Tema 02. Introducción al lenguaje Java

Índice

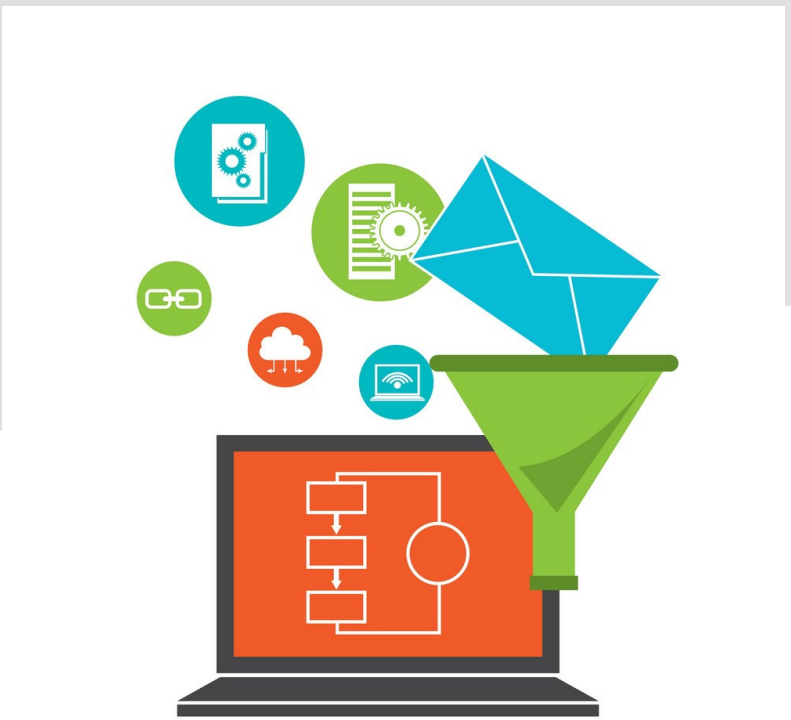
1.- Características Java.

- **Orientado a objeto.**
- **Simple y seguro:** funcionalidad de un lenguaje potente sin las características más confusas de estos.
- **Interpretado**.
- **Arquitectura independiente**: se compila el código a un fichero objeto independiente de la arquitectura de la máquina en la que se ejecutará.
- **Difusión y portabilidad:** lenguaje más usado actualmente.
- Proporciona herramientas para realizar programas distribuidos y programas multitarea.

NOTA: Cualquier máquina que tenga el sistema de ejecución (JRE – Java Runtime Enviroment) puede ejecutar el código objeto.

2.- Tipos de datos básicos.

- **Enteros**: byte, short, int, long.
- **Reales**: float, double.
- **Caracter**: char.
- **Lógico**: boolean.
- **Vacio**: void.



2.- Tipos de datos básicos.

TIPO	TAMAÑO	DESCRIPCIÓN
byte	8	Entero de -128 a 127
short	16	Entero de -32.768 a 32.767
int	32	Entero de -2.147.483.648 a 2.147.483.647
long	64	Entero con mayor rango (no cabe aquí)
float	32	Real en notación coma flotante (6 o 7 dígitos importantes)
double	64	Real en notación coma flotante (15 dígitos importantes)
boolean	1	Lógico con valores True o False
char	16	Caracteres ASCII
void	-	Tipo especial “vació”

3.- Identificadores y comentarios.

Identificadores:

- Empiezan por una letra, un subrayado (_) o un dólar ($). Los siguientes pueden ser letras o números.
- Distingue entre mayúsculas y minúsculas.
- No puede coincidir con palabras clave del lenguaje.

Comentarios:

- De bloque */* Comentario */*
- De línea *// Comentario*

```
Edad: ENTERO
2edad: REAL // NO VALIDO
_edad: ENTERO

SI expLogica
  /* En este caso entra
  solo si se cumple la
  Condición */
  Instrucción 1
  Instrucción 2
FIN SI
```

4.- Variables, constantes y expresiones.

Declaración de variables

Pseudocódigo

```
idVariable: ENTERO
```

Java

```
int idVariable;
```

Declaración de constantes

Pseudocódigo

```
PI = 3.14

MAXIMO = 10
```

Java

```
private static final float PI = 3.14;

private static final int MAXIMO = 10;
```

4.- Variables, constantes y expresiones.

Expresiones y operaciones en Java

TIPO	***SÍMBOLO***	***DESCRIPCIÓN***
Aritméticos	+ - * /	Suma, resta, multiplicación y división.
	%	Resto de división (módulo).
	Math.sqrt(x)	Raíz cuadrada.
	Math.pow(x,y)	Potencia (x elevado a y)
	++ --	Incremento, decremento
Relacionales	< > <= >=	Mayor que, menor que, menor o igual, mayor o igual
	== !=	Igual, distinto
Lógicos	&& || !	AND (y), OR (o), NOT (no)

5.- Formato programa en Java.

Librerías utilizadas

Nombre del archivo

Punto de inicio

Argumentos de entrada

```java
import java.util.*;

public class PrimerPrograma {
    // Declaración de constantes

    public static void main(String[] args) {
        // Declaración de variables
        /* Instrucciones */
    }
}
```

6.- Instrucciones de E/S.

Escribir en pantalla

Pseudocódigo

```
ESCRIBIR “HOLA”
```

Java

```java
//Escribe Hola y un salto de línea
System.out.println("Hola");
```

```
ESCRIBIR “Su saldo es:”, saldo
```

```java
System.out.print("Su saldo es"+saldo);
```

6.- Instrucciones de E/S.

Escribir en pantalla

Pseudocódigo

```
ESCRIBIR_SS “HOLA”
```

Java

```java
//Escribe Hola
System.out.print("Hola");
```

Escribir con formato

```java
//Escribe el saldo con 2 decimales
System.out.printf("Su saldo es: %.2f \n", saldo);
//  \n es un salto de linea
```

6.- Instrucciones de E/S.

Leer datos desde el teclado

En Java, para poder leer datos a través de la consola hay que crear un objeto que representa al teclado. Esto solo habrá que hacerlo una vez, e irá en la línea justo anterior al main.

```
private static Scanner teclado = new Scanner(System.in);
```

Leer un carácter

```
char caracter;
caracter = teclado.next().charAt(0);
```

6.- Instrucciones de E/S.

Leer un entero

```
int num;
System.out.println("Introduce un numero:");
num = Integer.parseInt(teclado.nextLine());
```

```
int num;
System.out.println("Introduce un numero:");
num = teclado.nextInt();
```

Leer un doble

```
double num;
System.out.println("Introduce un numero:");
num = Double.parseDouble(teclado.nextLine());
```

```
double num;
System.out.println("Introduce un numero:");
num = teclado.nextDouble();
```

7.- Conversión de tipos.

En Java se realiza conversión de tipos implícita entre tipos numéricos de menor a mayor rango. También entre los tipos int y char.

Explicita: permite convertir un tipo de mayor jerarquía a otro de menor jerarquía.

```
variable1 = ( tipo ) variable2;
```

Ejemplo:

```java
int numEntero = 3;
double numDouble = 5.2;
numEntero = numDouble; // ERROR
numEntero = (int) numDouble; // entero valdrá 5. Explícita
numDouble = numEntero; // real valdría 5.0. Implícita
```

7.- Conversión de tipos.

Conversión de tipos numéricos

El resultado de cualquier expresión es del tipo correspondiente al del operando de mayor jerarquía, en el orden: double, float, int

Ejemplo

```
int numEntero = 10, resDivision;
double numReal = 2.0;
resDivision = numEntero / numReal;  //ERROR, El resultado es real (double)
                                    // no se puede asignar a un entero
```

Una variable de un tipo puede recibir un valor de otro tipo si son tipos compatibles (los números lo son) y el tipo del destino es de “mayor jerarquía” que el de origen.

Ejemplo

```
int numEntero = 2;
float numReal = numEntero;
```

7.- Conversión de tipos.

Conversión implícita entre carácter y entero

En Java, una variable entera puede asignarse a un carácter y viceversa. La conversión se realizará a través del código ASCII.

Ejemplos:

```
int entero = 65;
char caracter;
caracter = (char) entero; //El carácter toma el valor 'A'

caracter = 'Z';
entero = caracter; //El entero toma el valor 90
```

7.- Conversión de tipos.

Cuidado: la división entre variables enteras da un entero, aunque el resultado se asigne a un double o float.

Para que la división entre enteros tenga un resultado con decimales, tendrá que hacerse un ***casting.***

Ejemplo:

```
double numDouble;

numDouble = 10 / 4; //toma el valor 2.0
numDouble = 10.0 / 4; //toma el valor 2.5
numDouble = (double) 10 / 4; //toma el valor 2.5
numDouble = (double) ( 10 / 4 ); //toma el valor 2.0
```

8.- Estructuras de control.

Alternativa simple

Pseudocódigo

```
SI expLogica
  Instrucción 1
  Instrucción 2
  Instrucción 3
  …
FIN SI
```

Java

```java
if ( condicion ) {
    //Instrucción 1
    //Instrucción 2
    //Instrucción 3
    // …
}
```

8.- Estructuras de control.

Alternativa doble

Pseudocódigo

```
SI expLogica
  Instrucción 1
  Instrucción 2
SI NO
  Instrucción 3
  Instrucción 4
FIN SI
```

Java

```
if ( condicion ) {
    //Instrucción 1
    //Instrucción 2
    // …
}
else {
    //Instrucción 3
    //Instrucción 4
    // …
}
```

8.- Estructuras de control.

Alternativa múltiple

Pseudocódigo

```
SEGÚN_VALOR expLogica
  Valor1:
    Bloq. Instr. 1
  Valor2:
    Bloq. Instr. 2
  ...
  OTROS:
    Bloq. Instr. otros
FIN SEGÚN_VALOR
```

Java

```
switch ( expresion ) {
    case valor1:
        //Bloque Instrucción 1
        break;
    case valor2:
        //Bloque Instrucción 2
        break;
    // ...
    default:
        //Bloque instrucciones "otros"
}
```

8.- Estructuras de control.

Mientras (while)

Pseudocódigo

```
MIENTRAS condición
  Instrucción 1
  Instrucción 2
  Instrucción 3
  …
FIN MIENTRAS
```

Java

```java
while ( condicion ) {
    //Instrucción 1
    //Instrucción 2
    //…
    //ACTUALIZAR CONDICIÓN
}
```

LA CONDICIÓN DEBE ACTUALIZARSE PARA NO CREAR UN BUCLE INFINITO

8.- Estructuras de control.

Repetir...mientras (do...while)

Pseudocódigo

```
REPETIR
  Instrucción 1
  Instrucción 2
  Instrucción 3
  …
MIENTRAS expLogica
```

Java

```
do {
    //Instrucción 1
    //Instrucción 2
    //…
    //ACTUALIZAR CONDICIÓN

} while ( condicion );
```

LA CONDICIÓN DEBE ACTUALIZARSE PARA NO CREAR UN BUCLE INFINITO

8.- Estructuras de control.

Para o desde (for)

Pseudocódigo

```
PARA vcont DE vIni A vFin CON INC = num
  Bloque de instrucciones
FIN PARA
```

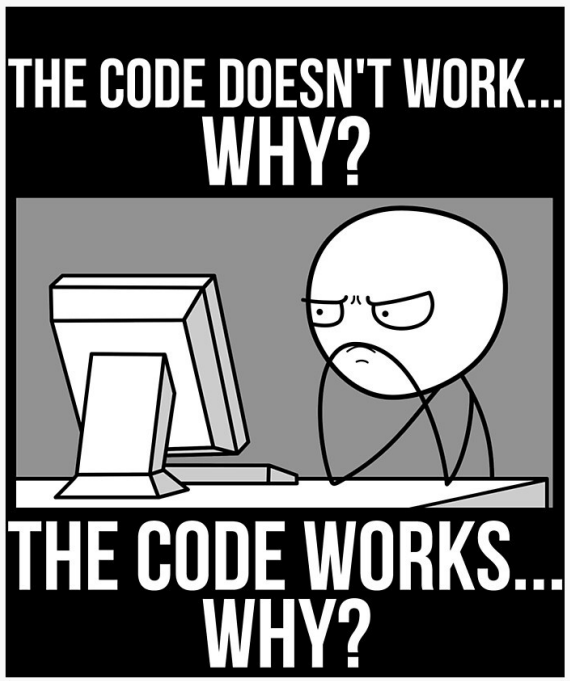
Java

```
for ( inicializa, condicion, incrementa ) {
    /* Bloque de instrucciones */
}
```

Ejemplo:

```
acum <-  0
PARA i DE 1 A 5 CON INC = 1
  acum <- acum + i
FIN PARA
```

```
int acum = 0;
for ( int i = 1; i <= 5; i++ ) {
    acum = acum + i;
}
```


THE CODE DOESN'T WORK...
WHY?
THE CODE WORKS...
WHY?