

UT5 ejercicios 6 y 7

Ejercicio 6

Substring()

```
String s = "Hello World!";

// Retorna un substring que comienza desde el indice beginInd
System.out.println(s.substring(0,5)); // Imprime 'Hello'
// Retorna un substring que comienza desde el indice beginInd
System.out.println(s.substring(6)); // Imprime 'World!'
```

Split()

```
String s = "Hello World!";

// Separa el string usando el argumento dentro del método spl
String[] split = s.split(" ");
System.out.println(split[0] + ", " + split[1]); // Imprime He
System.out.println(split.length); // Imprime 2, la cantidad d

// Se puede aclarar con un argumento int extra, la cantidad m
String[] splitAgain = s.split("l",3);
System.out.println(splitAgain[0] + splitAgain[1] + splitAgain
System.out.println(splitAgain.length); // Imprime 3, la canti
```

Subsequence()

```
String s = "Hello World!";

// Retorna una nueva secuencia de caracteres desde beginIndex
CharSequence subSequence = s.subSequence(3, 7);
System.out.println(subSequence); // Imprime "lo W"
```

Trim()

```
// Elimina los espacios iniciales y finales del string
String r = "    Hello, World!    ";
String trimmedString = r.trim();
System.out.println(trimmedString); // Imprime "Hello, World"
```

ToLowerCase()

```
String s = "Hello World!";

// Devuelve una copia del string convertido en mayúsculas o minúsculas
String sToLower = s.toLowerCase();
System.out.println(sToLower); // Imprime "hello world!"
```

ToUpperCase()

```
String s = "Hello World!";

// Devuelve una copia del string convertido en mayúsculas
String sToUpper = s.toUpperCase();
System.out.println(sToUpper); // Imprime "HELLO WORLD!"
```

IndexOf()

1. Con números de argumento:

```
String s = "Hello World!";

// Devuelve el índice de la primer ocurrencia del caracter e
int i = s.indexOf(7);
System.out.println(i);
// Devuelve el índice de la última ocurrencia del caracter e
i = s.lastIndexOf(7);
System.out.println(i);
```

2. Con argumento de caracter e índice:

```
String s = "Hello World!";

// Devuelve el índice de la primer ocurrencia del caracter es
// de atrás hacia delante a partir del índice específico.
int p = s.indexOf(7, 3);
System.out.println(p);

// Devuelve el índice de la última ocurrencia del caracter es
// de adelante hacia atrás a partir del índice específico.
p = s.lastIndexOf(7, 3);
System.out.println(p);
```

3. Con argumento de String:

```
String s = "Hello World!";

// Devuelve el índice de la primer ocurrencia del caracter es
int j = s.indexOf("o");
System.out.println(j);
// Devuelve el índice de la última ocurrencia del caracter es
j = s.lastIndexOf("o");
System.out.println(j);
```

4. Con argumento de String y número de índice:

```
String s = "Hello World!";

// Devuelve el índice de la primer ocurrencia del caracter es
// de atrás hacia delante a partir del índice específico.
int t = s.indexOf("o", 7);
System.out.println(t);
// Devuelve el índice de la última ocurrencia del caracter es
// buscando de adelante hacia atrás a partir del índice espec.
t = s.lastIndexOf("o", 7);
System.out.println(t);
```

Contains()

```
String s = "Hello World!";

// Retorna true si el string contiene la subsecuencia de caracteres
boolean contiene = s.contains("subsecuencia");
System.out.println(contiene); // Retorna false
```

Replace()

```
String s = "Hello World!";

// Retorna un nuevo string, resultado de reemplazar todas las
// de oldChar con newChar en el mismo.
String reemplazado = s.replace("o", "i");
System.out.println(reemplazado); // Imprime "Helli Wirld!"

//Reemplaza cada subsecuencia de caracteres con otra
reemplazado = s.replace("World", "People");
System.out.println(reemplazado); // Imprime "Hello People!"

//Se pueden usar expresiones regulares (regex), más compleja
reemplazado = s.replaceAll("l+", "w" );
System.out.println(reemplazado); // Imprime "Hewo Worwd!"

// Reemplaza el primer substring del string que coincida con
reemplazado = s.replaceFirst("l", "w");
System.out.println(reemplazado); // Imprime "Hewo World!"
```

Ejercicio 7

Constructores

```
// Crea un string builder vacío con capacidad de 16 elementos
StringBuilder sb = new StringBuilder();

// Crea un string builder inicializado con una secuencia de caracteres
CharSequence cs = "Hello World";StringBuilder sb2 = new StringBuilder(cs);

// Crea un constructor vacío con la capacidad inicial especificada
StringBuilder sb3 = new StringBuilder(2);

// Crea un constructor cuyo valor es inicializado por el string
StringBuilder sb4 = new StringBuilder("s");
```

SetLength()

```
// Establece el largo de la secuencia de caracteres
// Si newLength < length
sb2.setLength(5);
System.out.println(sb2); // imprime "Hello"

//Si newLength > length
sb2.setLength(12);
System.out.println(sb2.length());

// imprime el "mismo" contenido, pero se agregan caracteres vacíos
// teniendo ahora 12 elementos.
```

EnsureCapacity()

```
// Asegura que la capacidad sea por lo menos igual a lo mínimo
sb2.ensureCapacity(3);
System.out.println(sb2.length());
```

Append()

```
// Hace un append del argumento al string builder. Los datos
// que se lleve la operación a cabo.
sb2.append(55);
sb2.append("Appending");
sb2.append(33.5f);
System.out.println(sb2);
```

Delete()

```
// Elimina la subsecuencia desde el inicio hasta el final-1
sb2.delete(7, 10);
System.out.println(sb2);

// Elimina el caracter localizado en el indice especificado.
sb2.deleteCharAt(3);
System.out.println(sb2);
```

Insert()

```
// Inserta el segundo argumento en el string builder
// El primer argumento indica el indice anterior al que la da
// La data se convierte a string antes de que la operación se
sb2.insert(6, "Inserting");
sb2.insert(3, true);
sb2.insert(9, 33.45f);
System.out.println(sb2);
```

Reverse()

```
// mueve en reversa la secuencia de caracteres en el string b
sb2.reverse();
System.out.println(sb2);
```