

Fundamentos del Lenguaje Java

1. Estructura de un Programa Java

1.1 Estructura Básica

Todo programa Java debe seguir una estructura específica. El elemento fundamental es la **clase**, y dentro de ella encontramos el método `main`, que es el punto de entrada del programa.

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

Elementos clave:

- `public class MiPrimerPrograma`: Declaración de la clase. El nombre debe coincidir con el nombre del archivo (.java)
- `public`: Modificador de acceso que indica que es accesible desde cualquier lugar
- `static`: El método pertenece a la clase, no a una instancia
- `void`: El método no devuelve ningún valor
- `main`: Nombre del método principal
- `String[] args`: Parámetro que recibe argumentos desde la línea de comandos

1.2 Comentarios

Los comentarios son esenciales para documentar el código:

```
public class EjemploComentarios {  
    // Comentario de una línea  
  
    /* Comentario de  
       múltiples líneas */  
  
    /**  
     * Comentario de documentación (JavaDoc)  
     * Se usa para generar documentación automática  
     */  
    public static void main(String[] args) {  
        System.out.println("Programa con comentarios");  
    }  
}
```

2. Tipos Primitivos

Java dispone de 8 tipos primitivos que representan valores simples:

2.1 Tipos Numéricos Enteros

```

public class TiposEnteros {
    public static void main(String[] args) {
        // byte: 8 bits (-128 a 127)
        byte edad = 25;

        // short: 16 bits (-32,768 a 32,767)
        short temperatura = -273;

        // int: 32 bits (-2^31 a 2^31-1)
        int habitantes = 47000000;

        // long: 64 bits (-2^63 a 2^63-1)
        long distanciaEstrella = 9460730472580800L; // Nota la 'L' al
final

        System.out.println("Edad: " + edad);
        System.out.println("Temperatura: " + temperatura);
        System.out.println("Habitantes: " + habitantes);
        System.out.println("Distancia: " + distanciaEstrella);
    }
}

```

2.2 Tipos Numéricos Decimales

```

public class TiposDecimales {
    public static void main(String[] args) {
        // float: 32 bits (precisión simple)
        float precio = 19.99f; // Nota la 'f' al final

        // double: 64 bits (precisión doble)
        double pi = 3.141592653589793;

        System.out.println("Precio: " + precio);
        System.out.println("Pi: " + pi);
    }
}

```

2.3 Tipo Carácter y Booleano

```

public class TiposCaracterBooleano {
    public static void main(String[] args) {
        // char: 16 bits (caracteres Unicode)
        char inicial = 'A';
        char simbolo = '\u00A9'; // © en Unicode

        // boolean: true o false
        boolean esMayorDeEdad = true;
        boolean tieneLicencia = false;

        System.out.println("Inicial: " + inicial);
        System.out.println("Símbolo: " + simbolo);
        System.out.println("Mayor de edad: " + esMayorDeEdad);
        System.out.println("Tiene licencia: " + tieneLicencia);
    }
}

```

3. El Tipo String

String no es un tipo primitivo, sino una clase que representa cadenas de texto.

3.1 Declaración y Uso Básico

```
public class EjemploString {
    public static void main(String[] args) {
        // Diferentes formas de declarar Strings
        String nombre = "Carlos";
        String apellidos = "García López";
        String vacio = "";
        String nulo = null; // Sin valor asignado

        // Concatenación
        String nombreCompleto = nombre + " " + apellidos;
        System.out.println("Nombre completo: " + nombreCompleto);

        // Métodos útiles
        System.out.println("Longitud: " + nombreCompleto.length());
        System.out.println("Mayúsculas: " +
nombreCompleto.toUpperCase());
        System.out.println("Minúsculas: " +
nombreCompleto.toLowerCase());
        System.out.println("Contiene 'García': " +
nombreCompleto.contains("García"));
    }
}
```

3.2 Métodos Importantes de String

```
public class MetodosString {
    public static void main(String[] args) {
        String texto = "  Programación en Java  ";

        // Eliminar espacios al inicio y final
        String sinEspacios = texto.trim();
        System.out.println("Sin espacios: '" + sinEspacios + "'");

        // Subcadenas
        String subcadena = sinEspacios.substring(0, 12);
        System.out.println("Subcadena: " + subcadena);

        // Reemplazar
        String reemplazado = sinEspacios.replace("Java", "Python");
        System.out.println("Reemplazado: " + reemplazado);

        // Comparación
        String str1 = "Hola";
        String str2 = "hola";
        System.out.println("Iguales: " + str1.equals(str2)); // false
        System.out.println("Iguales (ignorando mayúsculas): " +
            str1.equalsIgnoreCase(str2)); // true

        // Dividir cadena
        String frutas = "manzana,pera,naranja";
        String[] arrayFrutas = frutas.split(",");
        for (String fruta : arrayFrutas) {
            System.out.println("Fruta: " + fruta);
        }
    }
}
```

```
}
```

4. Expresiones

Las expresiones son combinaciones de operadores y operandos que producen un resultado.

4.1 Operadores Aritméticos

```
public class OperadoresAritmeticos {
    public static void main(String[] args) {
        int a = 10;
        int b = 3;

        // Operaciones básicas
        System.out.println("Suma: " + (a + b));           // 13
        System.out.println("Resta: " + (a - b));         // 7
        System.out.println("Multiplicación: " + (a * b)); // 30
        System.out.println("División: " + (a / b));       // 3
        (división entera)
        System.out.println("Módulo: " + (a % b));         // 1 (resto)

        // División decimal
        double c = 10.0;
        double d = 3.0;
        System.out.println("División decimal: " + (c / d)); //
        3.333...

        // Operadores de incremento/decremento
        int contador = 5;
        System.out.println("contador++: " + contador++); // 5 (post-
incremento)
        System.out.println("contador: " + contador);      // 6
        System.out.println("++contador: " + ++contador); // 7 (pre-
incremento)
    }
}
```

4.2 Operadores Relacionales y Lógicos

```
public class OperadoresRelacionalesLogicos {
    public static void main(String[] args) {
        int x = 5;
        int y = 10;

        // Operadores relacionales
        System.out.println("x == y: " + (x == y)); // false
        System.out.println("x != y: " + (x != y)); // true
        System.out.println("x < y: " + (x < y));    // true
        System.out.println("x > y: " + (x > y));    // false
        System.out.println("x <= y: " + (x <= y)); // true
        System.out.println("x >= y: " + (x >= y)); // false

        // Operadores lógicos
        boolean condicion1 = true;
        boolean condicion2 = false;
```

```

        System.out.println("AND (&&): " + (condicion1 && condicion2));
// false
        System.out.println("OR (||): " + (condicion1 || condicion2));
// true
        System.out.println("NOT (!): " + !condicion1);
// false

        // Ejemplo práctico
        int edad = 20;
        boolean tieneCarnet = true;
        boolean puedeConducir = (edad >= 18) && tieneCarnet;
        System.out.println("¿Puede conducir? " + puedeConducir); //
true
    }
}

```

4.3 Operadores de Asignación

```

public class OperadoresAsignacion {
    public static void main(String[] args) {
        int numero = 10;
        System.out.println("Valor inicial: " + numero);

        numero += 5; // numero = numero + 5
        System.out.println("Después de +=5: " + numero); // 15

        numero -= 3; // numero = numero - 3
        System.out.println("Después de -=3: " + numero); // 12

        numero *= 2; // numero = numero * 2
        System.out.println("Después de *=2: " + numero); // 24

        numero /= 4; // numero = numero / 4
        System.out.println("Después de /=4: " + numero); // 6

        numero %= 4; // numero = numero % 4
        System.out.println("Después de %=4: " + numero); // 2
    }
}

```

5. Conversiones de Tipo

5.1 Conversión Implícita (Widening)

La conversión automática ocurre cuando no hay pérdida de información.

```

public class ConversionImplicita {
    public static void main(String[] args) {
        // De menor a mayor capacidad
        byte b = 100;
        short s = b;    // byte -> short
        int i = s;       // short -> int
        long l = i;      // int -> long
        float f = l;     // long -> float
        double d = f;    // float -> double

        System.out.println("byte: " + b);
        System.out.println("short: " + s);
    }
}

```

```

        System.out.println("int: " + i);
        System.out.println("long: " + l);
        System.out.println("float: " + f);
        System.out.println("double: " + d);

        // En expresiones mixtas
        int entero = 10;
        double decimal = 5.5;
        double resultado = entero + decimal; // int se convierte a
double
        System.out.println("Resultado: " + resultado); // 15.5
    }
}

```

5.2 Conversión Explícita (Narrowing/Casting)

Se requiere cuando puede haber pérdida de información.

```

public class ConversionExplicita {
    public static void main(String[] args) {
        // Casting de tipos numéricos
        double decimal = 9.78;
        int entero = (int) decimal; // Se pierden los decimales
        System.out.println("double " + decimal + " -> int " + entero);

// 9

        // Casting con pérdida de precisión
        long numeroGrande = 1000000000000000L;
        int numeroReducido = (int) numeroGrande;
        System.out.println("long: " + numeroGrande);
        System.out.println("int (con overflow): " + numeroReducido);

        // Casting de char a int y viceversa
        char letra = 'A';
        int codigoAscii = (int) letra;
        System.out.println("Carácter: " + letra);
        System.out.println("Código ASCII: " + codigoAscii); // 65

        int codigo = 97;
        char caracter = (char) codigo;
        System.out.println("Código " + codigo + " -> Carácter: " +
caracter); // 'a'
    }
}

```

5.3 Conversión de String a Tipos Primitivos

```

public class ConversionString {
    public static void main(String[] args) {
        // String a números
        String textoEntero = "123";
        String textoDecimal = "45.67";
        String textoBooleano = "true";

        // Parsing
        int entero = Integer.parseInt(textoEntero);
        double decimal = Double.parseDouble(textoDecimal);
        boolean booleano = Boolean.parseBoolean(textoBooleano);
    }
}

```

```

System.out.println("String \"123\" -> int: " + entero);
System.out.println("String \"45.67\" -> double: " + decimal);
System.out.println("String \"true\" -> boolean: " + booleano);

// Números a String
int numero = 100;
String textoNumero = String.valueOf(numero);
// O también:
String textoNumero2 = Integer.toString(numero);
// O concatenando con cadena vacía:
String textoNumero3 = numero + "";

System.out.println("int 100 -> String: " + textoNumero);

// Manejo de errores en conversión
try {
    String textoInvalido = "abc";
    int invalido = Integer.parseInt(textoInvalido);
} catch (NumberFormatException e) {
    System.out.println("Error: No se puede convertir 'abc' a
número");
}
}
}

```

6. Enumerados (enum)

Los enumerados son tipos especiales que representan un conjunto fijo de constantes.

6.1 Declaración y Uso Básico

```

// Definición del enum
enum DiaSemana {
    LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, SABADO, DOMINGO
}

public class EjemploEnum {
    public static void main(String[] args) {
        // Usar el enum
        DiaSemana hoy = DiaSemana.MIERCOLES;
        DiaSemana finDeSemana = DiaSemana.SABADO;

        System.out.println("Hoy es: " + hoy);
        System.out.println("Fin de semana: " + finDeSemana);

        // Comparación
        if (hoy == DiaSemana.MIERCOLES) {
            System.out.println("Mitad de semana");
        }

        // Switch con enum
        switch (hoy) {
            case LUNES:
            case MARTES:
            case MIERCOLES:
            case JUEVES:
            case VIERNES:
                System.out.println("Día laborable");
                break;
        }
    }
}

```

```

        case SABADO:
        case DOMINGO:
            System.out.println("Fin de semana");
            break;
    }

    // Recorrer todos los valores
    System.out.println("\nTodos los días:");
    for (DiaSemana dia : DiaSemana.values()) {
        System.out.println("- " + dia);
    }
}
}

```

6.2 Enumerados con Propiedades y Métodos

```

enum EstadoPedido {
    PENDIENTE("El pedido está pendiente de procesar"),
    PROCESANDO("El pedido se está procesando"),
    ENVIADO("El pedido ha sido enviado"),
    ENTREGADO("El pedido ha sido entregado"),
    CANCELADO("El pedido ha sido cancelado");

    private String descripcion;

    // Constructor privado
    EstadoPedido(String descripcion) {
        this.descripcion = descripcion;
    }

    // Método getter
    public String getDescripcion() {
        return descripcion;
    }
}

public class EjemploEnumAvanzado {
    public static void main(String[] args) {
        EstadoPedido estado = EstadoPedido.ENVIADO;

        System.out.println("Estado: " + estado);
        System.out.println("Descripción: " + estado.getDescripcion());

        // Obtener el índice
        System.out.println("Posición: " + estado.ordinal()); // 2

        // Convertir String a enum
        String textoEstado = "PROCESANDO";
        EstadoPedido estadoConvertido =
        EstadoPedido.valueOf(textoEstado);
        System.out.println("Estado convertido: " + estadoConvertido);
    }
}

```

7. Petición de Valores Primitivos al Usuario

Para solicitar datos al usuario, utilizamos la clase `Scanner`.

7.1 Uso Básico de Scanner

```
import java.util.Scanner;

public class EntradaDatos {
    public static void main(String[] args) {
        // Crear objeto Scanner
        Scanner scanner = new Scanner(System.in);

        // Leer diferentes tipos de datos
        System.out.print("Introduce tu nombre: ");
        String nombre = scanner.nextLine();

        System.out.print("Introduce tu edad: ");
        int edad = scanner.nextInt();

        System.out.print("Introduce tu altura en metros: ");
        double altura = scanner.nextDouble();

        System.out.print("¿Eres estudiante? (true/false): ");
        boolean esEstudiante = scanner.nextBoolean();

        // Mostrar los datos
        System.out.println("\n--- Datos introducidos ---");
        System.out.println("Nombre: " + nombre);
        System.out.println("Edad: " + edad);
        System.out.println("Altura: " + altura);
        System.out.println("Estudiante: " + esEstudiante);

        // Cerrar el scanner
        scanner.close();
    }
}
```

7.2 Gestión de Problemas Comunes

```
import java.util.Scanner;
import java.util.InputMismatchException;

public class EntradaDatosRobusta {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Problema del buffer con nextLine después de nextInt
        System.out.print("Introduce tu edad: ");
        int edad = scanner.nextInt();
        scanner.nextLine(); // Limpiar el buffer

        System.out.print("Introduce tu nombre: ");
        String nombre = scanner.nextLine();

        // Validación de entrada
        int numero = 0;
        boolean entradaValida = false;

        while (!entradaValida) {
            try {
                System.out.print("Introduce un número entero: ");
                numero = scanner.nextInt();
                entradaValida = true;
            } catch (InputMismatchException e) {
                // Manejo de excepción (no se muestra en el código original)
            }
        }
    }
}
```

```

        } catch (InputMismatchException e) {
            System.out.println("Error: Debes introducir un número
entero.");
            scanner.nextLine(); // Limpiar el buffer
        }
    }

    System.out.println("Número introducido: " + numero);

    scanner.close();
}
}

```

7.3 Ejemplo Completo: Calculadora Simple

```

import java.util.Scanner;

public class CalculadoraSimple {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("=== CALCULADORA SIMPLE ===");

        // Pedir primer número
        System.out.print("Introduce el primer número: ");
        double num1 = scanner.nextDouble();

        // Pedir operación
        System.out.print("Introduce la operación (+, -, *, /): ");
        char operacion = scanner.next().charAt(0);

        // Pedir segundo número
        System.out.print("Introduce el segundo número: ");
        double num2 = scanner.nextDouble();

        // Calcular resultado
        double resultado = 0;
        boolean operacionValida = true;

        switch (operacion) {
            case '+':
                resultado = num1 + num2;
                break;
            case '-':
                resultado = num1 - num2;
                break;
            case '*':
                resultado = num1 * num2;
                break;
            case '/':
                if (num2 != 0) {
                    resultado = num1 / num2;
                } else {
                    System.out.println("Error: División por cero");
                    operacionValida = false;
                }
                break;
            default:
                System.out.println("Operación no válida");
                operacionValida = false;
        }
    }
}

```

```

    }

    if (operacionValida) {
        System.out.println("Resultado: " + num1 + " " + operacion
+
                                " " + num2 + " = " + resultado);
    }

    scanner.close();
}
}

```

7.4 Lectura de Múltiples Valores

```

import java.util.Scanner;

public class LecturaMultiple {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Leer varios números en una línea
        System.out.println("Introduce 3 números separados por
espacios:");
        int n1 = scanner.nextInt();
        int n2 = scanner.nextInt();
        int n3 = scanner.nextInt();

        int suma = n1 + n2 + n3;
        double promedio = suma / 3.0;

        System.out.println("Suma: " + suma);
        System.out.println("Promedio: " + promedio);

        // Leer hasta encontrar una condición
        System.out.println("\nIntroduce números (0 para terminar:");
        int numero;
        int total = 0;
        int contador = 0;

        do {
            numero = scanner.nextInt();
            if (numero != 0) {
                total += numero;
                contador++;
            }
        } while (numero != 0);

        if (contador > 0) {
            System.out.println("Total: " + total);
            System.out.println("Números introducidos: " + contador);
            System.out.println("Media: " + (total / (double)
contador));
        }

        scanner.close();
    }
}

```

Ejercicios Propuestos

Ejercicio 1: Información Personal

Crea un programa que solicite al usuario su nombre, edad, altura y si trabaja. Muestra la información formateada y calcula cuántos años le faltan para jubilarse (considera 67 años como edad de jubilación).

Ejercicio 2: Conversor de Temperaturas

Desarrolla un programa que convierta temperaturas entre Celsius, Fahrenheit y Kelvin. Usa un enum para las unidades y permite al usuario elegir la conversión.

Ejercicio 3: Calculadora de Descuentos

Implementa una calculadora que solicite el precio de un producto y aplique diferentes descuentos según el día de la semana (usa enum). Por ejemplo: lunes 10%, miércoles 15%, viernes 20%.

Ejercicio 4: Validador de DNI

Crea un programa que valide si un DNI español es correcto. Debe tener 8 dígitos y una letra. Calcula la letra correspondiente y compárala con la introducida.

Ejercicio 5: Gestión de Notas

Desarrolla un sistema que permita introducir las notas de varios exámenes, calcule la media, determine si está aprobado (≥ 5), y muestre la calificación en texto (Suspenso, Aprobado, Notable, Sobresaliente).

Resumen de Buenas Prácticas

1. **Nomenclatura:**
 - Clases: PascalCase (MiClase)
 - Variables y métodos: camelCase (miVariable, miMetodo)
 - Constantes: MAYUSCULAS_CON_GUIONES (MI_CONSTANTE)
2. **Comentarios:** Documenta tu código de forma clara y concisa
3. **Inicialización:** Siempre inicializa las variables antes de usarlas
4. **Conversiones:** Ten cuidado con las pérdidas de precisión al hacer casting
5. **Scanner:** Recuerda cerrar el Scanner y limpiar el buffer cuando sea necesario
6. **Validación:** Siempre valida las entradas del usuario
7. **Enumerados:** Úsalos cuando tengas un conjunto fijo de valores posibles

Este tema sienta las bases fundamentales para programar en Java. Es esencial dominar estos conceptos antes de avanzar a estructuras de control, arrays y programación orientada a objetos.