



APRENDERPROGRAMAR.COM

COPIAR Y COMPARAR
ARRAYS EN JAVA.
CONCEPTOS DE IDENTIDAD
E IGUALDAD. EJERCICIOS
EJEMPLOS RESUELTOS.
(CU00675B)

Sección: Cursos

Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

Resumen: Entrega nº75 curso Aprender programación Java desde cero.

Autor: Alex Rodríguez

COPIAR ARRAYS Y COMPARAR ARRAYS. IDENTIDAD E IGUALDAD

Al trabajar con arrays de tipos primitivos o de objetos se nos puede plantear la necesidad de copiar arrays. La copia de arrays está permitida pero conviene ser cauto cuando realicemos procesos de este tipo. Recordar que un array es un objeto (aunque sea especial) y por tanto la variable que lo nombra en realidad contiene un puntero al objeto, no el objeto en sí mismo.



Al hacer una operación del tipo `array1 = array2`, el puntero de `array1` apunta al mismo objeto que `array2` mientras que el objeto al que apuntaba `array1` queda inaccesible. A partir de ese momento existe identidad entre los arrays y la comparación usando `==` nos devolverá `true`. A través de código, vamos a plantearnos distintas situaciones y a comentar cuáles son los resultados.

EJERCICIO EJEMPLO RESUELTO 1. COPIAR ESTABLECIENDO UNA RELACIÓN DE IDENTIDAD ENTRE ARRAYS (aplicable a arrays de tipos primitivos y a arrays de objetos)

```
//Test de copia de arrays aprenderaprogramar.com
public class TestCopiaArrays {

    public static void main (String [ ] Args) {
        int [ ] miArray1 = {2, -4, 3, -7};
        for (int i=0; i<miArray1.length; i++) {
            System.out.print ("miArray1[" + i +"]= " + miArray1[i]+"; ");
        }
        System.out.println("");
        int [ ] otroArray = {1, 2, 4, 8};
        for (int i=0; i<otroArray.length; i++) {
            System.out.print ("otroArray[" + i +"]= " + otroArray[i]+"; ");
        }
        System.out.println("");

        System.out.println ("¿Son el mismo objeto? ... " + (miArray1==otroArray) );
        System.out.println("");

        otroArray = miArray1; //otroArray pasa a ser el mismo objeto que miArray1

        for (int i=0; i<otroArray.length; i++) {
            System.out.print ("otroArray[" + i +"]= " + otroArray[i]+"; ");
        }
        System.out.println("");
        System.out.println ("¿Son el mismo objeto? ... " + (miArray1==otroArray) );
    } //Cierre del main
} //Cierre de la clase
```

Trata de predecir tú mismo el resultado y compáralo con el que ofrecemos. Resultado:

```
miArray1[0]= 2; miArray1[1]= -4; miArray1[2]= 3; miArray1[3]= -7;
otroArray[0]= 1; otroArray[1]= 2; otroArray[2]= 4; otroArray[3]= 8;
¿Son el mismo objeto? ... false
otroArray[0]= 2; otroArray[1]= -4; otroArray[2]= 3; otroArray[3]= -7;
¿Son el mismo objeto? ... true
```

EJERCICIO EJEMPLO RESUELTO 2. INTENTO DE COMPARAR ARRAYS (RELACIÓN DE IGUALDAD)

Supongamos que queremos comparar la igualdad entre dos arrays. Una idea (mala idea) podría ser aplicar el método equals directamente sobre el array. Un código de prueba podría ser este:

```
//Test uso incorrecto de equals aprenderaprogramar.com
public class TestUsoIncorrectoEquals {
    public static void main (String [ ] Args) {
        int [ ] miArray1 = {2, -4, 3, -7};
        for (int i=0; i < miArray1.length; i++) {
            System.out.print ("miArray1[" + i +"]= " + miArray1[i]+");
        }
        System.out.println();
        int [ ] otroArray = {2, -4, 3, -7};
        for (int i=0; i < otroArray.length; i++) {
            System.out.print ("otroArray[" + i +"]= " + otroArray[i]+");
        }
        System.out.println ("¿Son el mismo objeto? ... " + (miArray1==otroArray) );
        System.out.println ("¿Tienen el mismo contenido (relación de igualdad)? ... " + (miArray1.equals(otroArray) ) );
        otroArray = miArray1; //otroArray pasa a ser el mismo objeto que miArray1

        for (int i=0; i<otroArray.length; i++) { System.out.print ("otroArray[" + i +"]= " + otroArray[i]+"); }

        System.out.println ("¿Son el mismo objeto? ... " + (miArray1==otroArray) );
        System.out.println ("¿Tienen el mismo contenido (relación de igualdad)? ... " + (miArray1.equals(otroArray) ) );
    } //Cierre del main
} //Cierre de la clase
```

Trata de predecir tú mismo el resultado y compáralo con el que ofrecemos. Resultado:

```
miArray1[0]= 2; miArray1[1]= -4; miArray1[2]= 3; miArray1[3]= -7;
otroArray[0]= 2; otroArray[1]= -4; otroArray[2]= 3; otroArray[3]= -7; ¿Son el mismo objeto? ... false
¿Tienen el mismo contenido (relación de igualdad)? ... false
otroArray[0]= 2; otroArray[1]= -4; otroArray[2]= 3; otroArray[3]= -7; ¿Son el mismo objeto? ... true
¿Tienen el mismo contenido (relación de igualdad)? ... true
```

Con este ejemplo comprobamos que el método equals aplicado directamente sobre arrays no funciona ya que en el primer caso teniendo el mismo contenido nos dice que la relación de igualdad es falsa.

¿Por qué? Porque ya dijimos en su momento que los arrays son unos objetos especiales que no tienen una clase que los implemente. Si no tienen clase, no tienen métodos (al menos directamente), y el método que se aplica es el usado por defecto por Java con resultados imprevisibles o no deseados. La solución para comparar arrays es usar otra clase que permite realizar operaciones con arrays. Esta clase es la **clase Arrays**. Una vez más la terminología Java nos puede inducir a confusión: hablamos de array como colección de tipos primitivos u objetos, y al mismo tiempo Arrays (con mayúsculas) es el nombre de una clase Java que sirve para manipular arrays.

EJERCICIO

Crea un programa Java donde declares un array de enteros tipo int miArray1 cuyo contenido inicial sea {2, -4, 3, -7}. Muestra su contenido por pantalla. Copia el contenido de este array a un ArrayList denominado lista1 y muestra su contenido por pantalla. ¿Qué tipo de datos almacena el array? ¿Qué tipo de datos almacena el ArrayList?

Puedes comprobar si tu código y respuestas son correctas consultando en los foros aprenderaprogramar.com.

Próxima entrega: CU00676B

Acceso al curso completo en aprenderaprogramar.com --> Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188