Java

Tema 2

La clase String

Cadenas de caracteres. La clase String

- En Java las cadenas de caracteres son **objetos** de la clase String.
- Existen distintas formas de crear un String:

```
String s = new String("Hola");

String s = "Hola";
```

La clase String: características

- Un objeto de la clase String es **inmutable**, es decir no se puede modificar. Los cambios realizados con sus métodos deben almacenarse en otro objeto String

```
String palabra = "Hola";

palabra.toUperCase();

String enMayusculas = palabra.toUperCase();
```

La variable palabra no cambia su valor

Debemos almacenar el String devuelto

3

Operador de concatenación +

- El operador + puede utilizarse para concatenar cadenas.

```
String nombre = "Francisco";

String resultado =
 nombre + " tiene " + nombre.length() + " caracteres";

System.out.println(resultado);
```

4

Cadenas de caracteres. La clase String

❑ La clase String contiene varios **métodos** para operar sobre cadenas, se ejecutan sobre un objeto de tipo String (cadena):

Devuelve	Método	Descripción
int	length()	Devuelve la longitud de la cadena, total de caracteres
char	charAt(int indice)	Devuelve el carácter situado en la posición indicada por *índice*. El primer carácter de una cadena ocupa la posición es 0 y el último length() - 1
boolean	equals(String cad)	Si las dos cadenas son iguales devuelve true, sino false
boolean	equalsIgnoreCase (String cad)	Comprueba si dos cadenas son iguales, sin distinguir entre mayúsculas y minúsculas. Devuelve true o false
String	toLowerCase()	Convierte la cadena a minúsculas
String	toUpperCase()	Convierte la cadena a mayúsculas
String	concat(String cad)	Concatena por la derecha cad
String	substring(int *inicio*, int *fin*)	Extrae una subcadena entre las posiciones *inicio* y *fin*, excluyendo el carácter de la posición *fin*
String	subString(int inicio)	Extrae una subcadena de *inicio* hasta el *final*

5

Cadenas de caracteres. La clase String

❑ Más **métodos** de la clase String:

Devuelve	Método	Descripción
int	compareTo(String *cad*)	Compara la cadena con *cad*. Devuelve 0 sin son iguales, negativo si cadena va antes en orden alfabético que *cad,* y positivo si va después
int	compareToIgnoreCase(String cad)	Igual que compareTo, pero no tiene en cuenta mayúsculas ni minúsculas
int	indexOf(String cad)	Devuelve la posición que ocupa la **primera** vez que encuentra *cad* (-1 si no se encuentra).
Int	lastIndexOf(int ch)	Retorna la posición que ocupa la **última** vez que encuentra el carácter *ch*
String	replace(char viejo, char nuevo)	Reemplaza el carácter *viejo* por el *nuevo*
String	trim()	Elimina espacios en blanco de la cadena
boolean	contains(String texto)	Si texto está contenido en cadena devuelve true, sino false

6

Ver documentación de la clase String en la api java

Cadenas de caracteres. La clase String

Ejemplo de uso de algunos métodos de la clase String

```
public static void main(String[] args){

  String nombre = "Francisco";
  System.out.println("Total caracteres " + nombre.length());
  System.out.println("En mayúsculas es " + nombre.toUpperCase());
  System.out.println("En minúsculas es " + nombre.toLowerCase());
  System.out.println( nombre.concat(" Quevedo"));

  System.out.println(nombre);  //muestra Francisco
  nombre = nombre.concat(" Quevedo"); //almacenamos el texto contatenado
  System.out.println(nombre);  //muestra Francisco Quevedo

  String cadena =  nombre.substring(4);
  System.out.println(cadena);

  System.out.println(nombre.contains("Que"));
}
```

7

Cadenas de caracteres. La clase String

La forma correcta de comparar objetos String es utilizando los métodos **equals**() o **equalsIgnoreCase**()

```
String str1 = "El lenguaje Java";
String str2 = new String("El lenguaje Java");

if (str1 == str2) {  //COMPARA LAS REFERENCIAS A OBJETO
    System.out.println("Los mismos objetos");
} else {
    System.out.println("Distintos objetos");
}

if (str1.equals(str2)) {  //COMPARA EL CONTENIDO
    System.out.println("El mismo contenido");
} else {
    System.out.println("Distinto contenido");
}
```

muestra por pantalla

muestra por pantalla

8

Ejecuta el código para comprobar lo que devuelve por pantalla

La clase StringBuilder

- Almacena cadenas de caracteres. Una forma de creación es:

```
StringBuilder variable = new StringBuilder("Hola");
```

- Es **mutable**, su contenido se puede modificar directamente:

```
variable.append(" Mundo!");
```

variable cambia su valor y sería "Hola Mundo!"

9

La clase StringBuilder

Algunos de los métodos más utilizados:

Retorno	Método	Explicación
StringBuilder	**append**(...)	Añade al final del StringBuilder a la que se aplica, un String o la representación en forma de String de un dato asociado a una variable primitiva
int	**length**()	Devuelve el número de caracteres del StringBuilder
StringBuilder	**reverse**()	Invierte el orden de los caracteres del StringBuilder
char	**charAt**(int posicion)	Devuelve el carácter asociado a la posición que se le indica en posicion
StringBuilder	**insert**(int indiceIni, String cadena)	Añade la cadena del segundo argumento a partir de la posición indicada en el primero indiceIni
StringBuilder	**deleteChar**(int indice)	Borra el carácter indicado en índice
StringBuilder	**replace**(int indiceIni, int indiceFin, String str)	Reemplaza los caracteres comprendidos entre los dos indices por la cadena que se le pasa en el argumento str

10