

DIFERENCIAS ENTRE STRINGBUILDER, STRINGBUFFER Y STRING

La concatenación de cadena de caracteres, representada por la clase String en Java, es una tarea muy habitual en prácticamente cualquier programa. Sea para construir un mensaje, errores, excepciones o log de la aplicación, la concatenación de String está siempre presente.

En Java hay varias formas de concatenar los String, y su rendimiento difiere muchísimo. Vamos a ver las tres formas más habituales de concatenar String: la "clásica" con el operador de suma, usando StringBuffer y usando StringBuilder.

Los contendientes

El operador suma

El operador de suma está sobrecargado para los String, y es la forma más básica de concatenar dos cadenas de caracteres. Y también, es la peor forma en cuanto a rendimiento.

Al concatenar dos String con el operador suma, se crea un nuevo String resultante, con lo que constantemente estamos creando objetos nuevos.

La concatenación con el operador de suma se ve algo así:

```
String hola = "Hola, ";  
String mundo = "mundo";  
  
String holamundo = hola + mundo;
```

La clase StringBuffer

StringBuffer es una secuencia de caracteres mutable; es decir, es una clase que nos permite concatenar String de una manera simple y muy eficiente.

Los métodos de StringBuffer son sincronizados, por lo cual la podemos usar de manera segura en un ambiente de multihilos.

StringBuffer se usa de la siguiente manera:

```
StringBuffer holamundoBuffer = new StringBuffer();
holamundoBuffer.append("Hola, ");
holamundoBuffer.append("mundo");
```

```
String holamundo = holamundoBuffer.toString();
```

La clase **StringBuilder**

StringBuilder es una clase que aparece con el JDK 5, y es la clase recomendada para la concatenación de **String** en la mayoría de los casos. Su uso es idéntico al **StringBuffer** (tienen el mismo API), por lo resulta relativamente simple reemplazar a los **StringBuffer** por **StringBuilder**.

Los métodos de **StringBuilder** no son sincronizados, por lo que tiene mejor rendimiento que **StringBuffer**. En general, la concatenación de **String** ocurre con variables locales a un método, por lo que es seguro usar **StringBuilder** en lugar de **StringBuffer**. En métodos que hacen uso intensivo de la concatenación, la diferente en rendimiento puede ser importante.

StringBuilder se usa de la siguiente manera:

```
StringBuilder holamundoBuilder = new StringBuilder();
holamundoBuilder.append("Hola, ");
holamundoBuilder.append("mundo");
```

```
String holamundo = holamundoBuilder.toString();
```

Comparación

A modo de ejemplo, vamos a concatenar un millón de **String** ("zim") a un **StringBuilder** y a un **StringBuffer**, y compararemos los tiempos. El código que ejecutaremos será el siguiente:

```
StringBuffer sbuffer = new StringBuffer();
    inicio = System.currentTimeMillis();
    for (int i=0; i<1000000; i++) {
        sbuffer.append("zim");
    }
    fin = System.currentTimeMillis();
    System.out.println("Tiempo del StringBuffer: " +
(fin-inicio));
```

```
StringBuilder sbuilder = new StringBuilder();
    inicio = System.currentTimeMillis();
    for (int i=0; i<1000000; i++) {
        sbuilder.append("zim");
    }
    fin = System.currentTimeMillis();
```

```
System.out.println("Tiempo del StringBuilder: " +  
(fin-inicio));
```

Con varias ejecuciones, los tiempos son constantemente menores para el StringBuilder:

StringBuffer: 93 milisegundos en concatenar un millón de String

StringBuilder: 47 milisegundos en concatenar un millón de String

Es decir, en la mayoría de los casos, StringBuilder puede resultar un 50% más rápido para concatenar String.

Operador de suma

La misma prueba con el operador de suma se lleva una eternidad, ya que la creación constante de nuevos objetos hace que la JVM tenga que empezar a limpiar continuamente el Heap. A modo de referencia, concatenar tan sólo 100.000 String con el operador de suma se demora 99812 milisegundos, por lo que su rendimiento no tiene comparación.

Conclusión

En la amplia mayoría de los casos es recomendable utilizar la clase StringBuilder para concatenar cadenas de caracteres. La clase StringBuilder tiene el mismo API que StringBuffer, por lo que reemplazar código que usa StringBuffer es relativamente sencillo.

La única salvedad sería si la concatenación ocurre en un entorno multihilos, donde diferentes hilos van modificando la misma concatenación. De todas formas, en la mayoría de los casos la concatenación ocurre en variables locales dentro de métodos, por lo que el uso de StringBuilder sigue siendo la opción recomendada.