

Clase 3

PROGRAMACIÓN 1

Objetivos del tema

- Describir los operadores (aritméticos, incrementales, de relación, lógicos y de asignación) y los tipos de dato primitivos sobre los que actúan
- Evaluar expresiones que empleen datos primitivos, operadores y paréntesis
- Construir expresiones que empleen combinaciones de datos primitivos, operadores y paréntesis

Operador asignación

//El operando de la izquierda tiene que ser el
//identificador de una variable.

```
public class Programa {  
    public static void main(String[] args) {  
        int i,j;  
        String s;  
        i = 15;  
        j = i;           //j toma el valor que tiene i  
        s = "Hola";  
        System.out.println("i = " + i); //convierte y concatena  
        System.out.println("j = " + j);  
        System.out.println("s = " + s);  
    }  
}
```

Operadores aritméticos

- El resultado exacto depende de los tipos de operando involucrados.

Operador	Descripción	Ejemplo de expresión	Resultado
-	Cambio de signo (unario)	-4	-4
+	Suma	2,5 + 7,1	9,6
-	Resta	235,6 – 103,5	132,1
*	Producto	1,2 * 1,1	1,32
/	División (entera o real)	0,05 / 0,2 7 / 2	0,25 3
%	Resto	20 % 7	6

Ejemplo

```
public class Programa {  
    public static void main(String[] args) {  
        int i,j;  
        double a;  
        i = 7;  
        j = 3;  
        System.out.println(" Operador suma: i + j  
= " + (i+j));  
        System.out.println(" Operador resto: i % j  
= " + (i%j));  
    }  
}
```

Operadores aritméticos incrementales

Operador	Descripción	Ejemplo de expresión	Resultado del ejemplo
++	<ul style="list-style-type: none">• i++. Primero se usa el valor de la variable y luego se incrementa.• ++i. Primero se incrementa el valor y después se utiliza.	a++; ++a; a = 5;b = a++; a = 5;b = ++a;	5 a vale 6 b vale 5 a vale 6 b vale 6
--	Decremento. Funciona de manera análoga al incremento.	a--; --a; a = 5;b = a--; a = 5;b = --a;	a vale 4 b vale 5 a vale 4 b vale 4

```
public class Programa {  
    public static void main(String[] args) {  
        int i = 7;  
        int j = i++;  
        System.out.println(" Operador suma: i + j = " + (i+j));  
    }  
}
```

Operadores aritméticos combinados

Operador	Descripción	Ejemplo de expresión	Resultado
+=	Suma	a += b	a = a + b
-=	Resta	a -= b	a = a - b
*=	Multiplicación	a *= b	a = a * b
/=	División	a /= b	a = a / b
%=	Resto	a %= b	a = a % b

```
public class Programa {  
    public static void main(String[] args) {  
        int i;  
        i = 7;  
        i += 3;  
        System.out.println(" Suma combinada: i += 3 "  
        + " // i vale " + i);  
    }  
}
```

Operadores de relación

Operador	Descripción	Ejemplo de expresión	Resultado
==	Igual a	7 == 38	false
!=	Distinto de	'a' != 'k'	true
<	Menor que	'G' < 'B'	false
>	Mayor que	'b' > 'a'	true
<=	Menor o igual que	7,5 <= 7,38	false
>=	Mayor o igual que	38 >= 7	true

```
public class Programa {  
    public static void main(String[] args) {  
        int i,j;  
        i = 7;  
        j = 3;  
        System.out.println(" Operador igualdad: i == j es " +  
            (i==j)) ;  
    }  
}
```


Operadores lógicos

Operador	Descripción	Ejemplo de expresión	Resultado
!	Negación – Not	!false !(5 == 5)	true false
	Suma lógica – Or	true false (5 == 5) (5 < 4)	true true
&&	Multiplicación lógica - And	false && true (5 == 5) && (5 < 4)	false false

Concatenación y separadores

Operador	Descripción	Ejemplo	Resultado
+	Concatenación	"Hola" + "Juan"	"HolaJuan"

```
public class Programa {  
    public static void main(String[] args) {  
        System.out.println(" Concatenación: " + "cadena") ;  
    }  
}
```

Separador	Descripción
()	Permiten modificar la prioridad de operadores en una expresión.
{ }	Permiten definir bloques de código.
;	Permite separar sentencias.
,	Permite separar identificadores en una declaración de una sola línea.

Ejemplo

```
public class Programa {  
    public static void main(String[] args) {  
        int i,j,k;  
        i = 7;  
        j = 3;  
        k = 1;  
        System.out.println(" (i==j) || (i==k)  "+  
            ((i==j) || (i==k)));  
        System.out.println(" (i==j) || (i==k) && (i==2)  "+  
            ((i==j) || (i==k) && (i==2)));  
        System.out.println(" (i==j) || ((i==k) && (i==2))  "+  
            ((i==j) || ((i==k) && (i==2))));  
    }  
}
```

Diagram annotations:

- Inicio de primer bloque (points to the opening curly brace of the class)
- Inicio de segundo bloque (points to the opening curly brace of the main method)
- Fin de segundo bloque (points to the closing curly brace of the main method)
- Fin de primer bloque (points to the closing curly brace of the class)

Práctico

- Escribir un programa que dados tres números reales ingresados por el usuario divida el primero por el segundo y al resultado obtenido le reste el tercero.
- Escribir un programa que permita el ingreso de un número entero por teclado e imprima el cociente de la división de dicho número con 2, 3, y 4.
- Escribir un programa que permita el ingreso de dos números enteros por teclado e imprima:
 - _ si el primero es mayor al segundo.
 - _ si ambos son múltiplos de 2.
- Escribir un programa que ingrese un número entero por teclado e imprima el resultado de determinar:
 - _ si es múltiplo de 6 y de 7,
 - _ si es mayor a 30 y múltiplo de 2, o es menor igual a 30,
 - _ si el cociente de la división de dicho número con 5 es mayor que 10.