

JAVA

TEMA 02:

ESTRUCTURA Y SINTAXIS DE UN PROGRAMA EN JAVA

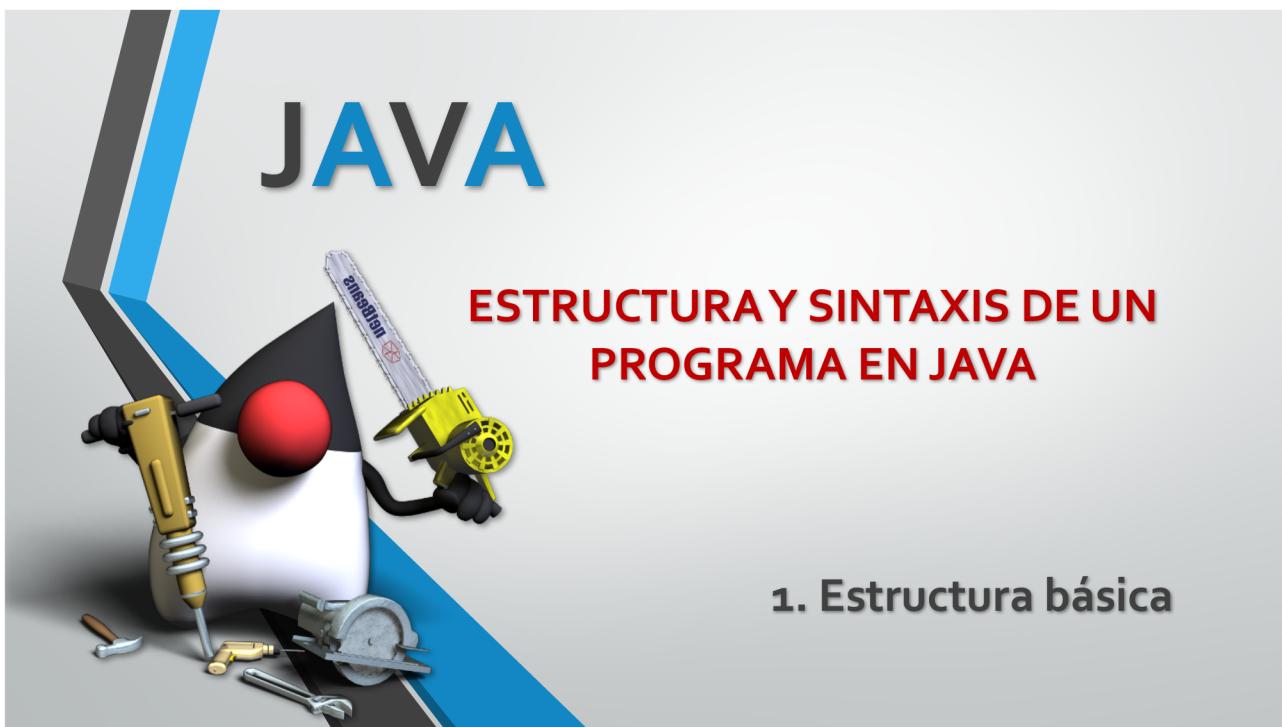


"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

ÍNDICE

1. Estructura básica
2. Los tipos de datos simples
3. Las variables
4. Las constantes
5. Los tipos de operadores
6. Prioridad entre operadores
7. La conversión de tipos (Cast)
8. Paquetes y sentencia import
9. Entrada de datos del usuario
10. Ejercicios de consolidación





1. Estructura básica

 "Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Estructura básica

- Todo programa en JAVA consta de una o más Clases, que a su vez constan de uno o más Métodos.
- Una de las clases que forman el programa (la clase principal) deberá contener el método **main** (o principal).
- Todo programa comienza ejecutándose por el método **main()** y, desde el cual, es posible llamar a otros métodos.

Oscar Laguna García
Ana M. Arribas Arjona

4
PROGRAMACIÓN

1.- Estructura básica

- La estructura general de una clase principal en Java suele ser la siguiente:

```
/**  
 * Estructura de una clase en Java  
 */  
  
public class NombreDeClase {  
    // Declaración de los atributos de la clase  
    // Declaración de los métodos de la clase  
  
    // El método main, que indica donde empieza la ejecución  
    public static void main(String[] args) {  
        // Declaración de las variables del método  
        // Sentencias de ejecución del método  
    }  
}
```



Ejemplo:

```
package vehiculo;  
  
public class Vehiculo {  
    int ruedas;  
    private int velocidad;  
    String nombre;  
  
    public void Arrancar() { //void porque no devuelve ningun valor  
        velocidad = 0; //Inicializa velocidad a 0  
    }  
    public void Acelerar(int cantidad) { //void porque no devuelve ningun valor  
        //Acelerar recibe como argumento un entero al que se le llama cantidad  
        velocidad = velocidad + cantidad;  
    }  
    public void Frenar(int cantidad) {  
        velocidad = velocidad - cantidad;  
    }  
    public int ObtenerVelocidad() { //no recibe argumentos  
        //ObtenerVelocidad es de tipo int porque devuelve un valor de tipo entero  
        return velocidad; // asigna el valor del atributo velocidad al metodo ObtenerVelocidad  
    }  
    public static void main(String[] args) {  
        Vehiculo miCoche = new Vehiculo();  
        miCoche.Arrancar(); // Esta llamada no pasa argumentos  
        miCoche.Acelerar(12); //Le pasamos 12 como argumento  
        miCoche.Frenar(5); //Le pasamos 5 como argumento  
        System.out.println(miCoche.ObtenerVelocidad());  
    }  
}
```

1.- Estructura básica

- En el caso de ser una clase distinta de la clase principal, el método main() no aparece.
- Los elementos que forman una clase principal(indicados en los comentarios) son 5:
 - Atributos de la clase: Son variables que definen el estado de los objetos. Los veremos en el tema 5.
 - Declaración de métodos: Definen la acciones que puede realizar el objeto. Los veremos en el tema 5.

1.- Estructura básica

- El método principal main(): Todo programa empieza ejecutando este método.
- Declaración de variables: Sirven para almacenar valores. Las veremos a continuación.
- Sentencias: Son instrucciones de ejecución del programa. Se ejecutan una detrás de otra hasta el final del programa y finalizan con un punto y coma ;



2. Los tipos de datos simples

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

V IES Valle del Jerte Bilingual Section IE

2.- Los tipos de datos simples

- Los tipos de datos se utilizan generalmente al declarar variables (luego veremos lo que son las variables).
- Son necesarios para que el intérprete o compilador conozca de antemano el tipo de información que va a contener una variable y cuando espacio debe reservar en memoria.
- Los tipos de datos simples en Java son los siguientes:

Oscar Laguna García
Ana M. Arribas Arjona

10
PROGRAMACIÓN



UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa". Colaboración a
cargo del Programa Operativo del FSE 2014-2020
para Extremadura gracias a Ciclos Formativos de
Grados Medio y Superior.



2.- Los tipos de datos simples

Tipo de datos	Información representada	Rango	Descripción
byte	Datos enteros	-128 ↔ +127	Se utilizan 8 bits (1 byte) para almacenar el dato.
short	Datos enteros	-32768 ↔ +32767	Dato de 16 bits de longitud (independientemente de la plataforma).
int	Datos enteros	-2147483648 ↔ +2147483647	Dato de 32 bits de longitud (independientemente de la plataforma).
long	Datos enteros	-9223372036854775808 ↔ +9223372036854775807	Dato de 64 bits de longitud (independientemente de la plataforma).
char	Datos enteros y caracteres	0 ↔ 65535	Este rango es para representar números en unicode, los ASCII se representan con los valores del 0 al 127. ASCII es un subconjunto del juego de caracteres Unicode.
float	Datos en coma flotante de 32 bits	Precisión aproximada de 7 dígitos	Dato en coma flotante de 32 bits en formato IEEE 754 (1 bit de signo, 8 para el exponente y 24 para la mantisa).
double	Datos en coma flotante de 64 bits	Precisión aproximada de 16 dígitos	Dato en coma flotante de 64 bits en formato IEEE 754 (1 bit de signo, 11 para el exponente y 52 para la mantisa).
boolean	Valores booleanos	true/false	Utilizado para evaluar si el resultado de una expresión booleana es verdadero (true) o falso(false).

Oscar Laguna García
Ana M. Arribas Arjona

11

PROGRAMACIÓN

JAVA

ESTRUCTURA Y SINTAXIS DE UN PROGRAMA EN JAVA



3. Las variables

3.- Las variables

- Las variables sirven para almacenar datos asociados a un nombre (identificador).

Ejemplo: **edadAlumno = 23;**

- Antes de empezar a utilizar una variable deberemos declararla, esto es, indicar el nombre que le vamos a poner (identificador) y el tipo de datos (los que vimos en la diapositiva anterior) que va a albergar.
- Cuando declaramos una variable, estamos reservándole un espacio en la memoria RAM del ordenador.

3.- Las variables

- Ejemplos de declaración de variables:

int edad; //declaro la variable edad de tipo entero

boolean respuesta; //declaro la variable respuesta de tipo boolean

float precio; /*declaro la variable precio de tipo float, ya que puede tener decimales */

3.- Las variables

- Una vez que tengamos la variable declarada, si lo deseamos, ya podríamos asignarle un valor. Para ello utilizaríamos el símbolo =.
- Ejemplos:

```
edad=24; /*asigno a la variable edad, que es  
de tipo entero, el valor 24.*/
```

```
respuesta=true; /*asigno a la variable respuesta, que es de tipo booleano,  
el valor verdadero.*/
```

```
precio= 4.99f; /*asigno a la variable precio, que  
es de tipo float, el valor 4.99.*/
```

3.- Las variables

- Otra opción, que le gusta a muchos programadores, es la de declarar la variable y, seguidamente, asignarle un valor. Ejemplos:

```
int edad=24; /*declaro la variable edad de tipo  
entero y le asigno el valor 24.*/
```

```
boolean respuesta=true; /*declaro la variable respuesta de tipo  
booleano y le asigno el valor verdad.*/
```

```
float precio= 4.99f; /*declaro la variable precio de tipo  
float y le asigno el valor 4.99.*/
```

- Incluso, los siguientes casos, también serían válidos:
`int edad, peso, altura; /*declaro las variables edad, peso
y altura de tipo entero*/`
- Y, por último, esto también sería válido:
`int edad=23, peso=80, altura=182; /*declaro las variables
edad, peso y altura de tipo
entero y les asigno un valor*/`

- ### 3.- Las variables
- Para nombrar a las variables, a las clases y a los métodos, no todos los identificadores son válidos. Hay ciertas normas que deben cumplir:
 - Tienen que ser significativos. (esto es, que el nombre tenga relación con lo que almacenen)
 - No llevan espacios en blanco.
 - No pueden empezar por un número.
 - No es recomendable que lleven acentos.
 - No es recomendable que contengan la letra ñ/Ñ.
 - Distinguen entre mayúsculas y minúsculas.
 - Aceptan _ y \$.

3.- Las variables

- Los identificadores de las variables (al igual que el de los métodos) empiezan por letra minúscula. Ejemplo: int **precio**; double **pagolImpuesto**; public int **reservarPlaza()**;
- Si son un nombre compuesto cada palabra empieza con mayúscula. (Ejem: **precioCoche**)
- En el caso de tratarse del identificador de una **constante** (en el punto siguiente veremos las constantes) se escribirán todas las letras en mayúsculas. (Ejem: final static int **DIAS_SEMANA** = 7;)
- Los identificadores de las **clases** comienzan siempre por letra mayúscula. Ejemplo: class **Coche** {

3.- Las variables

- **Ámbito de las variables:** Es importante el sitio del programa donde declares una variable.
- Por ejemplo, no es lo mismo declarar una variable dentro del bloque de la clase que dentro de un método. Cuando declaras una variable dentro de una clase la pueden utilizar todos los métodos de la clase.
- Veamos un ejemplo:

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

VIES Valle del Jerte
Bilingual section

3.- Las variables

```

package suma;
/*
 * @author OLG
 */
public class Suma {

    static int numero1 = 50; /*variable global o
                                miembro de la clase*/

    public static void metodoPrueba() {
        int numero3=10; //variable local
    }

    public static void main(String[] args) {
        int numero2, suma; // variables locales
        numero2=30;
        suma=numero1+numero2;
        System.out.println("La suma es: "+suma);
        suma=numero3+numero2;
    }
}

```

Oscar Laguna García
Ana M. Arribas Arjona

Declaramos `numero3` dentro de `metodoPrueba`, por lo que solo es accesible dentro ese método y no es accesible en el método `main`.

¡ERROR!

21

PROGRAMACIÓN

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

VIES Valle del Jerte
Bilingual section

EJERCICIOS

- Si vas a realizar prácticas en tu ordenador de casa, por comodidad, te recomiendo descargar el pack "JDK with Netbeans" en el que ya viene todo lo necesario incluido y configurado para empezar a trabajar:

Oscar Laguna García
Ana M. Arribas Arjona

Java SE
Software para Java SE
Java Embedded
Java EE
Java ME
Java FX
Java DB
Java for
Java Card
Community

Resumen Descargas Documentación Comunidad Tecnología Formación

JDK 8u111 with NetBeans 8.2

The Java Development Kit (JDK) is the Java SE runtime environment for developing applications on the Java platform. Learn more.

You must accept the JDK 8u111 and NetBeans 8.2 Cobundle License Agreement to download this software.

Thank you for accepting the JDK 8u111 and NetBeans 8.2 Cobundle License Agreement; you may now download this software.

Java 8 DKPs and Tools

- Java SE
- Java EE
- Java ME and GlassFish
- Java XC
- Java Card
- NetBeans IDE
- Java Vista or Control

Java Resources

- Java API
- Technical Articles
- Umls and Videos
- Events
- Java Vagabond
- Developer Training
- Tutorials
- Java.com

Java 8 and NetBeans Cobundle (JDK 8u111 and NB 8.2)

Product / File Description	Download	
Linux x64	507.71 KB	jdk-8u111-b14-2-linux-x64.tar.gz
Linux x86	262.87 MB	jdk-8u111-b14-2-linux-i586.tar.gz
Mac OS X 64	518.21 MB	jdk-8u111-b14-2-macosx-x64.dmg
Windows 64	387.21 MB	jdk-8u111-b14-2-windows-x64.exe
Windows 32	326.63 MB	jdk-8u111-b14-2-windows-i586.exe

22

PROGRAMACIÓN

EJERCICIOS

- **Ejercicio 01.- (OBLIGATORIO)** Utilizando el entorno de desarrollo NetBeans, corrige el error (solo uno) que hay en el siguiente programa:

```

1 package suma;
2 /**
3 * @author OLG
4 */
5 public class Suma {
6
7     public static void main(String[] args) {
8         int numero1=30;
9         boolean numero2;
10        int resultado;
11        numero2=700;
12        resultado=numero1+numero2;
13        System.out.println("El resultado de la suma es: "+resultado);
14    }
15 }

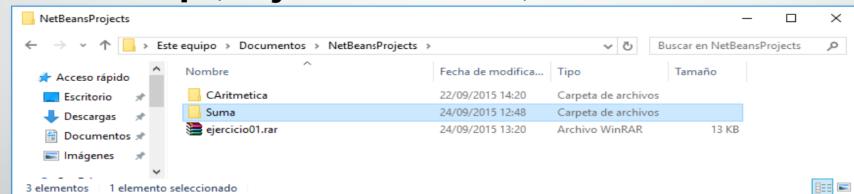
```

23

PROGRAMACIÓN

EJERCICIOS

- Cuando lo descubras vete a la carpeta donde se guardan los proyectos de NetBeans y comprímelo con el nombre de **ejercicio01.zip** (o **ejercicio01.rar**).



- Por último, súbelo a tu carpeta de Google Drive, dentro de una carpeta llamada **Tema02**.

24

PROGRAMACIÓN

EJERCICIOS

- **Ejercicio 02.- (OBLIGATORIO)** Utilizando el entorno de desarrollo NetBeans, corrige los errores que hay en el siguiente programa:

```
package producto;
/**
 * @author OLG
 */
public class Producto {
    public static void main(String[] args) {
        float numero1=0.5F;
        long numero2;
        float resultado;
        numero2=178823419991L;
        resultado=numero1*numero2;
        System.out.println("El resultado del multiplicar " +numero 1 "y numero 2 es igual a"+ resultado);
    }
}
```

EJERCICIOS

- Cuando lo descubras vete a la carpeta donde se guardan los proyectos de NetBeans y comprímelo con el nombre de **ejercicio02.zip** (o **ejercicio02.rar**).
- Por último, súbelo a tu carpeta de Google Drive, dentro de una carpeta llamada **Tema02**.



UNION EUROPEA

Fondo Social Europeo

"Una manera de hacer Europa"

Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



EJERCICIOS

- **Ejercicio 03.- (OBLIGATORIO)** Realiza un programa en JAVA que escriba por pantalla tu nombre completo en una línea, en la línea siguiente tu edad y en la siguiente tu altura (con decimales). Para ello deberás, previamente, almacenar tu edad y altura en dos variables que sean del tipo más óptimo posible (aprovechando la memoria).
- Recuerda subirlo a Google Drive (ejercicio03.zip)



UNION EUROPEA

Fondo Social Europeo

"Una manera de hacer Europa"

Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



EJERCICIOS

- **Ejercicio 04.- (OPTATIVO)** Realiza un programa en JAVA que escriba por pantalla el nombre de una asignatura, la nota del primer examen, la nota del segundo examen y la nota media de la asignatura. Para ello deberás, previamente, almacenar las notas en variables que sean del tipo más óptimo posible (aprovechando la memoria).
- Recuerda subirlo a Google Drive (ejercicio04.zip)



UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



EJERCICIOS

- **Ejercicio 05.- (OPTATIVO):** En el tema hemos visto que los identificadores nos sirven para nombrar a las variables, a las clases, a los métodos y a las constantes. Además, los identificadores sólo pueden estar formados por letras y dígitos, y deben comenzar exclusivamente por una letra. Recuerda que en JAVA también se consideran los caracteres subrayado y \$ como letra.
- Dados los siguientes identificadores que se van a utilizar en un programa escrito en JAVA, indica cuáles de ellos son correctos y cuáles no justificando tu respuesta:
- Recuerda subirlo a Google Drive (ejercicio05.zip)



UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



EJERCICIOS

- | | |
|------------------------|----------------------|
| a) mi carta | h) mi-programa |
| b) unacarta | i) ¿cuantos? |
| c) Mis2escritos | j) el%Descontado |
| d) 4cientos | k) a2ooPORAÑO |
| e) es_un_mensaje | l) TengoMUCHOS\$\$\$ |
| f) no_vale nada | m) LOS400GOLPES |
| g) _____ ejemplo _____ | n) quieroUNAsolución |



UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa". Cofinanciación a
cargo del Programa Operativo del FSE 2014-2020
para Extremadura gastos de Ciclos Formativos de
Grados Medio y Superior.



EJERCICIOS

- **Ejercicio 06.- (OPTATIVO):** Dados los siguientes identificadores que se van a utilizar en un programa escrito en JAVA, indica cuáles de ellos son correctos y cuáles no justificando tu respuesta:
- Recuerda subirlo a Google Drive (ejercicio06.zip)

- a) descartaz
- b) cuántosQuerrás
- c) çaVaBienAvec\$\$
- d) Ègressa
- e) österreich
- f) Ñosýevan
- g) EsôÇäépùog



UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa". Cofinanciación a
cargo del Programa Operativo del FSE 2014-2020
para Extremadura gastos de Ciclos Formativos de
Grados Medio y Superior.

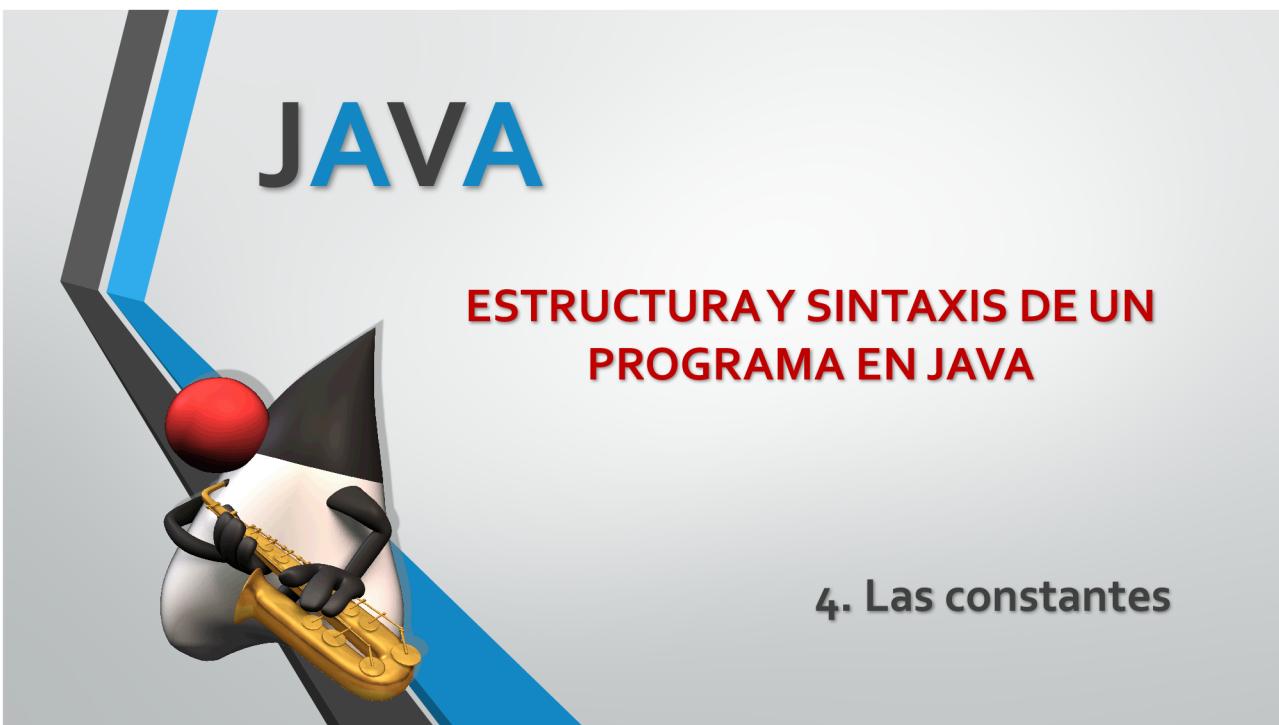


EJERCICIOS

- **Ejercicio 07.- (OPTATIVO):** Abre el libro "Programación en Java 2" (Aprueba tu examen con Schaum) y realiza los siguientes ejercicios (*intenta no mirar las soluciones*) para terminar de practicar sobre los tres primeros puntos que acabamos de ver en este tema:

- Ejercicios 1.6, 1.7 y 1.8 (página 20)

Recuerda subirlo a Google Drive (ejercicio07.zip)



4. Las constantes

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

V IES Valle del Jerte Bilingual Section IE

4.- Las constantes

- Se utilizan para almacenar datos que nunca varían durante el desarrollo del programa (IVA, PI, VELOCIDAD_LUZ, GRAVEDAD, PESO_MINIMO, CAPACIDAD_MAXIMA, etc) y así nos aseguramos de que su valor no es cambiado nunca.
- Si alguna vez queremos cambiar el valor de una constante bastará con cambiar la línea de código donde se le asignó el valor y afectará automáticamente al resto del programa.

Oscar Laguna García
Ana M. Arribas Arjona

34
PROGRAMACIÓN

4.- Las constantes

- Las constantes se declaran siguiendo el siguiente formato:
final static tipo de dato NOMBRE_DE_LA_CONSTANTE = valor;
- Ejemplos:

final static double PI = 3.141592;

final static short IVA = 21;

final static long VELOCIDAD_LUZ = 299792458;

final static int CAPACIDAD_MAXIMA = 88932;

4.- Las constantes

- Ejemplo:

```
package distancia;
/*
 * @author Oscar Laguna Garcia
 */
public class Distancia {

    final static int VELOCIDADLUZ = 299792;

    public static void main(String[] args) {
        int distanciaMarte,distanciaSaturno;
        int tiempo;
        distanciaMarte = 225300000;
        distanciaSaturno = 1429400000;
        tiempo= distanciaMarte/VELOCIDADLUZ;
        System.out.println("A la velocidad de la luz tardará "+tiempo+" segundos en llegar a Marte");
        tiempo= distanciaSaturno/VELOCIDADLUZ;
        System.out.println("A la velocidad de la luz tardará "+tiempo+" segundos en llegar a Saturno");
    }
}

<----->
tancia.Distancia > VELOCIDADLUZ >
- Distancia (run) x
run:
A la velocidad de la luz tardará 751 segundos en llegar a Marte
A la velocidad de la luz tardará 4767 segundos en llegar a Saturno
```

JAVA

ESTRUCTURA Y SINTAXIS DE UN PROGRAMA EN JAVA

5. Los tipos de operadores

  UNION EUROPEA
Fondo Social Europeo
"De Europa A las Tierras"

"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



5.- Los tipos de operadores

- En JAVA existen 6 tipos de operadores:
 - Los Operadores Aritméticos
 - Los Operadores Relacionales
 - Los Operadores Lógicos
 - Los Operadores Unarios o Unitarios
 - Los Operadores de Bits
 - Los Operadores de Asignación

5.- Los tipos de operadores

- Los Operadores Aritméticos:

Operador	Uso	Operación
+	A + B	Suma
-	A - B	Resta
*	A * B	Multiplicación
/	A / B	División
%	A % B	Módulo o resto de una división entera

5.- Los tipos de operadores

- Los Operadores Relacionales:

Operador	Uso	Operación
<	A < B	A menor que B
>	A > B	A mayor que B
<=	A <= B	A menor o igual que B
>=	A >= B	A mayor o igual que B
!=	A != B	A distinto que B
= =	A == B	A igual que B

5.- Los tipos de operadores

- Los Operadores Lógicos:

Operador	Uso	Operación
&& o &	A&& B o A&B	A AND B. El resultado será true si ambos operandos son true y false en caso contrario.
o	A B o A B	A OR B. El resultado será false si ambos operandos son false y true en caso contrario.
!	!A	Not A. Si el operando es true el resultado es false y si el operando es false el resultado es true.
^	A ^ B	A XOR B. El resultado será true si un operando es true y el otro false, y false en caso contrario.

5.- Los tipos de operadores

- Los Operadores Unarios (o Unitarios):

Operador	Uso	Operación
~	~A	Complemento a 1 de A
-	-A	Cambio de signo del operando
--	A--	Decremento de A
++	A++	Incremento de A
!	! A	Not A (ya visto)



5.- Los tipos de operadores

- Los Operadores de bits:

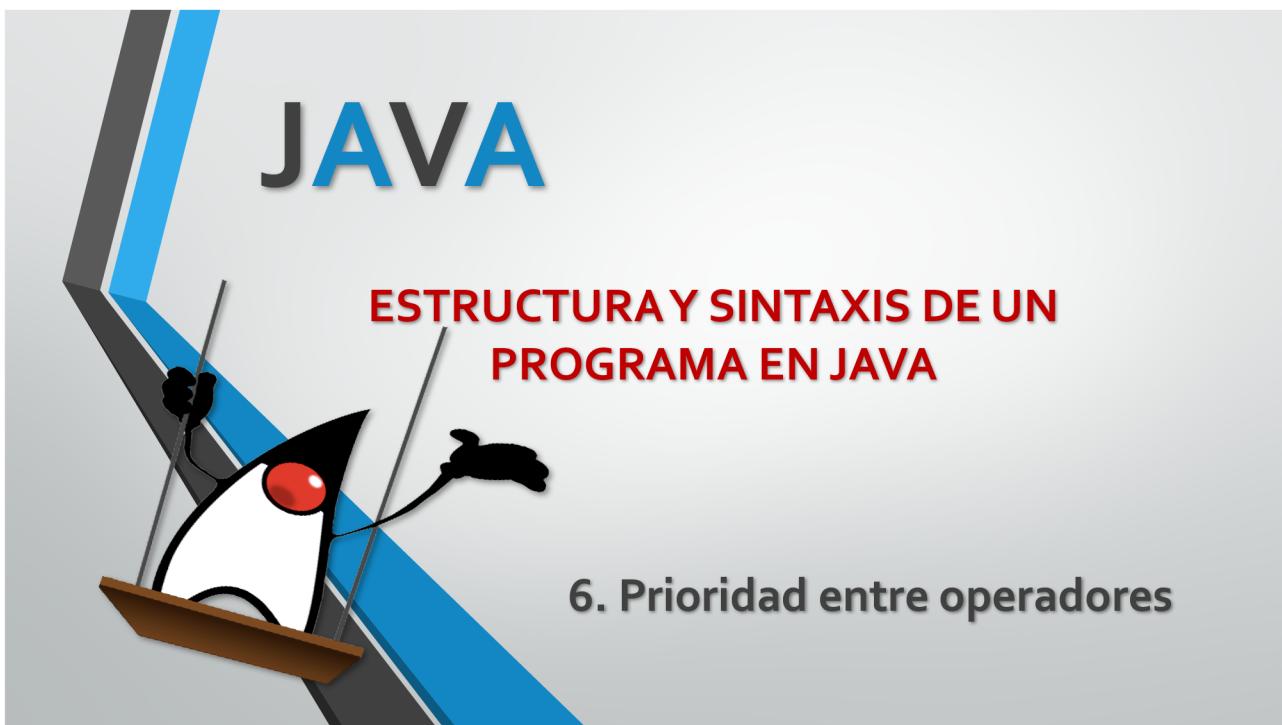
Operador	Uso	Operación
&	A & B	AND lógico. A AND B.
	A B	OR lógico. A OR B.
^	A ^ B	XOR lógico. A XOR B.
<<	A << B	Desplazamiento a la izquierda de A B bits rellenando con ceros por la derecha.
>>	A >> B	Desplazamiento a la derecha de A B bits rellenando con el BIT de signo por la izquierda.
>>>	A >>> B	Desplazamiento a la derecha de A B bits rellenando con ceros por la izquierda.



5.- Los tipos de operadores

- Los Operadores de Asignación:

Operador	Uso	Operación
=	A = B	Asignación. Operador ya visto.
*=	A *= B	Multiplicación y asignación. La operación A*=B equivale a A=A*B.
/=	A /= B	División y asignación. La operación A/=B equivale a A=A/B.
%=	A %= B	Módulo y asignación. La operación A%=B equivale a A=A%B.
+=	A += B	Suma y asignación. La operación A+=B equivale a A=A+B.
-=	A -= B	Resta y asignación. La operación A-=B equivale a A=A-B.



6. Prioridad entre operadores

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

V IES Valle del Jerte Bilingual Section IE

6.- Prioridad entre operadores

- De mayor a menor prioridad:

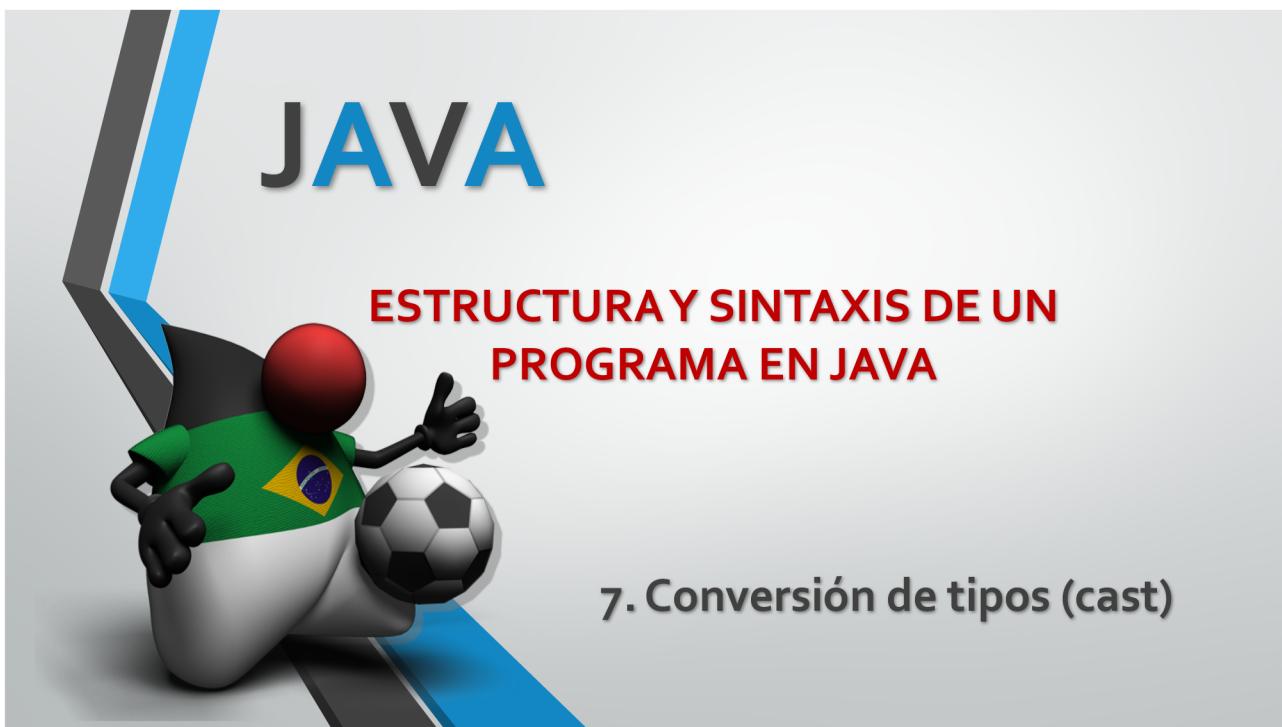
MAS PRIORIDAD ↑ ↓ MENOS PRIORIDAD

OPERADORES
() []. -- ~ ! ++ -- new (tipo)expresión * / % + - << >> >>> < <= > >= instanceof == != & ^ && ?: = *= /= %= += -= <<= >>= >>>= &= = ^=

Oscar Laguna García
Ana M. Arribas Arjona

PROGRAMACIÓN

46



7. Conversión de tipos (cast)

 "Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



7.- Conversión de tipos (cast)

- En JAVA existen dos tipos de conversiones: *Implícita* y *Explícita*
- **Conversión Implícita:** Se realiza de forma **automática** entre dos tipos de datos diferentes. Requiere que la variable destino (la colocada a la izquierda) tenga más precisión que la variable origen (situada a la derecha).
- Ejemplo:

```
short dato1;
int dato2;
dato1 = 88;
dato2 = dato1; //Conversión implícita
System.out.println (dato2); //Muestra 88 por pantalla
```

Oscar Laguna García
Ana M. Arribas Arjona

7.- Conversión de tipos (cast)

- **Conversión Explicita:** En este caso, es el programador el que **fuerza** la conversión mediante una operación llamada cast con el formato: **(tipo) expresion;**
- Ejemplo:

```
int dato1;
byte dato2;
dato1 = 88;
dato2 = (byte) dato1; //Conversión EXPLICITA
System.out.println(dato2); //Muestra 88 por pantalla
```

EJERCICIOS

- Ejercicio 08.- (OPTATIVO):

Recuerda subirlo a Google Drive (ejercicio08.zip)

EJERCICIOS

- **Ejercicio 09.- (OBLIGATORIO):** Realiza un programa que calcule la longitud de una circunferencia de radio 3,55 metros. Para ello utiliza la constante PI.
- Muestra por pantalla el resultado de la siguiente forma:
*La longitud de una circunferencia cuyo radio vale 3 sería igual a: **XXX** metros.*

Recuerda subirlo a Google Drive (ejercicio09.zip)

EJERCICIOS

- **Ejercicio 10.- (OPTATIVO):**

The screenshot shows a Java code editor with a file named Suma.java. The code is as follows:

```
1 package suma;
2 /**
3  * @author OLG
4  * Modificalo el siguiente programa para hacer que compile y funcione
5 */
6 public class Suma {
7
8     static int n=50;
9
10    public static void main(String[] args) {
11        int n2=30, suma=0, n3;
12        suma=n1+n2;
13        System.out.println("LA SUMA ES: " + suma);
14        suma = suma + n3;
15        System.out.println(suma);
16    }
17}
```

Recuerda subirlo a Google Drive (ejercicio10.zip)

EJERCICIOS

- **Ejercicio 11.- (OPTATIVO):**

The screenshot shows a Java code editor with the file 'Cuadrado.java' open. The code is as follows:

```
1 package cuadrado;
2 /**
3 * @author OLG
4 * El siguiente programa tiene fallos.
5 * Averigua cuáles son y modifica el programa
6 * para que funcione
7 */
8 public class Cuadrado {
9
10    public static void main(String[] args) {
11        int numero = 2,
12            cuad = numero * número;
13        System.out.println("EL CUADRADO DE " + NUMERO + " ES: " + Cuad);
14    }
15}
```

Recuerda subirlo a Google Drive (ejercicio11.zip)

EJERCICIOS

- **Ejercicio 12.- (OPTATIVO):**

//¿Qué mostrará el siguiente código por pantalla?

```
int num = 5;
num += num - 1 * 4 + 1;
System.out.println(num);
num = 4;
num %= 7 * num % 3 * 7 >> 1;
System.out.println(num);
```

Recuerda subirlo a Google Drive (ejercicio12.zip)



EJERCICIOS

- **Ejercicio 13.- (OBLIGATORIO):**

"¿Recuerdas la prueba de agilidad mental de los dos vasos? El vaso A tiene líquido azul y B líquido rojo. ¿Cómo pasarías el líquido de un vaso a otro de forma que el vaso A se quede con líquido rojo y el B con líquido azul?"

- Realiza un programa en el que tengas dos variables de tipo entero, num1 que contiene un 1 y num2 que contiene un 2. ¿Cómo pasarías el contenido de una variable a otra de forma que num1 contenga el 2 y num2 contenga el 1?
- Muestra por pantalla el resultado de la siguiente forma:

La variable num1 contiene el valor 1 y la variable num2 contiene el valor 2.

Ahora, la variable num1 contiene el valor 2 y la variable num2 contiene el valor 1.

- Recuerda subirlo a Google Drive (ejercicio13.zip)



EJERCICIOS

- **Ejercicio 14.- (OPTATIVO):** Realiza un programa que calcule el área de una circunferencia de radio 5,2 centímetros. Para ello utiliza la constante PI.

- Muestra por pantalla el resultado de igual forma que el ejercicio anterior.

Recuerda subirlo a Google Drive (ejercicio14.zip)

EJERCICIOS

- **Ejercicio 15.- (OBLIGATORIO):** Realiza un programa en el que tengas una variable entera llamada tiempo que contiene un tiempo en segundos igual a 10000 (diez mil). Queremos conocer ese tiempo, pero expresado en horas, minutos y segundos.
- Muestra por pantalla el resultado de la siguiente forma:
10.000 segundos hacen un total de: xxx horas, xxx minutos y xxx segundos.

Recuerda subirlo a Google Drive (ejercicio15.zip)

EJERCICIOS

- **Ejercicio 16.- (OPTATIVO):** Suponemos que llevas en tu cartera 130 euros en billetes de 50 euros y de 10 euros. Realiza un programa que sea capaz de decirte cuantos billetes de cada.
- Muestra por pantalla el resultado de la siguiente forma:

130 euros hacen un total de: 2 billetes de 50 euros y 3 billetes de 10 euros.

Recuerda subirlo a Google Drive (ejercicio16.zip)

EJERCICIOS

- **Ejercicio 17.- (OPTATIVO):** Dadas las siguientes expresiones aritméticas, calcule cuál es el resultado de evaluarlas.

- a) $25 + 20 - 15$
- b) $20 * 10 + 15 * 10$
- c) $20 * 10/2 - 20 / 5 * 3$
- d) $15 / 10 * 2 + 3 / 4 * 8$

Recuerda subirlo a Google Drive (ejercicio17.zip)

EJERCICIOS

- **Ejercicio 18.- (OPTATIVO):** Dadas las siguientes expresiones aritméticas, calcule cuál es el resultado de evaluarlas. Supón que las variables **a** y **b** son de tipo int y **a** tiene el valor **2** y **b** tiene el valor **4**.

- a) $-a + 5 \% b - a * a$
- b) $5 + 3 \% 7 * b * a - b \% a$
- c) $(a + 1) * (b + 1) - b/a$

Recuerda subirlo a Google Drive (ejercicio18.zip)

EJERCICIOS

- **Ejercicio 19.- (OPTATIVO):**
Dado el siguiente código, indique qué escribe en pantalla. Justifique su respuesta.
- Recuerda subirlo a Google Drive (ejercicio19.zip)

```
public static void main(String[] args) {  
    int a = 3, b = 6, c;  
    c = a / b;  
    System.out.println("El valor de c es: " + c);  
  
    c = a % b;  
    System.out.println("El valor de c es: " + c);  
  
    a++;  
    System.out.println("El valor de a es: " + a);  
  
    ++a;  
    System.out.println("El valor de a es: " + a);  
  
    c = ++a + b++;  
    System.out.println("El valor de a es: " + a);  
    System.out.println("El valor de b es: " + b);  
    System.out.println("El valor de c es: " + c);  
  
    c = ++a + ++b;  
    System.out.println("El valor de a es: " + a);  
    System.out.println("El valor de b es: " + b);  
    System.out.println("El valor de c es: " + c);  
}
```

61

PROGRAMACIÓN

Oscar Laguna García
Ana M. Arribas Arjona

JAVA

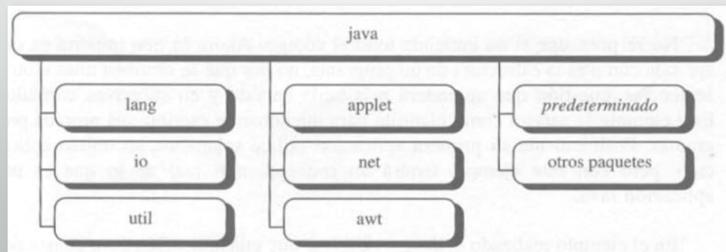
ESTRUCTURA Y SINTAXIS DE UN PROGRAMA EN JAVA



8. Paquetes y sentencia import

8.- Paquetes y sentencia import

- Un paquete es un conjunto de clases, relacionadas entre sí y agrupadas bajo un nombre.
- La propia biblioteca de Java está organizada en paquetes dispuestos jerárquicamente:



63

PROGRAMACIÓN

8.- Paquetes y sentencia import

- Para utilizar las clases públicas de estos paquetes de la biblioteca de Java (java.io, java.util, java.net...), la opción más cómoda es utilizar la sentencia import.
- *¡Ojo!!* El paquete java.lang es importado automáticamente en todos los programas java, por eso nos ha funcionado siempre la instrucción System.out.println. (System es una clase del paquete java.lang)

64

PROGRAMACIÓN

8.- Paquetes y sentencia import

PRINCIPALES CLASES EN EL PAQUETE JAVA.LANG

Boolean	Math
Double	String
Float	StringBuffer
Integer	System

8.- Paquetes y sentencia import

- Supongamos que el paquete java.lang no fuera importado directamente. Si lo quisieramos importar lo haríamos de la siguiente manera:

```
import java.lang.*; //Importamos todas las clases del paquete java.lang
public class HolaMundo { // Declaro la clase HolaMundo de nuestro programa
    public static void main(String[] args) {
        System.out.println("Hola Mundo!");
    }
}
```

8.- Paquetes y sentencia import

- No es obligatorio importar un paquete entero. Existe la opción de importar solo la clase que nos interese de un paquete. Lo haríamos de la siguiente manera:

```
import java.lang.System; //Importamos solo la clase System del paquete  
java.lang  
public class HolaMundo { //Declaro la clase HolaMundo de nuestro programa  
    public static void main(String[] args) {  
        System.out.println("Hola Mundo!");  
    }  
}
```

8.- Paquetes y sentencia import

- No es obligatorio utilizar la sentencia import para importar clases públicas de otros paquetes. Lo podríamos hacer directamente, de la siguiente manera:

```
public class HolaMundo //Declaro la clase HolaMundo de nuestro programa  
{  
    public static void main(String[] args) {  
        java.lang.System.out.println("Hola Mundo!");  
    }  
}
```

8.- Paquetes y sentencia import

- Resumen de librerías JAVA:

Paquete o librería	Descripción
java.io	Librería de Entrada/Salida. Permite la comunicación del programa con ficheros y periféricos.
java.lang	Paquete con clases esenciales de Java. No hace falta ejecutar la sentencia import para utilizar sus clases. Librería por defecto.
java.util	Librería con clases de utilidad general para el programador.
java.applet	Librería para desarrollar applets.
java.awt	Librerías con componentes para el desarrollo de interfaces de usuario.
java.swing	Librerías con componentes para el desarrollo de interfaces de usuario. Similar al paquete awt.
java.net	En combinación con la librería java.io, va a permitir crear aplicaciones que realicen comunicaciones con la red local e Internet.
java.math	Librería con todo tipo de utilidades matemáticas.
java.sql	Librería especializada en el manejo y comunicación con bases de datos.
java.security	Librería que implementa mecanismos de seguridad.
java.rmi	Paquete que permite el acceso a objetos situados en otros equipos (objetos remotos).
java.beans	Librería que permite la creación y manejo de componentes <i>javabeans</i> .

69

PROGRAMACIÓN

Oscar Laguna García
Ana M. Arribas ArjonaThe logo features the word "JAVA" in a bold, sans-serif font. The letters are colored blue and grey, with a slight shadow effect. The letter "J" has a vertical bar extending upwards from its top.

ESTRUCTURA Y SINTAXIS DE UN PROGRAMA EN JAVA



9. Entrada de datos del usuario

9.- Entrada de datos del usuario

- Hasta ahora hemos visto que para mostrar datos al usuario por pantalla, utilizamos la instrucción System.out.println().
- Lo habitual en un programa es que el usuario también pueda introducirle datos al programa cuando se los solicitemos por pantalla. Esto se puede hacer con el método System.in.read().
- System.in.read() tiene “el problema” de que solo lee *bytes*, por lo que luego debemos convertirlos a lo que necesitemos. Por ejemplo, si tecleamos una letra A mayúscula, el *byte* leído es el 65, correspondiente a la A mayúscula en código ASCII. (Ya haremos algunos ejemplos en temas posteriores)

9.- Entrada de datos del usuario

- A partir de JAVA 5, se introdujo la utilización de la clase Scanner, por lo que lo normal es utilizar sus métodos para la entrada de datos del usuario.
- La clase Scanner pertenece al paquete java.util.(paquete que debemos importar)
- Veamos un ejemplo de su utilización:

```

1 /*
2  * Programa en JAVA que pide al usuario los datos
3  * del radio y la altura de un cilindro y calcula su volumen
4  */
5 package cilindro;
6 import java.util.Scanner;
7 /**
8  * @author OLG
9  */
10 public class Cilindro {
11
12     public static void main(String[] args) {
13         // Valor del numero pi:
14         final double PI = 3.1415926536;
15
16         double radio;
17         double altura;
18
19         Scanner entrada = new Scanner (System.in);
20
21         System.out.println("Introduzca los datos del cilindro:");
22         System.out.print("Radio: ");
23         radio = entrada.nextDouble();
24         System.out.print("Altura: ");
25         altura = entrada.nextDouble();
26         System.out.println("El área del cilindro es: ");
27         System.out.println(PI * radio * radio * altura);
28     }
29 }

```

Importa la clase Scanner

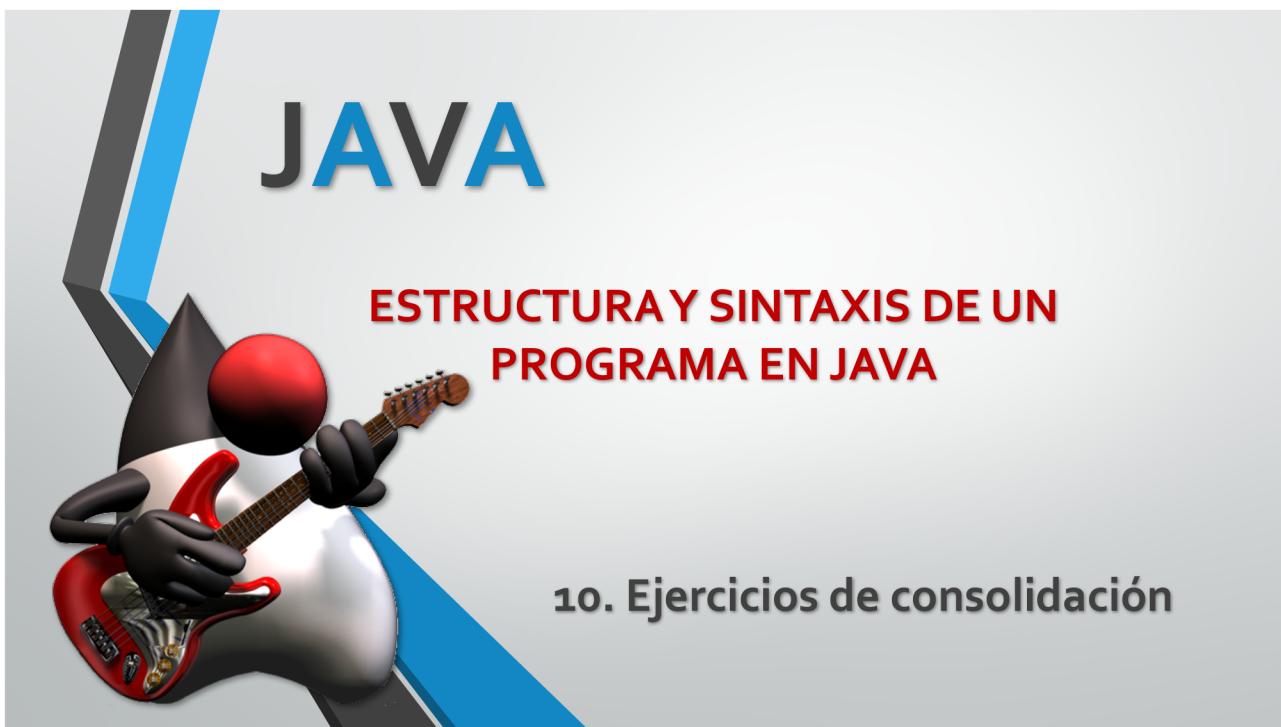
Crea el objeto entrada.
Lo veremos en el Tema 5

Utiliza el método
nextDouble() para leer el
valor de radio, ya que es
de tipo double

9.- Entrada de datos del usuario

- Si declaramos un objeto llamado "entrada" de la clase Scanner, estos son algunos de los tipos de datos que podemos leer con los métodos de la clase Scanner:

Tipo	Método a invocar
byte	entrada.nextByte();
short	entrada.nextShort();
int	entrada.nextInt();
long	entrada.nextLong();
float	entrada.nextFloat();
double	entrada.nextDouble();
boolean	entrada.nextBoolean();



10. Ejercicios de consolidación

 UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

 IES Valle del Jerte
Bilingual Section



EJERCICIOS

- Ejercicio 20.- (OBLIGATORIO):** Corrige los errores del siguiente programa, teniendo en cuenta que la declaración de las variables es correcta.
- Recuerda subirlo a Google Drive (ejercicio20.zip)

```
package suma;
import java.lang.Scanner; // el programa usa la clase Scanner

// Programa que muestra la suma de dos enteros.
public class Suma {

    // el método main empieza la ejecución de la aplicación en Java
    public static void main(String args[]) {
        // crea objeto Scanner para obtener la entrada de la ventana de comandos
        Scanner entrada = new Scanner(System.in);

        double numero1; // primer número a sumar
        double numero2; // segundo número a sumar
        int suma;//suma de numero1 y numero2 haciendo cast (Conversión Explicita)

        System.out.print("\t Escriba el primer entero: "); // indicador
        numero1 = entrada.nextInt(); // lee el primer número del usuario
        System.out.print("\t Escriba el segundo entero: "); // indicador
        numero2 = entrada.nextFloat(); // lee el segundo número del usuario

        suma = numero1 + numero2; // suma los números
        System.out.println("La suma es: ", suma); // muestra la suma
    } // fin del método main
} // fin de la clase Suma
```

Oscar Laguna García
Ana M. Arribas Arjona

76

PROGRAMACIÓN



"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



EJERCICIOS

- **Ejercicio 21.- (OBLIGATORIO):** Implementa un programa que solicite al usuario una cantidad en segundos y la convierta a días, horas, minutos y segundos.
- Muestra por pantalla el resultado de la siguiente forma:

Por favor, introduzca un número de segundos:

XXXXXXX

**xxx segundos hacen un total de : xxx días, xxx horas, xxx minutos y
xxx segundos.**

- Recuerda subirlo a Google Drive (ejercicio21.zip)



"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



EJERCICIOS

- **Ejercicio 22.- (OPTATIVO):** Diseña un programa que solicite al usuario el tamaño del lado de un triángulo equilátero, y calcule su perímetro y su área.
- Muestra por pantalla el resultado de la siguiente forma:

Por favor, introduzca la medida de un lado: XXXXX

El área de un triángulo de lado: xxx es: xxx

El perímetro de un triángulo de lado: xxx es: xxx

- Recuerda subirlo a Google Drive (ejercicio22.zip)



UNION EUROPEA

Fondo Social Europeo

"Una manera de hacer Europa"

"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



EJERCICIOS

- **Ejercicio 23.- (OPTATIVO):** Realiza un programa en JAVA en el que le preguntes al usuario por el precio de un producto y cuantas unidades quiere comprar y el programa le calculará el total de la compra.
- Muestra por pantalla el resultado de la siguiente forma:
Por favor, introduzca el precio del modelo de ordenador que desea comprar: XXXX.XX

¿Cuántas unidades quiere llevarse? XX

El precio total de su compra es de: XXXX.XX Euros.

Oscar Laguna García
Ana M. Arribas Arjona

PROGRAMACIÓN

79



UNION EUROPEA

Fondo Social Europeo

"Una manera de hacer Europa"

"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



EJERCICIOS

- **Ejercicio 24.- (OPTATIVO):** Escribe un programa en JAVA en el que le pidas al usuario las notas de las 6 asignaturas del Ciclo de DAM y te calcule la nota media del curso.
- Muestra por pantalla el resultado de la siguiente forma:

Por favor, introduzca la nota de Programación: X.X

Introduzca la nota de Lenguajes de Marcas: X

Introduzca la nota de Bases de Datos: X.X

Introduzca la nota de Entornos de Desarrollo: X

Introduzca la nota de Sistemas Informáticos: X

Por último, introduzca la nota de Formación y Orientación Laboral: X.X

Su nota media del curso es de: X.X

Oscar Laguna García
Ana M. Arribas Arjona

PROGRAMACIÓN

80



UNION EUROPEA

Fondo Social Europeo

"Una manera de hacer Europa"

"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



EJERCICIOS

- **Ejercicio 25.- (OPTATIVO):** Realiza un programa en JAVA en el que le pidas por pantalla 3 números al usuario y le dé como resultado la suma y el producto de estos.
- Muestra por pantalla el resultado de la siguiente forma:

Por favor, introduzca el primer número: XXX

Por favor, introduzca el segundo número: XX

Por favor, introduzca el tercer número: XXX

La suma de los números introducidos es: XXX

El producto de los números introducidos es: XXXXX



UNION EUROPEA

Fondo Social Europeo

"Una manera de hacer Europa"

"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



EJERCICIOS

- **Ejercicio 26.- (OBLIGATORIO):** Desarrolla un programa en el que le pidas al usuario un número de 4 cifras y muestre por pantalla cada una de las cifras que lo forman.
- Muestra por pantalla el resultado de la siguiente forma:

Por favor, introduzca un número de 4 cifras: XYZW

La primera cifra es: X

La segunda cifra es: Y

La tercera cifra es: Z

La cuarta cifra es: W



UNION EUROPEA

Fondo Social Europeo

"Una manera de hacer Europa"

"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



EJERCICIOS

- **Ejercicio 27.- (OPTATIVO):** Escribe un programa en JAVA para calcular el cuadrado y el cubo de un número entero introducido por teclado y que muestre los resultados.
- Muestra por pantalla el resultado de la siguiente forma:

Por favor, introduzca un número: XXX

El doble de XXX es: XXXXXX

El cubo de XXX es: XXXXXXXX

Oscar Laguna García
Ana M. Arribas Arjona

83

PROGRAMACIÓN



UNION EUROPEA

Fondo Social Europeo

"Una manera de hacer Europa"

"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



EJERCICIOS

- **Ejercicio 28.- (OBLIGATORIO):** Implementa un programa al que introduciéndole el tamaño de un archivo en MB y la velocidad de tu ADSL (en megabits) te calcule el tiempo en minutos que tardarías en descargártelo.
- Recuerda subirlo a Google Drive (ejercicio26.zip)

Oscar Laguna García
Ana M. Arribas Arjona

84

PROGRAMACIÓN



UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



EJERCICIOS

- **Ejercicio 29.- (OPTATIVO):** Escribe un programa en JAVA en el que se le pida al usuario que introduzca la longitud de los catetos de un triángulo rectángulo y que posteriormente el programa calcule la longitud de la hipotenusa.
- Recuerda subirlo a Google Drive (ejercicio29.zip)



UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.



EJERCICIOS

- **Ejercicio 30.- (OBLIGATORIO):** Utilizando la conversión Explícita (cast) es fácil averiguar el código ASCII de un carácter cualquiera (Por ejemplo (int) 'A';). Realiza un programa que muestre el código ASCII de los siguientes caracteres: A B C a b c 0 1 2 \$ * + / y el carácter en blanco.
- Muestra por pantalla el resultado de la siguiente forma:

El código ASCII de A es: XX

El código ASCII de B es: XX

El código ASCII de C es: XX

El código ASCII de a es: XX ... etc

EJERCICIOS

- **Ejercicio 31.- (OPTATIVO):** Escriba una aplicación JAVA que muestre un cuadro, un óvalo, una flecha y un diamante usando asteriscos (*), como se muestra a continuación:



EJERCICIOS

- **Ejercicio 32.- (OBLIGATORIO):** Realiza un programa que dado un importe en euros nos indique número óptimo de billetes de 50, 20, 10 y 5, así como la cantidad sobrante en monedas de 2 y de 1 euro.
- Por ejemplo:

Por favor, indique una cantidad de dinero: 232

232 Euros se descomponen en **4** billetes de 50, **1** billetes de 20, **1** billetes de 10, **0** billetes de 5, **1** monedas de 2 euros y **0** monedas de 1 euro.