

Programación en Lenguaje Java

Tema 5. Entrada/Salida



Michael González Harbour
Mario Aldea Rivas

Departamento de Matemáticas,
Estadística y Computación

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Programación en Java

1. Introducción a los lenguajes de programación

2. Datos y expresiones

3. Estructuras algorítmicas

4. Datos Compuestos

5. *Entrada/salida*

- Entrada salida con números y textos. Mensajes. Gráficas. Menús. Dibujos. Texto multilínea.

6. Clases, referencias y objetos

7. Modularidad y abstracción

8. Herencia y polimorfismo

9. Tratamiento de errores

10. Entrada/salida con ficheros

11. Pruebas

Paquete fundamentos

La entrada/salida de texto en Java es muy general y flexible, pero compleja

- podemos usar clases que encapsulen las operaciones

El paquete `fundamentos` (no estándar) contiene las clases:

- `Lectura`: Para leer datos desde una ventana
- `Escritura`: Para escribir resultados en una ventana
- `Mensaje`: Para escribir un texto sencillo en una ventana
- `Grafica`: Para hacer gráficas de x-y
- `Menu`: Para presentar varios botones en una ventana y saber cuál es el que el usuario pulsa
- `Dibujo`: Proporciona ventana para dibujos e imágenes
- `CajaTexto`: Para leer texto multilínea

Lectura

<code>new Lectura (String t)</code>	Crea una ventana con el título t
<code>void creaEntrada(String e,double x)</code>	Crea una caja para leer un double
<code>void creaEntrada(String e, int i)</code>	Crea una caja para leer un int
<code>void creaEntrada(String e,String s)</code>	Crea una caja para leer un String
<code>void espera (String mensaje)</code> <code>void espera ()</code>	Muestra un mensaje en su caso, y espera a que el usuario teclee datos y pulse aceptar
<code>void esperaYCierra (String mensaje)</code> <code>void esperaYCierra ()</code>	Igual que espera, pero además cierra la ventana
<code>double leeDouble (String e)</code>	Lee un double de la caja de etiqueta e
<code>int leeInt (String e)</code>	Lee un int de la caja con etiqueta e
<code>String leeString (String e)</code>	Lee un String de la caja de etiqueta e
<code>void println (String mensaje)</code>	Muestra un mensaje en la cabecera

Ejemplo

Hacer un programa para calcular la distancia entre dos puntos del globo, y que acepte datos de entrada con **Lectura**:

```
import fundamentos.*;
public class Dist {
    public static void main(String[] args) {
        double dist; // Kilometros
        double lon1,lat1,lon2,lat2; // grados

        // Paso 1: crear el objeto
        Lectura pantalla = new Lectura("Círculo Máximo");
        // Paso 2: crear las entradas para los datos a leer
        pantalla.creaEntrada("Latitud 1",0.0);
        pantalla.creaEntrada("Longitud 1",0.0);
        pantalla.creaEntrada("Latitud 2",0.0);
        pantalla.creaEntrada("Longitud 2",0.0);
```

Ejemplo (cont.)

```
// Paso 3: esperar a que el usuario teclee
pantalla.espera
    ("Introduce coordenadas y pulsa Aceptar");
// Paso 4: leer los datos tecleados
lat1=pantalla.leeDouble("Latitud 1");
lon1=pantalla.leeDouble("Longitud 1");
lat2=pantalla.leeDouble("Latitud 2");
lon2=pantalla.leeDouble("Longitud 2");

// Trabajar con los datos leídos
lat1=Math.toRadians(lat1);
lat2=Math.toRadians(lat2);
lon1=Math.toRadians(lon1);
lon2=Math.toRadians(lon2);
```

Ejemplo (cont.)

```
// Calcular la distancia y mostrarla en pantalla
dist=Math.toDegrees(Math.acos(Math.sin(lat1)*
    Math.sin(lat2)+
    Math.cos(lat1)*Math.cos(lat2)*
    Math.cos(lon1-lon2)))*
    60.0*1.852;
pantalla.println("La distancia es: "+dist+" Km");
}
```

Escritura

<code>new Escritura (String t)</code>	Crea una ventana con su título
<code>void insertaValor (String e, double x)</code>	Crea una caja con la etiqueta y el valor
<code>void insertaValor(String e, int)</code>	Crea una caja con la etiqueta y el valor
<code>void insertaValor(String e, String s)</code>	Crea una caja con la etiqueta y el valor
<code>void espera ()</code>	Espera a que el usuario pulse aceptar, y cierra la ventana

En el ejemplo anterior, podríamos cambiar la última línea por:

```
Escritura res = new Escritura ("Círculo Máximo");  
res.insertaValor("Distancia (Km)", dist);  
res.espera();
```


Mensaje

<code>new Mensaje ()</code>	Constructor que crea la ventana
<code>new Mensaje (String t)</code>	Constructor que crea la ventana y le pone título
<code>void escribe(String m)</code>	Muestra el mensaje y espera a que se pulse el botón OK

Ejemplo:

```
Mensaje mens = new Mensaje();  
mens.escribe("Esto es un mensaje");
```

Grafica

<code>new Grafica ()</code>	Constructor que crea la ventana con títulos en blanco
<code>new Grafica (String titulo, String tituloX, String tituloY)</code>	Constructor alternativo que pone los títulos de la ventana y los ejes X e Y
<code>void inserta (double x, double y)</code>	Inserta el punto (x,y) en la gráfica actual
<code>void otraGrafica ()</code>	Crea una nueva gráfica en la misma ventana
<code>void pinta()</code>	Pinta todas las gráficas
<code>void ponLineas (boolean b)</code>	Pone o quita la opción de mostrar la gráfica con líneas entre cada punto
<code>void ponSimbolo (boolean b)</code>	Pone o quita la opción de símbolos en la gráfica actual
<code>void ponTitulo (String t)</code>	Pone el título de la gráfica actual
<code>void ponColor (int c)</code>	Pone el color de la gráfica actual

Ejemplo de creación de una gráfica

```
Grafica g = new Grafica ("Valores", "x", "y");  
// El primer grafico  
g.ponSimbolo(true);  
g.ponColor(Grafica.azul);  
g.ponTitulo("graf1");  
g.inserta(x1,y1);  
g.inserta(x2,y2);  
...  
// El segundo grafico  
g.otraGrafica();  
g.ponColor(Grafica.rojo);  
g.ponTitulo("graf2");  
g.inserta(p1,q1);  
g.inserta(p2,q2);  
...  
// Pintar las dos gráficas  
g.pinta();
```

Menu

<code>new Menu (String t)</code>	Constructor que crea la ventana con su título
<code>void insertaOpcion (String e, int code)</code>	Crea un botón con la etiqueta y el código numérico indicados
<code>int leeOpcion ()</code>	Espera a que el usuario pulse un botón, y retorna el código numérico del botón pulsado
<code>int leeOpcion (String e)</code>	Muestra un mensaje, espera a que el usuario pulse un botón, y retorna el código numérico del botón pulsado
<code>void println (String s)</code>	Muestra un mensaje en la cabecera

Ejemplo con Menu

```
Menu menu= new Menu("Prueba de Menu");
int op;

// prepara la lista de opciones
menu.insertaOpcion("Insertar libro",1);
menu.insertaOpcion("Borrar libro",2);
menu.insertaOpcion("Prestar libro",3);
menu.insertaOpcion("Consultar libro",4);
menu.insertaOpcion("Salir",5);

// espera a que el usuario elija una opción
op=menu.leeOpcion("Elige una opción");
```

Dibujo

<code>new Dibujo (String titulo)</code>	Constructor que crea la ventana con su título y tamaño 640x480
<code>new Dibujo (String titulo, int alto, int ancho)</code>	Constructor que crea la ventana con su título y el tamaño indicado
<code>void espera()</code>	Pinta el dibujo y espera que se pulse OK
<code>void pinta()</code>	Hace el dibujo de forma inmediata, sin esperar
<code>void ponColorLapiz (ColorFig color)</code>	Pone el color del lápiz con el que se pinta.
<code>void ponRelleno (ColorFig color)</code>	Pone el color del relleno de las figuras cerradas
<code>void ponLetra(int tamaño)</code>	Pone el tamaño de la letra del texto dibujado
<code>void hazLienzoSinLimite()</code>	Hace que si se dibuja fuera del lienzo no salgan mensajes de error
<code>void borra()</code>	Borra el dibujo poniendo el fondo gris claro
<code>void borra(ColorFig color)</code>	Borra el dibujo poniendo el fondo del color indicado

Dibujo (cont.)

```
void dibujaTexto(String texto,
    int xOrigen,int yOrigen)
void dibujaPunto(int x,int y)
void dibujaLinea(int xOrigen,int yOrigen,
    int xDestino,int yDestino)
void dibujaLineas (int[] x,int[] y);
void dibujaRectangulo(int x1,int y1,
    int x2, int y2)
void dibujaElipse(int x1,int y1,
    int x2, int y2)
void dibujaArco(int x1,int y1,
    int x2, int y2,int angulo1,int angulo2);
void dibujaPoligono(int[] x, int[] y)
void dibujaImagen(int xOrigen,int yOrigen,
    String nombreFichero)
```

Operaciones de
dibujar

Colores disponibles

<code>ColorFig.negro</code> <code>ColorFig.azul</code> <code>ColorFig.grisOscuro</code> <code>ColorFig.gris</code> <code>ColorFig.verde</code> <code>ColorFig.grisClaro</code> <code>ColorFig.magenta</code> <code>ColorFig.naranja</code> <code>ColorFig.rosa</code> <code>ColorFig.rojo</code> <code>ColorFig.blanco</code> <code>ColorFig.amarillo</code>	Constantes estáticas de la clase <code>ColorFig</code>
---	---

Ejemplo de Dibujo

```
Dibujo dib=new Dibujo
    ("Prueba de dibujo",480,300);

dib.borra(ColorFig.blanco);
dib.ponGrosorLapiz(4);
dib.ponColorLapiz(ColorFig.magenta);

dib.dibujaElipse(50,100,90,140);
dib.dibujaArco(50,200,90,240,-45,45);

dib.dibujaImagen(250,30,"mi-foto.jpg");

dib.espera();
//necesario para que se pinte la ventana
```

CajaTexto

<code>new CajaTexto (String s, int filas, int columnas)</code>	Crea una ventana con el título <code>s</code> , y el tamaño en caracteres indicado por <code>filas</code> y <code>columnas</code>
<code>void espera ()</code>	Espera a teclear datos y pulsar aceptar; luego se coloca en la primera línea del texto
<code>boolean hayMas()</code>	Retorna un booleano indicando si hay mas líneas por leer o no
<code>double leeDouble ()</code>	Lee un <code>double</code> de la línea actual
<code>int leeInt ()</code>	Lee un <code>int</code> de la línea actual
<code>String leeString ()</code>	Lee un <code>String</code> de la línea actual
<code>void avanzaLinea ()</code>	Avanza a la siguiente línea

Ejemplo con una caja de texto

Este fragmento de código muestra en pantalla las líneas tecleadas en una caja de texto

```
// crea el objeto
CajaTexto caja = new CajaTexto ("Prueba de Caja de
Texto", 20, 50);

// espera a que el usuario teclee texto
caja.espera();

// lazo para mostrar todas las líneas tecleadas
while (caja.hayMas()) {
    System.out.println(caja.leeString());
    caja.avanzaLinea();
}
```

Otros métodos de la clase CajaTexto

<code>void esperaYCierra ()</code>	Igual que espera, y además cierra la ventana
<code>void reinicia ()</code>	Se coloca en la primera línea del texto
<code>double leeDouble (int pos)</code>	Lee el <code>double</code> que ocupa la posición <code>pos</code> de la línea actual; se separan con espacios en blanco; el primer numero es el de <code>pos=0</code>)
<code>int leeInt (int pos)</code>	Lee el <code>int</code> que ocupa la posición <code>pos</code> de la línea actual
<code>void borra()</code>	Borra el texto de la caja de texto
<code>void println (String s)</code>	Añade una línea con el <code>String s</code> a la caja de texto

Ejemplo: Lectura de datos en 3 columnas

```
caja = new CajaTexto ("Lectura en columnas", 20, 50);

caja.esperaYCierra();
int fila=0;
while (caja.hayMas()) {
    // leer datos de las tres columnas
    //y ponerlos en pantalla
    for (int col=0; col<3; col++) {
        System.out.println("Fila:" + fila + " Col: " + col +
            " =" + caja.leeDouble(col));
    }
    caja.avanzaLinea();
    fila++;
}
```