

La clase *String*

Hemos aprendido a diferenciar entre clase y objetos, a acceder desde un objeto a los miembros datos y a las funciones miembro. Vamos a utilizar clases importantes en el lenguaje Java y a crear objetos de dichas clases. Empezaremos por la clase *String* una de las más importantes del lenguaje Java. Más adelante, volveremos a estudiar otros ejemplos para que el lector se acostumbre a crear sus propias clases.

La clase *String*

Dentro de un objeto de la clases *String* o *StringBuffer*, Java crea un array de caracteres de una forma similar a como lo hace el lenguaje C++. A este array se accede a través de las funciones miembro de la clase.

Los strings u objetos de la clase *String* se pueden crear explícitamente o implícitamente. Para crear un string implícitamente basta poner una cadena de caracteres entre comillas dobles. Por ejemplo, cuando se escribe

```
System.out.println("El primer programa");
```

Java crea un objeto de la clase *String* automáticamente.

Para crear un string explícitamente escribimos

```
String str=new String("El primer programa");
```

También se puede escribir, alternativamente

```
String str="El primer programa";
```

Para crear un string nulo se puede hacer de estas dos formas

```
String str="";  
String str=new String();
```

Un string nulo es aquél que no contiene caracteres, pero es un objeto de la clase *String*. Sin embargo,

```
String str;
```

se está declarando un objeto *str* de la clase *String*, pero aún no se ha creado ningún objeto de esta clase.

Cómo se obtiene información acerca del string

Una vez creado un objeto de la clase *String* podemos obtener información relevante acerca del objeto a través de las funciones miembro.

Para obtener la longitud, número de caracteres que guarda un string se llama a la función miembro *length*.

```
String str="El primer programa";  
int longitud=str.length();
```

Extraer un substring de un string (opcional)

En muchas ocasiones es necesario extraer una porción o substring de un string dado. Para este propósito hay una función miembro de la clase *String* denominada *substring*.

Para extraer un substring desde una posición determinada hasta el final del string escribimos

```
String str="El lenguaje Java";  
String subStr=str.substring(12);
```

Se obtendrá el substring "Java".

Una segunda versión de la función miembro *substring*, nos permite extraer un substring especificando la posición de comienzo y la el final.

```
String str="El lenguaje Java";  
String subStr=str.substring(3, 11);
```

Se obtendrá el substring "lenguaje". Recuérdese, que las posiciones se empiezan a contar desde cero.

Convertir un número a string (opcional)

Para convertir un número en string se emplea la función miembro estática *valueOf* (más adelante explicaremos este tipo de funciones).

```
int valor=10;  
String str=String.valueOf(valor);
```

La clase *String* proporciona versiones de *valueOf* para convertir los datos primitivos: **int**, **long**, **float**, **double**.

Esta función se emplea mucho cuando programamos applets, por ejemplo, cuando queremos mostrar el resultado de un cálculo en el área de trabajo de la ventana o en un control de edición.

Convertir un string en número (opcional)

Cuando introducimos caracteres en un control de edición a veces es inevitable que aparezcan espacios ya sea al comienzo o al final. Para eliminar estos espacios tenemos la función miembro *trim*

```
String str=" 12 ";
String str1=str.trim();
```

Para convertir un string en número entero, primero quitamos los espacios en blanco al principio y al final y luego, llamamos a la función miembro estática *parseInt* de la clase *Integer* (clase envolvente que describe los números enteros)

```
String str=" 12 ";
int numero=Integer.parseInt(str.trim());
```

Para convertir un string en número decimal (**double**) se requieren dos pasos: convertir el string en un objeto de la clase envolvente *Double*, mediante la función miembro estática *valueOf*, y a continuación convertir el objeto de la clase *Double* en un tipo primitivo **double** mediante la función *doubleValue*

```
String str="12.35 ";
double numero=Double.valueOf(str).doubleValue();
```

Se puede hacer el mismo procedimiento para convertir un string a número entero

```
String str="12";
int numero=Integer.valueOf(str).intValue();
```

```
public class StringApp {

    public static void main(String[] args) {
//longitud del string
        String str="El primer programa";
        System.out.println("longitud "+str.length());
//substring
        str="El lenguaje Java";
        System.out.println("Substring "+str.substring(12));
        System.out.println("Substring "+str.substring(3, 11));
//eliminar los espacios en blanco
        str=" 12 ";
        System.out.println("string original "+str);
        System.out.println("string sin espacios en blanco "+str.trim());
//convertir un número en string
        int valor=24;
        str=String.valueOf(valor);
        System.out.println("número --> string "+str);
//convertir un string en número
        str=" 12 ";
        int numeroInt=Integer.parseInt(str.trim());
        System.out.println("string --> número "+numeroInt);
        str="12.35 ";
        double numeroDouble=Double.valueOf(str).doubleValue();
        System.out.println("string --> número "+numeroDouble);
    }
}
```

}	
---	--