



APRENDERAPROGRAMAR.COM

ESTUDIANDO EL CONCEPTO
DE MÉTODO JAVA.
EXPLORAR EL MÉTODO
SUBSTRING DE LA CLASE
STRING. (CU00649B)

Sección: Cursos

Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

Resumen: Entrega nº49 curso Aprender programación Java desde cero.

Autor: Alex Rodríguez

ESTUDIANDO EL CONCEPTO DE MÉTODO JAVA. EL MÉTODO SUBSTRING DE LA CLASE STRING

Vamos a centrar nuestra atención en la documentación del método substring de la clase String. La primera cuestión en que nos fijamos es que existen dos formas de invocar el método con el mismo nombre pero con distintos parámetros: se trata de un método sobrecargado.



En el resumen del método comprobamos que el método devuelve un tipo String y nos pide como parámetros uno o dos valores enteros.

<u>String</u>	substring (int beginIndex) Returns a new string that is a substring of this string.
<u>String</u>	substring (int beginIndex, int endIndex) Returns a new string that is a substring of this string.

Pero no queda muy claro qué son esos valores enteros que nos pide. Para aclararlo pinchamos sobre el nombre del método y vemos la documentación de detalle. Con la documentación de detalle ya queda claro cómo podemos usar el método. Vamos a centrarnos en la forma de uso empleando dos parámetros. Podemos extraer una fracción de una cadena usando esta sintaxis:

```
fraccionDeString = nombreDelString.substring (carácter Inicial Incluido, carácter Final Excluido)
```

Por ejemplo: `fraccion1 = miCadena.substring (0, 4)`; extrae los cuatro primeros caracteres de miCadena, que podemos nombrar como 0, 1, 2, 3 (el número de caracteres es cuatro). El carácter final indicado en la llamada al método, el número 4, queda excluido del substring.

Otro ejemplo: `return nombre.substring (0, 4) + id.substring (0, 3)`; devuelve una cadena donde los 4 primeros caracteres se han extraído del String denominado *nombre* y los tres siguientes del String denominado *id*. Tanto *nombre* como *id* tienen que ser obligatoriamente tipo String, ya que si alguno de ellos no lo fuera no sería posible aplicarles el método substring.

Otros ejemplos: `"hamburger".substring (4, 8)`; devuelve "urge".

`"smiles".substring (1, 5)` devuelve "mile".

`"coco".substring (2, 2)` devuelve "" (cadena vacía).

`"coco".substring (7, 3)` devuelve un error ya que no existe un carácter número 7 dentro de la cadena.

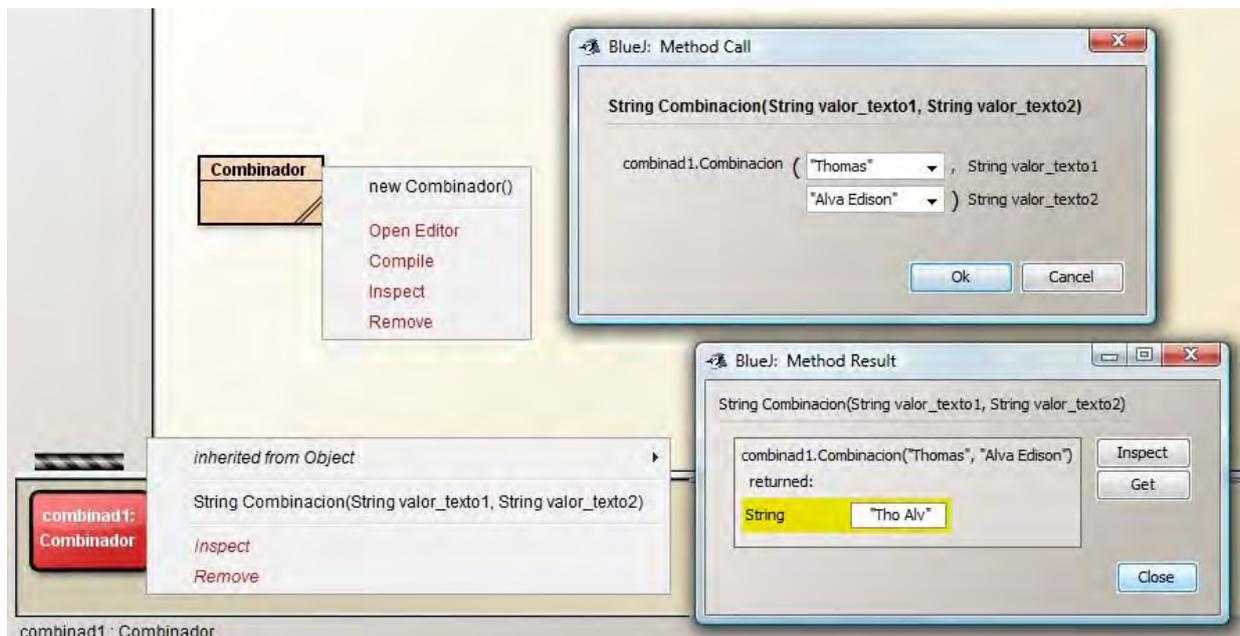
Un fragmento de código usando substring puede ser este:

```
//Combina las tres primeras letras de dos textos recibidos como parámetro en una sola cadena separada por un espacio
public class Combinador {
    //Campos de la clase
    private String texto1;
    private String texto2;

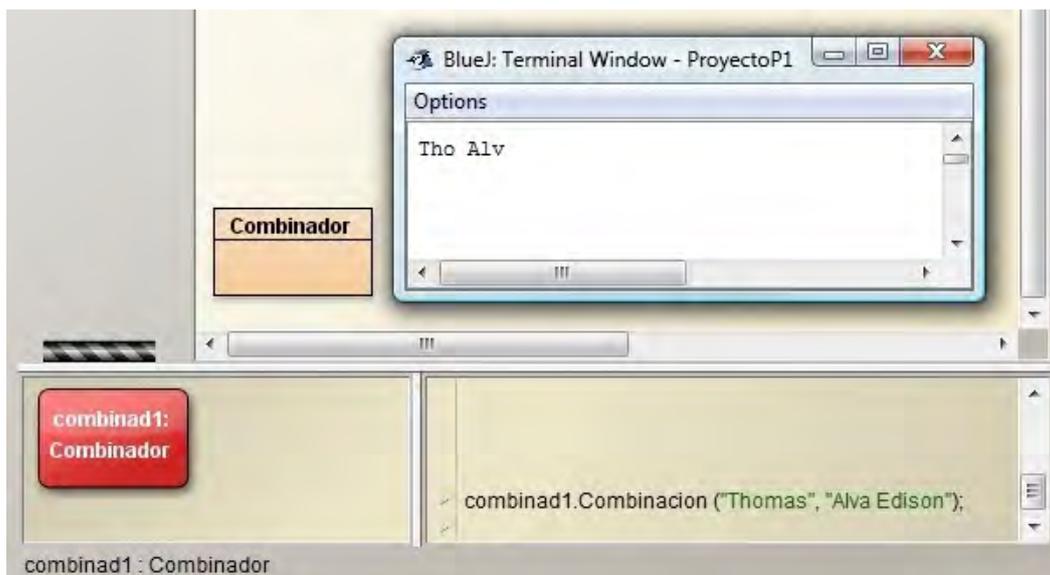
    //Constructor de la clase
    public Combinador () {
        texto1 = "";
        texto2 = "";
    } //Cierre del constructor

    //Método que combina las tres primeras letras de cada cadena
    String combinacion (String valor_texto1, String valor_texto2) {
        String combinacion = valor_texto1.substring (0,3) + " " + valor_texto2.substring (0,3);
        return combinacion; //combinacion es tanto el nombre del método como el de una variable local
    } //Cierre del método
} //Cierre de la clase
```

Crea un objeto de tipo Combinador y ejecuta el método *combinacion* introduciendo dos cadenas como "Thomas" y "Alva Edison" y comprueba sus resultados.



Para ir conociendo BlueJ, vamos a fijarnos ahora en la pequeña ventana que aparece en la parte inferior derecha de la ventana principal (si no está activa, actívala en el menú View → Show Codepad). Esta ventana, denominada Codepad, nos permite escribir código utilizando los objetos que tenemos en el banco de objetos. Si el objeto que has creado se llama combinad1, puedes escribir el siguiente código: *combinad1.combinacion ("Thomas", "Alva Edison");* y pulsa enter. Para comprobar que el método se está ejecutando introduce en el código una instrucción para que se muestre el resultado en la ventana de consola.



Como resumen de lo visto en este apartado podemos decir lo siguiente. String es una clase en Java y substring un método tipo función de la clase que devuelve una cadena de texto. La cadena devuelta es una fracción de la cadena sobre la que se invoca el método. El método substring está sobrecargado porque existe más de una forma de llamarlo. La signatura del método substring podemos consultarla en la documentación del API de Java, lo que nos resulta suficiente para utilizar el método sin necesidad de conocer el código de implementación del mismo (esta información nos queda oculta).

Si hacemos una llamada al método substring pidiendo que se extraiga un número de caracteres superior al número disponible en la cadena se produce un error de tipo "java.lang.StringIndexOutOfBoundsException. String index out of range". Podríamos evitar este tipo de errores si controláramos cuál es la longitud o número de caracteres de las cadenas con las que trabajamos. Para ello podemos recurrir una vez más al API de Java, en concreto al método *length* de los objetos tipo String.

EJERCICIO

Crea una clase (ponle el nombre que quieras) que carezca de atributos y cuyo constructor esté vacío. En dicha clase debe existir un método tipo función que reciba 3 cadenas de texto y devuelva la cadena de texto combinación de: las dos primeras letras de la primera cadena, seguidas por un espacio en blanco, seguidas de las cuatro primeras letras de la segunda cadena, seguidas de un guión medio y seguido de las 6 primeras letras de la tercera cadena. Para comprobar si es correcta tu solución puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU00650B

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