



TEMA 2 – ESTRUCTURAS DE CONTROL

Patricia Cilleros

IES Villaverde

1. ÍNDICE

1. Entrada desde teclado. Salida por pantalla
2. Estructuras de control. Selección. Repetición.



1. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA

- En Java la entrada y salida se lee y escribe en flujos (*streams*); la fuente básica de entrada de datos es el teclado, mientras que la de salida es la pantalla. La clase *System* define dos referencias a objetos *static* para la gestión de entrada y salida por consola:
 - **System.in** para **entrada** por teclado.
 - **System.out** para **salida** por pantalla.
- La primera es una referencia a un objeto de la clase *BufferedInputStream* en la cual hay diversos métodos para captar caracteres tecleados; la segunda es una referencia a un objeto de la clase *PrintStream* con métodos como *print()* para salida por pantalla.



1.1. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA.

SALIDA (SYSTEM.OUT)

- El objeto *out* definido en la clase *System* se asocia con el flujo de salida, que dirige los datos a consola y permite visualizarlos en la pantalla de su equipo; por ejemplo:
`System.out.println ("Esto es una cadena");`
- Entonces se visualiza: *Esto es una cadena*.
- *System.out* es una referencia a un objeto de la clase *PrintStream*, sus siguientes métodos se utilizan con mucha frecuencia:
 - **print()** transfiere una cadena de caracteres al buffer de la pantalla.
 - **println()** transfiere una cadena de caracteres y el carácter de fin de línea al buffer de la pantalla.
 - **flush()** el buffer con las cadenas almacenadas se imprime en la pantalla.
- Con estos métodos se puede escribir cualquier cadena o dato de los tipos básicos:
`System.out.println("Viaje relampago a " + " La comarca de Las " + "Hurdes");`
- Con el operador `+` se concatenan ambas cadenas, la formada se envía al buffer de pantalla para visualizarla cuando se termine el programa o se fuerce con el método *flush()*.

1.1. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA. SALIDA (SYSTEM.OUT)

- Como argumento de ***print()*** o ***println()*** no sólo se ponen cadenas, sino también constantes o variables de los **tipos básicos**, **int**, **double**, **char**, etc; el método se encarga de **convertir a cadena** esos datos; por ejemplo:

```
int x = 500;
```

```
System.out.print(x);
```

- De igual modo, se pueden concatenar cadenas con caracteres, enteros, etc; mediante el operador + que internamente realiza la conversión:

```
double r = 2.0;
```

```
double area = Math.PI*r*r;
```

```
System.out.println("Radio = " + r + ',' + "area: " + area);
```



1.1. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA.

SALIDA (SYSTEM.OUT)

- Java utiliza secuencias de escape para visualizar caracteres no representados por símbolos tradicionales, tales como `\n`, `\t`, entre otros; también proporcionan flexibilidad en las aplicaciones mediante efectos especiales; en la tabla se muestran las secuencias de escape.

Código de Escape	Significado
'\n'	nueva línea
'\r'	retorno de carro
'\t'	tabulación
'\b'	retroceso de espacio
'\f'	avance de página
'\\'	barra inclinada inversa
'\''	comilla simple
'\"'	doble comilla

```
System.out.print("\n Error - Pulsar una  
tecla para continuar \n");
```

```
System.out.print(" Yo estoy  
preocupado\n" +  
" no por el funcionamiento \n" +  
" sino por la claridad .\n");
```

La última sentencia se visualiza como sigue debido a que la secuencia de escape `'\n'` significa nueva línea o salto de línea:

```
Yo estoy preocupado  
no por el funcionamiento  
sino por la claridad.
```



1.2. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA.

SALIDA CON FORMATO: PRINTF

- La sentencia ***System.out.printf*** tiene un primer argumento cadena, conocido como especificador de formato; el segundo argumento es el valor de salida en el formato establecido; Ejemplo:
`System.out.printf("%8.3f", x);`
- Por ejemplo:
`double precio = 25.4;`
`System.out.printf("$");`
`System.out.printf("%6.2f", precio);`
`System.out.printf(" unidad");`
- Al ejecutar este código se visualiza \$ 25.40 *unidad*. El formato especifica el tipo (f de float), el ancho total (6 posiciones) y los dígitos de precisión (2).
- `double x = 10000.0/3.0;`
`System.out.printf("%9.3f", x);`
- Al ejecutar este código, se visualiza x con un ancho de nueve posiciones y una precisión de tres dígitos: 3333,333
- Cada uno de los especificadores de formato que comienza con un carácter % se reemplaza con el argumento correspondiente; el carácter de conversión con el que terminan indica el tipo de valor a dar formato: **f es un número en coma flotante, s una cadena y d un entero decimal.**

1.2. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA. SALIDA CON FORMATO: PRINTF

- Crea una clase en java llamada *PruebaSalida.java* donde pruebes distintos formatos usando printf. Prueba el ejemplo que se propone [aquí](#) y observa la salida
- Realiza la hoja de *Ejercicios salida por pantalla* del aula virtual.



1.3. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA.

ENTRADA (SYSTEM.IN)

- La clase *System* define un objeto de la clase *BufferedInputStream* cuya referencia resulta en *in*. El objeto se asocia al flujo estándar de entrada, que por defecto es el **teclado**; los elementos básicos de **este flujo son caracteres individuales** y no cadenas como ocurre con el objeto out; entre los métodos de la clase se encuentra **read()** que devuelve el carácter actual en el buffer de entrada; por ejemplo:

```
char c;
```

```
c = System.in.read();
```

- No resulta práctico el captar la entrada carácter a carácter, es preferible hacerlo **línea a línea**; para esto, se utiliza primero la clase **InputStreamReader**, de la cual se crea un objeto inicializado con **System.in**:

```
InputStreamReader en = new InputStreamReader(System.in);
```



1.3. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA.

ENTRADA (SYSTEM.IN)

- Este objeto creado se utiliza, a su vez, como argumento para inicializar otro objeto de la clase *BufferedReader* que permite captar líneas de caracteres del teclado con el método *readLine()*:

```
String cd;
```

```
BufferedReader entrada = new BufferedReader(en);
```

```
System.out.print("Introduzca una línea por teclado: ");
```

```
cd = entrada.readLine();
```

```
System.out.println("Línea de entrada: " + cd);
```



1.3. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA.

ENTRADA (SYSTEM.IN)

- El método *readLine()* crea un objeto cadena tipo *String* con la línea ingresada; la referencia a ese objeto se asigna, en el fragmento de código anterior, a *cd* porque es una variable de tipo *String*; el objeto de la clase *BufferedReader* se puede crear con una sola sentencia, como la forma habitual en que aparece en los programas:

```
BufferedReader entrada = new BufferedReader(new  
    InputStreamReader(System.in));
```



1.3. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA. ENTRADA (SYSTEM.IN)

- Ejemplo: ¿Cuál es la salida del siguiente programa si se introducen las letras **LJ** por medio del teclado?

```
public class Letras{  
    public static void main(String[] args) {  
        char primero, ultimo;  
        System.out.printf("Introduzca su primera y última inicial:  
        ");  
        primero = (char) System.in.read();  
        ultimo = (char) System.in.read();  
        System.out.println("Hola," + primero + "." + ultimo +  
        ".!\n");  
    }  
}
```



1.4. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA. ENTRADA CON LA CLASE SCANNER

- En la versión 5.0, Java incluyó una clase para simplificar la entrada de datos por el teclado llamada ***Scanner***, que se conecta a *System.in*; para leer la entrada a la consola se debe construir primero un objeto de Scanner.

```
Scanner entrada = new Scanner(System.in);
```

- Una vez creado el objeto *Scanner*, se pueden utilizar diferentes métodos de su clase para leer la entrada: ***nextInt*** o ***nextDouble*** leen enteros o de coma flotante.

```
System.out.print("Introduzca cantidad: ");
```

```
int cantidad;
```

```
cantidad = entrada.nextInt();
```

```
System.out.print("Introduzca precio: ");
```

```
double precio = entrada.nextDouble();
```



1.4. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA.

ENTRADA CON LA CLASE SCANNER

- Cuando se llama a uno de los métodos anteriores, el programa espera hasta que el usuario teclee un número y pulsa Enter.

- El método **nextLine** lee una línea de entrada:

```
Scanner entrada = new Scanner(System.in);
```

```
System.out.print("¿ Cual es su nombre?");
```

```
String nombre;
```

```
nombre = entrada.nextLine();
```

- El método **next** se emplea cuando se desea leer una palabra sin espacios:

```
String apellido = entrada.next();
```



1.4. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA.

ENTRADA CON LA CLASE *Scanner*

- La primera línea cerca del principio del archivo indica a Java dónde encontrar la definición de la clase *Scanner*:

```
import java.util.Scanner;
```

- Esta línea significa que la clase *Scanner* está en el paquete *java.util*; *util* es la abreviatura de *utility* (*utilidad o utilidad*), la cual siempre se utiliza en código Java.
- Los métodos **nextInt**, **nextDouble** y **next** leen respectivamente un valor de tipo **int**, un valor de tipo **double** y una **palabra**.



1.4. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA.

ENTRADA CON LA CLASE SCANNER

○ Ejemplo:

```
import java.util.Scanner;

public class EntradaTest{
    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);
        // obtener la primera entrada
        System.out.print("¿ Cual es su nombre?");
        String nombre = entrada.nextLine();
        // leer la segunda entrada
        System.out.print("¿ Cual es su edad?");
        int edad = entrada.nextInt();
        // visualizar salida
        System.out.println("Buenos días " + nombre + "; años " + edad);
    }
}
```



1.4. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA. ENTRADA CON LA CLASE SCANNER

- Ejemplo: a partir de la recogida de estos datos prepara un programa que pida los datos por teclado con Scanner y después formatea la salida para mostrarlos.

```
int edad;
```

```
edad = teclado.nextInt();
```

```
double precio;
```

```
precio = teclado.nextDouble();
```

```
String rio;
```

```
rio = teclado.next();
```



1.4. ENTRADA DESDE TECLADO. SALIDA POR PANTALLA. ENTRADA CON LA CLASE SCANNER

- Realiza los ejercicios de la hoja *Ejercicios Scanner*.

