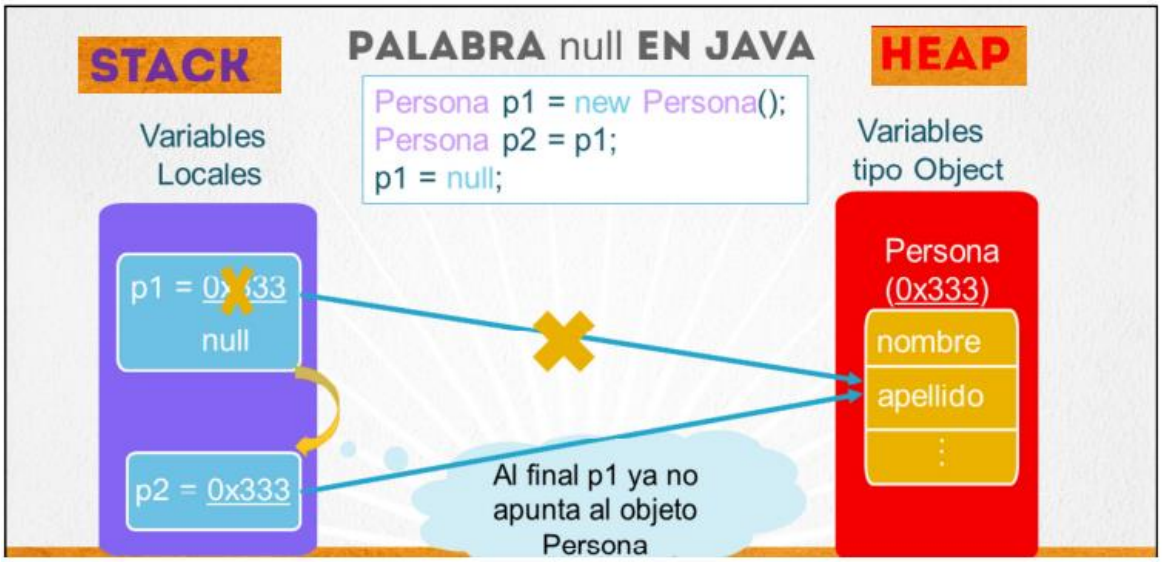
PALABRA NULL



La palabra null en Java la utilizamos con el objetivo de indicar que aún no se le ha asignado ninguna referencia de ningún objeto a una variable de tipo Object.

No es posible asignar el valor null a una variable de tipo primitivo.

Como podemos observar en la figura, en primer lugar creamos la variable p1 de tipo Persona, la cual se crea en la memoria heap y se le asigna la ubicación de memoria 0x333.

Posteriormente creamos la variable p2 y el asignamos la referencia almacenada por la variable p1, es decir, que ahora tanto la variable p1 como la variable p2 apuntan al mismo objeto y ambas variables pueden acceder a él.

Finalmente la variable p1 decidimos que ya no vamos a utilizarla, y para ello asignamos el valor de null, esto significa que pierde la referencia del objeto creado y por lo tanto solamente la variable p2 podrá ahora acceder al objeto Persona asignado en la ubicación de memoria 0x333.

¿Ahora, qué sucede si hacemos p2 = null ?

Esto significaría que el objeto Persona ya no lo apunta ninguna variable, y por lo tanto queda inaccesible. Desde el punto de vista del recolector de basura de Java sería un objeto candidato para ser eliminado de la memoria, ya que ninguna variable puede accederlo y por lo tanto es inservible, ya sólo queda que sea eliminado de la memoria Heap por medio del proceso de recolector de basura llamando al método System.gc(), es decir, garbage collector o recolector de basura.

Cabe aclarar que el recolector de basura no podemos obligarlo a que inicie su proceso de limpieza de objetos, sólo podemos mandar a llamar al método y esperar a que la misma máquina virtual de Java decida el momento en que es más conveniente iniciar este proceso, ya que es un proceso que utiliza muchos recursos. Sin embargo, si ya no vamos a utilizar un objeto, es conveniente y buena práctica que asignemos el valor de null a la variable que estaba apuntando al objeto creado previamente, con esto una vez que inicie el proceso de recolección de basura el objeto creado será candidato a ser eliminado.

Finalmente, podemos darnos cuenta que las variables creadas en la memoria Stack son temporales y de menor duración, y una vez concluido el método y/o programa en el que fueron creadas dichas variables, éstas se destruyen. Sin embargo las variables creadas en la memoria Heap, que son objetos Java, tienden a durar más tiempo en la memoria, y se destruyen hasta que se concluye el proceso de la máquina virtual de Java completo.

Por ello es que es buena práctica que si ya no vamos a utilizar un objeto eliminemos cualquier referencia de cualquier variable que apunte a él asignando el valor de null a dicha variable.

Podemos observar el código de la lámina, en la cual estamos creando un objeto de tipo Persona, y la referencia la asignamos inicialmente a la variable p1.

En la línea 5 mandamos a imprimir el nombre del objeto Persona cuya referencia está almacenada en la variable p1. Y podemos observar que el valor es Juan.

Posteriormente en la línea 7creamos la variable p2, esta variable le asignamos el valor de p1, es decir, ahora p2 también apunta al mismo objeto creado en la línea 4. Esto se comprueba en la línea 8 donde la variable p2 imprime el mismo nombre de la persona asignado en p1.

Posteriormente utilizamos la palabra null para indicar que la variable p1 NO apunta a ningún objeto, y de hecho si quisiéramos ejecutar el método de p1.obtenerNombre() nos arrojaría un error ya que esta variable ya no puede acceder ni a los métodos ni atributos del objeto Persona. El tema de manejo de excepciones lo estudiaremos a detalle en cursos posteriores, sin embargo es importante entender el uso de null y la razón por la cual nos marca este error conocido como NullPointerExcepcion, el cual será uno de los más comunes cuando trabajamos con Java, y básicamente significa que estamos tratando de acceder a un método o atributo de una clase en la cual la variable aún no se le ha asignado una referencia de un objeto válida y cuyo valor es null.

Finalmente en la línea 14 comprobamos que la variable p2 continua accediendo sin problemas al objeto Persona y puede seguir imprimiendo el nombre asignado a este objeto.