

# Clase String y lectura de datos

## Utilidades y elementos del JDK



# Declaración y creación de objetos

---

## □ Declaración:

```
nombre_clase  identificador_objeto;  
String palabra;
```

## □ Creación:

- Los objetos, se suelen crear con: new

```
identificador_objeto = new nombre_clase(argumentos);  
palabra = new String("Matemáticas Aplicadas");
```

## □ Uso de métodos:

```
identificador_objeto.nombre_metodo(argumentos);  
Palabra.length();
```

# Clase `String`: Cadena de caracteres

---

## □ Constantes

- Caracteres entre comillas dobles “a”
- Cadena vacía “”
- Se pueden colocar secuencias de escape

`“\t Uno \t 1 \n \t Dos \t 2 \n”`

- Se puede usar sin el comando `new`

```
String palabra;
```

```
palabra = new String("Ingenieria");
```

```
String palabra2 = "Hola";
```

**'A' ≠ "A"**

# Concatenando

---

- Única referencia que acepta la sobrecarga de un operador, el `+` y `+=`
  - Concatenar
    - `cadena1 += "bc";`
  - Si hay otro tipo de dato y un `String` se convierte a `String` automáticamente
    - `cadena2 = "bc" + 5; => "bc5"`
- E/S de datos
  - La mayoría de las instrucciones para escribir datos en pantalla y leer desde teclado asume que se están usando datos `String`

# Algunos métodos de la clase String

```
String ej = "Ingenieria";
```

<code>ej.length()</code>	10
<code>ej.charAt(4)</code>	n
<code>ej.substring(5)</code>	ieria
<code>ej.substring(2,5)</code>	gen
<code>ej.indexOf("e")</code>	3
<code>ej.indexOf("en")</code>	3
<code>ej.lastIndexOf('e')</code>	6
<code>ej.compareTo("Actuaria")</code>	8
<code>ej.compareTo("Psicologia")</code>	-7
<code>ej.equals("ingenieria")</code>	false
<code>ej.equalsIgnoreCase("ingenieria")</code>	true
<code>ej.toUpperCase()</code>	INGENIERIA
<code>ej.toLowerCase()</code>	ingenieria
<code>ej.toCharArray()</code>	{ 'I', 'n', 'g', 'e', 'n', 'i', 'e', 'r', 'i', 'a' }
<code>ej.split("e")</code>	{ "Ing", "ni", "ria" }
<code>ej.split("\\.")</code>	{ "Ingenieria" }

# Ejercicio: Relaciona el tipo de dato con el método que lo regresa como resultado

---

int

boolean

char

char[]

String

String[]

split("i")

length()

toCharArray()

equals("Hola")

charAt(3)

substring(3,6)

indexOf('A')

compareTo("Pepe")

toUpperCase()

# Conversión de `String` a Número

---

## □ Convertir de `String` al tipo de dato numérico

- `Byte.parseByte(String)`
- `Short.parseShort(String)`
- `Integer.parseInt(String)`
- `Float.parseFloat(String)`
- `Double.parseDouble(String)`

```
float f = Float.parseFloat("22.21");
```

# Ejemplo: Algunos métodos String

```
public class PruebaCadenas {  
    public static void main(String[] args) {  
        String frase = "Solo se que no se nada";  
        String respuesta = "SI";  
        String dato = "196";  
        String palabra;  
        char [] digi = new char[dato.length()];  
        int numero;  
        palabra= frase.substring(12,14);  
        System.out.println(palabra.equals(respuesta));  
        numero = Integer.parseInt(dato)/frase.lastIndexOf("e");  
        System.out.println(numero);  
        digi= dato.toCharArray();  
        System.out.println(digi[0]+"+"+digi[1]+"+"+digi[2]);  
    }  
}
```

run:

false

12

1+9+6

Carmen Villar / René Martínez

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



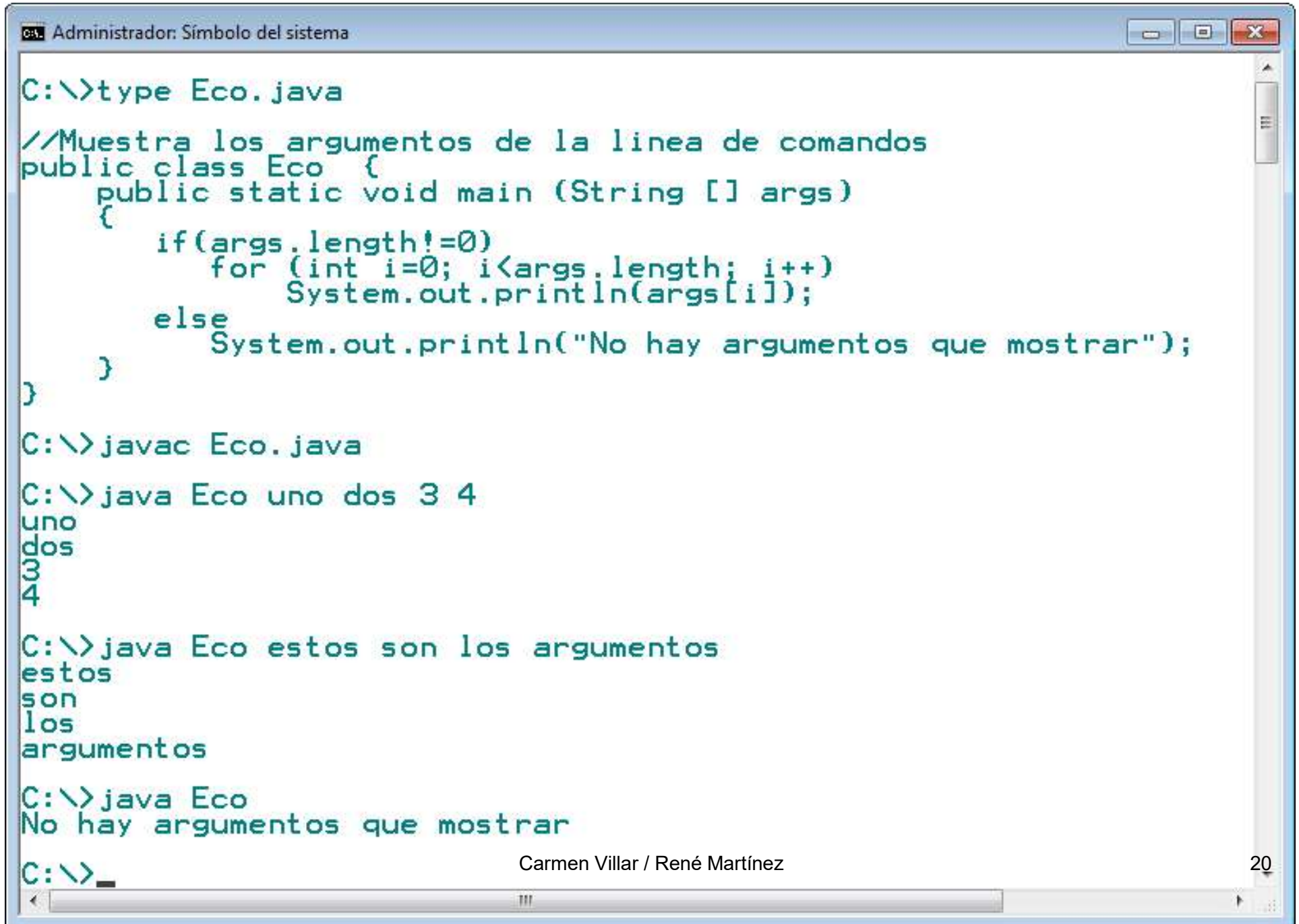
# Ejercicio

---

Identifica si la expresión subrayada es una clase (C), referencia a objetos (O), tipo de dato primitivo (P), variable (V) o un método (M). Indica la salida del siguiente código

```
public static void main(String[] args) {  
    String curso = "Lenguajes Orientados a  
Objetos";  
    String numero = "123.45";  
    String palabra;  
    int entero;  
    System.out.println(curso.charAt(4));  
    System.out.println(curso.length());  
    palabra= curso.substring(23).toUpperCase();  
    System.out.println(palabra);  
    entero = Integer.parseInt(  
        numero.substring(0,numero.indexOf(".")));  
    System.out.println(entero);  
}
```

# Ejemplo: Argumentos línea comandos



```
CA. Administrador: Símbolo del sistema

C:\>type Eco.java
//Muestra los argumentos de la linea de comandos
public class Eco {
    public static void main (String [] args)
    {
        if(args.length!=0)
            for (int i=0; i<args.length; i++)
                System.out.println(args[i]);
        else
            System.out.println("No hay argumentos que mostrar");
    }
}

C:\>javac Eco.java

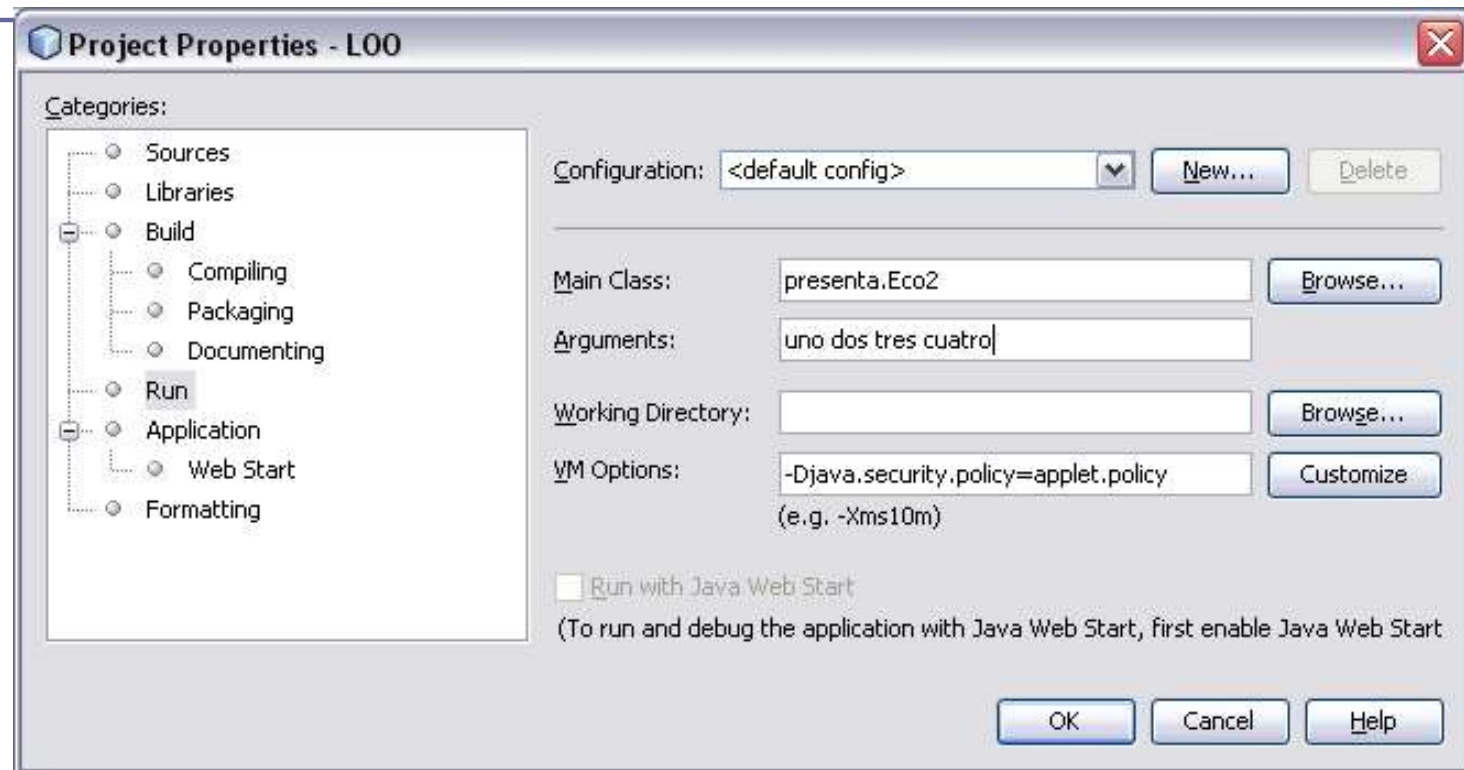
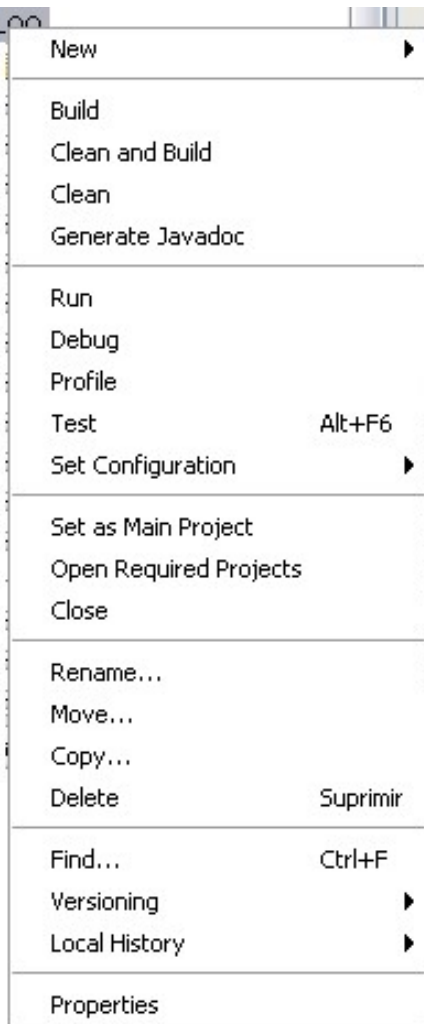
C:\>java Eco uno dos 3 4
uno
dos
3
4

C:\>java Eco estos son los argumentos
estos
son
los
argumentos

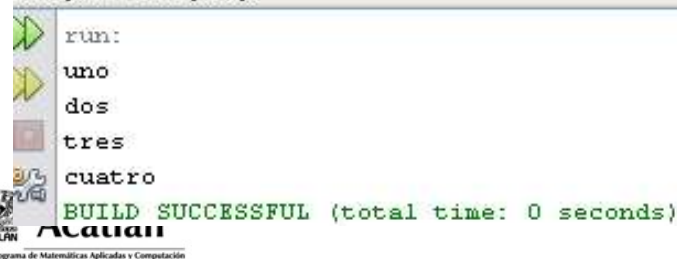
C:\>java Eco
No hay argumentos que mostrar

C:\>
```

# Argumentos en Netbeans



## Output - L00 (run)



```
public class Eco2 {  
    // Muestra los argumentos de la linea de comandos  
    public static void main (String [] args) {  
        if (args.length!=0) {  
            for(String i: args){  
                System.out.println(i);  
            }  
        }else{  
            System.out.println("No hay argumentos que mostrar");  
        }  
    }  
}
```

# Paquetes(package)

---

## □ Agrupan clases e interfaces relacionadas

- Indica al compilador donde encontrar las clases
- Se escribe al principio del archivo fuente

```
package <top_pkg_name>[.<sub_pkg_name>] *;
```

## □ Paquetes predefinidos más importantes

- `java.lang`
  - `System, String, Math, ...`
- `java.io`
  - `BufferedReader, InputStreamReader, FileInputStream, ...`
- `java.util`
  - `Date, Random, StringTokenizer, Scanner ...`

# Paquetes(package) (cont)

---

- `java.applet`
- `java.awt`

# Usando paquetes

---

## □ Nombre paquetes

### ■ dueño.paquete.clase

- dueño, sigue sintaxis de dominios internet, pero en orden inverso: com.ibm.ejs.client

## □ Importar paquetes

### ■ Para usar los métodos sin usar todo el nombre

```
import <pkg_name>[.<sub_pkg_name>]*.<class_name>;  
import <pkg_name>[.<sub_pkg_name>]*.*;
```

### ■ Ejemplos:

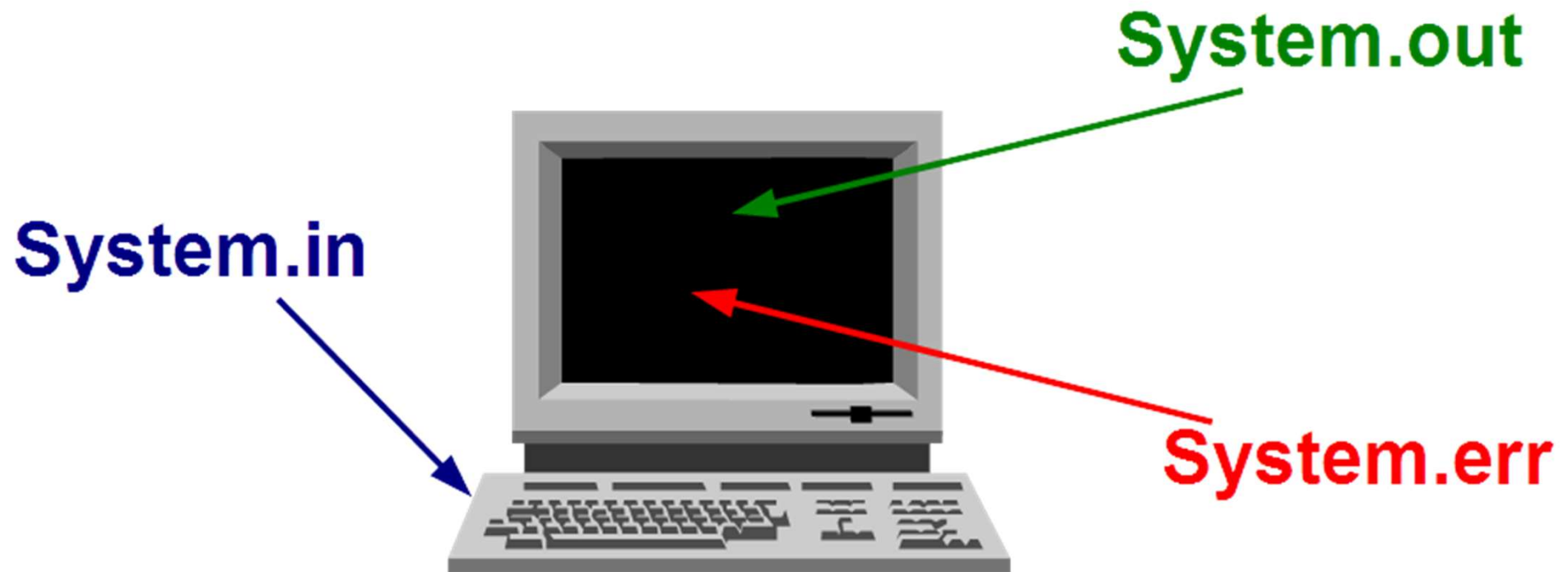
- `import java.util.Random;`
- `import java.awt.*;`

## □ El paquete `java.lang`, siempre es importado por el compilador.

# Operaciones Entrada/Salida

---

- Entrada estándar: teclado
- Salida estándar: pantalla
- Salida estándar errores: pantalla



# Clase Scanner: Entrada de datos

## □ Pertenece al paquete java.útil. Uso:

1. Importar paquete
2. Declarar un objeto de tipo Scanner
3. Crear el objeto enviando como parámetro System.in
4. Usar un método *nexttipo* para leer el dato y asignar a su variable

next()  
nextLine()  
nextByte()  
nextFloat()  
nextInt()  
nextLong()  
nextShort()

```
import java.util.* ; ①
public class LeeNumero {
    public static void main(String[] args) {
        ② Scanner lector = new Scanner(System.in); ③
        int numero;
        System.out.print("Introduce un numero: ");
        ④ numero = lector.nextInt();
        System.out.print("Numero: "+numero);
    }
}
```



# Ejemplo entrada

- Los delimitadores de campos por defecto son
  - Espacio en blanco y el <ENTER>

```
import java.util.*;
public class LeeNumeros {
    public static void main(String[] args) {
        Scanner escaner = new Scanner(System.in);
        int numerol, numero2;
        System.out.print("Introduce dos numeros: ");
        numerol=escaner.nextInt();
        numero2=escaner.nextInt();
        System.out.println("Suma: "+(numerol+numero2));
    }
}
```

```
run:
Introduce dos numeros: 3 4
Suma: 7
```

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

```
run:
Introduce dos numeros: 3
4
Suma: 7
```

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

```
Introduce dos numeros:
3
4
Suma: 7
```

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

# Leyendo cadenas

```
import java.util.Scanner;
public class LeeCadenas {
    public static void main(String[] args) {
        Scanner lector = new Scanner(System.in);
        String frase, autor;
        System.out.print("Introduce la frase: ");
        frase=lector.nextLine();
        System.out.print("Autor: ");
        autor=lector.next();
        System.out.println("La frase de "+autor+" tiene "+
                           frase.length()+" caracteres");
    }
}
```

run:

```
Introduce la frase: Donde hay soberbia habra ignorancia; mas donde hay humildad habra sabiduria
Autor: Salomon
La frase de Salomon tiene 75 caracteres
BUILD SUCCESSFUL (total time: 7 seconds)
```

run:

```
Introduce la frase: No se le puede dar la mano a quien tiene el puño cerrado
Autor: Indira Gandhi
La frase de Indira tiene 56 caracteres
BUILD SUCCESSFUL (total time: 28 seconds)
```