



PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

Preguntas de análisis

- A. ¿Qué pasaría si se pierde la referencia de un objeto de tipo Persona?
- B. ¿Cómo podríamos conocer el nombre del dueño de la variable auto de la **línea 11** del método main?
- C. ¿De qué manera podemos agregar un dueño al Vehículo de la **línea 13** del método main?
- 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
- 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?
- F. ¿Qué ocurre al momento de ejecutar la siguiente línea después de la **línea 16** `System.out.println(personas[1])` ? Explique
- 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?
- 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.

Solución

- A) Este objeto se convertiría en un candidato para ser recogido por el Garbage Collector. No obstante, no sería eliminado de inmediato, ya que esto depende de cuándo decida actuar. Aun así, se puede utilizar el método `finalize()`, haciendo que se imprima:
"Matando a: [nombre]"
- B) Se podría acceder a través de los métodos `get` de `getDueno` y `getNombre` de la clase Vehiculo, y se vería algo así:
`System.out.println(auto.getDueno().getNombre());`
- C) Utilizando el método `setDueno` de la clase Vehiculo, se podrá establecer, y el código se vería del siguiente modo:
`auto2.setDueno(personas[0]);`
(Lo que pondrá a como dueño al primer elemento del arreglo 'personas')
- D) Se podría conseguir la velocidad utilizando el enum Motor y su método `getVelocidadMaxima()`; el código se vería así:

`Int velocidadMaxima=auto2.getMotor().getVelocidadMaxima();`

`System.out.println(velocidadMaxima);`
- E) Cuando se pierde la referencia al objeto `personas[0]`, se ejecuta el método `finalize()` y se imprime "Matando a: Alejandro". Al llamar a `System.out.println(personas[3])`, se ejecuta el método `toString`, que imprime "Soy Santiago", por lo que la salida de consola se vería algo

así:

Matando a: Alejandro (al ejecutarse el método)

Soy Santiago(al llamarse en el main)

F) Después de ejecutar `System.out.println(personas[3])`, el programa muestra el texto devuelto por el método `toString()` del objeto en `personas[3]`. Aunque `finalize()` se haya ejecutado previamente, el objeto sigue existiendo y su referencia no se ha perdido. Por lo tanto, la consola imprimiría algo como: “Soy Santiago”.

G) se sobre escribe el método `toString` en la clase `Vehiculo`, y se propone el siguiente cambio:

`@Override`

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

H) Se puede utilizar el método `setMejorAmigo` de la clase `Persona` para establecerlo; el código sería algo como esto

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