

Programació

UT3.1 Tipus de dades compostos.
Introducció als arrays

Introducció

- Ja hem vist un tipus de dada compost, el tipus `String`.
- Recorda que un tipus de dada compost es crea a partir dels tipus de dades bàsics.
- Estos tipus de dades són capaços d'**emmagatzemar més d'un valor** dins d'una única variable. Per exemple, molts *char* junts formaven un *String*.

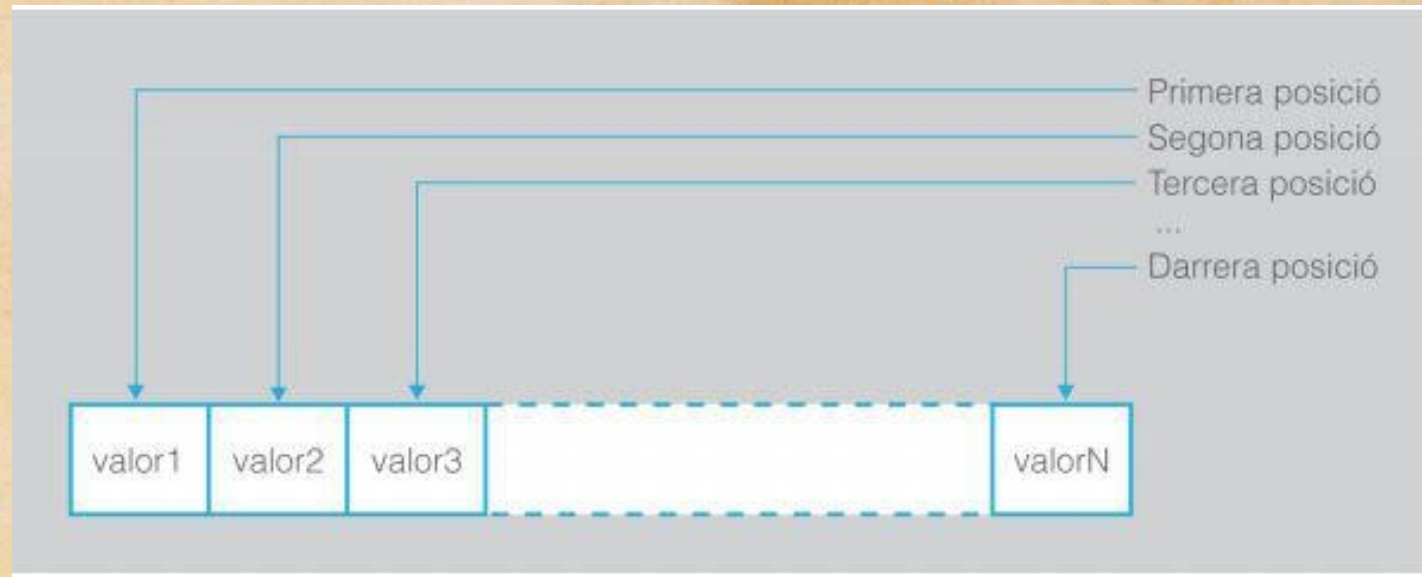
Què és un Array?

- És una estructura que permet emmagatzemar, en forma de seqüència, una quantitat determinada de valors pertanyents al **mateix tipus de dada**.



Arrays

- El seu contingut es gestiona d'acord amb les posicions seguint un ordre numèric: el valor emmagatzemat a la primera posició, a la segona, etc ..



Arrays

- La sintaxi per a utilitzar arrays pot ser diferent en altres llenguatges.
- El concepte d'accés a valors ordenats és aplicable a qualsevol llenguatge.
- Igual com en la resta de variables, un array necessita ser **declarat** per a poder ser utilitzat. La diferència és que **també necessita ser creat.**

Declaració

- La declaració és similar a la d'altres tipus, amb la particularitat de l'ús de els claudàtors ([])

- La sintaxi de declaració és la següent

tipusDeDada[] nomDelArray;

- Per exemple, per a declarar un array d'enters:

int[] enters;

- O també

int enters[];

Creació

- Cal, no només declarar l'array, sinó també crear-lo per a **determinar de quants elements disposarà**

- La sintaxi de creació d'un array és:

new tipusDeDada[nombreElements];

- Per exemple, per a crear l'array declarat a la diapositiva anterior, amb 5 elements de tipus enter:

enters = new int[5];

Declaració i creació

- Es pot (i es sol) declarar i crear l'array en la mateixa línia

```
int[] enters = new int[5];
```


Inicialització

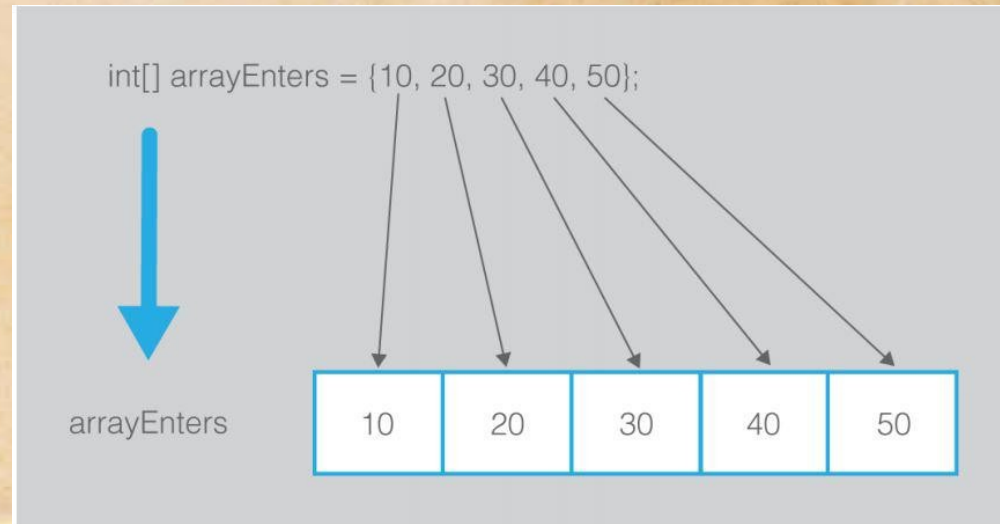
- La inicialització (total o parcial) dels elements d'un array té particularitats especials deguts a seua estructura.
- És una operació opcional, encara que **segons el tipus de dada de l'array, l'accés a un valor no inicialitzat provocarà errors.**

Declaració, creació i inicialització

- És possible realitzar les 3 operacions en una mateixa línia.

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

```
int[] arrayEnters = {10, 20, 30, 40, 50};
```



Manipulació de dades dins d'un array

- **No és possible utilitzar l'identificador d'un array directament per a invocar operacions i així manipular les dades contingudes**



```
...  
int[] a = {10, 20, 30, 40, 50};  
int[] b = {50, 60, 70, 80, 100};  
int[] c = a + b;  
...
```


Manipulació de dades dins d'un array

- Cada volta que es vulga fer alguna operació amb les dades emmagatzemats s'ha de fer amb **cadascuna de forma individual**.
- Cada valor de l' array té assignat un **índex: valor enter** que **indica l'ordre** fins el final de l'estructura. (Igual que passava amb els caràcters d'un String)
- La sintaxi d'accés a una dada de l' array és

`identificadorArray[índex]`

→ `enters[3]`

Manipulació de dades dins d'un array

- El rang d'índexs en Java van **des del 0**, per a la primera posició, fins el nombre de elements de l'array menys un, per a l'última.

```
int m = 5;  
int [] a = new int[5];
```

a[0]	a[1]	a[2]	a[3]	a[4]

```
a[1] = 2;
```

	2			
a[0]	a[1]	a[2]	a[3]	a[4]

```
a[2] = a[1];
```

	2	2		
a[0]	a[1]	a[2]	a[3]	a[4]

```
a[0] = a[1] + a[2] + 2;
```

6	2	2		
a[0]	a[1]	a[2]	a[3]	a[4]

```
a[0]++;
```

7	2	2		
a[0]	a[1]	a[2]	a[3]	a[4]

```
int m = 5;  
a[3] = m + 10;
```

7	2	2	15	
a[0]	a[1]	a[2]	a[3]	a[4]

Longitud de l' array

- Per a saber la longitud de l' array (la quantitat d'elements disponibles) es pot consultar amb la seua propietat `length`.

```
identificadorArray.length
```

- En l'exemple de la diapositiva anterior
`a.length` (és igual a 5)

Pregunta 1

- Quina és l'eixida del següent codi font?

```
int x = 30;  
int[] numbers = new int[x];  
x = 60;  
System.out.println("x is " + x);  
System.out.println("The size of numbers is " + numbers.length);
```

RECORDA: És important que proves el codi i anotes qualsevol dubte per a preguntar al professor.

Pregunta 2

- Indica si les següents afirmacions són certes o falses:
 - Tots els elements d'un array tenen el mateix tipus
 - La longitud de l' array es fixa quan es declara la referència de l'array
 - La longitud de l' array es fixa en la seua creació
 - Els elements d'un array han de ser de tipus de dada primitiu

Pregunta 3

- Quines sentències són vàlides?

```
int i = new int(30);  
double d[] = new double[30];  
char[] r = new char(1..30);  
int i[] = (3, 4, 3, 2);  
float f[] = {2.3, 4.5, 6.6};  
char[] c = new char();
```


Pregunta 4

- De quin tipus de dada és l'índex de referència dels elements d'un array?
- Quin és el mínim valor que pot tenir este índex?
- I el màxim?

Pregunta 5

- Escriu les sentències que fan el següent:
 - Crea un array que emmagatzeme deu valors de tipus double
 - Assigna el valor 5.5 a l'últim element de l'array
 - Mostra la suma dels primers 2 elements
 - Escriu un bucle que calcule la suma de tots els elements de l'array
 - Escriure un bucle que trobe el mínim element de l'array

Pregunta 6

- Quins errors de sintaxi hi ha al següent codi?

```
public class Test {  
    public static void main(String[] args) {  
        double[100] r;  
  
        for (int i = 0; i < r.length(); i++);  
            r(i) = Math.random * 100;  
        }  
    }
```


Pregunta 7

- Quina és l'eixida del següent codi?

```
public class Test {  
    public static void main(String[] args) {  
        int list[] = {1, 2, 3, 4, 5, 6};  
        for (int i = 1; i < list.length; i++)  
            list[i] = list[i - 1];  
  
        for (int i = 0; i < list.length; i++)  
            System.out.print(list[i] + " ");  
    }  
}
```


Exemple de lectura i emmagatzematge de deu nombres enters.

Declaración
y creación
del array

Almac.
de datos
en array

Recorrido del array

```
private static final int NUM_VALORES = 10;
/**
 * @param args the command line arguments
 */
public static void main(String[] args) {

    Scanner teclado = new Scanner(System.in);
    System.out.println("Escribe " + NUM_VALORES + " valores enteros:");

    //Creación del array "valores" con NUM_VALORES posiciones
    int[] valores = new int[NUM_VALORES];

    int numLeidos = 0;

    while (numLeidos < NUM_VALORES) {

        //Se lee un valor entero por teclado
        int valor = teclado.nextInt();

        // Se guarda en la posición correspondiente
        valores[numLeidos] = valor;

        // Se incrementa el número de valores leídos
        numLeidos++;

    }

    // Se recorre el array para mostrar cada valor almacenado
    for (int i = 0; i < valores.length; i++) {
        System.out.println("Valor " + i + ": " + valores[i]);
    }

    System.out.println("Fin");
}
```


Pràctica 3.1

Realitza un programa que **lliça un nombre determinat de notes de classe**. El funcionament serà el següent:

1. Es preguntarà **quantes notes es van a llegir** (esta dada serà introduïda per l'usuari, validada i "resol·licitada" en cas de que siga errònia)
2. Es demanarà a l'usuari que **introduïska una sèrie de valors** (ha de funcionar si les dades s'introdueixen en una línia separada per espais, o a diverses línies, o ambdues a la volta)
3. Cada valor **tindrà que ser validat com a nota vàlida** (un valor real entre 0 i 10)
4. Quan s'arribe al nombre de notes desitjades, el programa finalitzarà mostrant les notes introduïdes.

Pràctica 3.2

Realitza una **segona versió** del programa anterior. El programa funcionarà de la mateixa manera, però podrà parar la recollida de dades en cas de que l'usuari introduïska un **-1** per teclat, **i mostrarà al final només els elements introduïts.**

Pràctica 3.3

Realitza una **tercera versió** del programa anterior. Ara haurà de mostrar també la mitjana aritmètica de les notes i la nota màxima. Redueix el nombre de decimals mostrats a només 2.

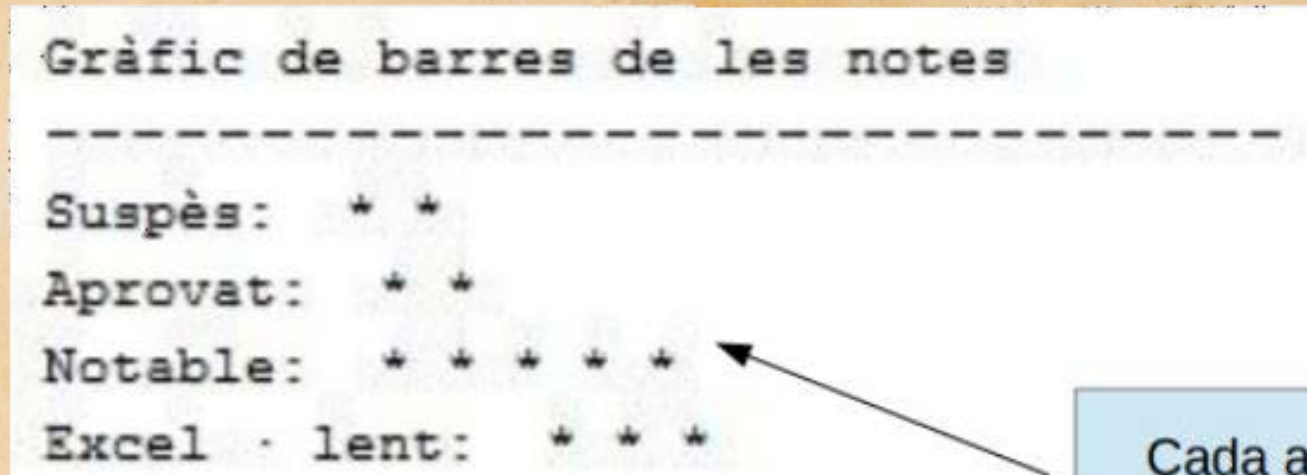
Pràctica 3.4

Amplia la funcionalitat del programa. En esta ocasió, **el programa no finalitzarà en mostrar la mitjana i la nota màxima**, sinó que després **el programa demanarà a l'usuari que introduïska una nota a buscar** (validant la dada introduïda).

Si la nota existeix, es mostrarà la posició (començant des del 1) de la nota trobada. **En cas contrari** es mostrarà a l'usuari "La nota X no existeix". (NOTA: pots usar la sentència break)

Pràctica 3.5

Elimina el que has realitzat a la pràctica 3.4 (opció de buscar) i ara mostra el següent esquema.



Cada asterisc és una nota trobada en eixa categoria.