DOC2.1: Programació estructurada avançada - Arrays.

1. Arrays

Un array és una estructura de dades que permet emmagatzemar múltiples valors del mateix tipus en una única variable. És útil quan necessitem treballar amb una col·lecció de dades, com llistes de números, noms, etc.

Imagina que tens un conjunt de dades, com ara les notes d'un grup d'estudiants o els números de telèfon d'un grup d'amics. Guardar cadascun d'aquests valors en variables individuals pot ser una tasca llarga i desordenada. Aquí és on els **arrays** es converteixen en una eina poderosa: permeten emmagatzemar diversos valors en una sola estructura organitzada.

Un **array** és com una taula amb una sèrie de caselles alineades. Cada casella té un índex, que comença sempre per **0**, i serveix per identificar quin valor hi ha guardat en aquella posició.

Els arrays són útils perquè:

- Ens permeten treballar amb un grup de dades relacionades de manera fàcil i ordenada.
- Podem accedir als valors ràpidament mitjançant el seu índex.
- Redueixen la complexitat del codi, ja que no cal crear una variable per a cada valor individual.

Per exemple, si necessites emmagatzemar les temperatures d'una setmana, en lloc de crear 7 variables diferents (`tempDilluns`, `tempDimarts`, etc.), pots utilitzar un array que guardi totes les temperatures de la setmana juntes.

Ex:

Cada celda es una temperatura de la setmana. Comença per la celda 0 i acaba a la 6. Per tant, tenim un array de llargada 7.

25	30	32	35	28	23	19

Formes per a declarar un array.

Les dues formes que tenim per a declarar un array son les seguents:

1. **Declarar i inicialitzar un array buit.** Aquesta forma crea un array amb una mida específica, però encara no té valors assignats:

   int[] temperatures = new int[7];

   En aquest cas, cada casella de l'array tindrà el valor per defecte del tipus de integer, en aquest cas 0.

2. **Declarar i inicialitzar un array amb valors.** Aquesta forma ens permet assignar directament els valors a l'array:

   int[] temperatures = {23, 25, 20, 22, 24, 21, 19};

   En aquest exemple, cada número representa la temperatura d'un dia de la setmana

Accedir als valors d'un array.

Per treballar amb un array, necessitem utilitzar el seu **índex** per accedir o modificar els valors que conté. La lógica es la mateixa que quan tenim un bucle for que tenim un index que ens permet recorrer tots els números fins arribar al final, doncs es el mateix. Cada número del index ens mostra el valor d'aquella posició de l'array.

int dilluns = temperatures[0];

La variable dillums tindrà el valor de la primera posició de l'array temperatures, en aquest cas, 23.

Els índexs comencen sempre a **0!!!!!!**, així que la primera posició és `0`, la segona és `1`, i així successivament.

Modificar o assignar un valor.

Indiquem la posició de l'array i el valor que volem guardar en aquella posició.

temperatures[0] = 26;

Recorrer un array.

Els arrays sovint es recorren per accedir a tots els seus elements. Això es fa habitualment amb un **bucle `for`**.

Per exemple, si vols imprimir totes les temperatures d'una setmana:

```
for (int i = 0; i < temperatures.length; i++) {
    System.out.println("Temperatura dia " + (i + 1) + ": " + temperatures[i]);
}
```

En aquest cas:

- `i` *és l'índex que utilitzem per accedir a cada posició de l'array.*
- `temperatures.length` *retorna la mida de l'array (el nombre de valors que conté).*

Exemples de Tipus d'Arrays en Java

Diversos exemples de com inicialitzar i utilitzar arrays de tipus ***String****,* ***char*** *i* ***boolean*** *en Java:*

1. Arrays de String

Els arrays de tipus String s'utilitzen per emmagatzemar cadenes de text.

Exemple 1.1: Declaració i inicialització buida

```
String[] noms = new String[3]; // Array amb 3 espais, inicialment buits
noms[0] = "Anna";
noms[1] = "Joan";
noms[2] = "Maria";
```

Exemple 1.2: Declaració i inicialització directa

```
String[] ciutats = {"Barcelona", "Girona", "Lleida", "Tarragona"};
```

Exemple 1.3: Accedir a un element:

```
System.out.println(noms[0]);
```

A la CMD mostra Mostra "Barcelona"

2. Arrays de char

Els arrays de tipus char s'utilitzen per emmagatzemar caràcters individuals.

Exemple 2.1: Declaració i inicialització buida

```
char[] vocals = new char[5]; // Array amb 5 espais, inicialment buits

vocals[0] = 'a';

vocals[1] = 'e';

vocals[2] = 'i';

vocals[3] = 'o';

vocals[4] = 'u';
```

Exemple 2.2: Declaració i inicialització directa

```
char[] lletres = {'A', 'B', 'C', 'D', 'E'};
```

3. Arrays de boolean

Els arrays de tipus boolean emmagatzemen valors true o false.

Exemple 3.1: Declaració i inicialització buida

```
boolean[] estats = new boolean[3]; // Inicialment tots els valors són `false`

estats[0] = true;

estats[1] = false;

estats[2] = true;
```

Exemple 3.2: Declaració i inicialització directa

```
boolean[] llums = {true, false, true, true};
```

IMPORTANT !!! Limitacions dels arrays

- **Mida fixa:** Quan creem un array, la seva mida no es pot canviar.
- **Tipus únic:** Tots els valors d'un array han de ser del mateix tipus.

Per resoldre algunes d'aquestes limitacions, podem utilitzar altres estructures de dades com ara **ArrayList** (que permet mides dinàmiques).