

0101010
0100101
1101010

UD2.- ELEMENTS D'UN PROGRAMA. EL LLENGUATGE JAVA

Programació - 1er DAW / DAM

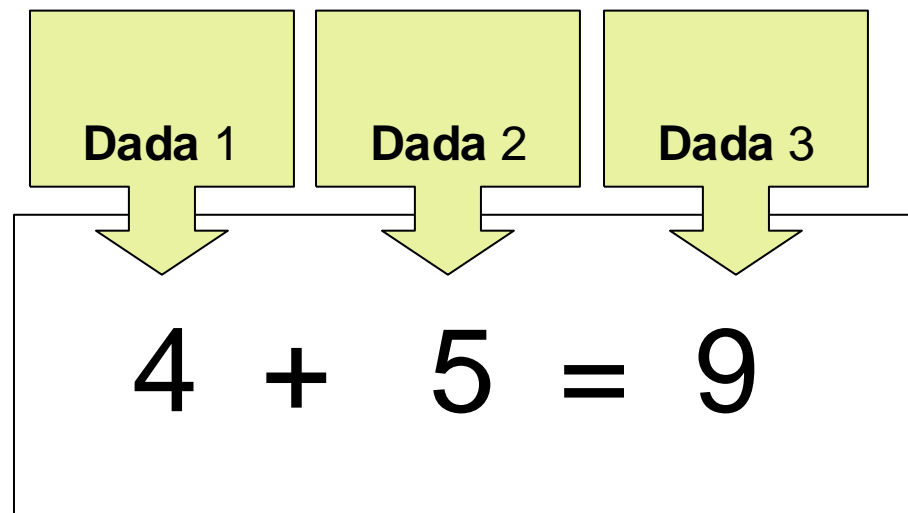
0. CONTINGUTS

- TIPUS DE DADES
- VARIABLES
- CONSTANTS
- OPERADORS

1. INTRODUCCIÓ

- Què és una **DADA**? Tota informació que l'ordinador utilitza per a dur a terme l'**execució d'un programa**
 - Cada dada és tractada **per separat**: (Ex: programa que suma dos nombres treballa amb tres dades)

- Operando1
- Operando2
- resultat



2. TIPUS DE DADES

- Conjunt de **valors vàlids** que pot **prendre una dada** i el **conjunt d'operacions** que podem portar a terme amb ell.
- **Exemples dades numèriques** (Vida quotidiana):
 - Una ciutat és a **8** km d'una altra.
 - Una persona té **30** anys.
 - Han passat **15** dies des d'un esdeveniment.
 - He pagat **5,50** € per a entrar al zoològic.

2. TIPUS DE DADES

- Exemple de **dades alfabètics** (A la vida real):
 - El meu gos es diu **Milo**
 - Estudie al **CIPFP Batoi**
 - Visc al carrer **Penàguila**

Els **valors** que **pot prendre** cada dada **són limitats** i vindran definits **pel seu tipus**.

2. TIPUS DE DADES

- El **tipus** de cada dada, **determinarà les operacions** que podem dur a terme:
 - **Tipus numèric:**
 - **Aritmètiques:** +, -, *, /, %, ...
 - **Comparació:** ==, <>, <,>, ...
 - **Tipus alfabètic:**
 - **Concatenació:** 'Hola '+' Món ' → 'Hola Món'
 - **Comparació:** ==, <>, <,>

2. TIPUS DE DADES

- **Tots els llenguatges de programació suporten els següents tipus:**
 - Enter
 - Reial
 - Caràcter
 - Booleà
 - Cadena

2.1 Tipus ENTER

- Representen un **valor numèric**; *positiu o negatiu*, **sense decimals**
 - Exemples de literals enters: **3, 0, -345,**
 - Exemples de dades sencers: **edat, dia del mes, any, nombre de fills, etc.**
 - Valor mínim i màxim: **-2147483647 - 2147483647**

2. TIPUS DE DADES

- **Cada dada d'un programa ha de pertànyer a algun tipus**
 - És tasca de programador decidir quin tipus s'acosta més a la informació a representar
 - **Tot llenguatge maneja tipus de dades bàsiques**
(O tipus primitius)
 - **A partir d'aquests tipus es poden definir altres més complexos**

2.1 Tipus ENTER

```
public class LiteralsEnter {  
  
    public static void main(String[] args) {  
  
        System.out.println(3);  
        System.out.println(0);  
        System.out.println(-345);  
        System.out.println(138764);  
        System.out.println(-345002);  
  
    }  
}
```

2.2 Tipus REAL

- Representa un **valor numèric**: *positiu o negatiu*, amb decimals.
 - **Exemples de literals**: **2.25, 4.0, .45, 3., 100.3,...**
 - **Exemples de dades**: *un preu en euros, el rècord mundial dels 100m llisos, la distància entre ciutats, etc ...*

2.2 Tipus REAL

```
public class LiteralsReal {  
    public static void main(String[] args) {  
        System.out.println(2.25);  
        System.out.println(4.0);  
        System.out.println(-9653.3333);  
        System.out.println(100.0003);  
    }  
}
```

2.3 Tipus CARÀCTER

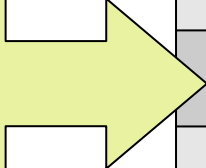
- Unitat **fonamental de text** usada en **qualsevol alfabet**, un **número** o un **signe de puntuació** o **exclamació**.
- **Exemples de literals:** **'a', 'A', '4', '>', '?', 'Γ'** (Lletra grega gamma majúscula), etc.
 - Es representen entre **cometes simples** (")
- **Exemples de dades:** cada un dels símbols individuals de l'alfabet, etc.

2.3. Tipus CARÀCTER

- Existeixen una serie de **caràcters especials** anomenats **de control** com són:

seqüència	significat
'\"'	Comillas simple
'\"'	Comillas doble
'\\'	contrabarra
'\b'	Backspace
'\n'	Salt de línia
'\r'	Retorn de carro
'\t'	Tab

Fixat que per a fer-los servir hem d'utilitzar una **seqüència d'escape**

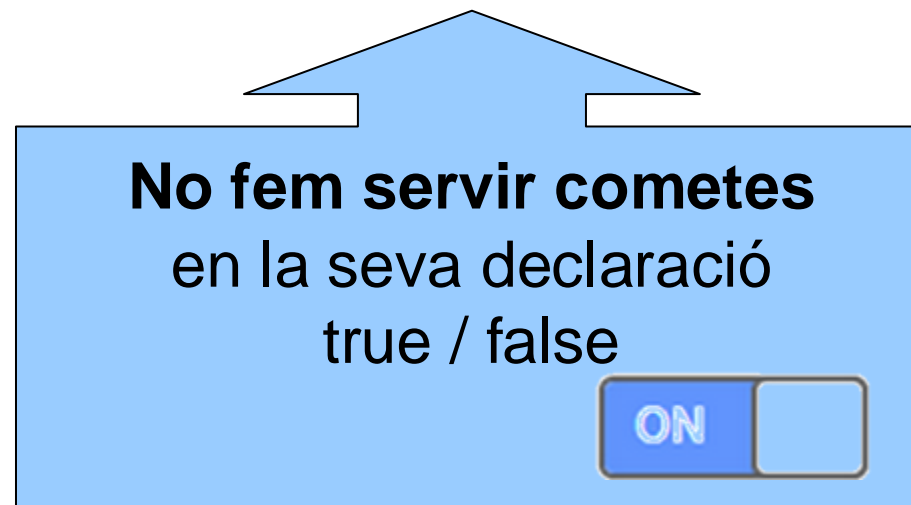


2.3. Tipus CARÀCTER

```
public class LiteralsCharacter {  
    public static void main(String[] args) {  
        System.out.println('a');  
        System.out.println('A');  
        System.out.println('4');  
        System.out.println('>');  
        System.out.println('?');  
        System.out.println('\\"');  
    }  
}
```

2.4. Tipus LÒGIC o BOOLEÀ

- Representa un **valor de tipus lògic** per establir la **veracitat o falsedat** d'un estat o afirmació.
 - **Valors possibles:** **true** i **false**
 - **Exemples de dades:** interruptor (on / off), casat (si / no), dret a vot, contrasenya és correcta, etc ...



2.4. Tipus LÒGIC o BOOLEÀ

- Literals lògic o boolean

```
public class LiteralsBoolea {  
  
    public static void main(String[ ] args) {  
        System.out.println (true);  
        System.out.println (false);  
    }  
  
}
```

3. Visualització d'informació per pantalla

- Mètodes ***print()*** i ***println()***
 - ***println()*** inclou un salt de línia a la fi de la sortida

```
public static void main(String[ ] args) {  
  
    System.out.print ("S'imprimeix sense el salt de línia");  
  
    System.out.println ("S'imprimeix amb un salt de línia");  
  
}
```

2.5 Tipus CADENA

- Representen una **cadena de caràcters** o **string**
 - **NO** formen part dels **tipus de dades primitius** de Java.
 - S'escriuen entre cometes dobles ("**cadena**")
 - Exemples: "**Hola Món**"

```
public static void main(String[ ] args) {  
  
    System.out.println("Primera línia \n Segona línia de l'string \n");  
    System.out.println("Ho\u00u0061");  
  
}
```

Activitat prèvia

- **Actividad 1.** Fes un programa que continga una expressió amb literals dels **quatre tipus de dades vistos** i ho mostre per pantalla. El programa ha de manejar almenys **10 literals**.

4. Variables

- Permeten emmagatzemar **dades, resultats i valors intermedis** d'un programa.
- Tenen associat un **tipus de dades**, que determina:
 - Conjunt de valors que la variable pot guardar.
 - Operacions que es poden realitzar amb ella.
- Declaració de **variables en Java**:

`int count;`

típo ← ← nombre

4.1 Tipus primitius

Tipus numèrics



TIPUS ENTERS

TIPUS DECIMALS

NOM	MIDA A BITS	VALOR MÀXIM
byte	8	127
short	16	32767
int	32	2147483647
long	64	9223372036854775807
float	32	3.4E + 38
double	64	1.7E + 308

4.1 Tipus primitius

Tipus caràcter

NOM	MIDA A BITS	REPRESENTA
char	16	Representa caràcters (<i>Lletres, nombres i símbols especials</i>) Java utilitza la codificació UNICODE de 2 bytes

Tipus booleà o lògic

NOM	MIDA A BITS	VALORS
boolean	8	Pot prendre a els valors true i false

4.1 Tipus primitius. Exemples

Declaració i Assignació

```
int x;
```

```
x = 234;
```

```
byte b = 89;
```

```
boolean esDivertit = true;
```

```
double d = 3456.98;
```

```
char c = 'f';
```

```
int z = x;
```

```
boolean esMusicaRock;
```

```
esMusicaRock = true;
```

```
boolean conectat;
```

```
conectat = true;
```

```
conectat = esDivertit;
```

```
long gran = 3456789L;
```

```
float f = 32.5f;
```

Posem la **lletra 'f'** per indicar que el literal **volem que siga declarat com un float** i no com un **double** (per defecte).

De la mateixa manera, la **lletra 'l'** indica que el literal **volem que siga declarat com un long** i no com un **int** (per defecte)

4.2 Identificadors de les variables

- Els **identificadors** són els **noms** que se'ls dona a les **variables** d'un programa:
 - han **començar** amb una **lletra**, **subratllat** (**_**) o el **caràcter \$**. **no** poden començar amb un **número**.
 - **Després del primer caràcter**, sí que es poden **utilitzar números**.
 - **No es poden utilitzar espais en blanc** ni els **símbols** dels operadors.
 - **El % no està permès**, **si** el **\$** i la **ç**. **Accents** i **ñ** estan **permesos**, però no són **recomanables**.
 - Recorda que Java és **case sensitive**.

4.3 Paraules reservades en Java

Els **identificadors** no poden coincidir amb cap **paraula reservada** ni els valors **true / false**.

abstract	continue	finally	int	public	throw
assert	default	float	interface	return	throws
boolean	do	for	long	short	transient
break	double	goto	native	static	true
byte	else	if	new	strictfp	try
case	enum	implements	null	super	void
catch	extends	import	package	switch	volatile
class	false	inner	private	synchronized	
const	final	instanceof	protected	this	while

Les paraules reservades tenen un **significat especial (semàntica)** dins el llenguatge

4.4 Convencions de noms

- Els noms de les **variables** han de començar per minúscula.
- Si el **identificador** està format **per diverses paraules**, La primera s'escriurà en minúscules i la resta de paraules comencen per majúscula. (**CamelCase**)
 - **anyoCreacion**
 - **resultadoOperacion**
 - **nombreApellidos**

Important utilitzar noms de variables que representen la seua finalitat.

Utilitzar noms de variables que reflectisquen la seua finalitat millora la llegibilitat del codi i facilita la seua comprensió.

4.4 Assignació de variables

- Assignar un **valor literal** després del símbol "="

```
x = 12;
```

```
isGood = true;
```

- Assignar el valor d'una variable a una altra

```
x = i;
```

- Utilitzant una combinació de les dues anteriors

```
x = i + 43;
```

- Assignant el valor en la mateixa sentència de declaració:

```
char car = 'A';
```

```
float preuPoma = 1.2f;
```

4.4. Assignació de variables. Exemples

```
int size = 32;
```

declara un entero llamado **size** asignándole el valor 32

```
char initial = 'j';
```

declara un carácter llamado **initial** asignándole el valor 'j'

```
double d = 456.709;
```

declara un doble llamado **d** asignándole el valor 456.709

```
boolean isCrazy;
```

declara un boolean llamado **isCrazy** sin asignarle valor

```
isCrazy = true;
```

asigna el valor true a la variable **isCrazy**, antes definida

```
int y = x + 456;
```

declara un entero llamado **y** y le asigna el valor de la suma del valor de x más el valor 456

4.5 Tipus Cadena

- En java, el **tipus cadena** no forma part dels **tipus primitius**.
- Les **cadena**s es defineixen fent ús de la classe **String**.

Declarem el tipus

Declaració: **String** alumne;

Inicialització: alumne = **"Pepe Batoi"**;

Per inicialitzar una cadena
sempre utilitzarem les
cometes dobles **"cadena"**

Declaración e inicialització: String alumne = **"Pepe Batoi"**;

Activitat Prèvia

Activitat 2. Donats els següents identificadors de variables. Quins són correctes i incorrectes?

- | | |
|---------------------------|--------------------------------|
| a) la meva carta | b) unacarta |
| c) min2escritos | d) 4cientos |
| e) es_un_mensaje | f) no_val res |
| g) ____ejemplo____ | h) mi-programa |
| i) Quants? | j) el% Descomptat |
| k) a150PORHORA | l) TengoMUCHO\$\$\$\$\$ |
| m) LOS400GOLPES | n) quieroUNAsolución |
| ñ) descarta2 | o) cuántosQuerrás |
| p) Carr3Mesas | q) çaVaBienAvec\$\$ |

Activitat Prèvia

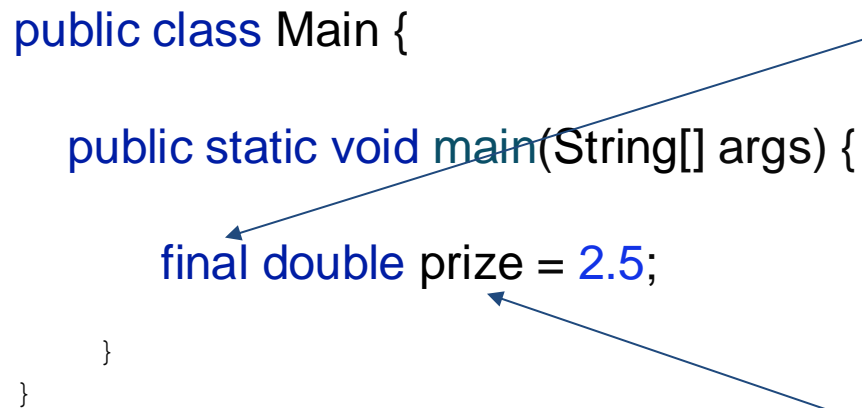
Activitat 3.- Crea un programa que declare **4 variables**, assigna'ls **el teu nom, cognoms, edat i si estàs casat o no**. Seguidament mostra el seu contingut per pantalla.

Utilitza el tipus de dades més d'acord en cada cas.

5. Constants

- Representen variables el **valor de les quals no pot canviar** al llarg de l'**execució del programa**.
- Per declarar una **constant** utilitzem el modificador ***final***

```
public class Main {  
    public static void main(String[] args) {  
        final double prize = 2.5;  
    }  
}
```



- Només podem donar-li un valor **quan la declarem**.

5. Constants

- Quan es declaren **fora del mètode main**, l'identificador d'aquestes van en **majúscules**. A més de `final`, afegirem la paraula reservada `static`

```
public class Main {
```

```
    final static double NUMERO_PI = 3.1415926536;
```

```
    public static void main(String[] args) {
```

```
        System.out.println(NUMERO_PI);
```

```
    }
```

```
}
```

Declare la variable

**Mostre el seu
valor**

6. Conversió de tipus de dades.

- Quan es realitza una assignació tant la variable com l'expressió han de ser del mateix tipus o de tipus compatibles

`int` nom = 25; ✓

`String` nom = 25; ✗

- Podem assignar un literal o el resultat d'una expressió a una variable sempre que tinga una grandària menor que la del tipus de la variable (**casting implícit**).

byte → *short* → *int* → *long* → *float* → *double*



int a = 2

long b = 2

float c = 12

double d = 25f

6.1 Què passa si el valor és més gran?

- No podem posar molta aigua en un pot petit.
 - En realitat sí podem, però se surt del pot.
- Exemple:

```
public static void main(String[] args) {  
    int x = 130;  
    byte b = x;  
}
```

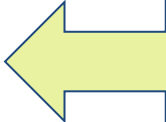
no **funcionarà**, si volem fer-ho
utilitzarem **càsting explícit** però
perdrem precisió



6.2 Conversió de tipus de dades (II)

- **Altres formes** de conversió de tipus **es poden realitzar** a través del ***casting explícit***.

```
int num = (int) 34.56;  
System.out.println(num);
```



Perdrem la
precisió
decimal

També podem utilitzar **funcions adequades d'alguns paquets de Java**. (Les estudiarem més endavant)

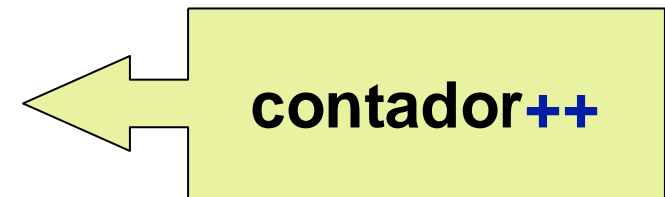
Activitat Prèvia

- **Activitat 4.-** Executa l'exemple de la diapositiva anterior. Què ocorre? Fes el mateix ara, però assignant a una **variable de tipus double** un **literal de tipus sencer**. Què ocorre?

7. Operadors

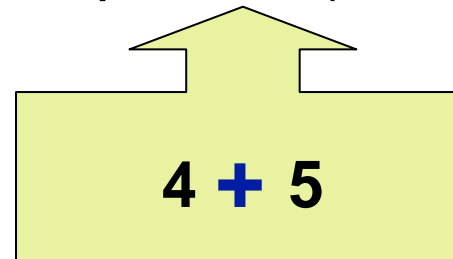
- Permeten **dur a terme transformacions** (Operacions) **sobre les dades** creant **nous valors**.
- Existeixen **2 tipus d'operadors**:

- **Unaris** (Actuen sobre un operant)



- **Binàries** (Actuen sobre dos operands)

- aritmètics
 - de comparació
 - lògics
 - assignació
 - concatenació



7.1 Operadors unaris (I)

Canvi de Signe

- Posar un **signe +** o **signe -** davant d'un literal.
- Exemples: **+45**, **-32**.

```
public static void main(String[ ] args) {  
    int i = -35;  
    int j = -45;  
}
```


7.1 Operadors unaris (II)

Increment (++) i decrement (--)

- Augmentar i disminuir a 1 el valor de la variable.
- Exemple:

```
public static void main(String[ ] args) {  
    int i = 5;  
    i++; // i val 6; és equivalent a i = i + 1  
    i--; // i val 5; és equivalent a i = i - 1  
}
```

7.1 Operadors unaris (II)

increment(++) i decrement (--)

- Poden anar **davant (pre)** o **darrere (post)** de la variable.
- la diferència **entre pre i post** apareix en una **instrucció complexa**:

```
public static void main(String[ ] args) {

    int valor;
    int i = 5;
    valor = i++; // valor val 5, i val 6
                // equival a  valor = i;
                i = i + 1;
    valor = ++i; // valor val 7, i val 7
                // equival a  i = i + 1;
                valor = i;

}
```

7.2 Operadors aritmètics

- Permeten realitzar **operacions aritmètiques** que impliquen al càlcul de **valors numèrics** representats per:

- **Literals**
- **Variables**
- **Altres expressions**



Aplicables sobre **tipus numèrics**

```
public static void main(String[] args)
{
    int x;
    int i = 4;
    x = 52 * 17;
    x = 120 / 4;
    x = 67 + 34;
    x = 32 - 12;
    x = 7 % 2;
    int resultat = x + 4
}
```

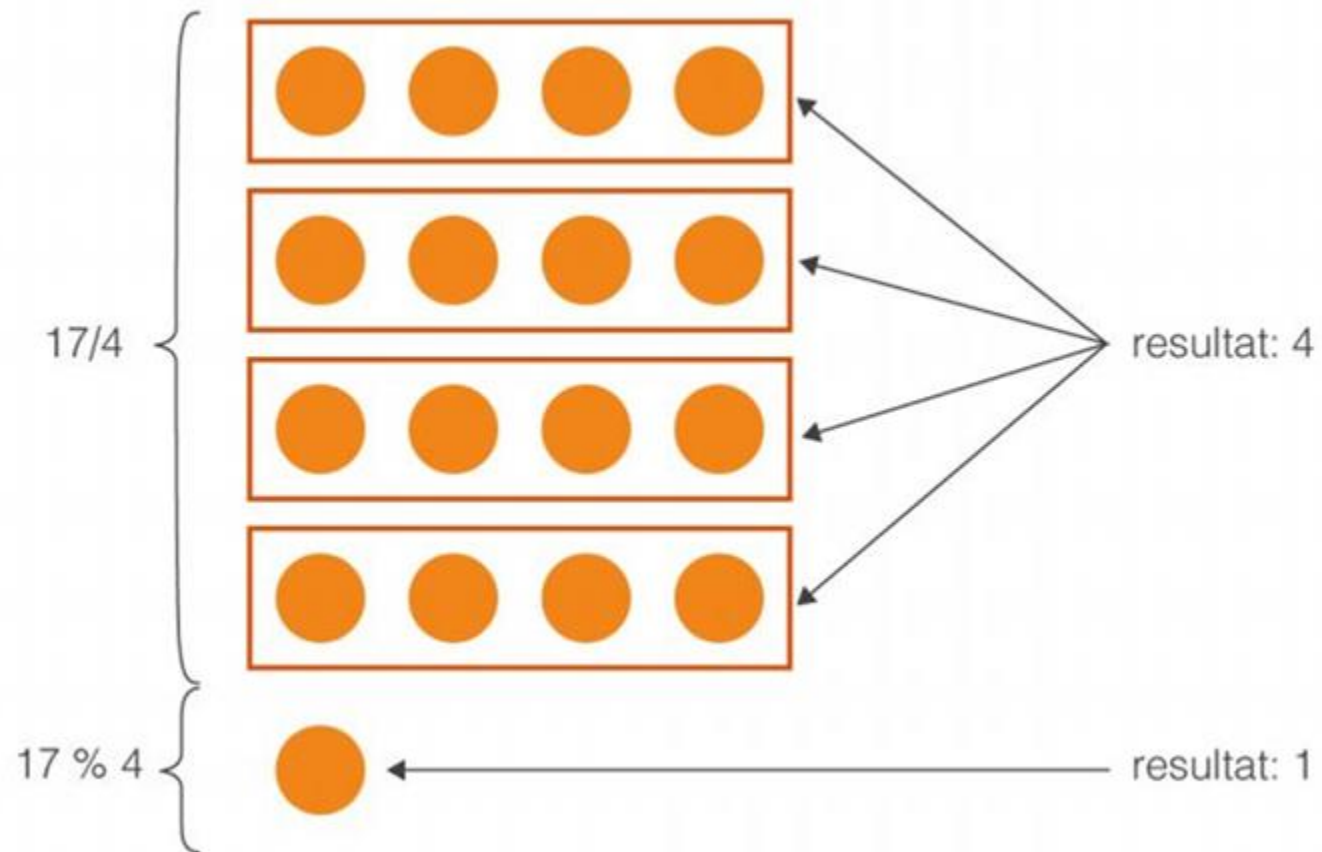
7.2 Operadors aritmètics

- + suma
- - resta
- * multiplicació
- / divisió
- % resta de la divisió entera

7.2 Operadors aritmètics

Operador de divisió i mòdul

● = 17



7.3 Operadors de concatenació

- Permet **generar una cadena de caràcters a partir d'altres dues cadenes.**

Exemple: "Hola" + "," + "bon dia" → "Hola,bon dia"

```
public static void main(String[ ] args) {  
    String cadena1 = "Hola";  
    String cadena2 = " Bon dia";  
    String cadena3 = cadena1 + cadena2;  
    System.out.println (cadena3); // Hola Bon dia  
}
```

Aplicables sobre **qualsevol tipus**,
sempre i quan **UN DELS OPERANDS**
siga una **CADENA**

7.4 Operadors de comparació

- Avaluen **expressions condicionals** i retornen un **booleà**.

Expressió operadorComparació **Expressió**

A	B	A == B	A > B	A < B	A >= B	A ≤ B
4	3	false	true	false	true	false
14	-2	false	true	false	true	false
-78	34	false	false	true	false	true
12	12	true	false	false	true	true

Aplicables sobre **tipus**
Enters i reals

7.4 Operadors de comparació

- **Exemple:**

```
public static void main(String[] args) {  
    int quantitat=100;  
    boolean pedidoGrande;  
    pedidoGrande = quantitat > 1000;  
  
    boolean testResult;  
    testResult = ( 45 < 35 );  
    testResult = ( 45 == 45 );  
    testResult = ( 4 != 3 );  
    testResult = ('A' > 'B');  
}
```

**Quin resultat valor s'assignarà a la variable
testResult en cada cas?**

7.5 Operadors lògics

- Realitzen una **avaluació lògica** d'expressions i retornen un valor **booleà**.
 - **Aplicable** sobre tipus Lògics o **booleans**

Expressió operadorComparació **Expressió**

Exemple:



7.5 Operadors lògics

Operador	funció
exp1 && exp2	Cada expressió ha de ser <i>true</i> perquè tota l'expressió sigui <i>true</i> .
exp1 exp2	Si una expressió és <i>true</i> , tota l'expressió és <i>true</i> .
!exp1	El negatiu lògic de l'entrada.

7.5 Operadors lògics

		conjunció	disjunció	
A	B	A && B	A B	! A
false	false	false	false	true
true	false	false	true	false
false	true	false	true	true
true	true	true	true	false

Aplicables sobre **tipus
booleans**

Activitat prèvia

- **Activitat 5.** Escriu un programa que mostre per pantalla la taula de veritat tabulada de la **expressió de conjunció** vista a la **diapositiva anterior**.

Per fer-ho, defineix 2 **variables A i B**. Seguidament assigna'ls els valors necessaris i fes les operacions adients per a que mostre per pantalla el següent resultat

A	B	A && B

false	false	false
true	false	false
false	true	false
true	true	true

7.6 Combinació operadors

- Podem combinar **operadors de comparació** i **operadors lògics** amb instruccions condicionals.

```
(edad >= 18) && (sexe == 'M')  
(edad >= 18) || (sexe == 'M')
```

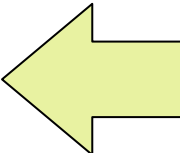
Aquesta expressió torna
un booleà

Aquesta expressió torna
un booleà

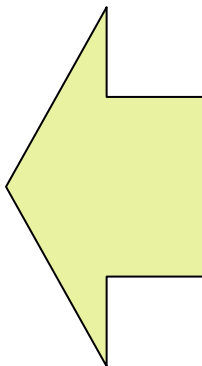
El resultat serà un **booleà**

7.7 Operadors d'assignació

Operador	exemple
=	edat = 34;
+=	edat + = 1; edat = edat + 1;
-=	edat -= 3; edat = edat - 3;
*=	edat *= 2; edat = edat * 2;
/=	edat /= 2; edat = edat / 2;
%=	edat% = 2; edat = edat%2;



Aplicable
sobre
**qualsevol
tipus**



Aplicables
sobre
**tipus
numèrics**

7.8 Prioritat d'operadors

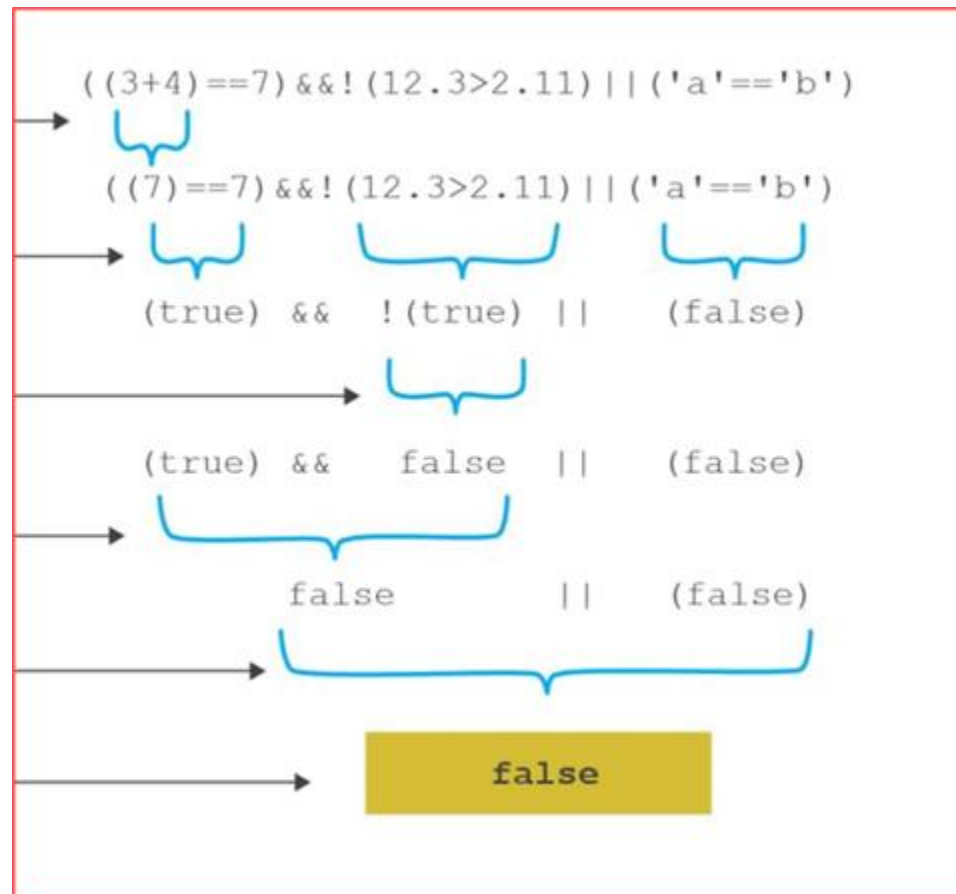
1. Parèntesi, de dins cap a fora.
2. Unaris (canvi de signe o increment/decrement)
3. Aritmètics
 - Multiplicatius: * / %
 - Sumatius: + -
4. De comparació
5. Lògics o booleans: NOT, AND, OR
6. D'assignació

Quan apareixen **operadors de la mateixa prioritat** junts, el compilador **avalua cada operació d'esquerra a dreta**.

7.9 Prioritat d'operadors

Exemple:

• **`((3 + 4) == 7) &&! (12.3 > 2.11) || (A == b)`**



Activitat Prèvia. Prioritat d'operadors

Activitat 6. Donades les següents expressions aritmètiques indica quin és el **resultat de la seva avaluació**:

a) $25 + 20 - 15$

b) $20 * 10 + 15 * 10$

c) $20 * 10 / 2 - 20 / 5 * 3$

d) $15 / 10 * 2 + 3 / 4 * 8$

e) $-a + 5\% b - a * a$

f) $5 + 3\% 7 * b * a - b\% a$

g) $(a + 1) * (b + 1) - b / a$

Per a les **3 últimes expressions** (e, f, g) suposa que les variables **a i b** són de tipus enter i que **a = 2 i b = 4**.

Activitat Prèvia. Prioritat d'operadors

- **Activitat 7.** A partir de les variables de tipus sencer amb valors **A = 5**, **B = 3** i **C = -12**, indica si l'avaluació de les següents expressions donaria com a resultat **true** o **false**:
 - **a)** $A > 3$
 - **b)** $A > C$
 - **c)** $A < C$
 - **d)** $B \neq C$
 - **e)** $A == 3$
 - **f)** $A * B == 15$
 - **g)** $A * B == -30$
 - **h)** $C / B < A$
 - **i)** $C / B == -10$
 - **j)** $C / B == -4$
 - **k)** $A + B + C == 5$
 - **l)** $(A + B == 8) \ \&\& \ (A * B == 2)$
 - **m)** $(A + B == 8) \ || \ (A * B == 6)$
 - **n)** $A > 3 \ \&\& \ B > 3 \ \&\& \ C < 3$

Activitat Prèvia

- **Activitat 8.** Escriu un programa en el qual es declaren les variables senceres **x** e **i**. assigna-li els **valors 150 i 2000** respectivament. A continuació, mostra per pantalla el valor de cada variable, *la suma, la resta, la divisió i la multiplicació*.
- **Activitat 9.-** Escriu un programa conversor de Mb a Kb. Els valor que ha de convertir estaran guardats en **2 variables** amb els valors 40.000 i 36.000 respectivament. Utilitza el **tipus més adient** i fés ús d'una **constant** per guardar el **ràtio de conversió**.

```
40000 Mb equivalen a 40960000 Kb  
36000 Mb equivalen a 36864000 Kb
```

8. Entrada de dades per teclat

- La **entrada de dades per teclat** en java la farem mitjançant la classe `Scanner` que pertany a la **biblioteca estándar** del llenguatge.

```
public class Exemple {
```

```
    public static void main(String[] args) {
```

```
        Scanner teclat = new Scanner(System.in);
```

```
        System.out.println("Escriu el teu nom");
```

```
        String nom = teclat.next();
```

```
        System.out.println("Escriu la teua edat");
```

```
        int edat = teclat.nextInt();
```

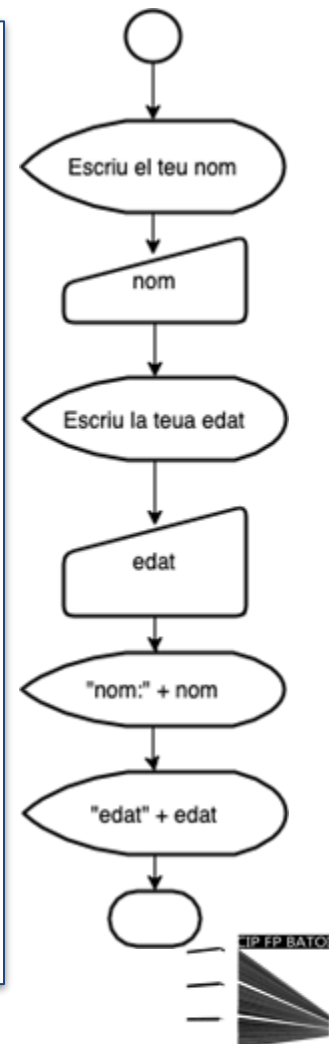
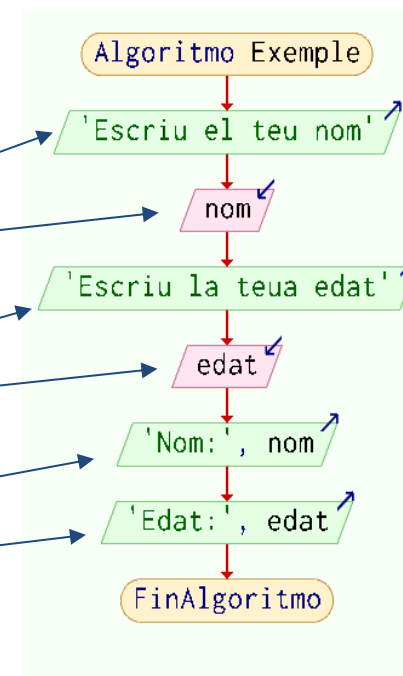
```
        System.out.println("Nom:" + nom);
```

```
        System.out.println("Edat:" + edat);
```

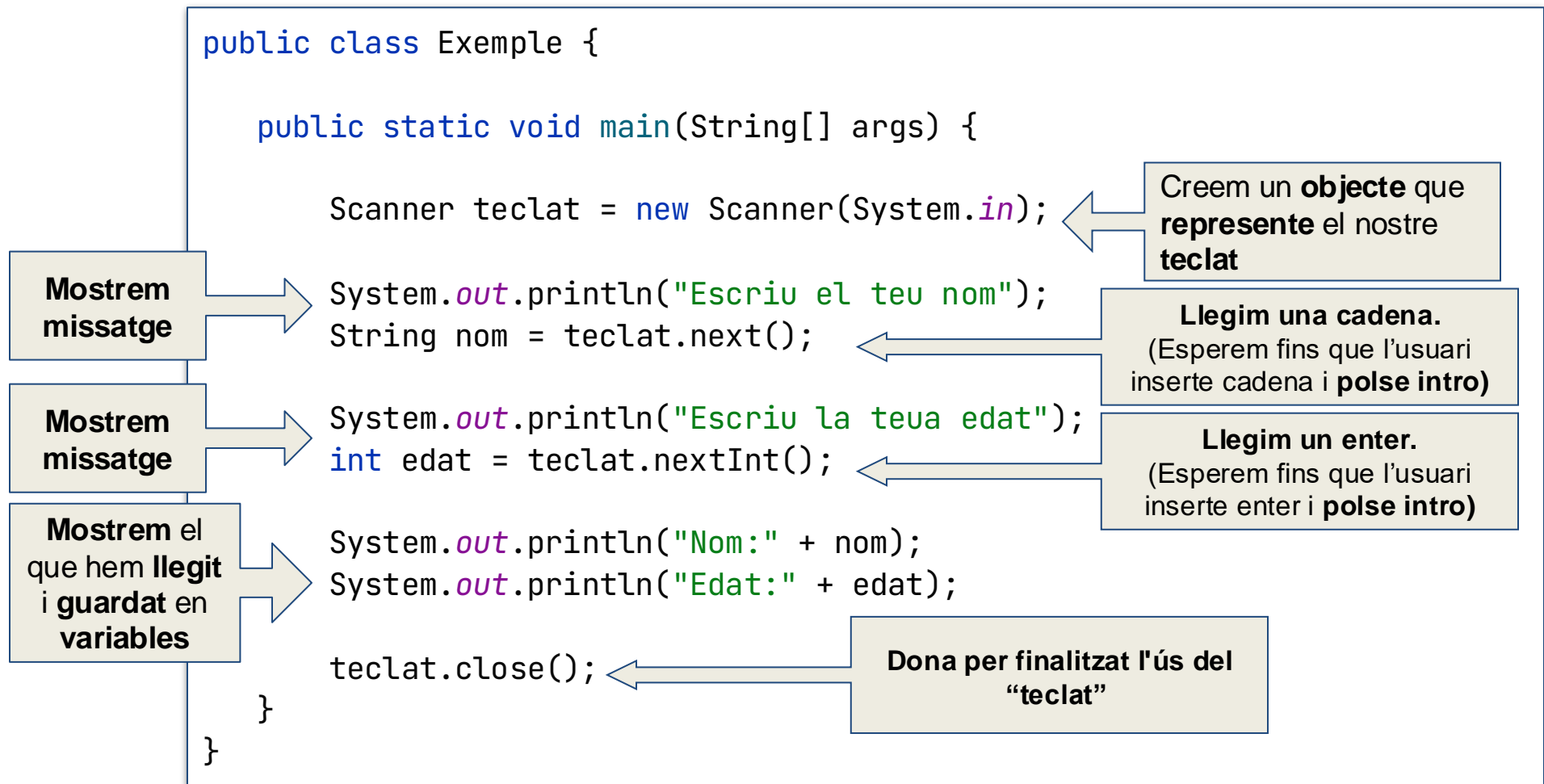
```
        teclat.close();
```

```
    }
```

```
}
```

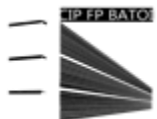


8.1 Analitzem el codi



8.2 Llegir altres Tipus de dades

mètode	Tipus de dades llegits
<code>teclat.nextByte()</code>	byte
<code>teclat.nextShort()</code>	short
<code>teclat.nextInt()</code>	int
<code>teclat.nextLong()</code>	long
<code>teclat.nextFloat()</code>	float
<code>teclat.nextDouble()</code>	double
<code>teclat.nextBoolean()</code>	boolean
<code>teclat.next()</code>	String
<code>teclat.nextLine()</code>	String



Activitat Prèvia

- **Activitat 10.** Implementa un programa que mostre per pantalla la **multiplicació de tres números reals** introduïts per teclat.
- **Activitat 11.** Implementa un programa que demane a l'usuari el número de monedes que té de: 2 cèntims, 5 cèntims, 10 cèntims, 20 cèntims, 50 cèntims, 1 euro y 2 euros. Al finalitzar ha de mostrar quants diners té en euros.

```
Quantes monedes tens de 2 cèntims? 5
Quantes monedes tens de 5 cèntims? 4
Quantes monedes tens de 10 cèntims? 2
Quantes monedes tens de 20 cèntims? 4
Quantes monedes tens de 50 cèntims? 2
Quantes monedes tens de 1€? 1
Quantes monedes tens de 2€? 2
En total tens 7 € y 30 cèntims
```

Això és tot ... de moment :-)