

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

Si se pierde la referencia de un objeto de tipo Persona y no hay otras referencias a él, será recolectado por el Garbage Collector. Esto libera la memoria ocupada por el objeto, haciéndolo inaccesible.

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

Podríamos obtenerlo con un método similar a este:

```
System.out.println(auto.getDueno().getNombre());
```

Asumiendo que existen métodos getDueno() en Vehiculo y getNombre() en Persona.

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

Podemos agregar un dueño usando un método como:

```
Auto2.setDueno(persona);
```

Esto requiere que auto2 esté inicializado y que la clase Vehiculo tenga definido el método setDueno().

Usando la variable auto2 de la línea 13 del método main, obtenga el valor del atributo velocidadMaxima del motor del vehículo.

El código puede ser:

```
Int velocidadMaxima = auto2.getMotor().getVelocidadMaxima();  
System.out.println("Velocidad máxima: " + velocidadMaxima);
```

Para esto es necesario que las clases Vehiculo y Motor tengan los métodos getMotor() y getVelocidadMaxima().

Suponga que, al momento de perder la referencia al objeto, se borra del sistema. ¿Qué imprimiría al ejecutar el método main por consola?

Si el Garbage Collector elimina objetos y el programa intenta acceder a ellos, se va a generar una excepción como NullPointerException. En caso contrario, imprimiría los valores definidos, o null si las referencias son nulas.

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

Si personas[1] tiene una referencia válida, imprimirá algo como Persona@hashcode, a no ser que se haya sobrescrito el método toString() en la clase Persona. Si es null, imprimirá null.

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

Se debe sobrescribir el método `toString()` en la clase `Vehiculo`:

```
@Override  
  
Public String toString() {  
    Return "Placa: " + this.placa + ", Dueño: " + (this.dueno != null ?  
    this.dueno.getNombre() : "Sin dueño");  
}
```

Usando la variable `auto` de la línea 11 del método `main`, y usando el atributo `dueno`, asigne de mejor amigo al tercer elemento del listado `personas`.

Primero, asegúrate de que la clase `Persona` tiene un atributo `mejorAmigo` y un método `setMejorAmigo()`:

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

Esto requiere que `auto.getDueno()` no sea `null` y que `personas[2]` esté inicializado.