

# FUNDAMENTOS DE PROGRAMACIÓN

## USO DEL ENTORNO ECLIPSE PARA JAVA

### 1- CONCEPTOS

#### Estructura de los directorios

Antes de comenzar, describiremos brevemente la estructura de directorios que vamos a utilizar durante todas las prácticas. Utilizar una estructura de directorios clara es importante cuando un proyecto utiliza varios ficheros de distintos tipos. Durante la elaboración de la práctica, manejaremos los siguientes directorios:

- **src**: directorio con el código fuente de la práctica (ficheros con extensión .java). Es el único directorio que contiene ficheros elaborados durante las prácticas. Los otros directorios los genera Eclipse automáticamente
- **bin**: directorio donde aparecerán los ficheros resultados de la compilación del código fuente anterior. Lo crea el IDE (el entorno de desarrollo, esto es, Eclipse) automáticamente.

#### Workspace

Antes de empezar a trabajar en la plataforma de Eclipse, debemos aprender el significado de *workspace*.

1. Un espacio de trabajo (*workspace*) es una carpeta donde estarán las carpetas correspondientes a los proyectos Java creados con Eclipse.
2. Una instancia en ejecución de Eclipse sólo puede trabajar con un workspace.
3. Cambiar el workspace requiere reiniciar Eclipse.

#### Proyecto

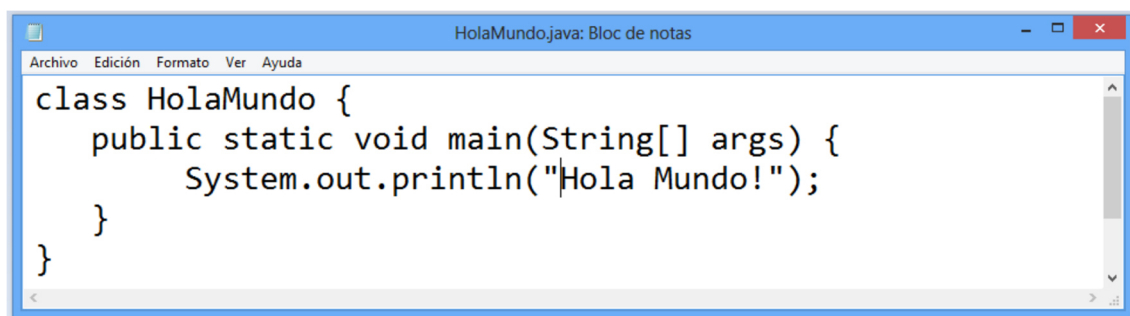
Un *proyecto* es una carpeta que se crea en el workspace. Cada práctica será un proyecto nuevo. Dentro de un proyecto nos encontraremos las siguientes carpetas:

- **src** Carpeta donde estará el código fuente (source code), es decir, los ficheros .java.
- **bin** Carpeta donde estarán los ficheros compilados, que tienen extensión .class y contienen los *bytecodes* interpretables por la máquina virtual de Java.

### 2- NUESTRO PRIMER PROGRAMA

A continuación haremos un primer ejemplo en la plataforma de trabajo Eclipse. Está será nuestra práctica 0, que será escribir "Hola Mundo".

El código es el siguiente:

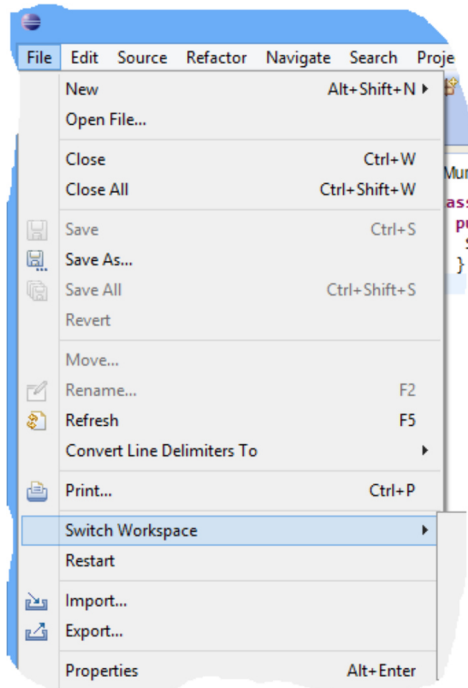


```
class HolaMundo {  
    public static void main(String[] args) {  
        System.out.println("Hola Mundo!");  
    }  
}
```

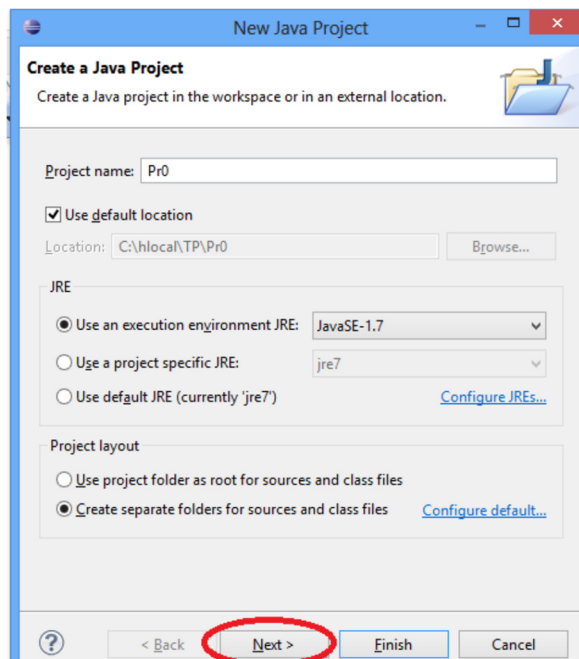
**Paso 1.** Desde el sistema operativo crear una carpeta que será nuestro espacio de trabajo (Workspace)

**Paso 2:** Ejecutar Eclipse. No se instala, se ejecuta cada vez que queremos usarlo.

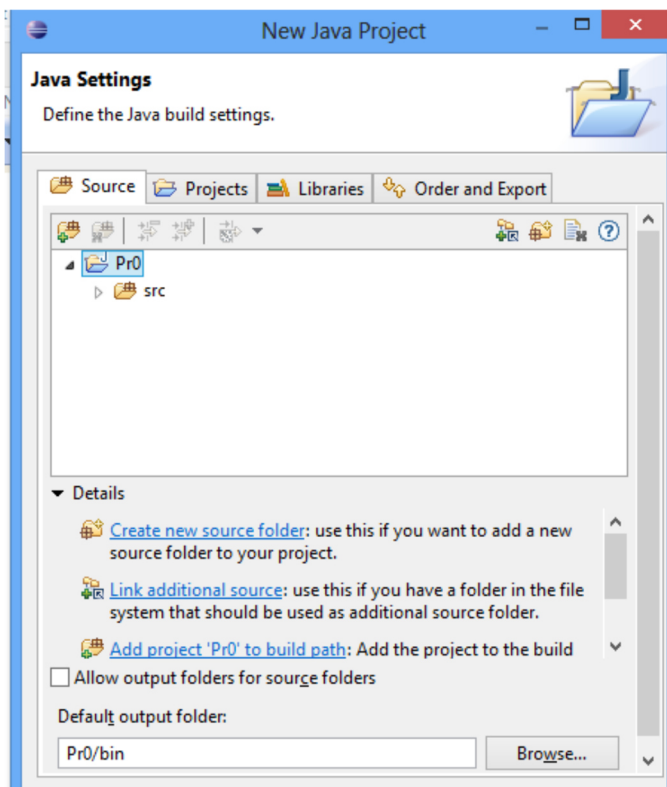
**Paso 3.** Cuando ya esté abierto Eclipse, cambiar al workspace deseado desde el menú **File → Switch Workspace** de Eclipse.



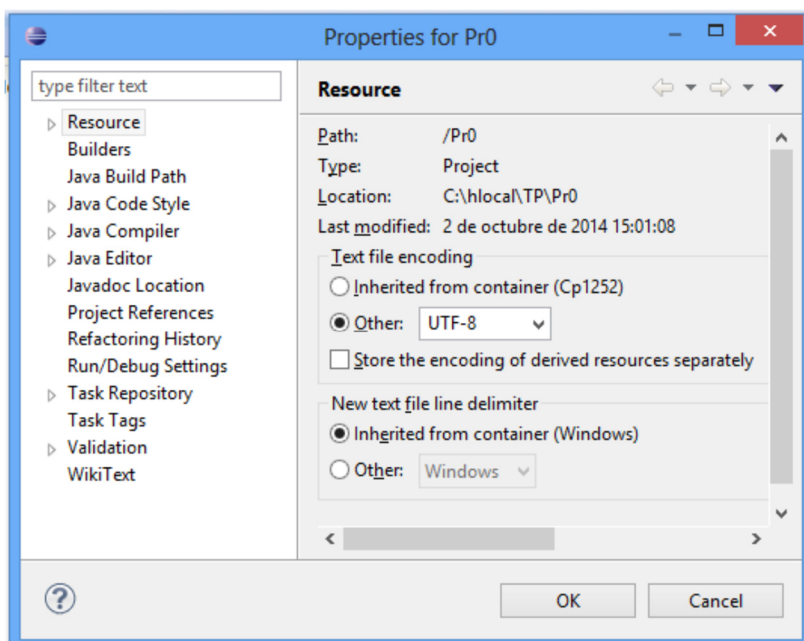
**Paso 4:** Crear un proyecto: **File → New → Java Project**. Recuerda que cada programa será un proyecto nuevo.



**Paso 5:** La creación de un proyecto requiere indicar las carpetas **src** y **bin**.

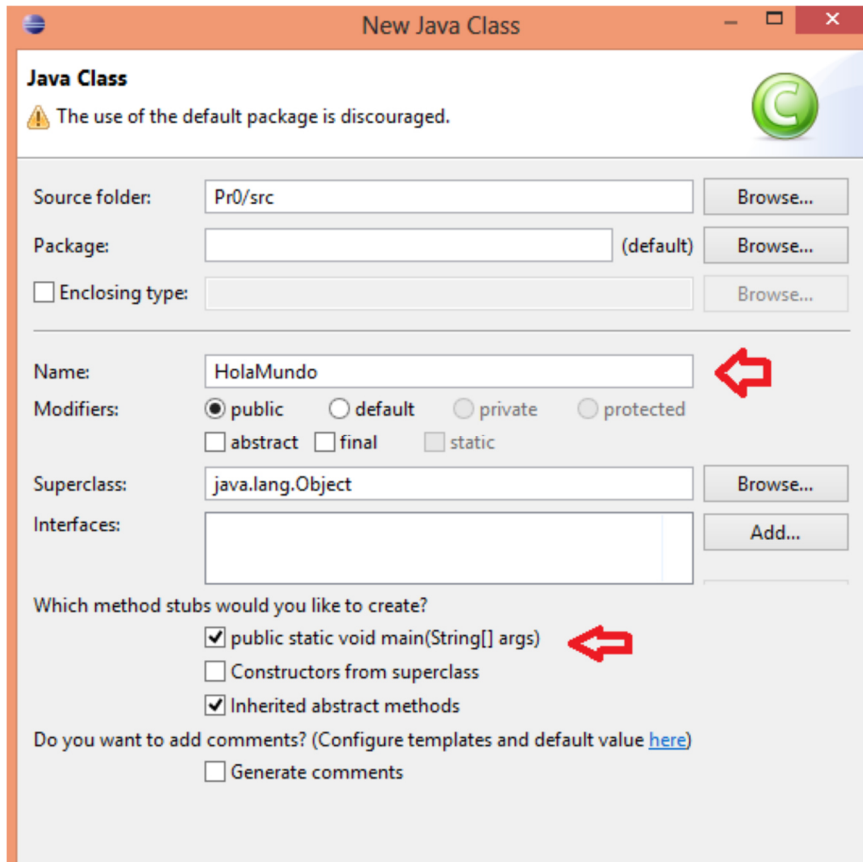


**Paso 6 (opcional) :** Para evitar problemas de compatibilidad entre plataformas e idiomas debemos configurar el proyecto para utilizar la codificación de caracteres **UTF-8**. Selecciona el proyecto y pulsa el botón derecho seleccionando la opción **Properties**.  
Elige el menú **Resource** y modifica **Text File Encoding** seleccionando **Other** e indicando **UTF-8**.

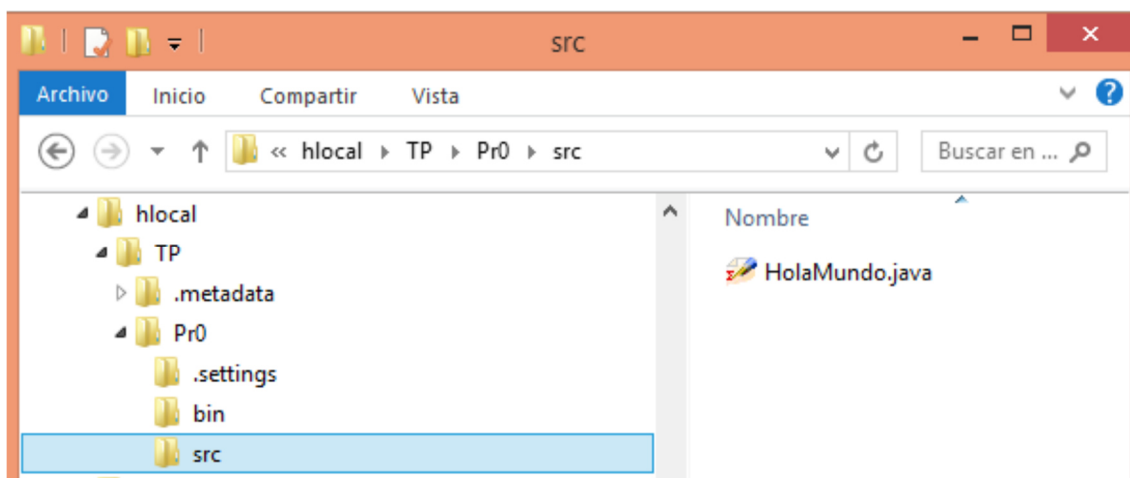


**Paso 7:** Crear una clase llamada “HolaMundo”: desde el menú **File** → **new Class**. Marcar la casilla:

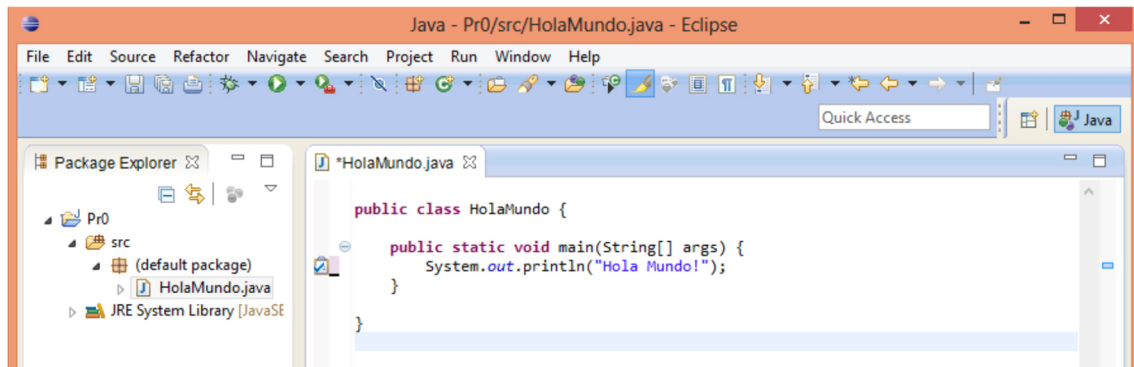
☐ public static void main (String[] args)



**Paso 8:** puedes comprobar que en tu directorio del workspace ha creado una carpeta Pr0 y dentro las carpetas src y bin.

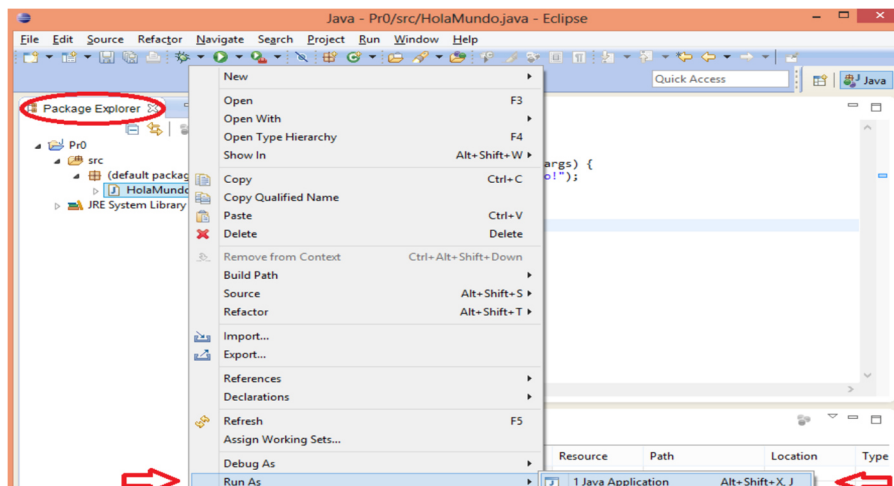


**Paso 9:** Escribir el código de la clase “HolaMundo”.

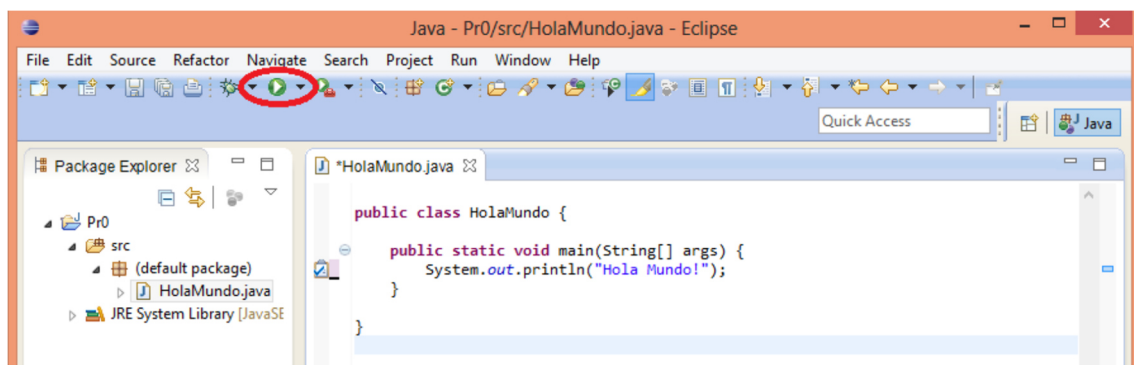


**Paso 10:** Para ejecutar la aplicación, Eclipse ofrece distintas formas. Una de ellas es:  
Opción 1: Seleccionar la clase HolaMundo.java en la pestaña *Package Explorer* (situado a la izquierda). Pulsar el botón derecho.

Seleccionar **Run As** → **Java Application**.



Opción 2: Otra alternativa es pulsar el icono verde de “Run” después de seleccionar la clase HolaMundo.java en la pestaña *Package Explorer*.



**Paso 11:** Eclipse ha compilado nuestro fichero de código fuente y el resultado de esa compilación ha sido un fichero con el mismo nombre y extensión .class. Podemos ver que en la carpeta C:\Workspace\Pr0\bin\ tenemos el fichero HolaMundo.class.

## 2- NUESTRO SEGUNDO PROGRAMA

A continuación haremos un segundo ejemplo en la plataforma de trabajo Eclipse incorporando las instrucciones (leer, escribir y condicional) que hemos visto.

Esta será la práctica 1: "se introducen dos números enteros por teclado mostrar el valor del mayor"

**Paso 1:** Crea un nuevo proyecto Pr1.

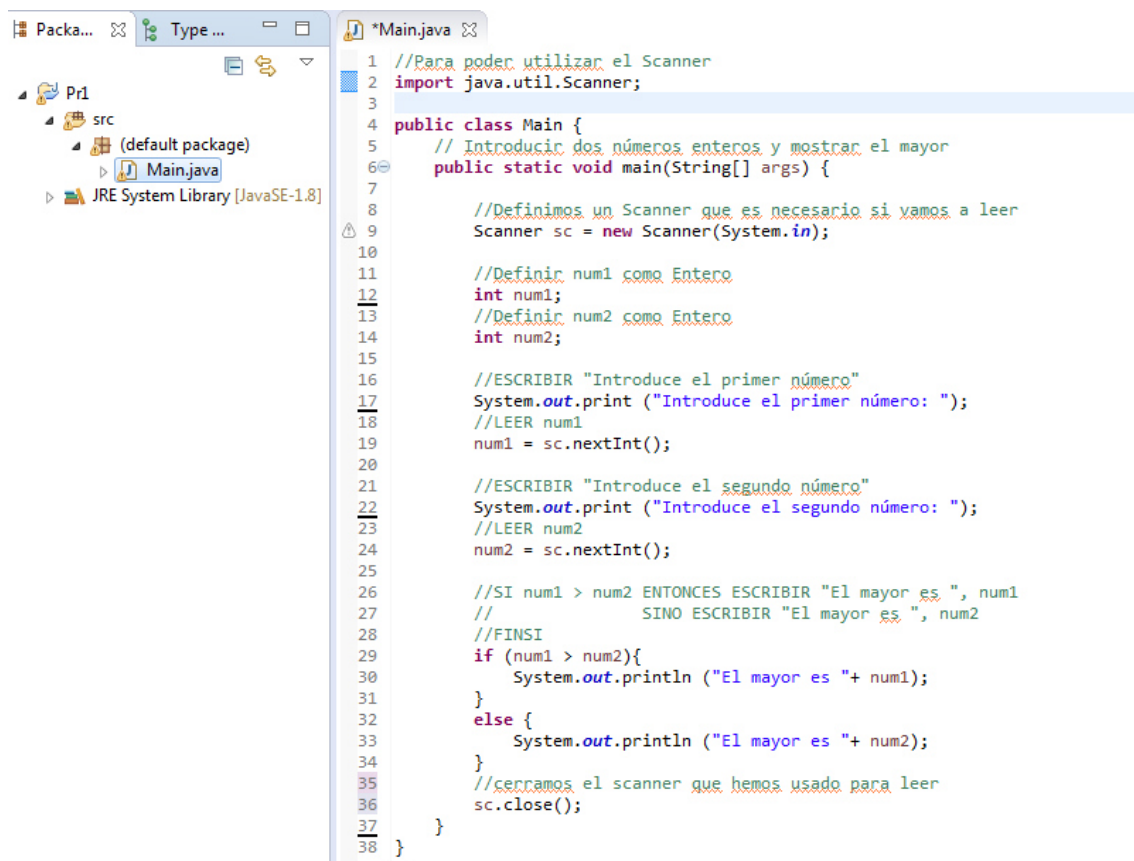
**Paso 2:** Crea el programa Main como se ha indicado antes.

**Paso 3:** Copia el código como aparece a continuación.

- Lo que empieza por // y está en verde son comentarios con el código en pseudocódigo
- Cada instrucción debe terminar en ;
- Fíjate en los márgenes y sigue la misma estructura
- Las líneas en blanco se eliminan puedes poner las que quieras.
- Las llaves delimitan los bloques, debe haber tantas de apertura como de cierre. Fíjate dónde se cierran, según dónde se han abierto.
- Las palabras en morado son palabras reservadas, siempre en minúsculas

**Paso 4:** Ejecútalo como se ha indicado antes.

Ejecútalo varias veces con diferentes valores



```
1 //Para poder utilizar el Scanner
2 import java.util.Scanner;
3
4 public class Main {
5     // Introducir dos números enteros y mostrar el mayor
6     public static void main(String[] args) {
7
8         //Definimos un Scanner que es necesario si vamos a leer
9         Scanner sc = new Scanner(System.in);
10
11         //Definir num1 como Entero
12         int num1;
13         //Definir num2 como Entero
14         int num2;
15
16         //ESCRIBIR "Introduce el primer número"
17         System.out.print ("Introduce el primer número: ");
18         //LEER num1
19         num1 = sc.nextInt();
20
21         //ESCRIBIR "Introduce el segundo número"
22         System.out.print ("Introduce el segundo número: ");
23         //LEER num2
24         num2 = sc.nextInt();
25
26         //SI num1 > num2 ENTONCES ESCRIBIR "El mayor es ", num1
27         //      SINO ESCRIBIR "El mayor es ", num2
28         //FINI
29         if (num1 > num2){
30             System.out.println ("El mayor es "+ num1);
31         }
32         else {
33             System.out.println ("El mayor es "+ num2);
34         }
35         //cerramos el scanner que hemos usado para leer
36         sc.close();
37     }
38 }
```

### 3- NUESTRO TERCER PROGRAMA

A continuación haremos un tercer ejemplo en la plataforma de trabajo Eclipse incorporando las instrucciones (asignar, leer, escribir y repetitiva) que hemos visto.

Esta será la práctica 2: *"se introduce un número entero por teclado, mostrar tantos "\*" como indique ese número"*

**Paso 1:** Crea un nuevo proyecto Pr1.

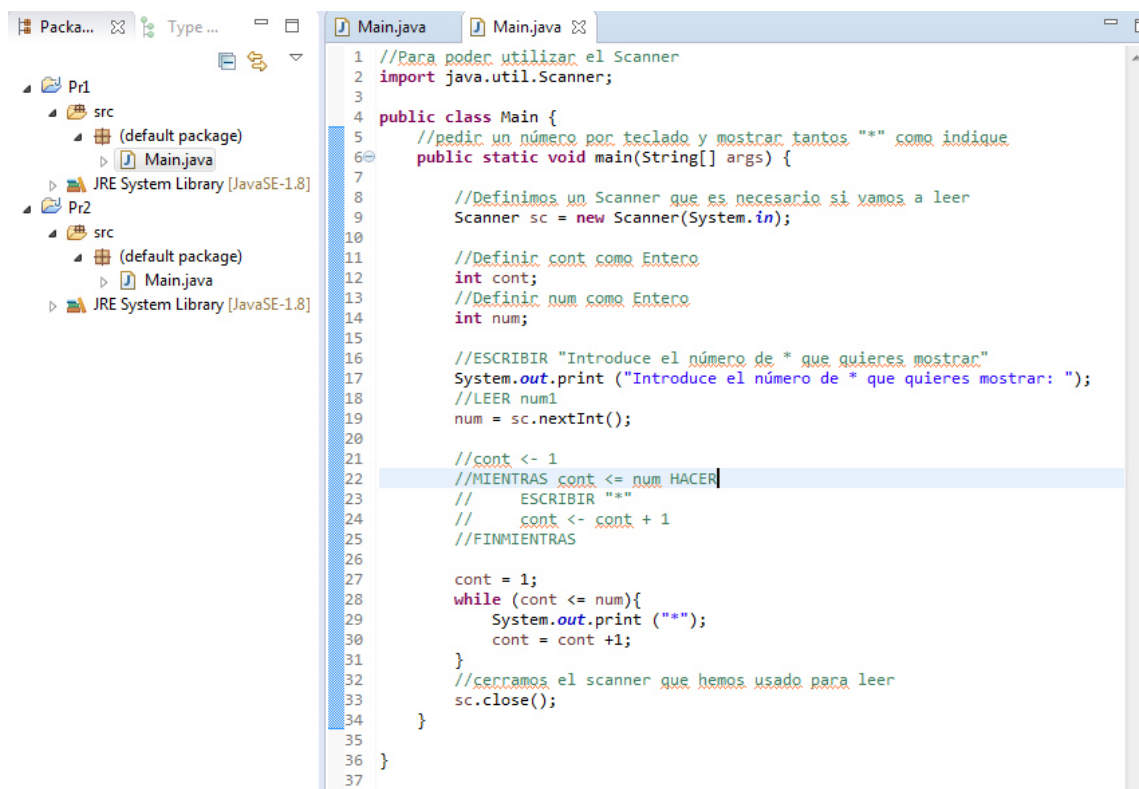
**Paso 2:** Crea el programa Main como se ha indicado antes.

**Paso 3:** Copia el código como aparece a continuación.

- Lo que empieza por // y está en verde son comentarios con el código en pseudocódigo
- Cada instrucción debe terminar en ";
- Fíjate en los márgenes y sigue la misma estructura
- Las líneas en blanco se eliminan puedes poner las que quieras.
- Las llaves delimitan los bloques, debe haber tantas de apertura como de cierre. Fíjate dónde se cierran, según dónde se han abierto.
- Las palabras en morado son palabras reservadas, siempre en minúsculas

**Paso 4:** Ejecútalo como se ha indicado antes.

Ejecútalo varias veces con diferentes valores



```
1 //Para poder utilizar el Scanner
2 import java.util.Scanner;
3
4 public class Main {
5     //pedir un número por teclado y mostrar tantos "*" como indique
6     public static void main(String[] args) {
7
8         //Definimos un Scanner que es necesario si vamos a leer
9         Scanner sc = new Scanner(System.in);
10
11         //Definir cont como Entero
12         int cont;
13         //Definir num como Entero
14         int num;
15
16         //ESCRIBIR "Introduce el número de * que quieres mostrar."
17         System.out.print ("Introduce el número de * que quieres mostrar: ");
18         //LEER num1
19         num = sc.nextInt();
20
21         //cont <- 1
22         //MIENTRAS cont <= num HACER
23         //     ESCRIBIR "*"
24         //     cont <- cont + 1
25         //FINMIENTRAS
26
27         cont = 1;
28         while (cont <= num){
29             System.out.print ("*");
30             cont = cont +1;
31         }
32         //cerramos el scanner que hemos usado para leer
33         sc.close();
34     }
35 }
36
37 }
```

#### 4- NUESTRO CUARTO PROGRAMA

A continuación haremos varios ejemplos de los que ya tienes hechos el pseudocódigo en clase. Elige los que quieras.... si no te da tiempo en clase práctica en casa.

Recordamos la sintaxis en java

##### 1. ESCRITURA

ESCRIBIR SIN SALTO "Texto"	System.out.print("Texto");
ESCRIBIR CON SALTO "Texto"	System.out.println("Texto");
ESCRIBIR NombreVariable	System.out.println(NombreVariable);
ESCRIBIR Expresión	System.out.println(EscribirExpresión);
ESCRIBIR "texto", NombreVariable	System.out.println("texto" + NombreVariable);

##### 2. LECTURA

Hay que tener definido un Scanner que debe cerrarse al final	Scanner sc = new Scanner (System.in); sc. close();
LEER nombreVariable	nombreVariable = sc.nextInt()      //si es un entero nombreVariable = sc.nextChar();    //si es un carácter nombreVariable = sc.nextFloat();    //si es un real nombreVariable = sc.nextDouble();   //si es un doble nombreVariable = sc.nextBoolean();   //si es un boeelano

##### 3. ASIGNACIÓN

nombreVariable <- valor	nombreVariable = valor;
nombreVariable <- expresión	nombreVariable = expresion;

##### 4. CONDICIONAL

SI condición	if (condición){
ENTONCES Instrucciones-V	Instrucciones-V;
	}
SINO Instrucciones-F	else {
FINSI	Instrucciones-F;
	}

##### 5. REPETITIVA

MIENTRAS condición HACER	while (condición ) {
Instrucciones	instrucciones;
FMIENTRAS	}