

7 - Analiza el siguiente código y responde: ¿Cuáles son expresiones y cuáles son instrucciones? Explica la diferencia en un breve párrafo. `int x = 10; // Línea 1`
`x = x + 5; // Línea 2`
`System.out.println(x); // Línea 3`

Análisis

Línea 1: `int x = 10;` Instrucción, declaración e inicialización de variable

Línea 2: `x = x + 5;` Instrucción, asignación

- `x + 5` → Expresión, dentro de la instrucción

Línea 3: `System.out.println(x);` Instrucción, llamada a método

- `x` → Expresión, dentro de la instrucción

Diferencia entre expresiones e instrucciones:

Las expresiones son combinaciones de valores, variables y operadores que producen un resultado. Las instrucciones son unidades completas de ejecución que realizan una acción (declaraciones, asignaciones, llamadas a métodos). Mientras las expresiones se "evalúan", las instrucciones se "ejecutan". Toda expresión puede ser parte de una instrucción, pero no toda instrucción es una expresión.

8-

a. División con números enteros (int)

```
import java.util.Scanner;

public class DivisionEnteros {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingrese el primer número entero: ");

        int num1 = scanner.nextInt();

        System.out.print("Ingrese el segundo número entero: ");

        int num2 = scanner.nextInt();

        int resultado = num1 / num2;

        System.out.println("Resultado de la división (int): " + resultado);

        scanner.close();

    }

}
```

b. División con números decimales (double)

```
import java.util.Scanner;

public class DivisionDecimales {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingrese el primer número: ");
```

```

double num1 = scanner.nextDouble();

System.out.print("Ingrese el segundo número: ");

double num2 = scanner.nextDouble();

double resultado = num1 / num2;

System.out.println("Resultado de la división (double): " + resultado);

scanner.close();

}

}

```

c. Programa completo que compara ambos resultados

```

import java.util.Scanner;

public class ComparacionDivision {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingrese el primer número: ");

        int num1Int = scanner.nextInt();

        double num1Double = num1Int;

        System.out.print("Ingrese el segundo número: ");

        int num2Int = scanner.nextInt();

        double num2Double = num2Int;

        int resultadoInt = num1Int / num2Int;

        System.out.println("División con int: " + num1Int + " / " + num2Int + "
= " + resultadoInt);

```

```

    double resultadoDouble = num1Double / num2Double;

    System.out.println("División con double: " + num1Double + " / " +
num2Double + " = " + resultadoDouble);

    double resultadoCasting = (double) num1Int / num2Int;

    System.out.println("División con casting: (double)" + num1Int + " / " +
num2Int + " = " + resultadoCasting);

    scanner.close();

}

}

```

Ejemplo de ejecución: Ingrese el primer número: 5; Ingrese el segundo número: 2 ,División con int: $5 / 2 = 2$

División con double: $5.0 / 2.0 = 2.5$

División con casting: $(\text{double})5 / 2 = 2.5$

Diferencias: División con int: elimina decimales; División con double: Mantiene decimal.

Casting: Conversión explícita que permite operaciones con diferente precisión

Tipos de conversión:

Implícita: `double num = 5;` (Java convierte automáticamente)

Explícita (casting): `double resultado = (double) num1 / num2;`

9- Código corregido:

```
import java.util.Scanner;

public class ErrorEjemplo {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingresa tu nombre: ");

        String nombre = scanner.nextLine();

        System.out.println("Hola, " + nombre);

        scanner.close();

    }

}
```

Error y solución:

Error: String nombre = scanner.nextInt();

Problema:

- scanner.nextInt() está diseñado para leer números enteros, no texto
- Al intentar leer un nombre (texto) con un método para números, se produce un InputMismatchException
- El programa falla cuando el usuario ingresa texto porque nextInt() espera números

Solución:

Cambiar scanner.nextInt() por scanner.nextLine()

nextLine() está diseñado específicamente para leer líneas de texto completas

Versión con manejo de errores:

```
import java.util.Scanner;

public class ErrorEjemplo {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        try {

            System.out.print("Ingresa tu nombre: ");

            String nombre = scanner.nextLine();

            System.out.println("Hola, " + nombre);

        } catch (Exception e) {

            System.out.println("Error al leer la entrada: " + e.getMessage());

        } finally {

            scanner.close();

        }

    }

}
```

Ejemplo de ejecución:

Antes, con error:

Ingresa tu nombre: Juan

```
Exception in thread "main" java.util.InputMismatchException
```

Después, corregido:

Ingresa tu nombre: Juan

Hola, Juan

Métodos comunes de Scanner:

- `nextLine()` - para leer texto
- `nextInt()` - para leer enteros
- `nextDouble()` - para leer decimales
- `next()` - para leer una palabra (hasta espacio)

10-

Prueba de Escritorio

Línea	Variable	Valor	Explicación
1	-	-	Inicio del programa
2	<code>a</code>	<code>5</code>	Se declara e inicializa la variable <code>a</code> con valor 5
3	<code>b</code>	<code>2</code>	Se declara e inicializa la variable <code>b</code> con valor 2
4	<code>resultado</code>	<code>2</code>	Se calcula <code>a / b = 5 / 2 = 2</code> (división entera)

5

-

-

Se imprime: "Resultado: 2"

Valor final de resultado: 2

Explicación:

- Las variables a y b son de tipo int (números enteros)
- En Java, cuando divides dos números enteros, el resultado también es un número entero
- La división entera trunca (elimina) la parte decimal
- $5 / 2$ matemáticamente es 2.5, pero como es división entera, solo conserva la parte entera: 2
- No hay redondeo, simplemente se descartan los decimales

Demostración del código:

```
public class PruebaEscritorio {  
  
    public static void main(String[] args) {  
  
        int a = 5;  
  
        int b = 2;  
  
        int resultado = a / b; // 5 / 2 = 2 (no 2.5)  
  
        System.out.println("Resultado: " + resultado);  
  
    }  
  
}
```


Resultado decimal:

```
double resultado = (double) a / b;
```

```
double resultado = a / (double) b;
```

Output del programa original:

Resultado: 2

La división entera es un comportamiento fundamental en Java y muchos otros lenguajes de programación cuando se trabaja con tipos de datos enteros.