

UT02-08 - La Clase String

La Clase String. Introducción

- Para Java las cadenas de texto son objetos especiales. Los textos deben manejarse creando objetos de tipo String. Ejemplo:

```
String texto1 = "¡Prueba de texto!";
```

- Las cadenas pueden ocupar varias líneas utilizando el operador de concatenación `"+"`

```
String texto2 ="Este es un texto que ocupa " + "varias líneas, no obstante se puede " + "perfectamente encadenar";
```

- También se pueden crear objetos String sin utilizar constantes entrecomilladas, usando otros constructores:

```
//Ejemplos de String
char[] palabra = {'P','a','l','a','b','r','a'};//Array de char
String cadena = new String(palabra);
byte[] datos = {97,98,99};
String codificada = new String (datos, "8859_1");
//En el último ejemplo la cadena codificada se crea desde un array
//de tipo byte que contiene números que serán interpretados como códigos Unicode.
//Al asignar, el valor 8859_1 indica la tabla de códigos a utilizar.
```

Comparación entre objetos String

- **No se puede usar** el operador de comparación : `==` .
- En su lugar se deben utilizar estas expresiones:
 - `cadena1.equals(cadena2)`.
 - El resultado es true si la cadena1 es igual a la cadena2. Ambas cadenas son variables de tipo String.
 - `cadena1.equalsIgnoreCase(cadena2)`.
 - Como la anterior, pero en este caso no se tienen en cuenta mayúsculas y minúsculas.
 - `s1.compareTo(s2)`.
 - Compara ambas cadenas, considerando el orden alfabético.
 - Si la primera cadena es mayor en orden alfabético que la segunda devuelve 1 o un número positivo,
 - Si son iguales devuelve 0
 - Si es la segunda la mayor devuelve -1 o un número negativo.
 - `s1.compareToIgnoreCase(s2)`.
 - Igual que la anterior, sólo que además ignora las mayúsculas.
- Hay que tener en cuenta que el orden no es el del alfabeto español, sino que usa la tabla ASCII, en esa tabla la letra ñ es mucho mayor que la o.

El método valueOf

- `String.valueOf(valor)`
- Este método se encuentra no sólo a la clase String, sino en otras también y siempre es un método que convierte valores de una clase a otra.
- En el caso de los objetos String, permite convertir valores que no son de cadena a forma de cadena.

- **Ejemplo :**

```
//Ejemplo
String numero = String.valueOf(1234);
String fecha  = String.valueOf(new Date());
```

- En el ejemplo se observa que este método pertenece a la clase String directamente.
- No hay que utilizar el nombre del objeto creado (como se verá más adelante, es un método estático).

Métodos más comunes de la Clase String

//METODO=====	DESCRIPCION=====
char charAt(int index)	Proporciona el carácter que está en la posición dada por el entero index.
int compareTo(string s)	Compara las dos cadenas. Devuelve un valor menor que cero si la cadena s es mayor que la original, devuelve 0 si son iguales y devuelve un valor mayor que cero si s es menor que la original.
int compareToIgnoreCase(string s)	Compara dos cadenas, pero no tiene en cuenta si el texto es mayúsculas o no.
String concat(String s)	Añade la cadena s a la cadena original.
String copyValueOf(char[] data)	Produce un objeto String que es igual al array de caracteres data.
boolean endsWith(String s)	Devuelve true si la cadena termina con el texto s
boolean equals(String s)	Compara ambas cadenas, devuelve true si son iguales
boolean equalsIgnoreCase(String s)	Compara ambas cadenas sin tener en cuenta las mayúsculas y las minúsculas.
byte[] getBytes()	Devuelve un array de caracteres que toma a partir de la cadena de texto
void getBytes(int srcBegin, int srcEnd, char[] dest, int dstBegin);	Almacena el contenido de la cadena en el array de caracteres dest.
	Toma los caracteres desde la posición srcBegin hasta la posición srcEnd y les copia en el array desde la posición dstBegin
int indexOf(String s)	Devuelve la posición en la cadena del texto s. Valor -1 si no lo encuentra.
int indexOf(String s, int primeraPos)	Devuelve la posición en la cadena del texto s, empezando a buscar desde la posición PrimeraPos
int lastIndexOf(String s)	Devuelve la última posición en la cadena del texto s. Valor -1 si no lo encuentra.
int lastIndexOf(String s, int primeraPos)	Devuelve la última posición en la cadena del texto s, empezando a buscar desde la posición PrimeraPos
int length()	Devuelve la longitud en caracteres de la cadena.
boolean startsWith(String s)	Devuelve true si la cadena comienza con el texto s.
String replace(char carAnterior, char carNuevo)	Devuelve una cadena idéntica al original pero que ha cambiado los caracteres iguales a carAnterior por carNuevo
String replaceFirst(String str1, String str2)	Cambia la primera aparición de la cadena str1 por la cadena str2
String replaceAll(String str1, String str2)	Cambia todas las apariciones de la cadena uno por la cadena dos
String split(String str[, int max])	Fracciona un String en varios hasta max según el patrón str o una expresión regular en str
String substring(int primeraPos, int segundaPos)	Devuelve el texto que va desde primeraPos a segundaPos.
char[] toCharArray()	Devuelve un array de caracteres a partir de la cadena dada
String toLowerCase()	Convierte la cadena a minúsculas
String toLowerCase()	Lo mismo pero siguiendo las instrucciones del argumento local
String toUpperCase()	Convierte la cadena a mayúsculas
String toUpperCase()	Lo mismo pero siguiendo las instrucciones del argumento local
String trim()	Elimina los blancos que tenga la cadena tanto por delante como por detrás
Static String valueOf(tipo elemento)	Devuelve la cadena que representa el valor elemento.

- Se recomienda probar los métodos de String en un programa de prueba

Retorno

- [Unidad 02. Identificación de Elementos](#)
- Índice [General de la Asignatura](#). Vuelta al [Índice General de Apuntes](#) de Asignaturas.

UT02-08 - La Clase String - [Andrés Ramos González](#) - IES Alonso de Avellaneda

Este obra está bajo una licencia: [Creative Commons Reconocimiento-CompartirIgual 3.0](#)

