

CURSO DE FUNDAMENTOS DE JAVA

PALABRA null EN JAVA



Por el experto: Ing. Ubaldo Acosta



CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

Hola, te saluda Ubaldo Acosta. Bienvenida o bienvenido nuevamente. Espero que estés listo para comenzar con esta lección.

Vamos a estudiar el tema de la palabra null en Java.

¿Estás listo? Ok, ¡Vamos!

PALABRA null EN JAVA

Variables Locales

```
Persona p1 = new Persona();
Persona p2 = p1;
p1 = null;
```

Variables tipo Object



STACK

CURSO DE FUNDAMENTOS DE JAVA

HEAP

www.globalmentoring.com.mx

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 le 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.

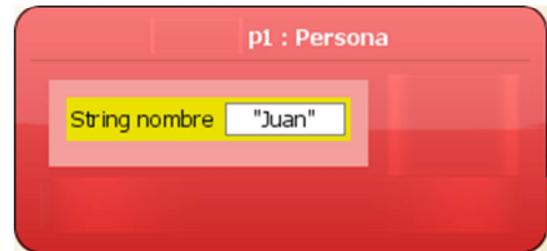
Por ello es que es buena práctica que si ya no vamos a utilizar un objeto eliminemos cualquier referencia de cualquier variable que apunte a él asignando el valor de `null` a dicha variable.

EJEMPLO PALABRA null EN JAVA

Uso palabra null:

```

1  public class PalabraNull {
2
3      public static void main(String[] args) {
4          Persona p1 = new Persona("Juan");
5          System.out.println("Nombre p1:" + p1.obtenerNombre());
6
7          Persona p2 = p1; //p2 apunta al mismo objeto que p1
8          System.out.println("Nombre p2:" + p2.obtenerNombre());
9
10         //Hacemos que p1 ya no apunte al objeto p1
11         p1 = null;
12
13         //Comprobamos que p2 sigue accediendo al objeto
14         System.out.println("Nombre p2:" + p2.obtenerNombre());
15     }
16 }
17
18 class Persona{
19     String nombre; //valor por default es null
20
21     public Persona(String nombre){
22         this.nombre = nombre;
23     }
24
25     public String obtenerNombre(){
26         return this.nombre; //Usa opcional de this
27     }
28 }
```



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 7 creamos 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 quisieramos 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 NullPointerException, 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 válido 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.

EJERCICIOS CURSO FUNDAMENTOS DE JAVA

- **ABRIR LOS ARCHIVOS DE EJERCICIOS EN PDF.**
- **EJERCICIO:** Ejercicio Uso de la palabra null en Java.

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

CURSO ONLINE

FUNDAMENTOS DE JAVA

Por: Ing. Ubaldo Acosta



CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

En Global Mentoring promovemos la Pasión por la Tecnología Java. Te invitamos a visitar nuestro sitio Web donde encontrarás cursos Java Online desde Niveles Básicos, Intermedios y Avanzados, y así te conviertas en un experto programador Java.

Además agregamos nuevos cursos para que continúes con tu preparación como programador Java profesional. A continuación te presentamos nuestro listado de cursos:

- | | |
|---|--|
| <ul style="list-style-type: none">✓ Lógica de Programación✓ Fundamentos de Java✓ Programación con Java✓ Java con JDBC✓ HTML, CSS y JavaScript✓ Servlets y JSP's✓ Struts Framework | <ul style="list-style-type: none">✓ Hibernate Framework✓ Spring Framework✓ JavaServer Faces✓ Java EE (EJB, JPA y Web Services)✓ JBoss Administration✓ Android con Java✓ HTML5 y CSS3 |
|---|--|

Datos de Contacto:

Sitio Web: www.globalmentoring.com.mx

Email: informes@globalmentoring.com.mx

