APRENDERAPROGRAMAR.COM

IMPRIMIR POR CONSOLA EN JAVA CON SYSTEM.OUT. CONCATENAR CADENAS. NOTACIÓN DE PUNTO. (CU00632B)

Sección: Cursos
Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

Resumen: Entrega nº32 curso Aprender programación Java desde cero.

Autor: Alex Rodríguez

IMPRIMIR POR CONSOLA EN JAVA (SYSTEM.OUT). CONCATENAR CADENAS. NOTACIÓN DE PUNTO.

En Java hay algunos objetos que existen por defecto (en cualquier entorno de desarrollo, llámese Eclipse, NetBeans, BlueJ, etc.). Uno de ellos es el objeto denominado *System.out*. Este objeto dispone de un método llamado *println* que nos permite imprimir algo por pantalla en una ventana de consola.

La sintaxis básica es: *System.out.println ("Mensaje a mostrar");*

Ten en cuenta que la primera S de *System.out* es mayúscula. Si la escribes minúscula obtendrás un error de compilación. Tenlo presente porque cualquier pequeño error de escritura (el simple cambio de una letra) en el nombre de un objeto, variable, método, etc. puede dar lugar a errores. Además, BlueJ en ocasiones te señalará el lugar del error pero en otras ocasiones no lo hará con exactitud y tendrás que buscarlo con paciencia. Ten también presente que los espacios dentro de las comillas cuentan, es decir, el resultado de escribir ("Mensaje a mostrar"); no es el mismo que el de escribir (" Mensaje a mostrar");.

Si queremos incluir variables concatenamos usando el símbolo + de esta manera:

System.out.println ("El precio es de " + precio + " euros");

El símbolo **+ se usa para concatenar cadenas de texto, o variables que representen texto.** ¿Qué ocurre si introducimos en una concatenación un número o una variable que no sea una cadena? Java por defecto realizará la conversión de aquello que hayamos concatenado a texto. Por ejemplo:

System.out.println ("El precio es de " + 42 + " euros");

Se verá en pantalla como: *El precio es de 42 euros*. Fíjate que hemos incluido los espacios necesarios antes y después del número para evitar que por pantalla nos apareciera *El precio es de42euros*.

Si queremos imprimir una línea en blanco escribiremos esto: *System.out.println ();*

El mismo resultado se obtiene escribiendo *System.out.println ("");*

Escribir esto es válido: *System.out.println ("Mi nombre es " + "Juan");* ya que al fin y al cabo estamos concatenando dos cadenas, aunque sea más lógico escribir *("Mi nombre es Juan");*

Fíjate que la sintaxis que estamos usando responde al siguiente esquema:

```
nombreDelObjeto.nombreDelMétodo (parámetro1, parámetro 2, … );
```

En el caso de un objeto Taxi, una invocación de un método podría ser *taxi1.getDistrito()*. Esta invocación devuelve una cadena, por lo que podríamos escribir:

System.out.println ("El distrito del taxi 1 es " + taxi1.getDistrito());

> ***Esta forma de invocar los métodos se denomina "notación de punto" porque se basa en escribir el nombre del objeto seguido de un punto y el nombre del método con los parámetros que procedan. La notación de punto es un aspecto básico de Java y otros lenguajes de programación.***

Otro uso sería el siguiente: taxi1.distrito = "Norte". En este caso no estamos invocando a un método, sino a un atributo. ¿Cómo sabemos si se trata de una invocación a un método o a un atributo? Simplemente hemos de fijarnos en si la invocación termina con unos paréntesis o no. Si no hay paréntesis, se trata de un atributo. Si los hay, se trata de un método.

Una última cuestión a comentar sobre la concatenación de cadenas es su uso en el resultado que puede devolver un método tipo función. Supongamos que precio es una variable de tipo int con valor 42. Si escribimos como línea final de un método *return precio;* estamos devolviendo 42, un entero. Si escribimos return "precio" estamos devolviendo la cadena de texto "precio" y no la variable precio. ¿Qué ocurre si escribimos una sentencia como return "" + precio? **Unas comillas que se abren y cierran son una cadena,** aunque sean una cadena vacía. El operador + después de una cadena, concatena como cadena aquello que venga a continuación, ya que Java realiza por defecto la conversión a cadena. Por tanto en este caso el valor devuelto sería "42" como cadena de texto y no como valor numérico. Fíjate que esto es una forma de convertir un tipo numérico a tipo texto.

EJERCICIO

Considera estás desarrollando un programa Java donde necesitas trabajar con objetos de tipo Medico (que representa a un médico de un hospital). Define una clase Medico considerando los siguientes atributos de clase: nombre (String), apellidos (String), edad (int), casado (boolean), numeroDocumentoIdentidad (String), especialidad (String). Define un constructor asignando unos valores de defecto a los atributos y los métodos para poder establecer y obtener los valores de los atributos. En cada método, incluye una instrucción para que se muestre por consola un mensaje informando del cambio. Por ejemplo si cambia la especialidad del médico, debe aparecer un mensaje que diga: "Ha cambiado la especialidad del médico de nombre … . La nueva especialidad es: …". Compila el código para comprobar que no presenta errores, crea un objeto, usa sus métodos y comprueba que aparezcan correctamente los mensajes por consola. Para comprobar si es correcta tu solución puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU00633B

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