

The logo for Oracle Academy. The word "ORACLE" is in a bold, orange, sans-serif font. Below it, the word "Academy" is in a smaller, dark gray, sans-serif font. The entire logo is centered on a light gray background, which is framed by dark gray horizontal bars at the top and bottom.

ORACLE

Academy

Java Foundations

7-2

Instanciación de Objetos

ORACLE
Academy



Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

Objetivos

- En esta lección se abordan los siguientes objetivos:
 - Comprender las consecuencias de instanciar objetos en la memoria
 - Comprender las referencias de objetos
 - Comprender la diferencia entre la memoria de pila y de montón
 - Comprender por qué los objetos String son especiales





Gracias por desarrollar software para mi banco. Sería un honor estrechar su mano.

Gracias a usted, nuestros clientes abren más cuentas que nunca.



Y los niños son mucho más felices.



Por la noche...



ORACLE
Academy

JFo 7-2
Instanciación de Objetos

Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

4



¡Ja! ¡Ja! ¡Ja! ¡Robar es divertido!

ORACLE
Academy

JFo 7-2
Instanciación de Objetos

Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

5

¡zas!

¡Deteneos,
ladrones!



Gracias, desconocido. ¿Desea
desarrollar software para
nuestra prisión?

No sé
programar...
Pero conozco
a alguien que
sí sabe.

ORACLE
Academy

JFo 7-2
Instanciación de Objetos

Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

6

Descripción de un preso


- Propiedades:
 - Nombre
 - Altura
 - Años de sentencia




- Comportamientos:
 - Piense en lo que han hecho

Ejercicio 1, parte 1

- Cree un nuevo proyecto Java
- Cree una clase `PrisonTest` con un método `main`
- Cree una clase `Prisoner` basada en la descripción de la diapositiva anterior
- Instancie dos presos y asígneles las propiedades siguientes:



Variable: bubba
Name: Bubba
Height: 6'10"
(2.08m)
Sentence: 4 years



Variable: twitch
Name: Twitch
Height: 5'8"
(1.73m)
Sentence: 3 years

ORACLE
Academy

JFo 7-2
Instanciación de Objetos

Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

8

Puede ser más sencillo programar la altura en metros.

Ejercicio 1, parte 2

- ¿Los presos pueden engañar al sistema de seguridad si se suplantán entre sí?
 - Escriba una sentencia print con una expresión booleana que pruebe si bubba == twitch
 - Cambie las propiedades de twitch para que coincidan con las de bubba
 - A continuación, vuelva a probar la igualdad de estos objetos



Variable: bubba
Name: Bubba
Height: 6'10"
(2.08m)
Sentence: 4 years



Variable: twitch
Name: Bubba
Height: 6'10"
(2.08m)
Sentence: 4 years

ORACLE
Academy

JFo 7-2
Instanciación de Objetos

Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

9

Programación de la clase Prisoner

- La clase podría ser como esta:

```
public class Prisoner {  
    public String name;  
    public double height;  
    public int sentence;  
  
    public void think(){  
        System.out.println("I'll have my revenge.");  
    }//end method think  
}//end class Prisoner
```

Suplantación de preso

- La expresión booleana `bubba == twitch` es `false`
 - Los presos que comparten las mismas propiedades no han engañado al sistema de seguridad
 - El sistema de seguridad ha sabido que cada preso era un objeto único
- ¿Cómo es posible?

```
public class PrisonTest {  
    public static void main(String[] args){  
        Prisoner bubba = new Prisoner();  
        Prisoner twitch = new Prisoner();  
  
        ...  
        System.out.println(bubba == twitch); //false  
    } //end method main  
} //end class PrisonTest
```

Ubicaciones de los presos

- Los presos viven en celdas
- A los presos nuevos se les asigna una celda para que vivan en ella
- Si un preso vive en una sola celda, es un objeto único



ORACLE
Academy

JFo 7-2
Instanciación de Objetos

Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

12

Ubicaciones de los objetos Prisoner

- Las celdas son como ubicaciones de la memoria
- Al instanciar un objeto Prisoner se rellena una ubicación disponible en la memoria con el nuevo objeto Prisoner

```
public class PrisonTest {  
    public static void main(String[] args){  
        Prisoner bubba = new Prisoner();  
        Prisoner twitch = new Prisoner();  
    } //end method main  
} //end class PrisonTest
```

bubba



Celda A1

A1	B1	C1	D1	E1	F1
A2	B2	C2	D2	E2	F2

twitch



Celda F2

Palabra clave new

- La palabra clave new asigna la memoria disponible para almacenar un objeto recién creado
- Los desarrolladores de Java no necesitan conocer la ubicación de un objeto en la memoria
 - Solo necesitamos conocer la variable del objeto
 - Pero podemos imprimir las direcciones de memoria

```
public class PrisonTest {  
    public static void main(String[] args){  
        Prisoner bubba = new Prisoner();  
        Prisoner twitch = new Prisoner();  
        System.out.println(bubba);    //prisontest.Prisoner@15db9742  
        System.out.println(twitch);    //prisontest.Prisoner@6d06d69c  
    }//end method main  
}//end class PrisonTest
```

Direcciones de memoria

Objetos con las mismas propiedades

- Los objetos pueden compartir las mismas propiedades
- Pero eso no significa que estos objetos sean iguales
- Siempre que use la palabra clave new durante la instanciación...
 - Tendrá objetos únicos
 - Cada objeto tendrá una ubicación diferente en la memoria



Variable: bubba
Name: Bubba
Height: 6'10"
(2.08m)
Sentence: 4 years



Variable: twitch
Name: Bubba
Height: 6'10"
(2.08m)
Sentence: 4 years

Comparación de objetos

- Si compara dos objetos con el operador ==...
 - Comprueba si sus direcciones de memoria son iguales
 - No comprueba si sus campos son iguales
- La expresión booleana `bubba == twitch` es `false` porque...
 - Las direcciones de memoria `@15db9742` y `@6d06d69c` son distintas
 - No importa si `bubba` y `twitch` comparten las mismas propiedades

```
public class PrisonTest {  
    public static void main(String[] args){  
        Prisoner bubba = new Prisoner();  
        Prisoner twitch = new Prisoner();  
  
        ...  
        System.out.println(bubba == twitch); //false  
    } //end method main  
} //end class PrisonTest
```

Acceso a objetos mediante una referencia



A los objetos se accede utilizando variables de referencia. Una analogía adecuada es usar un control remoto (la referencia) para utilizar una cámara (el objeto). Los botones del control remoto se pueden usar para disparar un determinado comportamiento de cámara. Por ejemplo, puede utilizar el control remoto para llamar a las funciones de parada, reproducción o grabación de la cámara.

Trabajar con referencias de objetos

1

Coger el control remoto para obtener acceso a la cámara



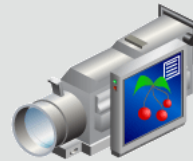
2

Pulsar los controles remotos para que la cámara haga algo

1

Crear un objeto Camera y obtener una referencia a él

```
Camera remotel = new Camera();
```



2

Llamar a un método para que el objeto Camera haga algo

```
remotel.play();
```

ORACLE
Academy

JFo 7-2
Instanciación de Objetos

Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

18

Examinemos la analogía del uso de un control remoto para manejar un dispositivo electrónico.

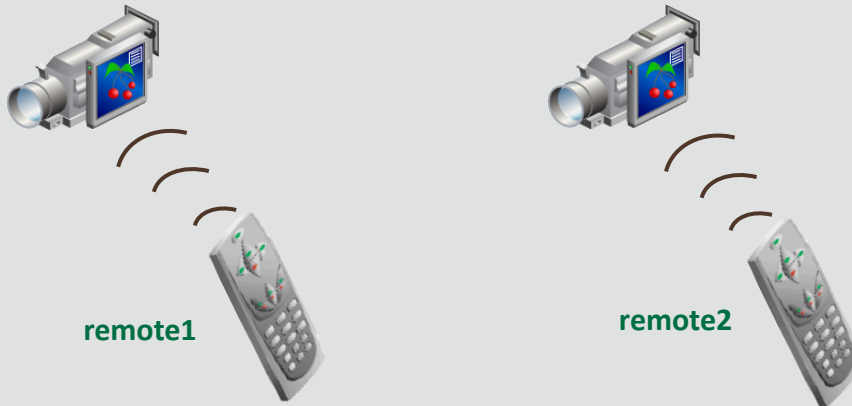
Para manejar un dispositivo electrónico con un control remoto, necesita:

1. Coger el control remoto (y posiblemente encenderlo).
2. Pulsar un botón del control remoto para hacer algo en la cámara.

Igualmente, para hacer algo con un objeto Java, necesita:

1. Obtener el “control remoto” (denominado referencia).
2. Pulsar los “botones” (denominados métodos).

Trabajar con referencias de objetos: Ejemplo 1

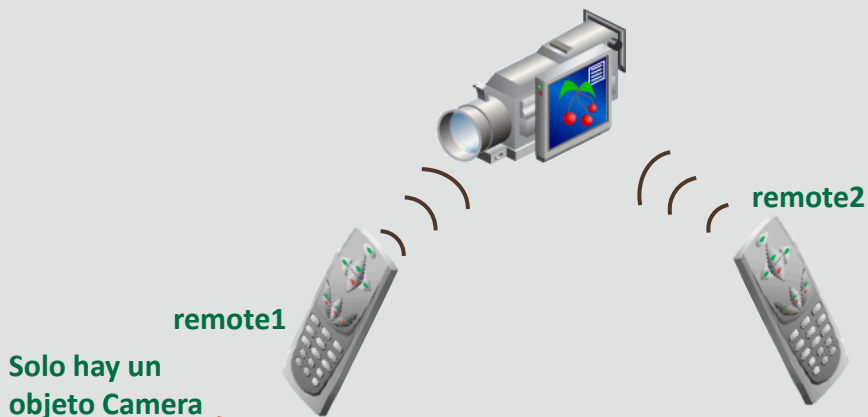


```
Camera remote1 = new Camera();  
Camera remote2 = new Camera();  
  
remote1.play();  
remote2.play();
```

Hay dos objetos
Camera

En este ejemplo hay dos objetos de cámara. Cada cámara tiene su propio control remoto único. `remote2` no funcionará en la cámara de `remote1`, y `remote1` no funcionará en la cámara de `remote2`. Esto refleja cómo, en Java, se pueden instanciar dos objetos distintos con sus referencias exclusivas. Estas referencias se pueden utilizar para llamar a los métodos en sus respectivos objetos.

Trabajar con referencias de objetos: Ejemplo 2



```
Camera remote1 = new Camera();  
  
Camera remote2 = remote1;  
  
remote1.play();  
remote2.stop();
```

ORACLE
Academy

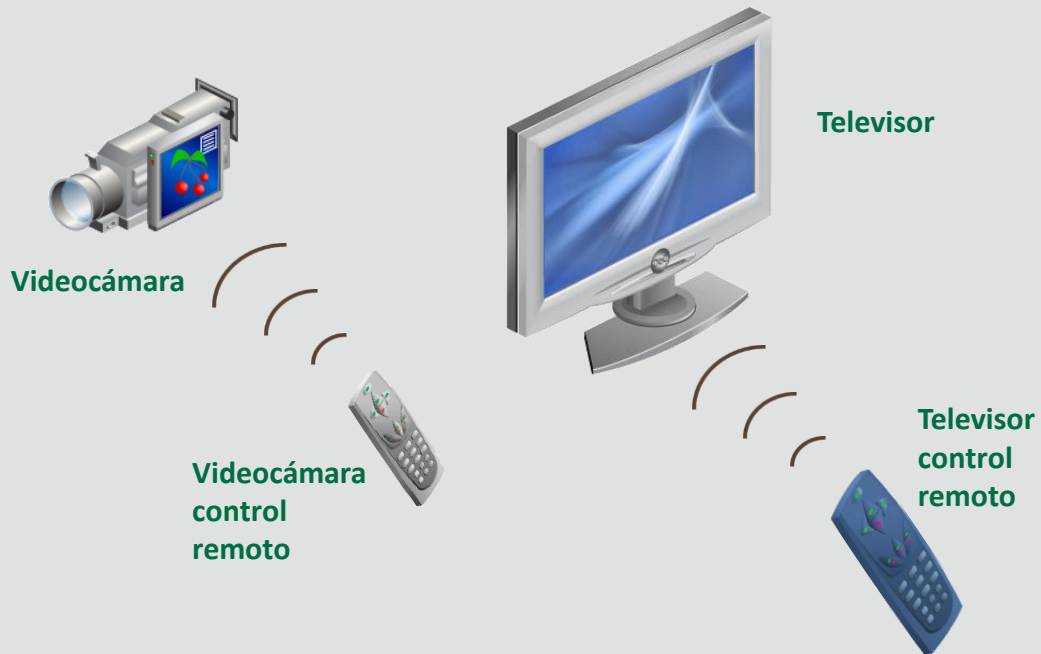
JFo 7-2
Instanciación de Objetos

Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

20

En el diagrama se muestra otro aspecto importante del funcionamiento de las referencias. En este ejemplo se crea un objeto `Camera` con la referencia `remote1`. A continuación, esta referencia se asigna a otra referencia de `Camera`, `remote2`. Las referencias `remote1` a `remote2` se asocian al mismo objeto `Camera`. Llamar a los métodos con cualquier referencia afecta al mismo objeto `Camera`. Llamar a `remote1.play()` es lo mismo que llamar a `remote2.play()`.

Referencias a diferentes objetos



Trabajar con distintos tipos de objetos (por ejemplo, una cámara y una televisión) requiere un control remoto específico de ese tipo de objeto. En Java, necesita una variable de referencia del tipo correcto para el objeto al que hace referencia.

Referencias a diferentes objetos: Ejemplo

Tipo de referencia Variable de referencia Tipo de objeto

```
Camera remote1 = new Camera();  
remote1.menu();
```

```
TV remote2 = new TV();  
remote2.menu();
```

```
Prisoner bubba = new Prisoner();  
bubba.think();
```

Un preso no puede suplantar a un televisor para engañar al sistema de seguridad.

Referencias a diferentes objetos: Ejemplo

- El siguiente ejemplo no se permite porque...
 - El tipo de referencia no coincide con el tipo de objeto
 - Un preso y un televisor son cosas completamente distintas



```
Prisoner twitch = new TV();
```

Un preso no puede suplantar a un televisor para engañar al sistema de seguridad.

Ejercicio 2

- Continúe experimentando con la clase `PrisonTest`
- ¿Se engaña al sistema de seguridad cuando cambian las variables de referencia?
 - Instancie dos presos y asígneles las propiedades siguientes
 - Pruebe la igualdad de estos objetos
 - A continuación, defina la variable de referencia para bubba igual a twitch
 - Vuelva a probar la igualdad de estos objetos



Variable: bubba
Name: Bubba
Height: 6'10"
(2,08m)
Sentence: 4 years



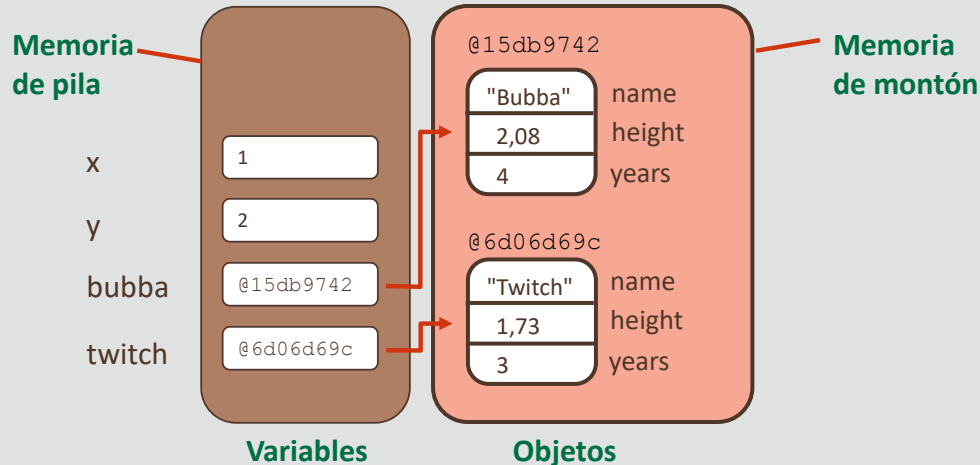
Variable: twitch
Name: Twitch
Height: 5'8"
(1,73m)
Sentence: 3 years

Memoria de pila y de montón

- Comprender los resultados del ejercicio 2 requiere conocer los tipos de memoria que utiliza Java
- La memoria de pila se utiliza para almacenar...
 - Variables locales
 - Primitivos
 - Referencias a ubicaciones en la memoria de montón
- La memoria de montón se utiliza para almacenar...
 - Objetos

Referencias y objetos en memoria

```
int x = 1;
int y = 2;
Prisoner bubba = new Prisoner();
Prisoner twitch = new Prisoner();
...
```



ORACLE
Academy

JFo 7-2
Instanciación de Objetos

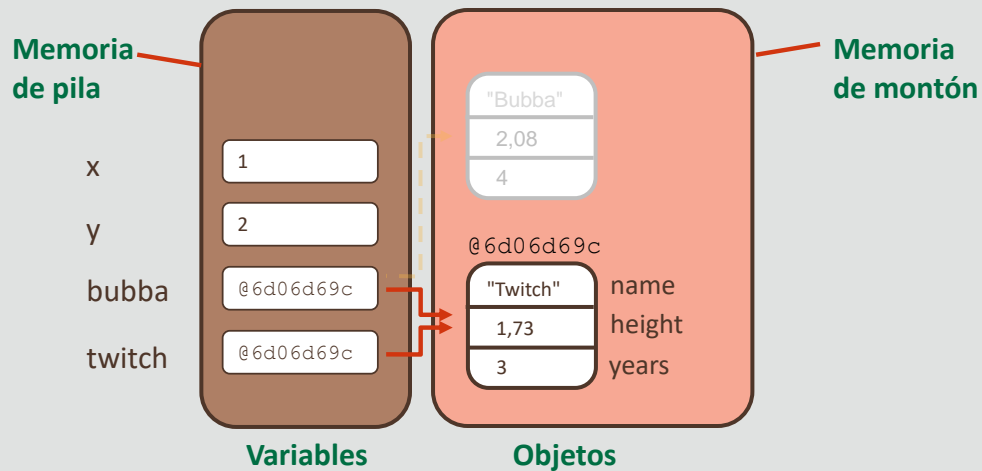
Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

26

En este diagrama se muestra cómo las variables de referencia apuntan a un objeto concreto en memoria. Hay dos referencias de objeto `Prisoner` que apuntan a dos objetos `Prisoner`. La memoria de pila contiene variables locales, primitivas o de referencia, y la de montón contiene objetos.

Asignación de una referencia a otra

```
bubba = twitch;
```



ORACLE
Academy

JFo 7-2
Instanciación de Objetos

Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

27

Las variables de referencia `bubba` y `twitch` ahora apuntan al mismo objeto.

Dos referencias, un objeto

- A partir de la línea 14, **bubba** y **twitch** hacen referencia al mismo objeto
- Cualquier variable de referencia se podría utilizar para acceder a los mismos datos

```
11 Prisoner bubba = new Prisoner();
12 Prisoner twitch = new Prisoner();
13
14 bubba = twitch;
15
16 bubba.name = "Bubba";
17 twitch.name = "Twitch";
19
20 System.out.println(bubba.name); //Twitch
21 System.out.println(bubba == twitch); //true
```

Al imprimir `bubba.name` provoca que se imprima "Twitch" porque `bubba.name` y `twitch.name` hacen referencia al mismo campo del mismo objeto.

Dos referencias, dos primitivas

- Las primitivas siempre son variables independientes
- Los valores primitivos siempre ocupan ubicaciones distintas en la memoria de pila
- La línea 14 hace brevemente que los valores primitivos x e y sean iguales

```
11 int x;  
12 int y;  
13  
14 x = y;  
15  
16 x = 1;  
17 y = 2;  
18  
19  
20 System.out.println(x);           //1  
21 System.out.println(x == y);     //false
```


¿Qué ha sucedido con Bubba?

- Si no hay más variables de referencia que apunten a un objeto...
- Java borra automáticamente la memoria que había ocupado ese objeto
 - Este proceso se denomina recolección de basura
 - Los datos asociados a este objeto se pierden para siempre



Variable: **bubba**
Name: Bubba
Height: 6'10"
(2,08m)
Sentence: 4 years

Memory Address:



Variable: **twitch,
bubba**
Name: Twitch
Height: 5'8"
(1,73m)
Sentence: 3 tahun

Memory Address: @6d06d69c

ORACLE
Academy

JFo 7-2
Instanciación de Objetos

Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

30

En lenguajes como C++, la memoria se tiene que borrar manualmente.

Los objetos String son especiales

- Al imprimir una referencia de String se imprime el objeto String real en vez de la dirección de memoria del objeto
- Los objetos String se pueden instanciar con la palabra clave new
 - Pero no debe hacerlo

```
String s1 = new String("Test");
```

- Los objetos String se deben instanciar sin new
 - De este modo, la memoria es más eficaz
 - Explicaremos el motivo en las diapositivas siguientes

```
String s2 = "Test";
```

Ejercicio 3

- Continúe experimentando con la clase `PrisonTest`
- Compruebe las consecuencias de los objetos `String` en la memoria por su cuenta
 - Instancie dos presos con los nombres que se muestran a continuación
 - Defina sus nombres con la palabra clave `new` y pruebe la igualdad de estos objetos `String` con `==`
 - Defina sus nombres sin usar la palabra clave `new` y pruebe la igualdad de estos objetos `String` con `==`



Variable: bubba
Name: Bubba
Height: 6'10"
(2.08m)
Sentence: 4 years



Variable: twitch
Name: Bubba
Height: 6'10"
(2.08m)
Sentence: 4 years

ORACLE
Academy

JFo 7-2
Instanciación de Objetos

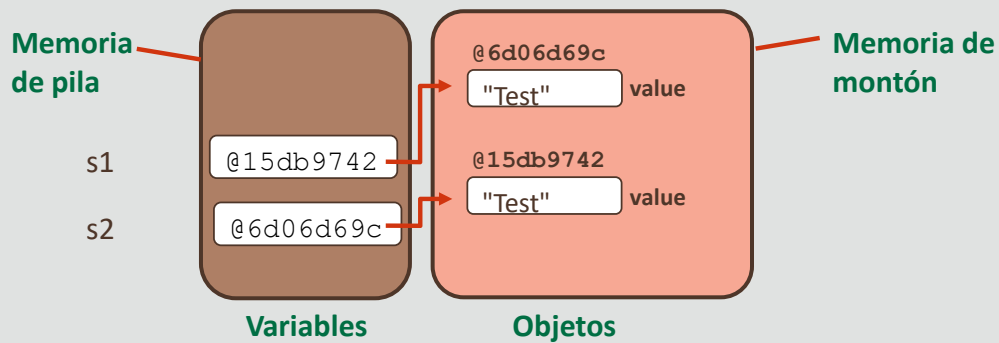
Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

32

Instanciación de objetos String con la palabra clave new

- Con la palabra clave new se crean dos referencias distintas a dos objetos diferentes

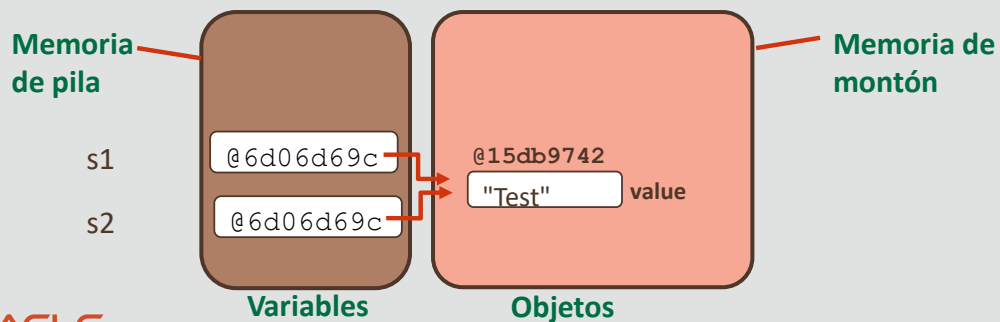
```
String s1 = new String("Test");  
String s2 = new String("Test");
```



Instanciación de objetos String sin la palabra clave new

- Java reconoce automáticamente los objetos String y reserva memoria para almacenar el objeto solo una vez
- De este modo se crean referencias distintas a un solo objeto

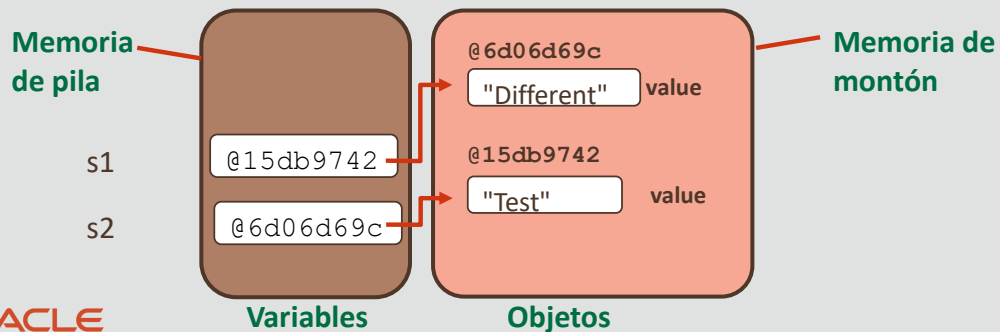
```
String s1 = "Test";  
String s2 = "Test";
```



Referencias al objeto String

- La modificación de un objeto String utilizando una referencia no afectará a las demás referencias
- Java asigna memoria nueva a distintos objetos String

```
String s1 = "Test";  
String s2 = "Test";  
s1 = "Different";
```



Resumen

- En esta lección, debe haber aprendido lo siguiente:
 - Comprender las consecuencias de instanciar objetos en la memoria
 - Comprender las referencias de objetos
 - Comprender la diferencia entre la memoria de pila y de montón
 - Comprender por qué los objetos String son especiales



