Manipular cadenas con String

Fundamentos String

- Un objeto de la clase String es una cadena de caracteres inmutable, no se pueden modificar
- Los métodos que operan con String devuelven una copia de la cadena modificada
- Se pueden crear:

```
String n1=new String("mi cadena"); //Deprecated
```

- O:

```
String n1="mi cadena";
```

En este caso puede reutilizar cadenas de un pool

Métodos String (I)

- int length(). Devuelve la longitud de la cadena

- String toLowerCase(), toUpperCase(). Devuelven la cadena convertida a minúsculas y mayúsculas, respectivamente

```
String n1="cadena";
System.out.println(n1.toUpperCase()); //muestra: CADENA
System.out.println(n1); //muestra: cadena, no ha cambiado
```

- String substring(int a, int b). Devuelve un trozo de cadena comprendido entre las posiciones a y b-1

```
String n1="esto es un texto";
System.out.println(n1.substring(3,9)); //muestra: o es u
```

Métodos String (II)

- **char charAt(int pos). Devuelve el carácter que ocupa la posición indicada**

```
String n1="esto es un texto";
System.out.println(n1.charAt(0)); //muestra: e
System.out.println(n1.charAt(20));  //StringIndexOutOfBoundsException
```

- **int indexOf(String cad). Devuelve la posición de la cadena parámetro. Si no existe, devuelve -1**

```
String n1="esto es un texto";
System.out.println(n1.indexOf("un")); //muestra: 8
```

- **String replace(CharSequence c1, CharSequence c2). Devuelve la cadena resultante de reemplazar la subcadena c1 por c2.**

```
String n1="esto es un texto";
System.out.println(n1.replace("es","de")); //muestra: deto de un texto
```

Métodos String (III)

- boolean startsWith(String s), endsWith(String s). Indica si la cadena empieza o termina, respectivamente, por el texto recibido:

```
String n1=“esto es un texto”;
System.out.println(n1.endsWith(“to”)); //muestra: true
System.out.println(n1.startsWith(“eso”)); //muestra: false
```

- String trim(). Devuelve la cadena resultante de eliminar espacios al principio y al final de la misma

```
String n1=“  cadena prueba nueva  ”;
System.out.println(n1.trim().length()); //muestra: 17
```

- String concat(String s). Mismo efecto que aplicar el operador +

- boolean isEmpty(). Devuelve true si es una cadena vacía. Equivale:

```
cad.equals(“”)
```

Nuevos métodos Java 11

- boolean isBlank(). Devuelve true si la cadena está vacía o contiene solamente espacios en blanco:

```
String s1="";
String s2="     ";
System.out.println(s1.isBlank()); //muestra: true
System.out.println(s2.isBlank()); //muestra: true
```

- String repeat(int n). Devuelve una cadena resultado de concatenar tantas veces la cadena actual como se indique en el parámetro

```
String cad="hello";
System.out.println(cad.repeat(3));  //imprime  "hellohellohello"
```

- String strip(). Devuelve una cadena resultante de eliminar espacios a izquierda y derecha. Similar a trim(), pero reconoce más caracteres en blanco