

U.D. 3 EL LENGUATGE JAVA

3.1 Variables i constants

3.1.1 Tipus de dades (literals)

3.1.2 Variables

3.1.3 Constants

3.1.4 Àmbit d'una variable

3.1.5 Els identificadors

3.2 Operadors i expressions

3.2.1 Operadors aritmètics

3.2.2 Operadors de bits

3.2.3 Operadors relacionals

3.2.4 Operadors lògics

3.2.5 Prioritat d'operadors

3.3 Conversió de tipus

3.4 Taules

3.4.1 Declaració i creació

3.4.2 Inicialització explícita

3.4.3 Accés al contingut

3.4.4 Taules de taules

3.4.5 Còpia de taules

3.5 Cadenes

3.5.1 Operadors de les cadenes de text

3.5.2 Caràcters de control

3.6 Sortida de dades per pantalla i entrada per teclat

3.6.1 Sortida de dades per pantalla

3.6.2 Entrada de dades per teclat

3.1 Variables i constants

3.1.1 Tipus de dades (literals)

Tipus de dades primitius:

- Booleà (boolean)
- Numèric
 - Enter (int)
 - Real (double)
- Alfanumèric
 - Caràcter (char)
 - Cadena de caràcters

3.1.1.1 Booleà

```

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

```

Il·lustració 3-1 Literals booleà

3.1.1.2 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);
    }
}

```

Il·lustració 3-2 Literals enter

3.1.1.3 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);
    }
}

```

Il·lustració 3-3 Literals real

3.1.1.4 Caràcter

```

public class LiteralsCaracter {
    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('¡');
    }
}

```

Il·lustració 3-4 Literals caràcter

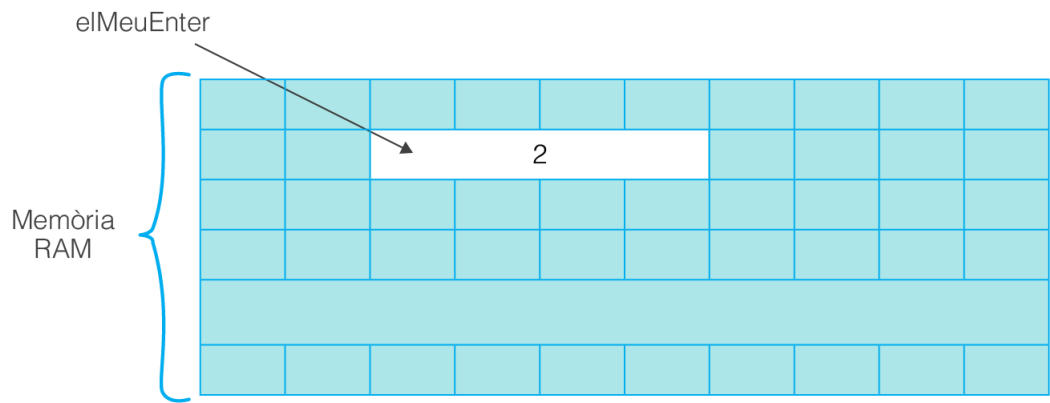
Tipus de dades numèriques

| Tipus | Paraula clau Java | Mida (bits) | Rang |
|-------------------------|-------------------|-------------|---|
| byte | byte | 8 | -128 a 127 |
| enter curt | short | 16 | -32768 a 32767 |
| enter simple | int | 32 | -2147483648 a 2147483648 |
| enter llarg | long | 64 | -9,223,372,036,854,775,808 a 9,223,372,036,854,775,808 |
| real de simple precisió | float | 32 | -3.40292347*10^38 a 3.40292347*10^38 |
| real de doble precisió | double | 64 | -4.94065645841246544*10^24 a 1.79769313486231570*10^308 |

3.1.2 Variables

```
//El programa que només declara una variable de tipus enter
public class DeclararEnter {
    public static void main(String[] args) {
        int elMeuEnter = 2;
        //Ara hi ha una variable en memòria que conté el valor enter 2.
    }
}
```

Il·lustració 3-5 Declaració d'una variable de tipus enter



Il·lustració 3-6 Una variable dins la memòria

3.1.3 Constants

```
private static final int NUMERO = 5;
```

3.1.4 Àmbit d'una variable

```
//Un programa que calcula una divisió i una suma, completat
public class DivideixISuma {
    public static void main(String[] args) {
        double dividend = 20.0;
        double divisor = 6.0;
        double sumarAlFinal = 3.0;
        //Les variables es poden usar com a literals dins una expressió.
        //Aquí és equivalent a fer (20.0/6.0) + 3.0
        System.out.println((dividend/divisor) + sumarAlFinal);
    }
}
```

Il·lustració 3-7 Programa divisió més suma

```
//Un programa que calcula una divisió i una suma, però a poc a poc
public class DivideixISumaDetallat {
    public static void main(String[] args) {
        double dividend = 20.0;
        double divisor = 6.0;
        double sumarAlFinal = 3.0;
        double resultat = 0.0;
        //El resultat serà una variable que s'anirà modificant.
        resultat = dividend/divisor;
        //Una expressió que usa la mateixa variable que es modifica!
        resultat = resultat + sumarAlFinal;
        System.out.println(resultat);
    }
}
```

Il·lustració 3-8 Divisió i suma detallat

3.1.5 Els identificadors

Un identificador:

- No pot contenir espais.
- No pot començar amb un nombre.
- Es desaconsella usar accents.
- No pot ser igual que alguna de les paraules clau del llenguatge.

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

Il·lustració 3-9 Paraules clau en Java

```

//Programa A. Un codi poc entenedor.
public class DivideixISuma {
    public static void main(String[] args) {
        double x = 20.0;
        double y = 6.0;
        double z = 3.0;
        //Ara vindria la resta del codi
        ...
    }
}

//Programa B. Un codi més entenedor.
public class DivideixISuma {
    public static void main(String[] args) {
        double dividend = 20.0;
        double divisor = 6.0;
        double sumarAlFinal = 3.0;
        //Ara vindria la resta del codi
        ...
    }
}

```

Il·lustració 3-10 La importància dels identificadors

3.2 Operadors i expressions

3.2.1 Operadors aritmètics

| Nom | Operador |
|--------------------------|----------|
| Canvi de signe (unari) | - |
| Suma | + |
| Resta aritmètica | - |
| Multiplicació | * |
| Divisió (resultat enter) | / |
| Mòdul | % |
| Increment | ++ |
| Decrement | -- |

| Abreviatures | | |
|----------------------------------|---------------------|------------------------|
| Suma i assignació | <code>a += b</code> | <code>a = a + b</code> |
| Resta i assignació | <code>a -= b</code> | <code>a = a - b</code> |
| Multipliació i assignació | <code>a *= b</code> | <code>a = a * b</code> |
| Divisió i assignació | <code>a /= b</code> | <code>a = a / b</code> |
| Mòdul i assignació | <code>a %= b</code> | <code>a = a % b</code> |

```
public class OperacionsEnter {
    public static void main (String[] args) {
        //Operacions aritmètiques: el resultat és un enter
        System.out.println(3 + 2);
        System.out.println(4 - 10);
        System.out.println(3 * 8);
        System.out.println(10 / 3);
        System.out.println(10 % 3);
    }
}
```

Il·lustració 3-11 Operadors aritmètics entre enters

```
public class OperacionsReal {
    public static void main (String[] args) {
        //Operacions aritmètiques: el resultat és un real
        System.out.println(8.5 + 3.2);
        System.out.println(5.66 - 3.1);
        System.out.println(3.1 * 8.4);
        System.out.println(17.0 / 4.0);
    }
}
```

Il·lustració 3-12 Operadors aritmètics entre reals

```
public class OpAritmetiquesEnterReal {
    public static void main (String[] args) {
        //Operacions relacionals: el resultat és un real
        System.out.println(3 + 2.5 );
        System.out.println(4.7 - 10);
        System.out.println(3 * 8.33);
        System.out.println(10.0 / 3);
    }
}
```

Il·lustració 3-13 Operadors aritmètics entre enters i reals

3.2.2 Operadors de bits

| Operador | Significat |
|----------|------------|
|----------|------------|

| | |
|-----|---|
| & | AND |
| | OR |
| ~ | NOT |
| ^ | XOR |
| >> | Desplaçament a la dreta |
| << | Desplaçament a l'esquerra |
| >>> | Desplaçament a la dreta omplint amb 0 |
| <<< | Desplaçament a l'esquerra omplint amb 0 |

3.2.3 Operadors relacionals

| Operands | | Resultat de les operacions | | | | |
|----------|----|----------------------------|-------|-------|--------|-------|
| 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 |

Exemple:

```
public class OperacionsEnter {
    public static void main (String[] args) {
        //Operacions relacionals: el resultat és un booleà
        System.out.println(4 == 4);
        System.out.println(5 > 6);
        System.out.println(7 < 10);
    }
}
```

II-lustració 3-14 Operadors relacionals entre enters

```
public class OperacionsReal {
    public static void main (String[] args) {
        //Operacions relacionals: el resultat és un booleà
        System.out.println(1.2 == 1.2 );
        System.out.println(-4.3 > -12.45);
        System.out.println(3.14 < 5.126);
    }
}
```

II-lustració 3-15 Operacions relacionals entre reals

```

public class OpRelacionalsEnterReal {
    public static void main (String[] args) {
        //Operacions relacionals: el resultat és un booleà
        System.out.println(2 > 1.2 );
        System.out.println(-4.3 >= 4);
        System.out.println(3.0 > 3);
    }
}

```

Il·lustració 3-16 Operacions relacionals entre enters i reals

3.2.4 Operadors lògics

| Operands | | Resultats de l'operació | | |
|----------|-------|-------------------------|--------|-------|
| 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 |

Il·lustració 3-17 Taula de veritat dels operadors booleans

Exemple d'operacions booleanes:

```

public class OperacionsBoolea {
    public static void main (String[] args) {
        //Operacions lògiques: el resultat és un booleà
        System.out.println(!true);
        System.out.println(true && true);
        System.out.println(true && false);
        System.out.println(true || false);
        System.out.println(false || false);
        //Operacions relacionals: el resultat també és un booleà
        System.out.println(false == false);
        System.out.println(false != true);
    }
}

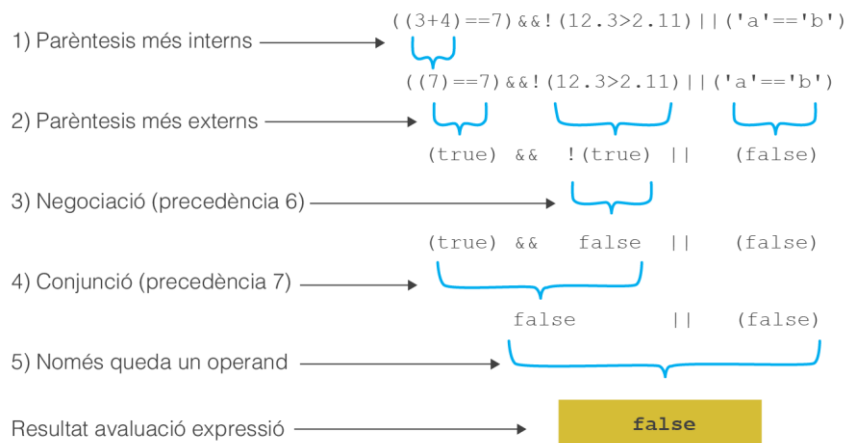
```

Il·lustració 3-18 Operacions booleanes

3.2.5 Prioritat d'operadors

| Ordre | Operació | Operador |
|-------|---------------------------|-----------|
| 1 | Canvi de signe | -(unari) |
| 2 | Producte, divisió i mòdul | * / % |
| 3 | Suma i resta | + - |
| 4 | Relacionals de comparació | > < <= >= |
| 5 | Relacionals d'igualtat | == != |
| 6 | Negació | ! (unari) |
| 7 | Conjunció | && |
| 8 | Disjunció | |

Il·lustració 3-19 Ordre de precedència dels operadors



Il·lustració 3-20 Exemple precedència

Exercici 3.1: Determineu el resultat de les expressions numèriques següents:

- 1) $30 \% 4 - 30 - 2 * 5$
- 2) $5 / 2 - 5.0 / 2$
- 3) $10 - 10.0 / 2 * 3 + 5$
- 4) $10 - 10.0 / (2 * 3) + 5$

Exercici 3.2: Calculeu el valor de les expressions següents:

$27 \% 4 + 15 / 4$

$9 * 2 / 3 * 25 * 3$

$25 \geq 7 \&\&! (7 \leq 2)$

$24 > 5 \&\& 10 \leq 10 || 10 == 5$

$(10 \leq 15 || 23 == 13) \&\&! (8 == 8)$

$!(6 / 3 > 3) || 7 > 7) \&\& (3 \leq 9 / 2 || 2 + 3 \leq 7 / 2)$

$'H' < 'J' || '9' == '7'$

Exercici 3.3: Diguen que fa aquest programa i què es mostrarà per pantalla

```
public class ProgramaMisterios {  
  
    public static void main (String[] args) {  
        int m = 23;  
        int n = 12;  
        int resultat = 0;  
        resultat = m + n;  
        System.out.println(resultat);  
    }  
}
```

Exercici 3.4: Modifica el codi del programa de l'exercici 3.3 perquè sobre la variable resultat es facin les transformacions següents, en aquest ordre:

1. Se li resti 5
2. Es multipliqui pel valor de la variable m
3. Es mostri el seu valor actual per pantalla
4. Es divideixi per n
5. Se li sumi 30
6. Es mostri el seu valor actual per pantalla

Exercici 3.5: Creeu un programa anomenat *CalculQuadrat* que, partint del valor d'un dels costats d'un quadrat, en calculi el perímetre i l'àrea.

Heu de definir el costat del quadrat amb la variable següent:

```
int costat;  
costat = 9;
```

Heu de tenir en compte que:

- El perímetre d'un quadrat és el seu costat multiplicat per quatre (costat per 4)
- L'àrea d'un quadrat es calcula multiplicant el valor del costat per ell mateix (costat per costat)

Exercici 3.6: Intercanvi de valors. L'objectiu d'aquesta activitat és fer partint de zero un programa en què es manipulen els valors emmagatzemats dins de variables i es reflexiona sobre quan cal usar-ne de noves.

Creeu un programa anomenat *IntercanviValors* que, partint de les variables següents mostri els seus valors, els intercanviï (és a dir, que numB contingui el valor de numA i numA contingui el valor de numB) i els torni a mostrar.

```
int numA = 58;  
int numB = 7;
```

Exercici 3.7: L'objectiu d'aquesta activitat és fer partint de zero un programa que fa un càlcul basat en expressions de diversos operands.

Creeu un programa anomenat *CalculRectangle* que, partint del valor dels dos costats d'un rectangle, en calculi el perímetre i l'àrea.

Heu de definir els costats del rectangle amb la variable següent:

```
int costatGran = 8;
int costatPetit = 3;
```

Heu de tenir en compte que:

- El perímetre d'un rectangle és el seu costat gran multiplicat per dos més el seu costat petit multiplicat per dos.
- L'àrea d'un rectangle es calcula multiplicant el valor del costat gran pel del costat petit.

Exercici 3.8: Avaluació d'expressions. L'objectiu d'aquesta activitat és fer partint de zero un programa que avalua expressions compostes per diversos operands, alguns d'aquests variables.

Creeu un programa anomenat *AvaluadorExpressions* que, partint de les variables següents,

```
int a=5;
int b=9;
int c=3;
int d=4;
```

executi i mostri el resultat d'avaluar les expressions següents:

```
2 - a * b + c
(2 - a) * b + c
a * b - c * a - d
a / 3 - b
a / (33 - b)
d * 23 - 1 + b
```

Exercici 3.9: Anàlisi d'operadors relacionals. L'objectiu d'aquesta activitat és entendre el funcionament d'una expressió relacional basada en variables.

Creeu un programa anomenat *VariableBooleana*. Cal que contingui les instruccions següents:

```
int a = 26;
int b = 13;
boolean resultat = a > b;
System.out.println(resultat);
```

Compileu-lo i executeu-lo. Quin valor mostra? Què indica aquest valor?

Tot seguit, modifiqueu-lo, canviant l'assignació a *resultat*:

```
resultat = a < b;
```

Compileu-lo i executeu-lo. Quin valor mostra ara? Què indica aquest valor?

Exercici 3.10: Completeu la taula omplint els espais en blanc segons l'evolució dels valors de les variables indicades:

| Expressió | Valor de a | Valor de b |
|--------------------------------|------------|------------|
| int a = 0 | | X |
| a = 1 | | X |
| a = a + 1 | | X |
| int b = 0 | | |
| a = 2 | | |
| a = a + a * (a + a * (a + a)) | | |
| a = 1 | | |
| a = a + a | | |
| b = 2 | | |
| b = a + 1 | | |
| b = a + b | | |
| b = a + b | | |
| a = 1 | | |
| b = 2 | | |
| b = b - 1 | | |
| b = b - a | | |
| a = 1 | | |
| a = a + 2 | | |
| a = 1 | | |
| a = a * 2 | | |
| a = 1 | | |
| b = 2 | | |
| c = 0 | | |
| d = 0 | | |
| c = a | | |
| d = b | | |
| a = d | | |
| b = c | | |

Exercici 3.11: Ús de constants. Fer dos programes, un que mostri per pantalla la taula de multiplicar del 3, i un altre, la del 5. Els dos programes han de ser exactament iguals, lletra per lletra, excepte en un únic literal dins de tot el codi.

3.3 Conversió de tipus

Hi ha càstings automàtics

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

En l'altre sentit és necessari fer un càsting explícit:

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

Exercici 3.12: Indicar si les següents conversions de tipus son necessàries, innecessàries o incompatibles.

```
int i = (int) 250;
```

```
int i = (int) 34F;
```

```
byte b = (byte) 30D;
```

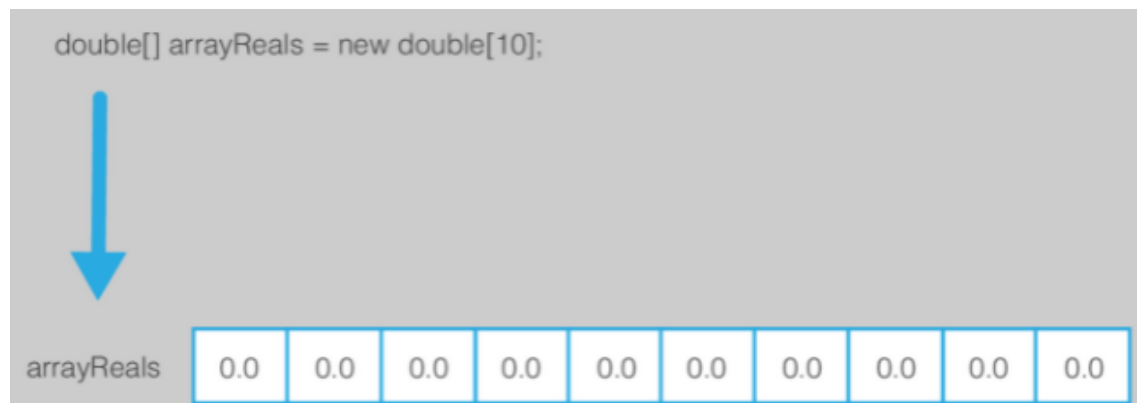
```
double d = (double) 47F;  
float f = (float) 350D;  
short s = (short) 250D;  
boolean b = (boolean) 1;  
double d = (double) 28;  
double d = (double) 390L;  
long l = (long)640F;
```

Exercici 3.13: Quin és el missatge d'error que us farà veure que és necessari un càsting explícit? Posau un exemple.

3.4 Taules (arrays)

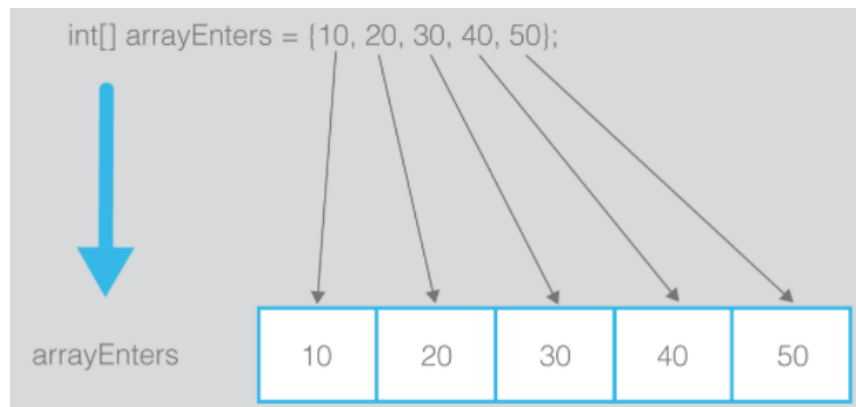
3.4.1 Declaració i creació

```
paraulaClauTipus[] identificadorVariable = new paraulaClauTipus[mida];
```



3.4.2 Inicialització explícita

```
paraulaClauTipus[] identificadorVariable = {valor1, valor2, ... , valorN};
```



3.4.3 Accés al contingut

```
identificadorArray[index]
```

L'índex va de 0 a n-1 on n és la mida de la taula.

Per saber la mida es pot utilitzar l'atribut **length**

```
identificadorArray.length
```

3.4.4 Taules de taules

```
paraulaClauTipus[] identificadorVariable = new paraulaClauTipus[nombreFiles][nombreColumnes];
```

```
paraulaClauTipus[][] identificadorVariable = {  
    {Fila0valor1, Fila0valor2, ... , Fila0valorN},  
    {Fila1valor1, Fila1valor2, ... , Fila1valorN},  
    ...,  
    {FilaNvalor1, FilaNvalor2, ... , FilaNvalorN}  
};
```

3.4.5 Còpia de taules

El tipus de dada array, al contrari que els tipus de dades primitius, no té cap operació disponible. Només es pot operar en les posicions individuals.

Una assignació `arrayA = arrayB` **no fa una còpia de la taula**, només fa que `arrayA` apunti a la mateixa posició de memòria que `arrayB`.

Exercici 3.14: Crear un array de 10 posicions d'enters i omplir-lo utilitzant inicialització explícita. Treure per pantalla el número ubicat en les posicions 1, 3 y 5.

Exemple del que ha de sortir per pantalla:

```
array[1] = 3
```

```
array[3] = 7
```

```
array[5] = 9
```

Exercici 3.15: Omplir una taula amb valors reals i treure per pantalla la suma de les posicions 0,2,4 i en la línia següent la suma de les posicions 1,3,5.

Exemple del que ha de sortir per pantalla:

```
Suma de les posicions 0,2,4 = 23.22
```

```
Suma de les posicions 1,3,5 = 48.68
```

Exercici 3.16: Crear una taula que emmagatzemi les notes obtingudes per alumne a cada trimestre i en calculi la nota final (mitjana de les 3 notes)

Exercici 3.17: Crea dues taules (`taulaA` i `taulaB`) amb 5 valors enters cadascuna. Després crea una altra taula (`taulaC`) que contengui la suma de cadascuna de les posicions de `A` i `B`. Es a dir, dins la posició 0 de la taula `C` hi ha d'haver la suma del contingut de la posició 0 de `A` més el contingut de la posició 0 de `B`. Mostra el contingut de la taula `C` per pantalla.

Exercici 3.18: Crear una taula de 5 valors reals i, utilitzant la propietat **length**, mostrar el contingut de la darrera posició. (No podeu emprar el número de darrer índex, heu d'emprar la propietat **length** per calcular la darrera posició).

3.5 Cadenes

Una cadena de text (String) és un tipus de dada que permet representar qualsevol seqüència de caràcters de longitud arbitrària.

3.5.1 Operadors de les cadenes de text

```
public class HolaMonConcatenat {
    public static void main(String[] args) {
        String hola = "Hola,";
        String mon = "món";
        String exclamacio = "!";
        //Mostra el text "Hola, món!" per pantalla
        System.out.println(hola + mon + exclamacio);
    }
}
```

3.5.2 Caràcters de control

| Seqüència d'escapament | Acció o símbol representat |
|------------------------|-------------------------------------|
| \t | Una tabulació. |
| \n | Un salt de línia i retorn de carro. |
| \' | El caràcter "cometa simple". |
| \" | El caràcter "cometes dobles" ("). |
| \\ | El caràcter "contrabarra" (\). |

3.5.3 Altres mètodes útils de String

El mètode **charAt()** permet seleccionar un caràcter individual dins d'una cadena (variable tipus String)

```
String cadena = "Hola Món";
char c;
c = cadena.charAt(0); //Selecció de la primera lletra 'H'
```

El mètode **toUpperCase()** passa una cadena a majúscules i el mètode **toLowerCase()** la passa tot a minúscules.

Exercici 3.19: Fer un programa que mostri en pantalla de manera tabulada la taula de veritat d'una expressió de disjunció entre dues variables booleanes.
Mostra:

| A | B | A B |
|-------|-------|--------|
| false | false | false |
| false | true | true |
| true | false | true |
| true | true | true |

3.6 Sortida de dades per pantalla i entrada per teclat

3.6.1 Sortida de dades per pantalla

Principalment, Java proporciona dues instruccions per mostrar dades a la pantalla:

```
System.out.print(x);
System.out.println(x);
```

3.6.2 Entrada de dades per teclat

```
Scanner lector = new Scanner (System.in);
int enter = lector.nextInt();
```

| Instrucció | Tipus de dada llegida |
|-----------------------------------|-----------------------|
| <code>lector.nextByte()</code> | byte |
| <code>lector.nextShort()</code> | short |
| <code>lector.nextInt()</code> | int |
| <code>lector.nextLong()</code> | long |
| <code>lector.nextFloat()</code> | float |
| <code>lector.nextDouble()</code> | double |
| <code>lector.nextBoolean()</code> | boolean |
| <code>lector.next()</code> | String |

`lector.next()` llegeix la pròxima paraula fins al primer espai. Si volem llegir una frase sencera s'ha d'emprar **`lector.nextLine()`**.

Exercici 3.20: Fer un programa que doni el resultat de multiplicar tres nombres reals entrats per teclat. Primer ha de demanar el primer nombre i s'ha d'emmagatzemar dins una variable de tipus real. Repetir fins a 3 vegades aquesta operació. Després calcular el producte dels valors de les variables i mostrar-lo per pantalla.

Exercici 3.21: Fer un programa que transformi graus Fahrenheit a graus Celsius. L'usuari ha d'entrar per teclat el valor en Fahrenheit i el programa li retornarà el valor en graus Celsius segons la següent fórmula:

$$C = F - 32 \cdot \left(\frac{5}{9}\right)$$

Exercici 3.22: Fer un programa que demani a l'usuari la introducció d'una frase per teclat i la mostri per pantalla tota en majúscules.

Exercici 3.23: Fer un programa que demani a l'usuari una cadena de caràcters per teclat i li retorni el darrer caràcter de la cadena. (S'han d'emprar els mètodes *charAt()* i *length()*)

Exercici 3.24: Fer un programa que demani un caràcter per pantalla i retorni el caràcter següent segons l'alfabet (Per exemple: si l'usuari entra una 'a' el programa retornarà una 'b'; si l'usuari entra una 'J' el programa retornarà una 'K') (Heu d'emprar el canvi de tipus de variable de *char* a *int* i al revés).

Exercicis complementaris (la seva entrega és voluntària):

Exercici 3C.1: Indicar quins d'aquests identificadors de variables són vàlids i quins no i el motiu:

- | | | | |
|----------------------|---------------------|--------------------|--------------------|
| 1) registro1 | 2) 1registro | 3) archivo_3 | 4) while |
| 5) \$impuesto | 6) año | 7) primer apellido | 8) primer_apellido |
| 9) primer-apellido | 10) primerApellido | 11) Tom's | 12) C3PO |
| 13) 123# | 14) PesoMáximo | 15) %descuento | 16) Weight |
| 17) \$\$precioMínimo | 18) _\$Único | 19) tamaño_máximo | 20) peso.maximo |
| 21) Precio____ | 22) matrícula? | 23) cuántoVale | 24) high |
| 25) java | 26) piragüista | 27) B_011 | 28) X012AB |
| 29) 70libro | 30) nombre&apellido | 31) 0X1A | 32) else |

Exercici 3C.2: Convertir en expressions les següents fórmules :

a) $\frac{3}{2} + \frac{4}{3}$

e) $\frac{a^2}{b-c} + \frac{d-e}{f-\frac{g*h}{j}}$

i) $\frac{m + \frac{n}{p}}{q - \frac{r}{s}}$

b) $\frac{1}{x-5} - \frac{3xy}{4}$

f) $\frac{m}{n} + p$

j) $\frac{3a + b}{c - \frac{d + 5e}{f + \frac{g}{2h}}}$

c) $\frac{1}{2} + 7$

g) $m + \frac{n}{p-q}$

d) $7 + \frac{1}{2}$

h) $\frac{a^2}{b^2} + \frac{c^2}{d^2}$

k) $\frac{a^2 + 2ab + b^2}{\frac{1}{x^2} + 2}$

Exercici 3C.3: A partir de la següent declaració de variables:

```
byte b;  
short s;  
long ln;  
int i, j;  
float f;  
double d;  
char c;
```

determinar el tipus de variable que s'obindrà com a resultat de les següents operacions:

- | | | |
|----------|----------------|----------------------|
| a) i + c | j) b + c | s) j - 4L |
| b) f - c | k) b / c + s | t) j - 4L * 2.5F |
| c) d + f | l) c + c | u) b + 2.5 * i + 35F |
| d) d + i | m) i + ln + d | v) 'a' + b |
| e) i / f | n) ln + c | w) 'a' + c |
| f) s + j | o) 5 / j | x) c + 2 |
| g) d + j | p) 5.2 / j | y) c - ln / 2 |
| h) s * c | q) i * f * 2.5 | z) 2 / i + 2.0 / j |

i) $d + c$ r) $\ln * f * 2.5F$

Exercici 3C.4: Si X, Y y Z son variables de tipus *boolean* con valores $X = true$, $Y = false$, $Z = true$, determinar el valor de les següents expressions lògiques:

- | | |
|--|---|
| a) $(X \&\& Y) \parallel (X \&\& Z)$ | b) $(X \parallel !Y) \&\& (!X \parallel Z)$ |
| c) $X \parallel Y \&\& Z$ | d) $!(X \parallel Y) \&\& Z$ |
| e) $X \parallel Y \parallel X \&\& !Z \&\& !Y$ | f) $!X \parallel !Y \parallel Z \&\& X \&\& !Y$ |

Exercici 3C.5: Si W, X, Y y Z son variables de tipus *boolean* amb valors $W = false$, $X = true$, $Y = true$, $Z = false$, determinar el valor de las següents expressions lògiques:

- | | |
|--|--|
| a) $W \parallel Y \&\& X \&\& W \parallel Z$ | b) $X \&\& !Y \&\& !X \parallel !W \&\& Y$ |
| c) $!(W \parallel !Y) \&\& X \parallel Z$ | d) $X \&\& Y \&\& W \parallel Z \parallel X$ |
| e) $Y \parallel !(Y \parallel Z \&\& W)$ | f) $!X \&\& Y \&\& (!Z \parallel !X)$ |

Exercici 3C.6: A partir de les següents declaracions de variables:

```
int i = 8, j = 5;
float x = 0.005F, y = -0.01F;
char c = 'c', d = 'd';
```

Determinar el valor de les següents expressions:

- | | |
|---|--|
| a) $i \leq j$ | b) $c > d$ |
| c) $x \geq 0$ | d) $x < y--$ |
| e) $j \neq 6$ | f) $c == 99$ |
| g) $!(i \leq j)$ | h) $!(c == 99)$ |
| i) $!(x > 0)$ | j) $-j == i - 13$ |
| k) $++x > 0$ | l) $y-- < 1$ |
| m) $c > d \parallel c > 0$ | n) $5 * (i + j) > 'c'$ |
| o) $2 * x + y == 0$ | p) $2 > x \&\& (y == 0)$ |
| q) $x + y \geq 0$ | r) $x < ++y$ |
| s) $-(i + j) \neq -i + j$ | t) $i \leq j \&\& i \geq c$ |
| u) $i > 0 \&\& j < 5$ | v) $i > 0 \parallel j < 5$ |
| w) $x > y \&\& i > 0 \parallel j < 5$ | x) $(3 * i - 2 * j) \% (2 * d - c) > 3 * d$ |
| y) $2 * ((i / 5) + (4 * (j - 3)) \% (i + j - 2)) \geq 10$ | z) $(i - 3 * j) \% (c + 2 * d) / (x - y) \geq 0$ |

Exercici 3C.7: Determinar el valor que sortirà per pantalla quan s'executi cada instrucció:

```
int x = 10;
System.out.println(x);
x++;
System.out.println(x);
System.out.println(++x);
System.out.println(x++);
System.out.println(x);
System.out.println(++x);
System.out.println(++x);
System.out.println(++x);
++x;
x++;
```

```
System.out.println(++x);
System.out.println(x++);
System.out.println(++x);
```

Exercici 3C.8: Si el valor inicial de les variables son $i = 1$, $j = 1$, $k = 1$, determinar el valor de cada variable després de cada operació. Cada operació és independent, el valor de totes les variables a l'inici de cada instrucció sempre és 1.

| | i=1 | j=1 | k=1 |
|---------------------------|-----|-----|-----|
| a) $i = ++j;$ | | | |
| b) $i = k++;$ | | | |
| c) $i = k + ++j;$ | | | |
| d) $i = i + j++;$ | | | |
| e) $i = j + ++k;$ | | | |
| f) $i = ++j + k++;$ | | | |
| g) $j = k-- + --i;$ | | | |
| h) $i = k + 1 + ++j;$ | | | |
| i) $i = ++i + --j + k--;$ | | | |
| j) $k = j-- + ++k;$ | | | |