Novedades en el lenguaje Java en las versiones Java 9, 10 y 11

En este documento vamos a describir las novedades que se han producido, en lo que a sintaxis del lenguaje Java se refiere, en las últimas versiones del mismo.

NOVEDADES JAVA 9

La versión Java 9 trajo bastantes novedades relacionadas con la estructuración de aplicaciones (modularidad) y APIs, aunque no muchas en lo que a sintaxis del lenguaje se refiere. Veamos las que afectan a lo que hemos estudiado en el curso:

Eliminación de carácter de subrayado como identificador de variable

El carácter de subrayado no puede utilizarse en solitario en Java 9 como identificador de variable, tal y como si podía hacerse en Java 8. Por ejemplo, la siguiente instrucción no es válida en Java 9:

```
int _=10;
```

El motivo es que se reserva para usos futuros

Si podríamos utilizar el doble subrayado:

```
int __=20;
```

También puede utilizarse en combinación con otros caracteres:

```
int _r=1;
```

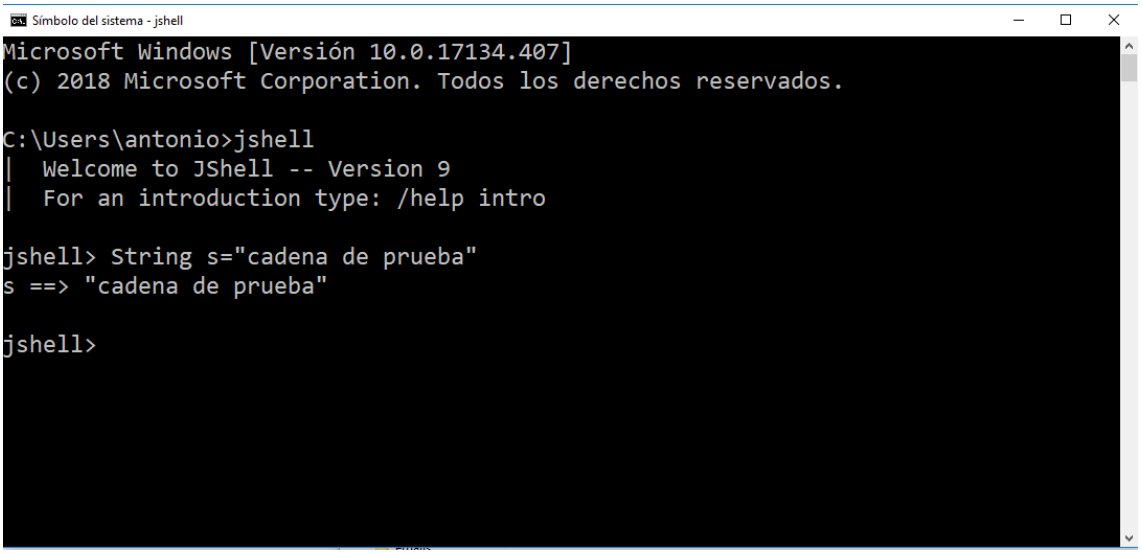
Herramienta Jshell

Java 9 incorpora la herramienta jshell, que es una aplicación basada en línea de comandos, a través de la que podemos probar de forma muy cómoda bloques de código Java, como métodos o expresiones, evitando tener que crear una clase con su método main para realizar estas tareas.

Para entrar en jshell simplemente tecleamos jshell en la línea de comandos:

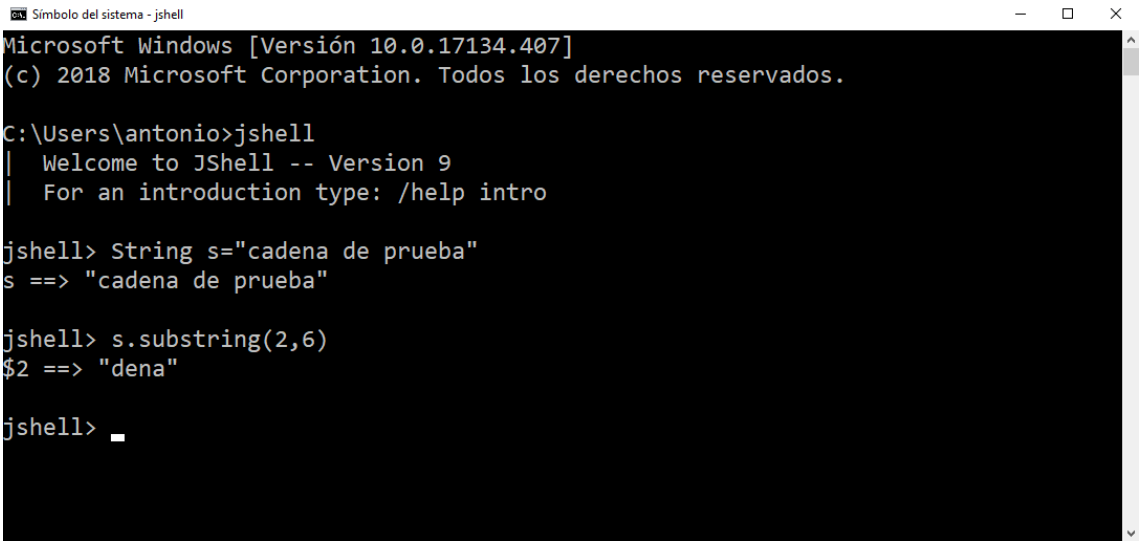
```
>jshell
```

A partir de ahí, nos aparece una consola en la que podemos teclear cualquier código Java que devuelva o produzca un resultado . Por ejemplo, si queremos declarar una variable y asignarle un valor, directamente escribiremos esta operación en la línea de comandos. En el siguiente ejemplo puedes ver como declaramos una variable String y le asignamos un texto:



Como vemos, inemdiatamente después, jshell nos muestra el contenido de dicha variable.

Después podemos aplicar un método sobre dicho objeto y ver el resultado devuelto por el mismo:



Para salir de la herramienta, simplemente escribiremos /exit

NOVEDADES JAVA 10

Inferencia de tipos

Una de las principales novedades de Java 10 es la inferencia de tipos, que consiste en declarar variables locales sin indicar el tipo de la misma, mediante el uso de la palabra reservada *var*:

var a=10;

A partir del valor asignado a la variable, el compilador "infiere" el tipo de la misma, en este caso int. La inferencia de tipos puede aplicarse a cualquier tipo Java, incluidas clases de cualquier tipo:

var datos=new ArrayList<Integer>();

Gracias a la inferencia de tipos, se evita tener que indicar nombres largos a la hora de definir el tipo de una variable cuando se trata de nombres de clases compuestos. Si se recomienda, a fin de claridad en el código, que el nombre de la variable sea lo suficientemente claro como para deducir de qué tipo de dato se trata:

var connection=DriverManager.getConnection(....);

A la hora de aplicar la inferencia de tipos, hay que tener en cuenta las siguientes consideraciones:

- Solo puede aplicarse con variables locales, nunca con atributos ni parámetros de métodos.
- La declaración y asignación de dato a la variable deben hacerse obligatoriamente en la misma instrucción, de manera que el compilador pueda determinar el tipo de la misma.

Al tratarse de una operación en tiempo de compilación, la inferencia de tipo **no empeora ni mejora el rendimiento** de la aplicación en tiempo de ejecución.

NOVEDADES JAVA 11

Compilación y lanzamiento de programas a través de java.exe

Antes de Java 11, para ejecutar un programa Java desde línea de comandos, primero había que compilar el archivo de código fuente con javac.exe y después ejecutar la clase principal con java.exe.

A partir de java 11, si un programa Java está formado por un único archivo de código fuente, ya no es necesario compilar el archivo desde línea de comandos con javac.exe y ejecutarlo después con java.exe, **es posible utilizar el comando java.exe sobre el archivo de código fuente**, lo que provocará la compilación y seguida ejecución del mismo:

>java Test.java //compila primero y luego ejecuta la clase Test

Nuevos métodos en la clase String

Java 11 incorpora nuevos métodos en la clase String. Estos son:

- **boolean isBlank().** Devuelve true si la cadena está vacía o solamente contiene espacios en blanco. No debemos confundir con isEmpty(), que devuelve true solo si la longitud de la cadena es 0**.**
- **Stream lines()**. Devuelve un stream con las líneas que forman la cadena.
- **String repeat(int n)**. Devuelve una cadena resultado de concatenar la cadena actual, tantas veces como se indica en el parámetro.
- **String strip()**. Devuelve una nueva cadena resultante de eliminar espacios a izquierda y derecha. Es prácticamente igual *a trim()*, se diferencian en el concepto de espacio en blanco de cada uno, pues hay determinados código unicode que representan espacios en blanco pero *trim()* no los reconoce como tal, como por ejemplo '\u2000'.

- **String stripLeading()**.Devuelve una nueva cadena resultante de eliminar espacios a izquierda
- **String stripTrailing()**.Devuelve una nueva cadena resultante de eliminar espacios a derecha