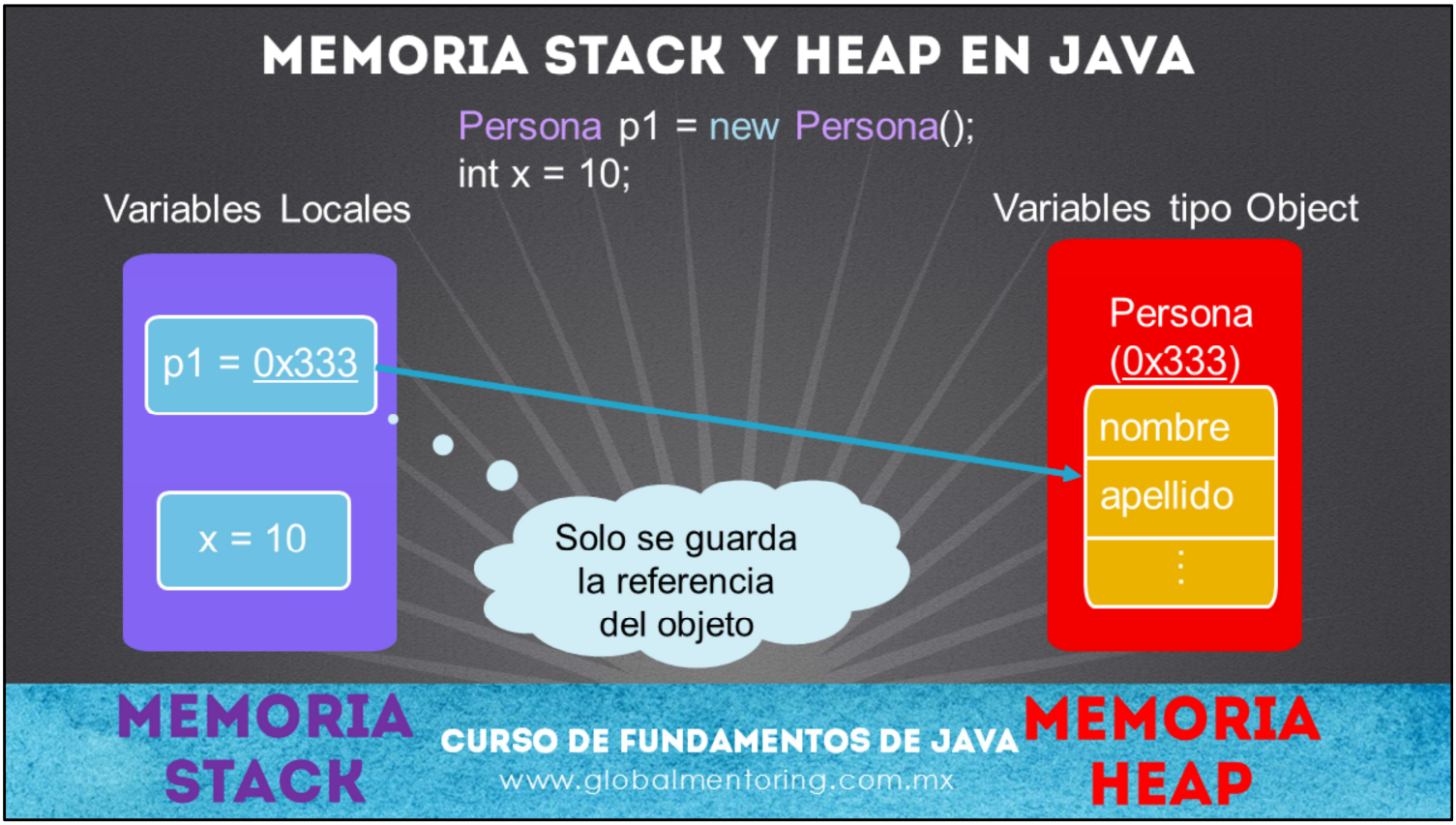
Hola, te saluda nuevamente Ubaldo Acosta. Espero que estés listo para comenzar con esta lección.

Vamos a estudiar el tema de métodos en Java.

¿Estás listo? ¡Vamos!



Como en muchos lenguajes de programación, la memoria RAM (Random Access Memory) se utiliza para almacenar la información de nuestro programa mientras éste se ejecuta. Java ejecuta un proceso, y a su vez internamente existen dos clasificaciones para almacenar los valores de nuestros programas, conocidos como memoria Stack y Memoria Heap.

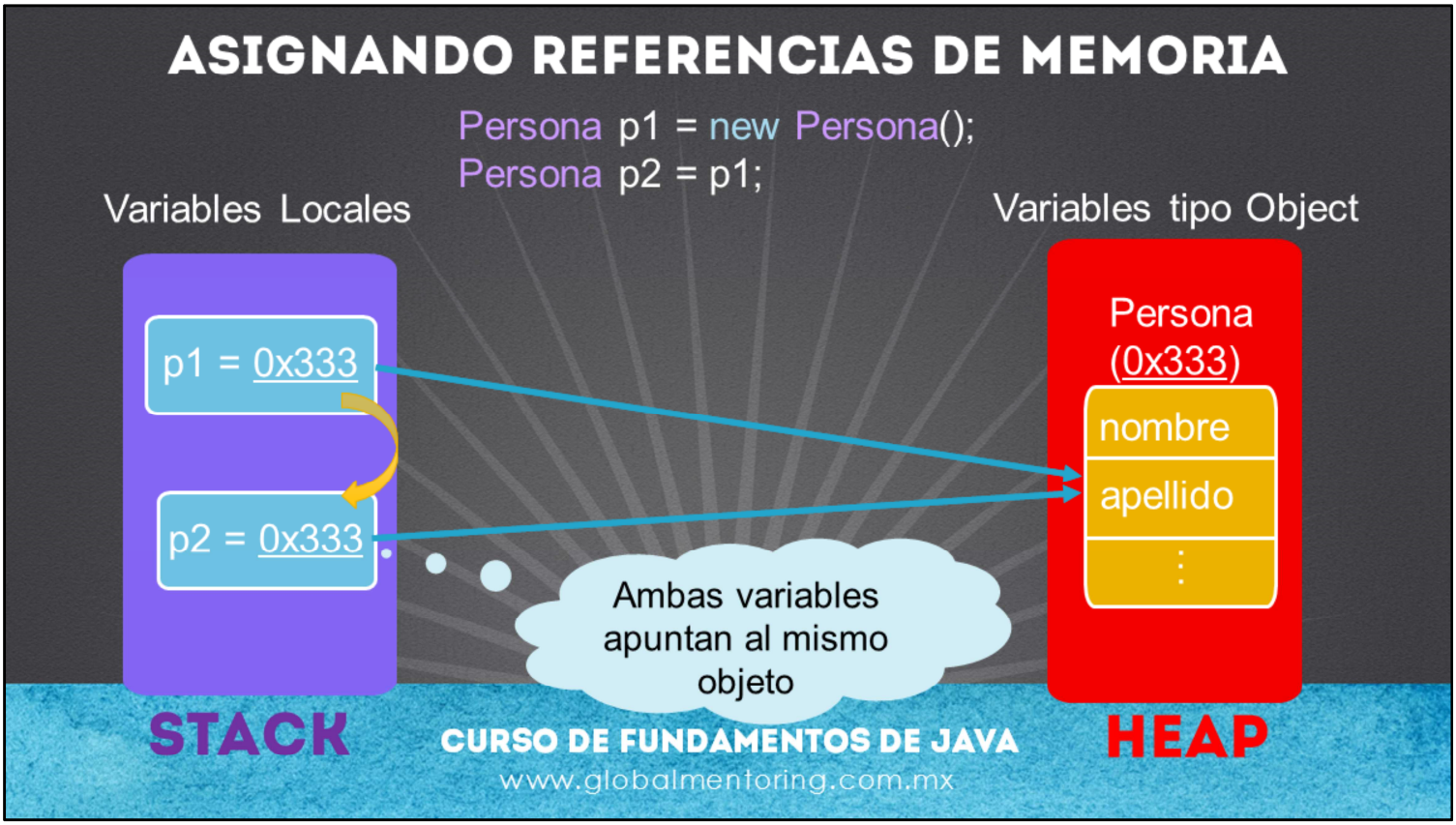
La memoria Stack se utiliza para almacenar las variables locales y las llamadas de funciones en Java. Las variables almacenadas en este espacio de memoria normalmente tienen un periodo de vida corto, únicamente mientras termina la función o método en el que se están ejecutando.

La memoria Heap se utiliza para almacenar los objetos Java, incluyendo sus atributos. Los objetos almacenados en este espacio de memoria normalmente tienen un tiempo de duración más prolongado.

Esto quiere decir que las varibles que creamos, como podemos observar en la figura, no almacenan el objeto en sí mismo, sino solo guarda la referencia del objeto. En la figura podemos observar que la referencia del objeto se representa por un valor hexadecimal (0x333), el cual contiene la dirección de memoria donde está almacenado el objeto, y por lo tanto la variable local p1 almacena únicamente esta referencia de memoria.

Esto es muy importante para temas como la recolección de objetos en Java, debido a que el recolector de basura únicamente podrá eliminar los objetos que no estén siendo apuntados por ninguna variable. De esta manera el recolector de basura (garbage collector) puede buscar aquellos objetos en la memoria heap que ya no estén siendo referenciados por ninguna otra variable y finalmente liberar el espacio en memoria que ocupaba dicho objeto.

Con esto podemos agregar otra importante distinción entre clase y objeto. Una clase define un tipo para ser posteriormente instanciado, es decir, con el objetivo de crear objetos. Esto quiere decir que un objeto ocupa un espacio real en memoria, y son los objetos los que utilizaremos en la mayoría de nuestros programas para enviar y recibir mensajes mediante el uso de sus métodos.



A diferencia de lo que pudiéramos pensar, en Java una variable de tipo Object únicamente almacena una referencia a los objetos que son creados, por lo que le segunda línea del código mostrado, únicamente lo que hace es pasar la referencia que está almacenando la variable p1. Y por lo tanto ahora son dos variables las que están apuntando al mismo objeto, es decir, que ambas variables podrán acceder a lo que el objeto permita según la clase definida, por ejemplo sus atributos o métodos que haya definido.

Es importante observar que no se está creando un segundo objeto, ya que solamente hay una llamada de la palabra new, y por lo tanto solo hay un objeto creado. Lo que almacena la variable p2 es sólo el valor de la referencia en memoria del objeto Persona creado. Cada que utilizamos la palabra new recibiremos una nueva referencia de memoria, es importante notar esto, ya que hasta ese momento es que se creará un nuevo objeto y no antes.

Si ejecutáramos la siguiente línea:

p1 = null;

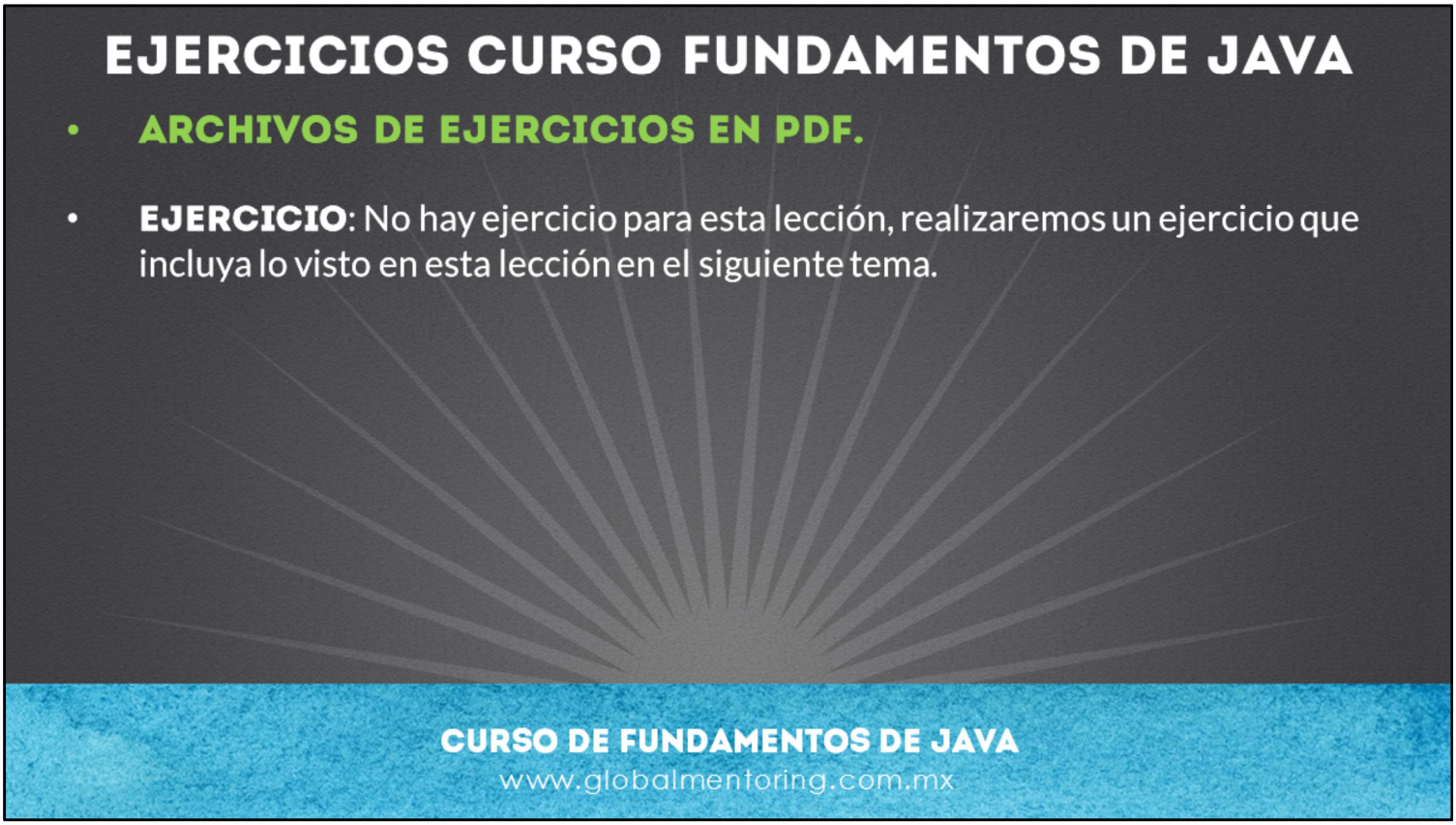
Lo que sucedería es que la variable p1 ya no tendría la referencia del objeto Persona creado ubicado en la referencia 0x333, y por lo tanto sólo la variable p2 es la que podría seguir accediendo a este objeto creado.

Si ejecutáramos finalmente:

p2 = null;

Lo que sucedería es que p2 ya no tendría más la referencia del objeto y por lo tanto ninguna variable tendría la referencia del objeto Persona creado, y por lo tanto el recolector de basura de Java podría disponer de este objeto y eliminarlo de la memoria. Un objeto es candidato a eliminarse de la memoria hasta que ninguna variable este apuntando al objeto, es decir, que ninguna variable contenga su referencia.

El proceso de recolector de basura es un proceso que demanda muchos recursos, por lo que llamara a System.gc() no garantiza que se ejecute el recolector de basura (garbage collector). Sin embargo con la llamada a esta función le estamos indicando a Java que en cuanto sea posible ejecute el recolector y por tanto elimine los objetos de memoria que ya no están siendo utilizados, es decir, que ya no existe ninguna variable apuntando a estos objetos.


EJERCICIOS CURSO FUNDAMENTOS DE JAVA
• ARCHIVOS DE EJERCICIOS EN PDF.
• EJERCICIO: No hay ejercicio para esta lección, realizaremos un ejercicio que incluya lo visto en esta lección en el siguiente tema.
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:

- Fundamentos de Java
- Fundamentos de Java
- Programación con Java
- Java con JDBC
- HTML, CSS y JavaScript
- Servlets y JSP's
- Struts Framework
- 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