

Array

Modulo 226A

Implementare in base alle classi (senza ereditarietà)

Dichiarazione

<tipo>[] <nome variabile>;

```
int[] numeri;
```

```
double[] note;
```

```
String[] nomi;
```

Allocazione in memoria

Esistono due modi per creare un nuovo array

Primo modo:

```
int[] numeri = new int[6]  
  
double[] temperature = new double[5]  
  
String[] nomi = new String[20]
```

Secondo modo (shortcut syntax)

```
int[] numeri = {1,2,3,5,7,11}  
  
String[] nomi = {"Luca","Simona","Fabio","Daniela"}
```

Dimensione array

`<nome array>.length`

```
int[] numeri = {1,2,3,5,7,11}  
System.out.println(numeri.length);
```

Accesso valori

<nome array>[<indice>]

```
int[] numeri = {1,2,3,5,7,11}  
numeri[0] = 9; //{0,2,3,5,7,11}
```

Tutti gli accessi agli array sono verificati in fase di esecuzione; il tentativo di accedere ad un elemento con indice inferiore a zero, o maggiore o uguale alla dimensione dell'array causa un **ArrayOutOfBoundsException**.

Copiare un Array

Esistono due modi per copiare un array

Primo modo :

```
int[] sorgente = {1, 2, 3, 4, 5};  
int[] destinazione = new int[sorgente.length];  
  
for (int i = 0; i < sorgente.length; i++) {  
    destinazione[i] = sorgente[i];  
}
```

Secondo modo (copy of)

```
int[] sorgente = {1, 2, 3, 4, 5};  
int[] destinazione = Arrays.copyOf(sorgente, 5);
```

The background features two large, decorative, curved lines. One line, in shades of blue and green, curves from the top right towards the center. Another line, in shades of green and blue, curves from the bottom left towards the center. Both lines have a soft, multi-layered appearance.

Array Multidimensionali

Dichiarazione multidimensionali

```
int[][] m = new int[3][3];
```

	0	1	2
0	0	0	0
1	0	0	0
2	0	0	0

Java prevede un approccio Row Major Order il quale indica che la prima dimensione dell'array definisce il numero di **righe** e la seconda dimensione il numero di **colonne**.

Inizializzare

```
int[][] matrice= { //scacchiera 8 x 8
    {1,0,1,0,1,0,1,0},
    {0,1,0,1,0,1,0,1},
    {1,0,1,0,1,0,1,0},
    {0,1,0,1,0,1,0,1},
    {1,0,1,0,1,0,1,0},
    {0,1,0,1,0,1,0,1},
    {1,0,1,0,1,0,1,0},
    {0,1,0,1,0,1,0,1}
};
```

```
matrice.length; //numero righe
```

```
matrice[0].length //numero di colonne sulla prima riga
```

```
matrice[n].length //numero di colonne sulla n-esima riga
```

Percorrere un array multidimensionale

```
public static void main(String[] args) {  
    int[][] m = { //matrice 8 x 8  
        {1,0,1,0,1,0,1,0},  
        {0,1,0,1,0,1,0,1},  
        {1,0,1,0,1,0,1,0},  
        {0,1,0,1,0,1,0,1},  
        {1,0,1,0,1,0,1,0},  
        {0,1,0,1,0,1,0,1},  
        {1,0,1,0,1,0,1,0},  
        {0,1,0,1,0,1,0,1}  
    };  
  
    for (int i = 0; i < m.length; i++) { //ciclo delle righe  
        for (int j = 0; j < m[i].length; j++) { //ciclo delle colonne  
            System.out.print(m[i][j]+" "); //stampo il j'esimo elemento della i'esima riga  
        }  
        System.out.println(""); // dopo ogni riga vado a capo  
    }  
}
```