UD5.1- ESTRUCTURES DE DADES: ARRAYS

Programació – **1er DAW/DAM**

GENERALITAT VALENCIANA
Conselleria d'Educació, Cultura i Esport

CPFP BATOI

C/ Societat Unió Musical, 8 - 03802 Alcoi
Tel. 966 52 76 60 - Fax 966 52 76 61
03012165.secret@gva.es
www.cipfpbatoi.es

GOBIERNO DE ESPAÑA
MINISTERIO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL

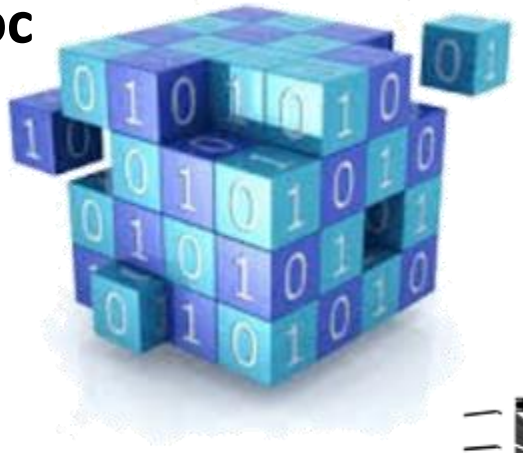
UNIÓ EUROPEA

CONTINGUTS

- INTRODUCCIÓ
- QUÈ ÉS UN ARRAY?
  - DECLARACIÓ CREACIÓ E INICIALITZACIÓ
  - ACCÉS ALS ELEMENTS
  - RECORREGUT
  - ESTRUCTURA FOR...EACH

1. INTRODUCCIÓ

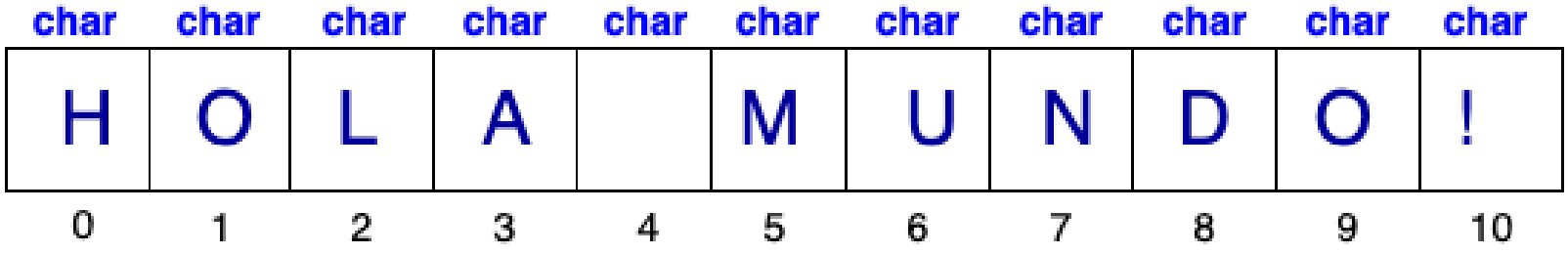
- Al món real **la informació útil** no **és tractada** de forma aïllada, sinó **d'una manera organitzada i estructurada** d'acord amb unes regles determinades.
  - Llistat d'**alumnes d'una classe.**
  - **Informació d'un alumne en concret**
  - **Possibles** opcions a **seleccionar** en un **formulari.**
  - Puntuacions **obtingudes** pels **usuaris en un joc**



1. INTRODUCCIÓ

- Els **tipus de dades compostos** o estructurats són capaços d'emmagatzemar **més d'un valor** sobre una única variable.
  - Aquests **es creen** a **partir** de **tipus bàsics**; `char`, `int`, `float`

- Ja **hem treballat** amb un **tipus estructurat:** La classe `String`

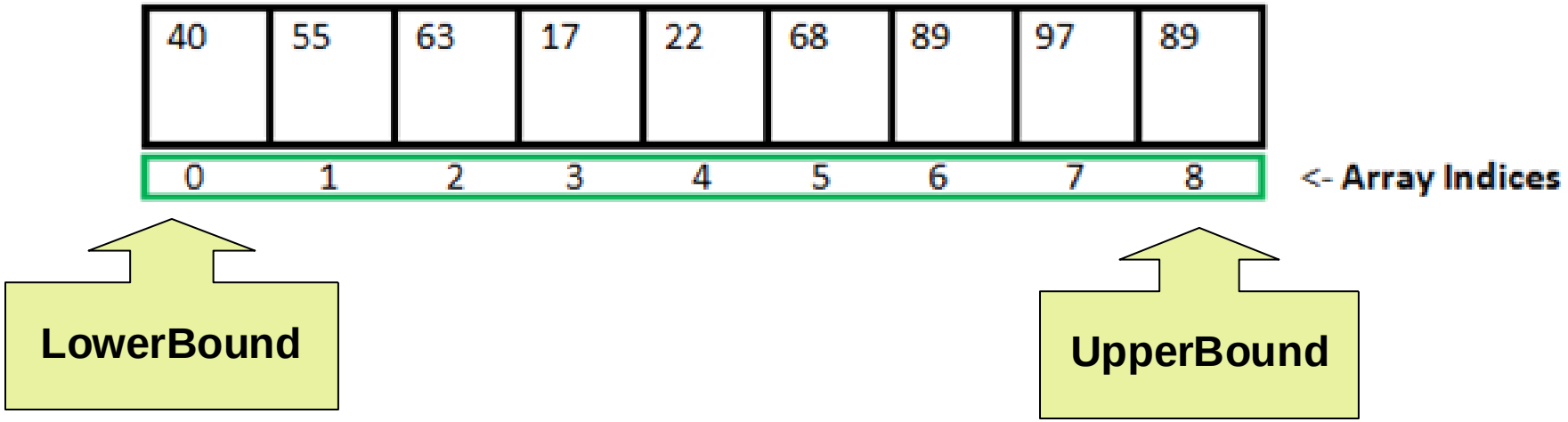


1. QUÈ ÉS UN ARRAY?

- Un ARRAY **permet** emmagatzemar i manipular **un conjunt d'elements** del **mateix tipus**.
  - Tots els elements **són emmagatzemats** en **posicions contigües de memòria.**
  - **S'accedeix** a cada element **mitjançant un índex** sencer.

1. QUÈ ÉS UN ARRAY?

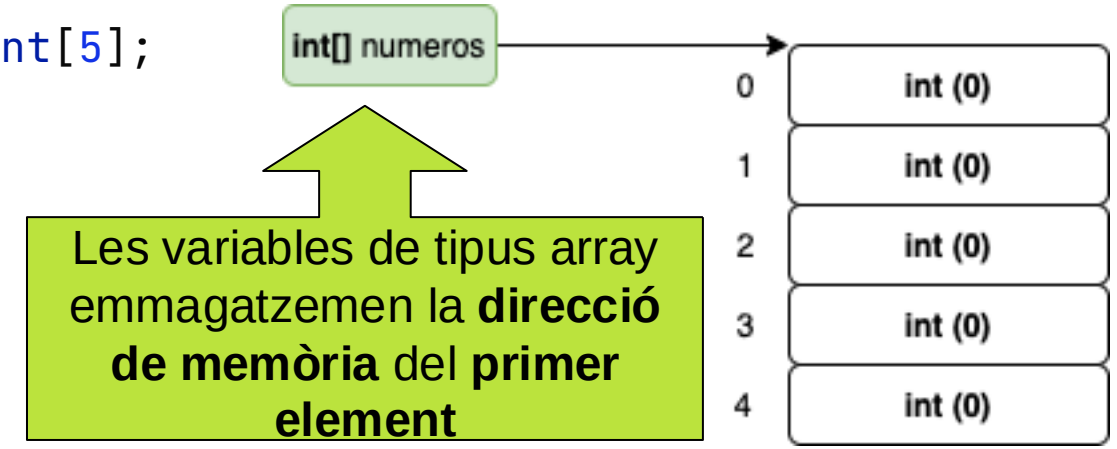
- Un **array** a Java és una **classe especial** (definida en **java.util.Arrays**).
  - Pot **contenir** tant tipus **primitius** com a tipus **referència**.
- Si intentem **accedir** a una **posició fora** de **l'array** es produirà un error i es llançarà una **excepció** (***java.lang.IndexOutOfBoundsException***)



1. QUÈ ÉS UN ARRAY?

- El nombre d'elements que pot albergar un array (**longitud** o **tamany)** es **determina** al **crear-lo**.
  - Aquesta longitud **no es pot modificar** en temps d'execució. **Els arrays són estructures estàtiques.**
    - No podrem **ni eliminar ni inserir** nous elements

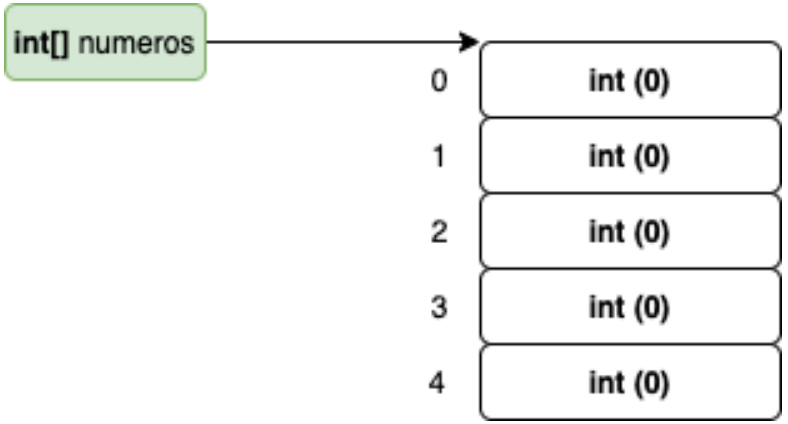
```
int[] numeros = new int[5];
```



1. QUÈ ÉS UN ARRAY?

- **Si al crear un array**, no s'especifiquen valors, **totes les posicions** són **inicialitzades** al **valor per defecte** del tipus de dades al que pertany.
  - **int** : 0
  - **float:** 0.0
  - **String**: null

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




```
float[] numeros = new float[5];
```

1.1 DECLARACIÓ, CREACIÓ I INICIALITZACIÓ

- **Declaració d'una variable array**
  - Es crea una **variable** capaç d'apuntar (**contenir l'adreça** de **memòria**) a un objecte de tipus **array**.
  - L'objecte encara no està creat i la variable apunta a **null** (conté valor null)

```
int[] myArray; // Declarem una variable de tipus array
```

1.1 DECLARACIÓN, CREACIÓN E INICIALIZACIÓN

- **Crear o instanciar un array**

  - **Creem** físicament **l'objecte i se** li **assignen les posicions** de **memòria** contigües.

  - S'utilitza la paraula reservada **new** i s'invoca al **constructor**.

```
int[] myArray ; // Declaramos una variable de tipo array
myArray = new int[10]; // instanciar un array de 10 elements
```

  - També podem **declarar** e **instanciar** l'array en la **mateixa instrucció**.

```
int[] myArray = new int[10];
```

1.1 DECLARACIÓN, CREACIÓN E INICIALIZACIÓN

declarar	tipus[] nomArray;	`int[] numeros;`
crear	nomArray = new tipus[numElements]	`numeros = new int[10];`
declarar + crear	tipus[] nomArray = new tipus[numElements]	`int[] numeros = new int[10];`
declarar + crear + inicialitzar	tipus[] nomArray = {v1, v2, v3, ...};	`int[] numeros = { 53, 15, 22, 60, 6, 8, 14, -75, 12, 64};`

1.2 ACCÉS ALS ELEMENTS

- Si un array té **n elements:**
  - **numeros[0]:** accedirem al **primer element**
  - **numeros[n-1]: Últim element.**

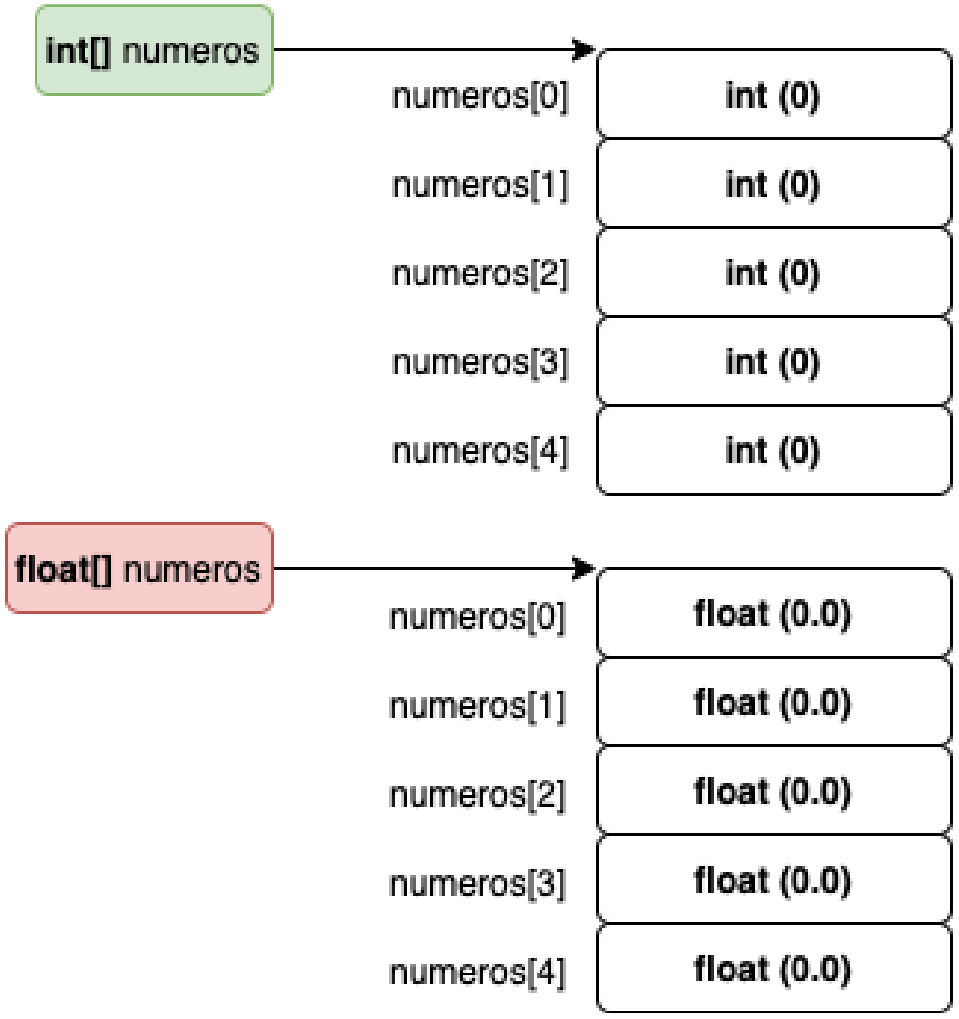
- **Accés lectura** (extraure el valor):

```
int[] numeros = { 53, 15, -22, 60, 6};
System.out.println(numeros[3]); //60
```

- **Accés escriptura** (assignar un valor):

```
int[] numeros = { 53, 15, -22, 60, 6};
numeros[3] = 100;
System.out.println(numeros[3]); //100
```

1.2 ACCÉS ALS ELEMENTS



1.2.1 EXEMPLE

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

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

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

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

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

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

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

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

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

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

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

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

1.3 RECORREGUT D'ARRAIS

- **Lectura dels elements**
- **Els objecte** de tipus **Array** disposen de la **propietat** `length` que ens torna el nombre d'elements que conté.
  - Podem **fer ús d'un bucle comptador** per **accedir a cadascun** dels seus elements

```
int[] dades = {1, 2, 3, 4};

for (int i = 0; i <dades.length; i ++) {
    System.out.println(dades[i]);
}
```

1.3 RECORREGUT D'ARRAIS

- **Assignació de valors**
- De la mateixa manera, podem dur a terme la **inicialització dels elements** d'un array de **forma dinàmica**

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

    int[] dades = new int[10];
    for (int i = 0; i <dades.length; i++) {
       dades[i] = 56 + i;
    }

}
```

ACTIVITATS PRÈVIES

- **Activitat 1.-** Escriu un programa que permeta introduir els valors d'un array de 10 **elements numèrics** i després els visualitzeu.

- **Activitat 2.-** Escriu un programa que permeta introduir els valors de **10 elements alfanumèrics** (tornant a demanar-lo si no es ni un dígit ni una lletra) i després visualitzeu aquells que ocupen una **posició parell**.

1.4 ESTRUCTURA for...each

- Una altra forma de **accés als elements** d'un array **consisteix** a **la iteració sobre els elements.**
  - **Per fer-ho s'introdueix l'estructura** `foreach...`

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

    int[] dades = new int[10];
    for (int dadaN: dades) {
        System.out.println(dadaN);
    }

}
```

A cada Iteració s'accedeix a l'element Situat a la **posició n (0,1,2,3,4,5,6,7,8,...n)**

ACTIVITATS PRÈVIES

- **Activitat 3.-** Modifica **l'activitat 1** perquè faça ús de la sentència `foreach` per a la visualització dels elements
- **Activitat 4.-** Escriu un programa que genere 200 números aleatoris del 0 al 300 i els guarde en un array. Seguidament calcula e imprimeix per pantalla la suma dels elements, el contingut dels quals siga senar (impar). Utilitza l'estructura `for` per a initzialitzar l'array creat i l'estructura `foreach` per a realitzar la suma que s'haurà de mostrar per pantalla.

ACTIVITATS PRÈVIES

- **Activitat 5.-** Escriu un programa que **genere un array de 20 elements** amb valors aleatoris de **tipus enter en el rang 0-9.** A continuació **els ha de mostrar de 4 en 4.** Utilitza, tant per a la generació com per a mostrar els valors, l'estructura `for`

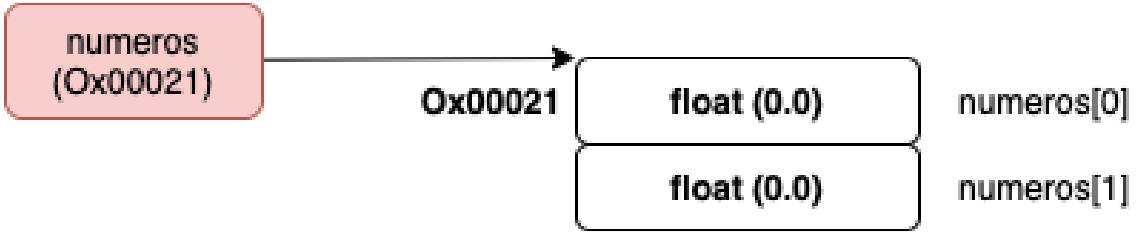
Exemple dexecució

```
1234 5465 1234 7862 4252
```

2. ARRAYS I MÈTODES

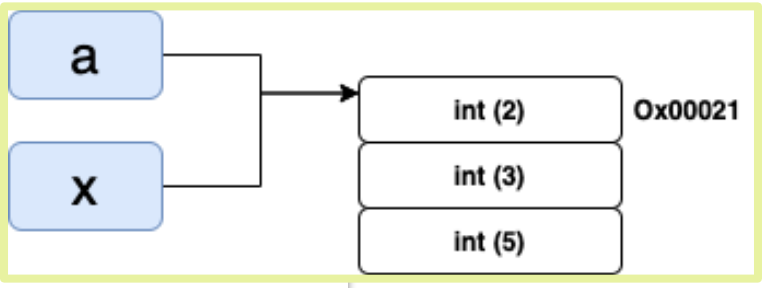
- Podem passar un **array com a paràmetre** a un mètode, però hem de tenir en compte que els arrays són un **tipus de dades referència.**
  - Els **canvis** que **realitzem sobre** els elements **de l'array es mantindran** en retornar **el flux d'execució** al procediment invocador.
  - Les variables de l'array són **punters** a la **zona de memòria** que conté l'array.

```
float[] números = new float[5];
```



2. ARRAYS I MÈTODES. EXEMPLE

- Quan es crida a un mètode i se li passa un array, el mètode fa **una còpia de la referència**, de manera que les 2 variables apuntaran a la mateixa zona de memòria.




```
public static void main (String[] args) {
    int[] a = {2, 3, 5};
    System.out.println (a[0] + "," + a[1] + "," + a[2]);
    canviarArray(a);
    System.out.println (a[0] + "," + a[1] + "," + a[2]);
}

public static void canviarArray (int[] x) {
    x[0] = x[0] * 10;
}
```

La variable **a** apunta a **0x00021**

La variable **x** apunta a **0x00021**

Execució

[2, 3, 5]

[20, 3, 5]

2.1 ACTIVITAT PRÈVIA

- **Activitat 6.-** Implementa els mètodes següents respectant les capçaleres donades:

- `public static int[] crearArray()`: declara, crea e inicialitza un array de 10 elements amb números aleatoris del 0 al 50.
- `public static void visualitzarArray(int[] vector)`: visualitza tot el contingut de l'array.
- `public static int cercarZero(int[] vector)`: cercarà el primer element amb contingut 0 i tornarà el seu índex. Si no hi ha cap element amb contingut 0, tornarà un -1.
- `public static void intercanvia(int[] vector)`: intercanvia els valors de la primera i última posició sempre que la seva longitud (número de elements) siga major o igual que 2.
- `public static boolean sonIguals(int[] vector1, int[] vector2)`: determina si els arrysrebuts son iguals o no ho son (mateix tamany i idèntics elements).

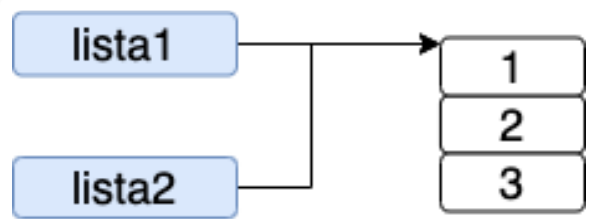
*Per finalitzar **crea un programa** que **ens permeta verificar** cadascun dels mètodes*

2.2 CÒPIA D'ARRAYS

- Quan una variable de **tipus array** s'iguala a una altra variable de **tipus array** es **còpia la referència** i, per tant, **ambdues variables** apuntaran al mateix array.

* La classe **Arrays** ofereix una sèrie de mètodes estàtics predefinits. Amb toString(...) podem imprimir tots els elements de l'array

```
public static void copiarArrays () {
    int[] llista1 = {1, 2, 3};
    System.out.println(Arrays.toString(llista1));
    int[] llista2 = llista1;
    System.out.println(Arrays.toString(llista2));
    llista1[0] = llista1[0] * 10;
    System.out.println(Arrays.toString(llista1));
    System.out.println(Arrays.toString(llista2));
}
```



Execució

[1, 2, 3]
[1, 2, 3]
[10, 2, 3]
[10, 2, 3]

2.2 CÒPIA D'ARRAYS

- Si volem dur a terme una **còpia de l'array**, haurem de **copiar un** a **un** cadascun d**els elements de l'array**.

```
public static void copiarArrays () {

    double[] arrayA = new double[100];
    double[] arrayB = new double[arrayA.length];

    for (int i = 0; i <arrayB.length; i++) {
        arrayB[i] = arrayA[i];
    }

}
```

2.3 ACTIVITAT PRÈVIA

- **Activitat 7.** Defineix un mètode `copiaArray` que, a partir d'un array d'elements cadena rebuts com a argument, cree un array nou i torne la seua còpia.

*Per finalitzar **crea un programa** que **ens permeta verificar el funcionament** per això hauràs de dur a terme les següents accions:*

1.- Declara, crea i inicialitza un array de cadenes `diesDeLaSetmana` que continguen els dies de la setmana (dilluns, dimarts,...diumenge).

2.- Declara, una nova variable `diesDeLaSemana2` i assignant-li el contingut de la variable anterior.

3.- Modifica el primer element de l'array *`diesDeLaSemana2`* traduint-ho a l'anglès (*Monday*) i comprova que s'ha modificat en les dues variables.

4.- Crea una còpia de l'array *`diesDeLaSemana2`* a partir del mètode `copiaArray` creat i assigna-ho a una variable *`diesDeLaSemana3`*. Tradueix el primer dia de la setmana a espanyol i comprova que no ha afectat les 2 variables declarades al **pas 1 i 2**.

2.3 ACTIVITAT PRÈVIA

Això és tot... de moment :-)