



APRENDERAPROGRAMAR.COM

CONVERSIÓN DE TIPOS DE DATOS EN JAVA (MOLDEADO O TIPADO). EJEMPLOS. MÉTODO VALUEOF (CU00670B)

Sección: Cursos

Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

Resumen: Entrega nº70 curso Aprender programación Java desde cero.

Autor: Alex Rodríguez

CONVERSIÓN DE TIPOS DE DATOS EN JAVA

En Java es posible transformar el tipo de una variable u objeto en otro diferente al original con el que fue declarado. Este proceso se denomina "**conversión**", "**moldeado**" o "**tipado**" y es algo que debemos manejar con cuidado pues un mal uso de la conversión de tipos es frecuente que dé lugar a errores.



Una forma de realizar conversiones consiste en colocar el tipo destino entre paréntesis, a la izquierda del valor que queremos convertir de la forma siguiente: Tipo VariableNueva = (NuevoTipo) VariableAntigua;

Por ejemplo: `int miNumero = (int) ObjetoInteger;` `char c = (char)System.in.read();`

En el primer ejemplo, extraemos como tipo primitivo `int` el valor entero contenido en un campo del objeto `Integer`. En el segundo caso, la función `read` devuelve un valor `int`, que se convierte en un `char` debido a la conversión `(char)`, y el valor resultante se almacena en la variable de tipo carácter `c`.

El tamaño de los tipos que queremos convertir es muy importante. No todos los tipos se convertirán de forma segura. Por ejemplo, al convertir un `long` en un `int`, el compilador corta los 32 bits superiores del `long` (de 64 bits), de forma que encajen en los 32 bits del `int`, con lo que si contienen información útil, ésta se perderá. Este tipo de conversiones que suponen pérdida de información se denominan "conversiones no seguras" y en general se tratan de evitar, aunque de forma controlada pueden usarse puntualmente.

De forma general trataremos de atenernos a la norma de que "**en las conversiones debe evitarse la pérdida de información**". En la siguiente tabla vemos conversiones que son seguras por no suponer pérdida de información.

TIPO ORIGEN	TIPO DESTINO
<i>byte</i>	<i>double, float, long, int, char, short</i>
<i>short</i>	<i>double, float, long, int</i>
<i>char</i>	<i>double, float, long, int</i>
<i>int</i>	<i>double, float, long</i>
<i>long</i>	<i>double, float</i>
<i>float</i>	<i>Double</i>

No todos los tipos se pueden convertir de esta manera. Como alternativa, existen otras formas para realizar conversiones.

MÉTODO VALUEOF PARA CONVERSIÓN DE TIPOS

El método `valueOf` es un método sobrecargado aplicable a numerosas clases de Java y que permite realizar conversiones de tipos. Veamos algunos ejemplos de uso.

EXPRESIÓN	INTERPRETACIÓN aprenderaprogramar.com
<code>miInteger = miInteger.valueOf (i)</code>	Con <code>i</code> entero primitivo que se transforma en <code>Integer</code>
<code>miInteger = miInteger.valueOf (miString)</code>	El valor del <code>String</code> se transforma en <code>Integer</code>
<code>miString = miString.valueOf (miBooleano)</code>	El booleano se transforma en <code>String</code> "true" o "false"
<code>miString = miString.valueOf (miChar)</code>	El carácter (<code>char</code>) se transforma en <code>String</code>
<code>miString = miString.valueOf (miDouble)</code>	El <code>double</code> se transforma en <code>String</code> . Igualmente aplicable a <code>float</code> , <code>int</code> , <code>long</code> .

No todas las conversiones son posibles. Muchas veces por despiste los programadores escriben instrucciones de conversión incoherentes como `miInteger = (int) miString;`. El resultado en este caso es un error de tipo "Inconvertible types". Un uso típico de `valueOf` es para convertir tipos primitivos en objetos.

EJERCICIO

El API de Java proporciona herramientas para pedir datos al usuario a través de ventanas. Un ejemplo de ello es el uso de la clase `JOptionPane` perteneciente al paquete `javax.swing.JOptionPane` del API Java. El método `showInputDialog` permite pedir un dato al usuario y almacena su respuesta en un objeto de tipo `String`. Queremos crear un programa que pida al usuario un número y muestre por pantalla el doble de ese número. Para ello hemos creado este código:

```
import javax.swing.JOptionPane;
public class ejemplo1 {
    public static void main (String[] Args) {
        String entradaUsuario = JOptionPane.showInputDialog ( "Introduzca un número:" );
        System.out.println ("El doble del número introducido es: " + 2*entradaUsuario);
    }
}
```

El problema que tenemos es que nos salta un mensaje de error "operator * cannot be applied to int, java.lang.String". ¿Qué interpretación haces de este mensaje de error? ¿Cómo se puede corregir el código para que a través de una conversión de tipos se ejecute el programa? Puedes comprobar si tu código y respuestas son correctas consultando en los foros [aprenderaprogramar.com](http://www.aprenderaprogramar.com).

Próxima entrega: CU00671B

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) --> Cursos, o en la dirección siguiente:

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