**La clase String**

La clase String representa una cadena de caracteres. Todos los literales de la cadena tales como “abc” son implementados como instancias de ésta clase.

La clase String incluye métodos que permiten examinar los caracteres individuales de una cadena para compararlos, ubicarlos extraerlos como sub cadenas y para crear copias de una cadena convirtiendo todos sus caracteres a mayúsculas o minúsculas.

Java provee un soporte especial para la concatenación (unión) de cadenas con el operador (+). y para convertir otros objetos a cadenas.La concatenación de cadenas es implementada a través de la clase StringBuffer y su método append(). Las conversiones son implementadas a través del método toString() definido en la clase Object la cual es heredada por todas las clases en Java.

**PARA TENER EN CUENTA :**

La clase String pertenece al paquete java.lang.

Existen diferentes formas para crear objetos String como las siguientes :

String str =”abc”;

char str[] = {‘a’,’b’,’c’}; //arreglo de caracteres

String str = new String (“abc”) ; //objeto con valor inicial “abc”

El lenguaje java proporciona el operador “+” para concatenar (unir) objetos String.

Un objeto String representa una cadena de caracteres no modificable;por lo tanto, una operación como convertir a mayúsculas o minúsculas no modificará el objeto original sino que devolverá un nuevo objeto con la cadena resultante de esa operación el cual deberá ser guardado en otro objeto string.

**JERARQUÍA DE CLASES**



**MÉTODOS MAS IMPORTANTES**

String (String valor) .- Se usa cuando se desea crear una cadena de caracteres :

String cadena = “Hola”

String cadena = new String (“Hola”); //Constructor de la clase String

cadena.**concat**(string txt) .- Devuelve como resultado un nuevo objeto String resultado de concatenar dos cadena :

String cadena = “Hola”

String result= cadena.**concat**(" Mundo");//El resultado será Hola mundo

cadena.**compareTo**(String otra cadena) .- Compara lexicográficamente la cadena con otra cadena , esta comparación es caracter por caracter teniendo en cuenta que las mayúsculas son mayores que las minúsculas – esto es de acuerdo al código ASCII o UNICODE – según esto devuelve :

< 0 : si cadena < otra cadena

= 0 : si cadena = otra cadena

> 0 : si cadena > otra cadena

Ejemplo :

JUAN > juan

Juan > Carlos

JUAN > JUAn

abcde > abcdefg

cadena.**toIgnoreCase**(String otra cadena) .- Hace lo mismo que compareTo pero con la única diferencia que no se hace diferencia entre mayúsculas y minúsculas ; es decir : JUAN = juan

cadena.**length**() .- Devuelve la longitud o número de caracteres de la cadena :

String cad="Hola Mundo";

int cant = cad.**length**();// El valor de cant será 10

cadena.**toLowerCase**() .- Convierte a minúsculas las letras mayúsculas de la cadena

cadena.**toUpperCase**().- Convierte en mayúsculas las letras minúsculas de la cadena.

String cad1="Esto Es Una Prueba";

String cad2=cad1.**toUpperCase**();//cad2 será igual a : ESTO ES UNA PRUEBA.

cadena.**trim** () .- Elimina los espacios en blanco que puedan haber al principio o al final de la cadena.

String cad1=" Esto Es Una Prueba ";

String cad2=cad1.**toLowerCase**();//cad2=Esto Es Una Prueba.

**MÉTODOS**

**CONSTRUCTORES**



**MÉTODOS**



