

Teoría JAVA I

Clase String

La **clase String** es fundamental en Java para representar y manipular cadenas de caracteres. En la programación Java, una cadena de texto, o String, es una secuencia de caracteres tratada como un objeto. Esta clase es una de las más utilizadas en el lenguaje y proporciona una amplia gama de métodos que permiten realizar operaciones comunes en cadenas de manera eficiente y efectiva.

Entre los métodos proporcionados por la clase String, se encuentran aquellos que permiten realizar operaciones básicas como la concatenación de cadenas, la búsqueda de caracteres o subcadenas, el reemplazo de caracteres, la extracción de subcadenas y la manipulación de caracteres individuales.

A continuación, te presentamos ejemplos de algunos métodos comunes de la clase String:

- **length():** Devuelve la longitud de la cadena, es decir, el número de caracteres que contiene.

```
String texto = "Hola Mundo";  
int longitud = texto.length(); // Devuelve 10
```

- **charAt(int index):** Devuelve el carácter en la posición especificada por el índice.

```
String texto = "Hola Mundo";  
char primerCaracter = texto.charAt(0); // Devuelve 'H'
```

- **substring(int beginIndex, int endIndex):** Devuelve una subcadena de la cadena original, desde el índice de inicio hasta el índice de fin (excluido).

```
String texto = "Hola Mundo";  
String subcadena = texto.substring(5, 10); // Devuelve "Mundo"
```

- **equals(String str):** Compara dos cadenas y devuelve true si son iguales, es decir, si contienen los mismos caracteres en el mismo orden.

```
String texto1 = "Hola";  
String texto2 = "Hola";  
boolean sonIguales = texto1.equals(texto2); // Devuelve true
```

- **indexOf(String str):** Devuelve el índice de la primera aparición de la subcadena especificada dentro de la cadena original.

```
String texto = "Hola Mundo";  
int indice = texto.indexOf("Mundo"); // Devuelve 5
```

Métodos de la clase String

En este anexo, puedes ver algunos de los distintos métodos que tiene la **clase String** explicados brevemente:

Método	Descripción
charAt(int index)	Devuelve el carácter en la posición específica (index).
chars()	Devuelve una secuencia de valores int que representan los caracteres de la cadena en orden.
codePointAt(int index)	Devuelve el punto de código Unicode en la posición especificada.
codePointBefore(int index)	Devuelve el punto de código Unicode inmediatamente antes de la posición especificada.
codePointCount(int beginIndex, int endIndex)	Devuelve la cantidad de puntos de código Unicode en un subintervalo de esta cadena.

codePoints()	Devuelve una secuencia de puntos de código Unicode en la cadena.
compareTo(String anotherString)	Compara alfabéticamente esta cadena con otra cadena.
compareToIgnoreCase(String str)	Compara lexicográficamente dos cadenas, ignorando las diferencias de mayúsculas y minúsculas.
concat(String str)	Concatena la cadena especificada al final de esta cadena.
endsWith(String suffix)	Comprueba si esta cadena termina con el sufijo especificado.
equals(Object anObject)	Compara esta cadena con el objeto especificado.
equalsIgnoreCase(String anotherString)	Compara esta cadena con otra cadena, ignorando las diferencias de mayúsculas y minúsculas.
formatted(Object... args)	Formatea la cadena utilizando los argumentos proporcionados.
getBytes(), getBytes(Charset charset), getBytes(String charsetName)	Convierte la cadena en una secuencia de bytes utilizando el conjunto de caracteres predeterminado o ingresado por parámetro.
getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin)	Copia los caracteres de esta cadena en el array de destino.
isBlank()	Devuelve true si la cadena está vacía o contiene solo espacios en blanco.
isEmpty()	Devuelve true si la longitud de la cadena es 0.

lastIndexOf(int ch), lastIndexOf(String str), lastIndexOf(int ch, int fromIndex), lastIndexOf(String str, int fromIndex)	Devuelve el índice de la última aparición del carácter o cadena especificada, comenzando la búsqueda hacia atrás desde el índice especificado.
length()	Devuelve la longitud de esta cadena.
lines()	Devuelve una secuencia de líneas de la cadena, separadas por saltos de línea.
matches(String regex)	Indica si esta cadena coincide con la expresión regular dada.
offsetByCodePoints(int index, int codePointOffset)	Devuelve el índice en esta cadena que es desplazado desde el índice dado por el número de puntos de código.
regionMatches(int toffset, String other, int ooffset, int len), regionMatches(boolean ignoreCase, int toffset, String other, int ooffset, int len)	Prueba si dos regiones de cadenas son iguales.
repeat(int count)	Devuelve una cadena cuyo valor es la concatenación de esta cadena repetida count veces.
replace(char oldChar, char newChar), replace(CharSequence target, CharSequence replacement)	Devuelve una nueva cadena que resulta de reemplazar todas las apariciones del carácter o la secuencia de caracteres de destino en esta cadena con el carácter o la secuencia de caracteres de reemplazo.
replaceAll(String regex, String replacement)	Reemplaza cada subcadena de esta cadena que coincide con la expresión regular dada con la cadena de reemplazo.
replaceFirst(String regex, String replacement)	Reemplaza la primera subcadena de esta cadena que coincide con la expresión regular dada con la cadena de reemplazo.

split(String regex), split(String regex, int limit)	Divide esta cadena alrededor de las coincidencias con la expresión regular dada.
startsWith(String prefix), startsWith(String prefix, int toffset)	Prueba si esta cadena comienza con el prefijo especificado.
strip(), stripLeading(), stripTrailing()	Devuelve una cadena cuyos valores son los de esta cadena, con todos los espacios en blanco líderes y/o finales eliminados.
subSequence(int beginIndex, int endIndex)	Devuelve una secuencia de caracteres que es una subsecuencia de esta secuencia.
substring(int beginIndex), substring(int beginIndex, int endIndex)	Devuelve una nueva cadena que es una subcadena de esta cadena.
toCharArray()	Convierte esta cadena en una nueva matriz de caracteres.
toLowerCase(), toLowerCase(Locale locale)	Convierte todos los caracteres de esta cadena a minúsculas utilizando las reglas del local predeterminado o del especificado.
toUpperCase(), toUpperCase(Locale locale)	Convierte todos los caracteres de esta cadena a mayúsculas utilizando las reglas del local predeterminado o del especificado.
translateEscapes()	Devuelve una cadena cuyo valor es el de esta cadena, con cualquier secuencia de escape traducida.
trim()	Devuelve una copia de la cadena, con los espacios en blanco iniciales y finales omitidos.

💡 Siempre es recomendable utilizar la **documentación oficial** para obtener más detalles sobre el material proporcionado, su uso y su implementación. Puedes acceder a la información sobre la Clase String desde [aquí](#).