

# Operadores

# Aritméticos

➤ Se emplean con tipos numéricos primitivos para realizar operaciones aritméticas en un programa: +, -, \*, /, %, ++, --

```
int a=3;
```

```
a++; //equivale a =(int)(a+1);
```

```
int x=6, y=10, z;
```

```
z=y/x; //el resultado es 1, la división entre enteros es un entero
```

➤ Los operadores ++ y -- se aplican solo con tipos enteros y pueden ir delante o detrás de la variable:

```
int a=3, b;
```

```
b=++a; //b toma el valor 4
```

```
int a=3, b;
```

```
b=a++; //b toma el valor 3
```

Primero asigna y luego incrementa !

# Asignación

- El operador = Asigna el resultado de una expresión a una variable.
- Los operadores +=, -=, \*=, /=, %=, realizan la operación entre un dato y la variable y asignan el resultado a la variable

```
int a=3;  
a+=10; //equivale a a=a+10;  
byte b=10;  
b+=5; //ojo, equivale a b=(byte)(b+5);  
b=b+5;
```

Esto provoca **error de compilación**,  
pues una operación con int (los  
literales enteros son int) siempre  
da como resultado int

# Condicionales

➤ Evalúan dos operandos y dan como resultado un valor boolean:

➤ <, >, <=, >=, ==, !=

➤ Salvo ==, que puede utilizarse con objetos, los demás solo se pueden emplear con tipos primitivos, y entre tipos compatibles:

```
int a=3;  
double c=9.5;  
boolean x=false;  
if(a>c) //ok  
if(a<x) //error de compilación
```

# Lógicos

➤ Evalúan expresiones de tipo boolean:

➤ &&, || y !

```
int a=3;  
int c=9;  
int n=0;  
if(a>n && a<c) //verdadero  
if(!(n==0)) //falso
```

➤ Los operadores && y || funcionan en modo cortocircuito:

```
boolean b=(a<n && a<++c);  
//muestra 9, la segunda condición no llega a evaluarse  
System.out.println(c);
```

# Otros

## ➤ new. Creación de objetos a partir de la clase

```
ArrayList lista=new ArrayList();
```

## ➤ instanceof. Comprueba si un objeto es de un tipo dado

```
String s="hello";  
//muestra true  
System.out.println(s instanceof String);
```

# Precedencia operadores

- Por regla general , las operaciones se ejecutan de izquierda a derecha
- Sin embargo, hay operaciones que tienen preferencia sobre otras

<i>Operador</i>
() []
++ --
* / %
+ -
> >= < <=
== !=
&&
?:
= += -= *= /= %=