



Programación III

POO JAVA

- ❖ Operadores Lógicos
- ❖ Conversión entre tipos numéricos

MTI. Guadalupe Ledesma Ramos
guadalupe.ledesma@upslp.edu.mx

Operadores Lógicos

```
public class Operadores {
    public void calcula(int a){
        System.out.println(a +=3);
        a=10;
        System.out.println(a -=24);
        a=10;
        System.out.println(a *=3);
        a=10;
        System.out.println(a /=3);
        a=10;
        System.out.println(a %=3);

        System.out.println(a +=3);
        System.out.println(a);

        System.out.println(a -=24);
        System.out.println(a);

        a=10;
        System.out.println(a--); // Post-decremento, primero imprime luego decrementa
        System.out.println(a);
        a=10;
        System.out.println(a++); //Post-incremento, primero imprime y luego incrementa
        System.out.println(a);
        a=10;
        System.out.println(--a); //Pre-decremento, primero decrementa y luego imprime
        System.out.println(a);
        a=10;
        System.out.println(++a); //Pre-incremento, primero incrementa y luego imprime
        System.out.println(a);
    }
}
```

```
public class OperadoresTipos {
    public static void main(String[] args) {
        Operadores o1 = new Operadores();
        o1.calcula(10);
    }
}
```

```
run:
13
-14
30
3
1
3
3
-24
-24
10
9
10
11
9
9
11
11
BUILD SUCCESS
```

Operadores Lógicos

```
public class Operador {
    private int a;
    private int b;
    private int c;
    private int d;

    public Operador() {
        a=8;
        b=31;
        c=3;
        d=77;
    }
}
```

```
public void calcula(){
    d= a-- + --b / c * 2;
    d= b %c++;
    c+= b + 10 / 2 + ++a;

    if(b %--c == 10 || a-- + --b <= 19 || --d / c++ <=10 && b*c >=30){
        System.out.println("true");
    }else{ System.out.println("false"); }
    System.out.println("Los valores finales son: ");
    System.out.println(a);
    System.out.println(b);
    System.out.println(c);
    System.out.println(d);
}
```

			Inician	Paso 1	Paso 2	Paso 3	Paso 4	Paso 5	Paso 6	Paso 7	
obj1	Operador	#98	a =	8	7	7	8	8	7	7	
a	int	8	b =	31	30	30	30	30	29	29	
b	int	31	c =	3	3	4	47	46	47	47	
c	int	3	d =	77	28	0	0	0	-1	-1	
d	int	77									

```
Output - Operadores (run) x
run:
true
Los valores finales son:
7
29
47
-1
BUILD SUCCESSFUL (total ti
```

Paso 1

d = a-- + --b / c * 2
d = 8 + 30 / 3 * 2
d = 8 + 10 * 2
d = 8 + 20
d = 28

#98
7
30
3
28

Paso 2

d = b % c++
d = 30 % 3
d = 0

#98
7
30
4
0

Paso 3

c += b + 10 / 2 + ++a
c = 4 + (30 + 5 + 8)
c = 47

#98
8
30
47
0

Paso 4

b % --c == 10 ||
30 % 46 == 10
F

Paso 5

a-- + --b <= 19 ||
8 + 29 <= 19
F

Paso 6

--d / c++ <= 10 &&
-1 / 46 <= 10
V

Paso 7

b * c >= 30
29 * 47 >= 30
V

#98
7
29
47
-1

Conversión entre tipos numéricos

- ▶ Las normas de conversión entre tipos numéricos son las habituales en un lenguaje de programación: sí en una operación se involucran varios datos numéricos de distintos tipos todos ellos se convierten al tipo de dato que permite una mayor precisión y rango de representación numérica;

- ▶ La “jerarquía” en las conversiones de mayor a menor es:

double <- float <- long <- int <- short <- byte

- ▶ Por ejemplo:

- ▶ Si cualquier operando es double todos se convertirán en double.
- ▶ Si cualquier operando es float y no hay ningún double todos se convertirán a float.
- ▶ Si cualquier operando es long y no hay datos reales todos se convertirán en long.
- ▶ Si cualquier operando es int y no hay datos long se convertirán en int.
- ▶ En cualquier otro caso el resultado será también un int.

Conversión entre tipos numéricos

- ▶ Estas conversiones sólo nos preocuparán a la hora de mirar en que tipo de variable guardamos el resultado de la operación; esta ha de ser de una jerarquía mayor o igual a la jerarquía de la máxima variable involucrada en la operación. Si es de rango superior no habrá problemas.
- ▶ Es posible convertir un dato de jerarquía “superior” a uno con jerarquía “inferior”, arriesgándonos a perder información en el cambio. Este tipo de operación (almacenar el contenido de una variable de jerarquía superior en una de jerarquía inferior) se denomina cast.

```
int i = 9,k;  
float j = 47.9F;  
  
k = (int)j; //empleo de un cast  
j = k; //no necesita cast
```

Conversión entre tipos numéricos

- ▶ Byte

- ▶ Su valor mínimo es -128 y el máximo 127

- ▶ Short

- ▶ Su valor mínimo es -32,768 y el máximo 32,767

- ▶ Int

- ▶ Su valor mínimo es -2,147,483,648 y el máximo 2,147,483,647

- ▶ Long

- ▶ Su valor mínimo es -9,223,372,036,854,775,808 y el máximo 9,223,372,036,854,775,807

- ▶ Float

- ▶ Se recomienda usar un float (en lugar de un double) si necesita ahorrar memoria en grandes array de números en coma flotante.

- ▶ Double es la elección predeterminada para valores decimales.

Conversión entre tipos numéricos

```
1 package bienvenida;
2
3 public class UsoCast {
4     //double <- float <- long <- int <- short <- byte
5     private double d, sumaDouble;
6     private float j, sumaFloat, u;
7     private long p, sumaLong, l;
8     private int i, sumaInt, r;
9     private short w, sumaShort, s;
10    private byte b, sumaByte;
11
12    public UsoCast() {
13    }
14
15    public void inicializa(){
16        d= 123.45;
17        j=68.5F;
18        sumaDouble = d+j;
19        sumaFloat=(float) (d+j);
20        u= 47.9F;
21        p = 456;
22        sumaLong= (long) (u+p);
23        l = 2345;
24        i=9;
25        sumaInt = (int) (l+i);
26        r = 345;
27        w = 234;
28        sumaShort =(short) (r+w);
29        s = 12;
30        b=1;
31        sumaByte=(byte) (s+b);
32    }
33
34    public void imprime(){
35        System.out.println("Suma double: " + sumaDouble);
36        System.out.println("Suma float: " + sumaFloat);
37        System.out.println("Suma long: " + sumaLong);
38        System.out.println("Suma int: " + sumaInt);
39        System.out.println("Suma short: " + sumaShort);
40        System.out.println("Suma byte: " + sumaByte);
41    }
42 }
```

```
1 package bienvenida;
2
3 public class Bienvenida {
4     public static void main(String[] args) {
5         UsoCast cast1 = new UsoCast();
6         cast1.inicializa();
7         cast1.imprime();
8     }
9 }
```

Output - Bienvenida (run) X

```
run:
Suma double: 191.95
Suma float: 191.95
Suma long: 503
Suma int: 2354
Suma short: 579
Suma byte: 13
BUILD SUCCESSFUL (total time: 0 seconds)
```

Class Cast

```
1 package bienvenida;
2
3 public class ClassCast {
4     private Object objeto = new Object();
5
6     public ClassCast() {
7     }
```

```
8
9
10     public void cambios() {
11         objeto = 5.5;
12         System.out.println("El valor es: " + (Double) objeto);
13         objeto = 5.6F;
14         System.out.println("El valor es: " + (Float) objeto);
15         objeto = 5L;
16         System.out.println("El valor es: " + (Long) objeto);
17         objeto = 90;
18         System.out.println("El valor es: " + (Integer) objeto);
19         objeto = (short) 3;
20         System.out.println("El valor es: " + (Short) objeto);
21         objeto = (byte) 3;
22         System.out.println("El valor es: " + (Byte) objeto);
23         objeto = "Saludo";
24         System.out.println("El valor es: " + (String) objeto);
25         objeto = 'x';
26         System.out.println("El valor es: " + (Character) objeto);
27     }
```

```
27 public void cambios1() {
28     objeto = 5.5;
29     System.out.println("El valor es: " + (double) objeto);
30     objeto = 5.6F;
31     System.out.println("El valor es: " + (float) objeto);
32     objeto = 5L;
33     System.out.println("El valor es: " + (long) objeto);
34     objeto = 90;
35     System.out.println("El valor es: " + (int) objeto);
36     objeto = (short) 3;
37     System.out.println("El valor es: " + (short) objeto);
38     objeto = (byte) 3;
39     System.out.println("El valor es: " + (byte) objeto);
40     /*objeto = "Saludo";
41     System.out.println("El valor es: " + (String) objeto); */
42     objeto = 'x';
43     System.out.println("El valor es: " + (char) objeto);
44 }
45
```

MTI. Guadalupe Ledesma Ramos guadalupe.ledesma@upslp.edu.mx

Output - Bienvenida (run) X



run:

El valor es: 5.5

El valor es: 5.6

El valor es: 5

El valor es: 90

El valor es: 3

El valor es: 3

El valor es: Saludo

El valor es: x

El valor es: 5.5

El valor es: 5.6

El valor es: 5

El valor es: 90

El valor es: 3

El valor es: 3

El valor es: x

BUILD SUCCESSFUL (total time: 0s)

Boolean

```
//boolean bandera = true;  
Boolean bandera = true;  
String valor = Boolean.toString(bandera);  
//String cambio = Boolean.toString(true);  
System.out.println(valor);
```

- Bienvenida (run) ×

run:

true

BUILD SUCCESSFUL (total time: 0 seconds)