto.

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

Como el atributo dueño es privado, la forma de acceder a este es mediante el método **getDueno()** que nos retornará un objeto de tipo persona (el dueño), ahora, para acceder al nombre tenemos que llamar también al método **getNombre()** porque el atributo nombre también es privado, así: **auto.getDueno().getNombre()**. Para que esto se muestre en pantalla habría que meterlo dentro de un **Sistem.out.println()**.

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

Como el **atributo** dueño es **privado**, para poder agregarle un dueño a **auto2**, debemos usar el método **setDueno()** y pasarle como argumento algún objeto de tipo persona, pero como las personas están almacenadas en una lista, se vería algo así: **auto2.setDueno(personas [])**, y dentro del corchete **[]** el índice que tiene el objeto que se desea poner como dueño (sin superar el largo de la lista-1). Incluso, se puede seguir la misma lógica, pero haciendo que el dueño sea un nuevo objeto que no pertenezca a la lista: **auto2.setDueno(new Persona (nombre))**.

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

Usando la variable **auto2**, podemos acceder al atributo **velocidadMáxima** de su respectivo motor así: **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?

Si el objeto pierde la referencia por haber ejecutado un método **finalize()** o por haber sido cambiado para apuntar a **null**, el código **imprimiría null cuando se intente acceder al apuntador**, pues **al objeto ya no sería posible acceder** de ninguna manera (no tiene apuntadores y por tanto el Garbage Collector lo elimina). Además, **el garbage antes de eliminar el objeto y se fija si hay un finalize()** para ejecutarlo primero, incluyendo los prints (Como es el caso de persona). En nuestro caso, imprimiría:



PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

Matando a: Alejandro //Su único apuntador pasa a apuntar a null

Matando a: Jaime //Se queda sin apuntador porque persona[1] deja de apuntarlo para apuntar a persona[2]

Matando a: Santiago //Se llama al finalize

Soy Santiago //Porque esta línea se ejecuta en paralelo, el programa no se pausa

//No se mata a Alexander porque, aunque persona[4] pasa a apuntar a null, el atributo dueño de auto sigue apuntando a ese objeto, evitando que se elimine.

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

Al intentar imprimir personas[1] el código se va hacia el método toString() que es el que indica el valor que será mostrado al imprimir para los objetos de esta clase (“Soy”+nombre), pero nombre en este caso sería “Daniel”, porque en la línea anterior de código (16) **le asigna al apuntador personas[1] el mismo objeto al que apunta personas[2]** y por tanto su información cambia.

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?

Agregar un método toString() para que así, cada que se imprima un objeto de tipo auto, se muestre en la consola su placa y dueño, de la siguiente manera:

```
public String toString() {  
    return “Placa del vehículo: ” +this.placa+ “Dueño del vehículo: ”+this.dueno;  
}
```

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 personas. Adjunte su propuesta.

Podríamos escribir el siguiente fragmento:

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

Accedemos primero al dueño de auto con el método getDueño(), el cuál devuelve un valor de tipo Persona y con el valor de esa Persona podemos acceder finalmente al método setMejorAmigo(). Se pone personas[3] porque es el tercer elemento de la lista.