

# PROGRAMACIÓN EN JAVA

---

## TEMA ANEXO 2: Clase para la entrada de datos en JAVA



## Bibliografía

---

- “JAVA 2 Lenguaje y aplicaciones”  
Fco. Javier Ceballos. Ed. Ra-Ma, 2006

# Contenido

---

- **Clase Leer**
- **Clase Leer – Ejemplo de utilización**

# Clase Leer

---

- El objetivo de esta clase es incluir como miembros un sin número de métodos, de manera que todos juntos nos proporcionen una interfaz que cualquier programa puede utilizar para obtener del teclado datos de cualquier tipo primitivo
- El código que define esta clase es el siguiente:

```
import java.io.*;
```

```
public class Leer {
```

```
    public static String dato() {
```

```
        String sdato = "";
```

```
        try {
```

```
            // Definir un flujo de caracteres de entrada: flujoE
```

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

```
            BufferedReader flujoE = new BufferedReader(isr);
```

```
            // Leer. La entrada finaliza al pulsar la tecla Entrar
```

```
            sdato = flujoE.readLine();
```

```
        }
```

```
        catch(IOException e) {
```

```
            System.err.println("Error: " + e.getMessage());
```

```
        }
```

```
        return sdato; // devolver el dato tecleado
```

```
    }
```

```
    public static short datoShort() {
```

```
        try {
```

```
            return Short.parseShort(dato());
```

```
        }
```

```
        catch(NumberFormatException e) {
```

```
            return Short.MIN_VALUE; // valor más pequeño
```

```
        }
```

```
    }
```

```
    public static int datoInt() {
```

```
        try {
```

```
            return Integer.parseInt(dato());
```

```
        }
```

```
        catch(NumberFormatException e) {
```

```
            return Integer.MIN_VALUE; // valor más pequeño
```

```
        }
```

```
    }
```

## Clase Leer (cont.)

```
    public static long datoLong() {
```

```
        try {
```

```
            return Long.parseLong(dato());
```

```
        }
```

```
        catch(NumberFormatException e) {
```

```
            return Long.MIN_VALUE; // valor más pequeño
```

```
        }
```

```
    }
```

```
    public static float datoFloat() {
```

```
        try {
```

```
            return Float.parseFloat(dato());
```

```
        }
```

```
        catch(NumberFormatException e) {
```

```
            return Float.NaN; // No es un Número; valor float.
```

```
        }
```

```
    }
```

```
    public static double datoDouble() {
```

```
        try {
```

```
            return Double.parseDouble(dato());
```

```
        }
```

```
        catch(NumberFormatException e) {
```

```
            return Double.NaN; // No es un Número; valor double.
```

```
        }
```

```
    }
```

```
}//Llave de cierre de la clase
```

**NOTA:** Código fuente (*Leer.java*) dentro de los archivos adjuntos del documento

## Clase Leer (cont.)

---

- En la clase *Leer*, se puede observar que todos los métodos, además de públicos, se han declarados como *static* con el fin de que estos puedan ser invocados allí donde se necesiten, sin necesidad de que exista un objeto de la clase. Hay que recordar que la sintaxis para invocar a un método *static* de una clase es:

`nombreClase.nombreMetodo`

- Una vez escrita la clase *Leer*, podemos utilizarla como soporte para otros programas. Debemos recordar que para que la clase aplicación que vamos a escribir pueda utilizar la clase *Leer*, deben estar ambas en la misma carpeta de trabajo

# Clase Leer – Ejemplo de utilización

Programa que utiliza la clase Leer que debe de estar almacenada en la misma carpeta de trabajo

```
public class LeerDatos
{
    public static void main(String[] args)
    {
        short dato_short = 0;
        int dato_int = 0;
        long dato_long = 0;
        float dato_float = 0;
        double dato_double = 0;

        System.out.print("Dato short: ");
        dato_short = Leer.datoShort();
        System.out.print("Dato int: ");
        dato_int = Leer.datoInt();
        System.out.print("Dato long: ");
        dato_long = Leer.datoLong();
        System.out.print("Dato float: ");
        dato_float = Leer.datoFloat();
        System.out.print("Dato double: ");
        dato_double = Leer.datoDouble();

        System.out.println(dato_short);
        System.out.println(dato_int);
        System.out.println(dato_long);
        System.out.println(dato_float);
        System.out.println(dato_double);
    }
}
```

**NOTA:** Código fuente (*LeerDatos.java*) dentro de los archivos adjuntos del documento