



Programación III POO JAVA

Clase String y Clase Scanner

MTI. Guadalupe Ledesma Ramos guadalupe.ledesma@upslp.edu.mx

String y StringBuffer

```
package bienvenida;
```

```
public class Cadena {  
    private static String salida;  
    private final StringBuffer buffer1;  
    private final StringBuffer buffer2;  
    private final StringBuffer buffer3;  
  
    public Cadena() {  
        buffer1 = new StringBuffer("Hola");  
        buffer2 = new StringBuffer(" ");  
        buffer3 = new StringBuffer("Mundo");  
    }  
  
    public void imprime() {  
        salida = buffer1.toString() + buffer2.toString() + buffer3.toString();  
        System.out.println(salida);  
    }  
  
    public static void main(String args[]) {  
        Cadena obj1 = new Cadena();  
        obj1.imprime();  
    }  
}
```

- ▶ El paquete java.lang contiene dos clases de cadenas: String y StringBuffer.
- ▶ Los Strings son más económicos (utilizan menos memoria) que los StringBuffers.

```
:- Bienvenida (run) x
```

```
run:
```

```
Hola Mundo
```

```
BUILD SUCCESSFUL (total time: 0 seconds)
```

String

- ▶ Cuando un método puede ser llamado en más de una forma se dice que el método está sobrecargado, por ejemplo el método substring está sobrecargado.

String substring(int beginIndex)

String substring(int beginIndex, int endIndex)

```

//compara si son iguales las cadenas
System.out.println(cadena.equals("Hola mundo"));
//imprime el carácter en la posición 6
System.out.println(cadena.charAt(6));
//regresa la primer posición en la que se encuentra el caracter o
System.out.println(cadena.indexOf('o'));
//busca el caracter o después de la posición 3
System.out.println(cadena.indexOf('o', 3));
//regresa la primer posición en la que encuentra la cadena
System.out.println(cadena.indexOf("do"));
//regresa la posición en la que encuentra la cadena después de la posición 3
System.out.println(cadena.indexOf("mi", 3));
//elimina espacios en blanco al inicio de la cadena
System.out.println(cadena.trim());
//muestra la cadena a partir de la posición 4
System.out.println(cadena.substring(4));
//crea la cadena substring a partir de la posición 3 y hasta una antes de la 7
System.out.println(cadena.substring(3, 7));
//une el string Hola al final de cadena
System.out.println(cadena.concat(" Hola"));
//reemplaza la cadena os por as
System.out.println(cadena.replace("os", "as"));
//reemplaza el caracter o por x
System.out.println(cadena.replace('o', 'x'));
//muestra true si la cadena termina con Hola de lo contrario false
System.out.println(cadena.endsWith("Hola"));
//muestra true si la cadena comienza con Hola de lo contrario false
System.out.println(cadena.startsWith("Hola"));
//muestra verdadero si la cadena inicia en la posición 5
System.out.println(cadena.startsWith("am", 5));
//muestra la longitud de la cadena
System.out.println(cadena.length());
//muestra la cadena
System.out.println(cadena.toString());
//cambia a mayúsculas
System.out.println(cadena.toUpperCase());
//cambia a minusculas
System.out.println(cadena.toLowerCase());

```

```

false
m
1
9
-1
6
Hola amigos
  amigos
a am
Hola amigos Hola
Hola amigas
Hxla amigxs
false
true
true
11
Hola amigos
HOLA AMIGOS
hola amigos

```

Considerando que la
cadena a utilizar es
“Hola amigos”

1

package bienvenida;

2

3

public class CadenaString {

4

private static String planeta;

5

6

public static void comparar(String planeta){

7

int valor;

8

CadenaString.planeta = planeta;

9

10

if(planeta.equals("TIERRA")){

11

System.out.println("Iguales con equals");

12

}

13

if (planeta.compareTo("tierra")==0){

14

System.out.println("Iguales con compareTo");

15

}

16

17

valor = planeta.compareTo("marte");

18

System.out.println(valor);

19

valor = "Saturno".compareTo(planeta);

20

System.out.println(valor);

21

}

22

23

public void imprime(){

24

System.out.println(planeta);

25

}

26

27

public static void main(String[] args) {

28

comparar("tierra");

29

comparar("TIERRA");

30

//tendría sentido solo estando desde otra clase

31

CadenaString.comparar("tierra");

32

//imprime();No puede ser referenciado desde un contexto estático

33

CadenaString obj = new CadenaString();

34

obj.imprime();

35

}

36

}

65	41h	A	97	61h	a
66	42h	B	98	62h	b
67	43h	C	99	63h	c
68	44h	D	100	64h	d
69	45h	E	101	65h	e
70	46h	F	102	66h	f
71	47h	G	103	67h	g
72	48h	H	104	68h	h
73	49h	I	105	69h	i
74	4Ah	J	106	6Ah	j
75	4Bh	K	107	6Bh	k
76	4Ch	L	108	6Ch	l
77	4Dh	M	109	6Dh	m
78	4Eh	N	110	6Eh	n
79	4Fh	O	111	6Fh	o
80	50h	P	112	70h	p
81	51h	Q	113	71h	q
82	52h	R	114	72h	r
83	53h	S	115	73h	s
84	54h	T	116	74h	t
85	55h	U	117	75h	u
86	56h	V	118	76h	v
87	57h	W	119	77h	w
88	58h	X	120	78h	x
89	59h	Y	121	79h	y
90	5Ah	Z	122	7Ah	z

Output - Bienvenida (run) ×

run:

Iguales con compareTo

7

-33

Iguales con equals

-25

-1

Iguales con compareTo

7

-33

tierra

BUILD SUCCESSFUL (total time

Clase Scanner

- ▶ La clase Scanner permite la entrada de información por:
 - ▶ Entrada de teclado.
 - ▶ Recibiendo los valores directamente en su constructor.
 - ▶ Lectura de archivo.

Clase Scanner

- Ejemplo Scanner recibiendo la información por teclado.

```
Scanner entra = new Scanner(System.in);  
System.out.print("Dame tu edad: ");  
int edad = entra.nextInt();  
System.out.println(edad);  
entra.close();
```

```
run:  
Dame tu edad: 14  
14  
BUILD SUCCESSFUL
```

Clase Scanner

- Ejemplo Scanner leyendo los argumentos recibidos en su constructor.

```
Scanner entrada = new Scanner("1 2 c 4.6 cinco cinco");
Object obj1 = entrada.nextInt();
Object obj2 = entrada.nextInt();
Object obj3 = entrada.next();
Object obj4 = entrada.nextFloat();
Object obj5 = entrada.nextLine();
```

```
System.out.println(obj1);
System.out.println(obj2);
System.out.println(obj3);
System.out.println(obj4);
System.out.println(obj5);
```

```
run:
1
2
c
4.6
cinco cinco
BUILD SUCCESSFUL
```


Clase Scanner

- Ejemplo Scanner leyendo la información contenida en un archivo.

```
Scanner entraFile=new Scanner(new File("alumnos.xls"));
while(entraFile.hasNextLine()){
    String regis = entraFile.nextLine();
    System.out.println(regis);
}
```

The screenshot displays two windows from a Java IDE. The left window, titled 'Output - EnumAlumno (run)', shows the output of a Java program. It lists student records with IDs, names, and scores, followed by a successful build message. The right window shows an Excel spreadsheet named 'alumnos.xls' with the same data.

Output - EnumAlumno (run)

```
run:
A100    Juan    34
A200    Guadalupe    18
A300    Arturo    38
A400    Miguel    18
A500    Magda    38
A600    Rosa    15
A700    Alejandro    14
A800    Victor    38
BUILD SUCCESSFUL (total time: 0 seconds)
```

	A	B	C
1	A100	Juan	34
2	A200	Guadalupe	18
3	A300	Arturo	38
4	A400	Miguel	18
5	A500	Magda	38
6	A600	Rosa	15
7	A700	Alejandro	14
8	A800	Victor	38
9			
10			
11			