

# **GRUP D'EXPERTS**

**BLOC B**

# Bloc B Final

## 1.1. Variables. Tipus i utilitat.

Una variable és una dada emmagatzemada a la memòria que pot veure modificat el seu valor en qualsevol moment durant l'execució del programa.

Tota variable dins del codi font d'un programa ha d'haver estat declarada prèviament pel programador abans de poder fer-la servir.

- Variables locals
  - Una variable definida dins d'un bloc, mètode o constructor es diu variable local.
  - Aquestes variables es creen quan el bloc ingressat o mètode es diu i destrueix després de sortir del bloc o quan la crida torna del mètode.
  - L'abast d'aquestes variables només existeix dins del bloc en el qual es declara la variable, és a dir, podem accedir a aquestes variables només dins d'aquest bloc
- Variables d'instància
  - Les variables d'instància són variables no estàtiques i es declaren en una classe fora de qualsevol mètode, constructor o bloc.
  - Com les variables d'instància es declaren en una classe, aquestes variables es creen quan un objecte de la classe es crea i es destrueix quan es destrueix l'objecte.
  - A diferència de les variables locals, podem usar especificadors d'accés per a variables d'instància. Si no especifiquem cap especificador d'accés, s'utilitzarà l'especificador d'accés predeterminat

### EXEMPLE

- `int numero = 2;`
- `String cadena = "Hola";`
- `long decimal = 2.4;`
- `boolean flag = true;`

## 1.2. Conversions de tipus de dades

Quan assigna valor d'un tipus de dades a un altre, els dos tipus poden no ser compatibles entre si. Si els tipus de dades són compatibles, llavors Java realitzarà la conversió (automàticament) que es coneix com Conversió automàtica de tipus i, en cas negatiu, haurà de fer un **càsting** o convertir-lo explícitament. Per exemple, assignant un valor a una variable.

- **Conversión automática de tipos o ampliación**

La conversión de ampliación tiene lugar cuando dos tipos de datos se convierten automáticamente.

**byte -> short -> int -> long -> float -> double**

- **Conversión explícita de tipo o casting**

Si volem assignar un valor de tipus de dada més gran a una mena de dada més petita, realitzem un casteo/càsting o el que es coneix com a conversió de tipus explícit

**double -> float -> long -> int -> short -> byte**

**EXAMPLE CASTING**

```
byte a = 20;  
int x = (int) a;
```

## 1.3. Constants. Tipus i utilitat

Una constant és una variable de sistema que manté un valor immutable al llarg de tota la vida de el programa. Les constants en Java es defineixen mitjançant el modificador final.

- Una constant es declara igual que una variable, posant la paraula final abans del tipus (enlloc del tipus podem usar var). La declaració té lloc al principi del mètode main(), igual com fem en les variables.

- El nom de les constants s'escriu en majúscules. Si és paraula composta les separarem en '\_'.
- El valor de les constants no pot canviar

Hi han dos tipus de constants:

- Constants predefinides
- Constants literals

## EXEMPLE

```
static final nomConstant = valor;
```

D'aquesta manera si volem definir les constants DIES\_SETMANA o DIES\_LABORALS, que sabem que són variables que no canviaran el seu valor al llarg de el programa, generarem el següent codi:

```
static final int DIES_SETMANA = 7;
static final int DIES_LABORALS = 5;
```

## 1.4. Operadors del llenguatge de programació

Java proporciona molts tipus d'operadors que es poden usar segons la necessitat. Es classifiquen segons la funcionalitat que brinden. Alguns dels tipus són els següents:

- Operadors aritmètics
- Operadors unaris
- Operador d'assignació
- Operadors relacionals
- Operadors lògics

**Operador aritmètics:** Aquests s'utilitzen per a realitzar les operacions matemàtiques.

OPERADOR	SIGNIFICADO
+	Suma
-	Resta
*	Multiplicación
/	División
^	Potencia. Puede variar de un lenguaje a otro, incluso en algunos está en forma de función.
DIV	División entera. Suele variar entre los distintos lenguajes de programación.
MOD	Resto de la división entera. También este operador suele tener distinta simbología según el lenguaje.

**Operadors relacionals:** Les relacionals son aquelles que solament es poden cumplir dos coses, és a dir que es compleix o no es compleix, sol tenen dos valors possibles

OPERADOR	SIGNIFICADO
<	Menor que
<=	Menor o igual que
>	Mayor que
>=	Mayor o igual que
=	Igual. En algunos lenguajes, se utiliza como operador el == (para diferenciarlo del operador de asignación que verás más abajo)
<>	Distinto. Hay lenguajes que usan el símbolo !=

**Operadors lògics:** Les operacions lògiques son aquelles que permeten que amb una sola expressió, contingue varies expressions que contenen operadors relacionals, això vol dir que el resultat final serà verdader o fals.

OPERADOR	SIGNIFICADO
OR (O)	Suma lógica o disyunción. Hay lenguajes que usan el símbolo de la barra vertical, simple ( ) o doble (  )
AND (Y)	Producto lógico o conjunción. En algunos lenguajes se usa el ampersand & o el doble ampersand (&&)
NOT (NO)	Negación. También este operador puede variar dependiendo del lenguaje, pudiendo ser ! o incluso ~

**Operadors alfanumeriques de cadenes.** Aquests operadors son els que possibiliten realitzar operacions sobre els textos. Una de les operacions que me s'utilitza en aquest tipus, es la concatenació de dos textos.

**float:** Es un tipo dato para almacenar números en coma flotante con precisión simple de 32 bits.

**double:** Es un tipo de dato para almacenar números en coma flotante con doble precisión de 64 bits

**boolean:** Sirve para definir tipos de datos booleanos. Es decir, aquellos que tienen un valor de true o false. Ocupa 1 bit de información.

**char:** Es un tipo de datos que representa a un carácter Unicode sencillo de 16 bits.

**Operadors d'assignació:** Aquest operador es el que serveix per assignar valor a variables i constants.

Operador	Descripció	Ejemplo de expresi3n	Resultado del ejemplo
!	Negaci3n - NOT (unario)	!false !(5==5)	true false
	Suma l3gica – OR (binario)	true   false (5==5)   (5<4)	true true
^	Suma l3gica exclusiva – XOR (binario)	true ^ false (5==5)   (5<4)	true true
&	Producto l3gico – AND (binario)	true & false (5==5) & (5<4)	false false

**Operadors unaris:** només necessiten un operador, s'utilitzen per incrementar, disminuir o negar un valor, aquests operadors són:

- : Unari menys, per negar valors
- + : Unari m3s, per donar valors positius.
- ++ : Operador d'increment, s'utilitza per incrementar el valor a 1. Hi ha dos varietats d'operador d'increment.
  - Pre-increment: el valor s'incrementa primer i despr3s es calcula el resultat.
  - Post-increment: el valor s'utilitza per primer cop per calcular el resultat i despr3s disminueix.
- : Operador de decrement, s'utilitza per disminuir el valor a 1. Hi ha dos varietats d'operador de decrement.
  - Pre-decrement: el valor disminueix primer i despr3s es calcula el resultat.
  - Post-decrement: el valor s'utilitza per primer cop per calcular el resultat i despr3s disminueix.
- ! : Operador l3gic “no”, s'utilitza per invertir un valor boolean.

## 1.5. Tipus de dades simples

Els tipus de dades en java son un format d'enmagatzemament de dades que poden contenir un tipus espec3fic o rang de valors.

En java ens podem trobar en diferents tipus de dades: les de tipus primitiu inclouen bytes, short, int, long, char, double, float i booleà.

### **Simples:**

Totes les variables tindran un tipus de dada assignat. El llenguatge Java d3na de base una s3rie de tipus de dades primitives.

- **byte**: Representa un tipus de dada de 8 bits amb signe. De tal manera que pot emmagatzemar els valors numèrics de -128 a 127 (ambdós inclosos).
- **short**: Representa un tipus de dada de 16 bits amb signe. D'aquesta manera emmagatzema valors numèrics de -32.768-32.767.
- **int**: És un tipus de dada de 32 bits amb signe per emmagatzemar valors numèrics. Que el seu valor mínim és  $-2^{31}$  i el valor màxim de  $2^{31} - 1$ .
- **long**: És un tipus de dada de 64 bits amb signe que emmagatzema valors numèrics entre  $-2^{63}$  a  $2^{63} - 1$
- **float**: És un tipus dada per emmagatzemar nombres en coma flotant amb precisió simple de 32 bits.
- **double**: És un tipus de dada per emmagatzemar nombres en coma flotant amb doble precisió de 64 bits.
- **boolean**: Serveix per definir tipus de dades booleans. És a dir, aquells que tenen un valor de true o false. Ocupa 1 bit d'informació.
- **char**: És un tipus de dades que representa un caràcter Unicode senzill de 16 bits.

Tipo de variable	Bytes que ocupa	Rango de valores
boolean	2	true, false
byte	1	-128 a 127
short	2	-32.768 a 32.767
int	4	-2.147.483.648 a 2.147.483.649
long	8	$-9 \cdot 10^{18}$ a $9 \cdot 10^{18}$
double	8	$-1,79 \cdot 10^{308}$ a $1,79 \cdot 10^{308}$
float	4	$-3,4 \cdot 10^{38}$ a $3,4 \cdot 10^{38}$
char	2	Caracteres (en Unicode)