



Integrantes:

Fernando José Flores Mendoza

Maestro:

Msc. Lesbia Valerío Lacayo

Carrera

Ingeniería en sistemas de Información

Asignatura:

Desarrollo de Aplicaciones 3

LA CLASE String

La clase String es un envoltorio para el tipo de datos primitivo de cadena y proporciona métodos y propiedades que permiten manipular tipos de valores de cadena primitivos. Puede convertir el valor de cualquier objeto en una cadena mediante la función `String()`.

Java posee gran capacidad para el manejo de cadenas dentro de sus clases String y StringBuffer. Un objeto String representa una cadena alfanumérica de un valor constante que no puede ser cambiada después de haber sido creada. Un objeto StringBuffer representa una cadena cuyo tamaño puede variar.



Los Strings son objetos constantes y por lo tanto muy baratos para el sistema. La mayoría de las funciones relacionadas con cadenas esperan valores String como argumentos y devuelven valores String.

Hay que tener en cuenta que las funciones estáticas no consumen memoria del objeto, con lo cual es más conveniente usar `Character` que `char`. No obstante, `char` se usa, por ejemplo, para leer ficheros que están escritos desde otro lenguaje.

Existen muchos constructores para crear nuevas cadenas:

```
String();  
String( String str );  
String( char val[] );  
String( char val[],int offset,int count );  
String( byte val[],int hibyte );  
String( byte val[],int hibyte,int offset,int count );
```

Tal como uno puede imaginarse, las cadenas pueden ser muy complejas, existiendo muchas funciones muy útiles para trabajar con ellas y, afortunadamente, la mayoría están codificadas en la clase String.

Funciones Básicas

La primera devuelve la longitud de la cadena y la segunda devuelve el carácter que se encuentra en la posición que se indica en índice:

```
Int length();  
Char charAt( int indice );
```

LA CLASE StringBuffer

Java posee gran capacidad para el manejo de cadenas dentro de sus clases String y StringBuffer. Un objeto String representa una cadena alfanumérica de un valor constante que no puede ser cambiada después de haber sido creada. Un objeto StringBuffer representa una cadena cuyo tamaño puede variar.

La clase StringBuffer dispone de muchos métodos para modificar el contenido de los objetos StringBuffer. Si el contenido de una cadena va a ser modificado en un programa, habrá que sacrificar el uso de objetos String en beneficio de StringBuffer, que aunque consumen más recursos del sistema, permiten ese tipo de manipulaciones.

Al estar la mayoría de las características de los StringBuffers basadas en su tamaño variable, se necesita un nuevo método de creación:

```
StringBuffer();  
StringBuffer( int len );  
StringBuffer( String str );
```



Se puede crear un StringBuffer vacío de cualquier longitud y también se puede utilizar un String como punto de partida para un StringBuffer.

```
StringBuffer Dos = new StringBuffer( 20 );  
StringBuffer Uno = new StringBuffer( "Hola Mundo" );
```

Cambio de Tamaño

El cambio de tamaño de un StringBuffer necesita varias funciones específicas para manipular el tamaño de las cadenas:

```
Int length();  
Char charAt( int index );  
Void getChars( int srcBegin,int srcEnd,char dst[],int dstBegin );  
String toString();  
Void setLength( int newlength );  
Void setCharAt( int index,char ch );  
Int capacity();  
Void ensureCapacity( int minimum );  
Void copyWhenShared();
```

Observar que una de las funciones devuelve una cadena constante normal de tipo String. Este objeto se puede usar con cualquier función String, como por ejemplo, en las funciones de comparación.

Modificación del Contenido

Para cambiar el contenido de un StringBuffer, se pueden utilizar dos métodos: `append()` e `insert()`.

En el ejemplo `CadAppend.java`, vemos el uso de estos dos métodos:

```
Class CadAppend {  
    Public static void main( String args[] ) {  
        StringBuffer str = new StringBuffer( "Hola" );  
        Str.append( " Mundo" );  
        System.out.println( str );  
    }  
}
```

CLASE STRINGBUILDER

La clase `StringBuilder` es similar a la clase `String` en el sentido de que sirve para almacenar cadenas de caracteres. No obstante, presenta algunas diferencias relevantes. Señalaremos como características de `StringBuilder` a tener en cuenta:

- Su tamaño y contenido pueden modificarse. Los objetos de éste tipo son mutables. Esto es una diferencia con los `String`.
- Debe crearse con alguno de sus constructores asociados. No se permite instanciar directamente a una cadena como sí permiten los `String`.
- Un `StringBuilder` está indexado. Cada uno de sus caracteres tiene un índice: 0 para el primero, 1 para el segundo, etc.
- Los métodos de `StringBuilder` no están sincronizados. Esto implica que es más eficiente que `StringBuffer` siempre que no se requiera trabajar con múltiples hilos (`threads`), que es lo más habitual.

La clase `StringBuilder` no ofrece ninguna garantía de sincronización, mientras que la clase `StringBuffer` sí lo hace. Por lo tanto, esta clase está diseñada para usarse como un reemplazo directo para `StringBuffer` en lugares donde `StringBuffer` estaba siendo utilizado por un solo subproceso (como suele ser el caso). Donde sea posible, se recomienda usar esta clase con preferencia a `StringBuffer`, ya que será más rápida en la mayoría de las implementaciones.

Las instancias de `StringBuilder` no son seguras para que las usen varios subprocesos. Si se requiere dicha sincronización, se recomienda utilizar `StringBuffer`. `String Builder` no es seguro para subprocesos y tiene un alto rendimiento en comparación con `String buffer`.

Constructores en la clase Java `StringBuilder`

`StringBuilder()`: construye un generador de cadenas sin caracteres y con una capacidad inicial de 16 caracteres.

`StringBuilder (capacidad int):` construye un generador de cadenas sin caracteres y una capacidad inicial especificada por el argumento de capacidad.

`StringBuilder(CharSequence seq):` construye un generador de cadenas que contiene los mismos caracteres que la `CharSequence` especificada.
`StringBuilder(String str):` construye un generador de cadenas inicializado con el contenido de la cadena especificada.

