

Tema 2. Introducción a Java

Segunda parte

Elementos básicos de Java

☐ Elementos básicos de Java

- ☐ Paquetes
- ☐ Clase
- ☐ Comentarios
- ☐ Identificadores
- ☐ Tipos simples
 - ☐ Enumerados
- ☐ Instrucciones
- ☐ Métodos
- ☐ Matrices
- ☐ String
- ☐ Entrada y salida

Paquetes en Java

- ❑ Los paquetes son agrupaciones de clases que pueden, a su vez, contener otros paquetes.
 - ❑ Las bibliotecas estándar se organizan en paquetes jerárquicos
 - ❑ java, javax, ...
 - ❑ Los paquetes garantizan la singularidad de los nombres de clase
 - ❑ Se puede dar nombre único a cualquier paquete añadiendo el nombre del dominio Internet en orden inverso
 - ❑ com.empresa.paquete
 - ❑ Se pueden acortar los nombres de las clases importándolas previamente
 - ❑ Usar java.util.Random
 - ❑ import java.util.Random;
 - ❑ usar Random

Tema 2.2 - 3

Paquetes en Java

- ❑ Creación de paquetes
 - ❑ Primera línea de cada clase:
 - ❑ package nombrePaquete;
 - ❑ Se asocia al directorio nombrePaquete
 - ❑ Todas las clases del paquete en ese directorio

Tema 2.2 - 4

Elementos básicos de Java – Clase

```
package tp.MisEjemplos;
```

```
public class NombreClase
```

```
{
```

```
    atributos
```

```
    método(parámetros)
```

```
{
```

```
        variables locales
```

```
        instrucciones
```

```
}
```

```
}
```

Tema 2.2 - 5

Elementos básicos de Java – Clase

❑ Ámbitos

- ❑ variables locales => bloque => método
 - ❑ No se inicializan por defecto (error compilación)
- ❑ parámetros => método
 - ❑ Son siempre por valor
- ❑ atributos => toda la clase
 - ❑ Se inicializan por defecto

Tema 2.2 - 6

Elementos básicos de Java – Comentarios

- ❑ En Java hay tres tipos de comentarios:

```
// comentarios para una sola línea
/* comentarios de una o
    más líneas
*/
```

Tema 2.2 - 7

Elementos básicos de Java – Comentarios de documentación

- ❑ Son utilizados por la herramienta javadoc para generar documentación en formato HTML

```
/** comentario de documentación, de una o
    más líneas justo antes del elemento
    (clase, atributo, método)
*/
```

- ❑ Puede llevar etiquetas

- ❑ En las clases

- ❑ @autor <nombre autor>
- ❑ @versión

- ❑ En los métodos

- ❑ @param, @return, @exception

Tema 2.2 - 8

Elementos básicos de Java – Identificadores

- ❑ Los identificadores nombran
 - ❑ variables, atributos, métodos, clases y objetos;
 - ❑ cualquier cosa que el programador necesite identificar
 - ❑ Por tanto, deben ser autoidentificativos.
- ❑ Un identificador comienza con
 - ❑ una letra, un subrayado (_) o un símbolo de dólar (\$).
 - ❑ Los siguientes caracteres pueden ser letras o dígitos.
 - ❑ Se distinguen las mayúsculas de las minúsculas y no hay longitud máxima.
 - ❑ No puede coincidir con un palabra reservada

Tema 2.2 - 9

Elementos básicos de Java – Identificadores

- ❑ Convenios:
 - ❑ nombre de CLASE empieza por Mayúsculas
 - ❑ nombre de MÉTODOS y ATRIBUTOS por minúsculas
 - ❑ las CONSTANTES con mayúsculas completamente
 - ❑ Comienzo de nueva palabra en mayúsculas: `dibujaRectangulo`
- ❑ <http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html>

Tema 2.2 - 10

Elementos básicos de Java – Tipos simples

❑ Tipos primitivos

| <u>Tipo</u> | <u>Tamaño</u> | <u>Valor mínimo</u> |
|-------------|---------------|---|
| byte | 8 bits | -128 |
| short | 16 bits | -32768 |
| int | 32 bits | -2147483648 |
| long | 64 bits | < -9×10^{18} |
| float | 32 bits | +/- 3.4×10^{38} con 7 dígitos significativos |
| double | 64 bits | +/- 1.7×10^{308} con 15 dígitos significativos |
| boolean | 1 bit | true false |
| char | 16 bits | Codificación UNICODE |

Tema 2.2 - 11

Elementos básicos de Java – Tipos simples

❑ Constantes

- ❑ `final int NUMERO = 3;`
- ❑ `final String SALUDO = "hola";`

Tema 2.2 - 12

Elementos básicos de Java – Tipos simples

❑ Expresiones

- ❑ Java es fuertemente tipado \Rightarrow cada expresión tiene un tipo que se determina durante la compilación
- ❑ Una expresión puede ser
 - ❑ Un literal: 1, 1.0, true, 'n', etc.
 - ❑ El tipo de esta expresión es el tipo del literal
 - ❑ Una variable o una constante: i, s, a
 - ❑ El tipo de esta expresión es el tipo con que se declaró aquella variable o constante
 - ❑ El resultado de una operación: i+1, etc.
 - ❑ El tipo se determina en función de los tipos de las expresiones que se indican como argumentos de la operación

Tema 2.2 - 13

Elementos básicos de Java – Tipos simples

- ❑ Un tipo A es de mayor rango que un tipo B si A es un superconjunto de B \Rightarrow las variables de tipo B siempre se pueden asignar a variables de tipo A
 - ❑ double > float > long > int > short > byte

Tema 2.2 - 14

Elementos básicos de Java – Tipos simples

❑ Expresiones – operadores aritméticos

- ❑ Al operar con byte y short, estos se convierten implícitamente a int
- ❑ Cuando los tipos de los operandos no coinciden, el operando de menor rango se convierte implícitamente al tipo de mayor rango
- ❑ El resultado de la operación es del tipo de mayor rango

| Símbolo | Operación |
|---------|----------------|
| + | Suma |
| - | Resta |
| * | Multiplicación |
| / | División |
| % | Resto |

Tema 2.2 - 15

Elementos básicos de Java – Tipos simples

❑ Ejemplos

```
int a = 1, b = 2;
int c = a + b;
short s = 1;
int d = s + c; // s se convierte a int
double f = 1.0 + a;
//a se convierte a double

float g = 1.0F;
long l = 23L;

char c = 'a';
```

Tema 2.2 - 16

Elementos básicos de Java – Tipos simples

Expresiones – operadores aritméticos

Operadores unarios sobre expresiones

| Símbolo | Operación | Argumento |
|---------|-----------------|-----------------------------|
| + exp | nada | int long float double |
| - exp | cambio de signo | int long float double |

Operadores sobre variables enteras

| Símbolo | Operación | Argumento |
|---------|----------------|--|
| ++ var | preincremento | int short byte long float double |
| -- var | predecremento | int short byte long float double |
| var ++ | postincremento | int short byte long float double |
| var -- | postdecremento | int short byte long float double |

Tema 2.2 - 17

Elementos básicos de Java – Tipos simples

❑ Ejemplos

```
int a = 1, b = 2;
```

```
int c = +a;
```

```
int d = -(c+1);
```

```
b = a++; // b = 1, a = 2
```

```
a = --b; // a = b = 0
```

```
(a+1)++; // error a+1 no es una variable
```

```
double f = 1.0;
```

```
f++; // f = 2.0
```

Tema 2.2 - 18

Elementos básicos de Java – Tipos simples

❑ Expresiones – operadores relacionales y booleanos

❑ (menor prioridad)

| Símbolo | Operación | Símbolo | Operación |
|---------|---------------|--------------|---------------------------|
| > | mayor | exp1 && exp2 | y-lógico (circuito corto) |
| < | menor | exp1 & exp2 | y-lógico (circuito largo) |
| = | igual | exp1 exp2 | o-lógico (circuito corto) |
| <= | menor o igual | exp1 exp2 | o-lógico (circuito largo) |
| == | igual | ! exp | negación |
| != | distinto | | |

Tema 2.2 - 19

Elementos básicos de Java – Tipos simples

❑ Prioridad de operadores

- ❑ [] . ()
- ❑ ! ~ ++ -- -(unario) () (casting) new
- ❑ * / %
- ❑ + -
- ❑ << >> >>>
- ❑ < <= > >= instanceof
- ❑ == !=
- ❑ &
- ❑ ^
- ❑ |
- ❑ &&
- ❑ ||
- ❑ = += -= *= /= %= ...

Tema 2.2 - 20

Elementos básicos de Java – Tipos simples

❑ Ejemplos (prioridad de operadores)

```
int a=1, b=2;
```

```
boolean v= ! a<=b && a==5 || b!=4;
```

```
boolean v = ! (a<=b && a==5) || b!=4;
```

```
boolean v= !(((a<=b) && (a==5)) || (b!=4));
```

```
boolean w= a<=b && a;
```

```
// error a no es boolean
```

Tema 2.2 - 21

Elementos básicos de Java – Tipos simples

❑ Conversiones \Rightarrow cast

❑ Java realiza conversiones implícitas de menor a mayor rango

❑ Todas las conversiones de mayor a menor rango deben ser explícitas por medio del operador cast

❑ (tipo) expresión

❑ Eventualmente hay pérdida de significación de punto flotante a entero

❑ También se puede hacer casting de int a char

❑ posición unicode

Tema 2.2 - 22

Elementos básicos de Java – Enumerados

- ❑ A partir de jdk1.5.0
 - ❑ Conjunto de valores constantes
- ❑ `public enum Dia {LUNES, MARTES, MIÉRCOLES, JUEVES, VIERNES};`
- ❑ `Dia dia = Dia.MARTES;`
- ❑ `System.out.println(dia);`
- ❑ En realidad se declara una clase cuyos atributos son los nombres de los valores.

Tema 2.2 - 23

Elementos básicos de Java – Instrucciones

- ❑ Instrucciones básicas
 - ❑ Asignación:
`var = expresión; // asignación`

`a = b + 1;`
 - ❑ Bloque:
`{ inst1 inst2 ... instn }`

Tema 2.2 - 24

Elementos básicos de Java – Instrucciones

❑ Ejecución condicional

```
if (expresión booleana)
    instrucción
```

```
int a, b;
```

```
...
```

```
if (a == 1) b = 2;
```

```
if (expresión booleana)
    instrucción
```

```
else (else if)
    instrucción
```

Tema 2.2 - 25

Elementos básicos de Java – Instrucciones

❑ Ejemplos

```
boolean v;
```

```
int a, b;
```

```
...
```

```
if (v) b = 2;
```

```
else b = 3;
```

```
if (a) b = 2;           // error, a no es boolean
```

```
if (a = 1) b = 4;       // error, a = 1 no es boolean
```

```
                        // es una asignación
```

Tema 2.2 - 26

Elementos básicos de Java – Instrucciones

❑ Selección switch/case

switch (expresión entera, carácter o enumerado)

```
{
    case expresión-constante: instrucciones
    case expresión-constante: instrucciones
    ...
    default: instrucciones
}
```

- ❑ Si no se especifica break al final de un caso, se continúa ejecutando las instrucciones del siguiente caso ⇒ se puede colocar un solo grupo de instrucciones para un conjunto de casos.

Tema 2.2 - 27

Elementos básicos de Java – Instrucciones

❑ Ejemplo

```
switch (c)
{
    case 'a':
    case 'b':
    case 'c':
        System.out.println("a, b o c");
        break;
    case 'd':
        System.out.println("d");
        // No termina en break
    case 'e':
        System.out.println("d o e!!");
        break;
    default:
        System.out.println("Ninguna");
}
```

Tema 2.2 - 28

Elementos básicos de Java – Instrucciones

❑ Ciclos simples

while (expresión booleana)
 instrucción

```
int a = 1;  
while (a <= 5) a++;
```

```
do { instrucciones }  
while (expresión booleana);
```

```
do  
    a--;  
while (a > 0);
```

Tema 2.2 - 29

Elementos básicos de Java – Instrucciones

❑ Ciclo for

```
for (instrucción;expresión booleana;instrucción)  
    instrucción  
for (int a = 1; a <= 5; a++)  
    b++;
```

❑ Para recorrer estructuras

```
for (variable : estructura)  
    instrucción  
for(String a : args)  
    System.out.println(a);
```

Tema 2.2 - 30

Elementos básicos de Java – Métodos

❑ Métodos de clase (estáticos)

```
static tipoDevuelto método (tipo1 nombre1; tipo2 nombre2)
```

```
{  
    instrucciones;  
    (return valor)  
}
```

```
static void imprimir (int valor)
```

```
{  
    System.out.println(valor);  
}
```

Tema 2.2 - 31

Elementos básicos de Java – Métodos

❑ Invocación de métodos de clase (estáticos)

❑ Desde algún método de la misma clase

```
public static void main(String[] args)
```

```
{  
    int valor = 23;  
    imprimir(valor);  
}
```

Tema 2.2 - 32

Elementos básicos de Java – Matrices

❑ Declaración de una matriz

```
int A[]; // A es una matriz de enteros
int A[10]; // error, el tamaño no va aquí
int[] A; // A es una matriz de enteros
int A,B[]; // A es un entero, B una matriz
int[] A,B; // A y B son matrices
int [][] C; // dos dimensiones

A[1] = 3; // error, la matriz no ha sido creada
```

❑ Creación de una matriz

```
int A[] = new int[10]; // 10 enteros
```

Tema 2.2 - 33

Elementos básicos de Java – Matrices

❑ Subindización de una matriz

```
A[0], A[1], ..., A[9] // correcto
A[10] // error, índice fuera de rango(excepción)
```

❑ Modificación de una matriz

```
A[3] = 8;
```

❑ Tamaño de una matriz

```
A.length // correcto
A.length() // error, sin paréntesis
```

Tema 2.2 - 34

Elementos básicos de Java – Matrices

❑ Inicialización de una matriz

```
for (int i = 0; i < A.length; i++)  
    A[i] = 2*i;  
for (int i: A)  
    System.out.println(i);
```

❑ Copia de una matriz

```
B = (int[]) A.clone();
```

Tema 2.2 - 35

Elementos básicos de Java – Strings

❑ Clase String => java.lang.String

❑ [api\java\lang\String.html](#)

❑ Declaración de una cadena

```
String s = "Hola, "; // asignación  
String v = new String("Hola");  
String r = "esto es un string";  
String t;
```

❑ Mostrar una cadena

```
System.out.println(s);  
System.out.println(t); // Error t no tiene valor
```

Tema 2.2 - 36

Elementos básicos de Java – Strings

❑ Asignación de cadenas

```
t = "Adiós";
```

❑ Concatenación de cadenas:

```
s + r // "Hola, esto es un string"
```

❑ El resultado de esta expresión es de tipo String

```
t = s + "que tal"; // "Hola, que tal"
```

Tema 2.2 - 37

Elementos básicos de Java – Strings

❑ Tamaño de una cadena

```
s.length() // 6
```

❑ El resultado de esta expresión es de tipo int

❑ Extracción de un carácter

```
s[0] => error  
s.charAt(0) // 'H'  
s.charAt(4) // ','
```

❑ El resultado de esta expresión es de tipo char

Tema 2.2 - 38

Elementos básicos de Java – Strings

❑ Extracción de subsecuencias

```
s.substring(2,4) // "la,"
```

❑ El resultado de esta expresión es de tipo String

❑ Comparación de cadenas

```
s.equals("Hola, ") // true  
s == "Hola, "      // false, compara referencias  
s.compareTo(r)     // < 0, == 0 ó > 0
```

Tema 2.2 - 39

Elementos básicos de Java – Strings

❑ Conversión a String

```
int n = 5;  
System.out.println("n = " + n); // n = 5  
String t = "El numero n es " + n
```

- ❑ Si uno de los operandos de + es de tipo String, el otro se convierte implícitamente a String (método toString)

❑ Operaciones prohibidas

```
s[0] = 'h'; // error - no se puede modificar un carácter  
s.charAt(0) = 'h'; // error - ídem  
s = s+1; // error
```

❑ método main(String args[]) => array de Strings

Tema 2.2 - 40

Entrada y salida

- ❑ Para imprimir por pantalla se utiliza `System.out` y `System.err` que hacen las veces de `std::cout` y `std::cerr`.
- ❑ Para escribir una línea, se utiliza el método `println`:
 - ❑ `System.out.println (" Hola mundo ");`
- ❑ Para leer por teclado se puede utilizar `System.in`, que permite leer carácter a carácter:
 - ❑ `System.in.read ();`
- ❑ En vez de eso, también se puede utilizar la clase `java.util.Scanner` que permite leer cómodamente enteros, reales, cadenas, etc.

```
Scanner entrada = new Scanner(System.in);  
int a = entrada.nextInt();
```