



## Declare, instancie, inicialize e use um array multi-dimensional

Podemos generalizar a ideia de array para construir arrays de duas dimensões, em outras palavras, **array de arrays**. Analogamente, podemos definir arrays de quantas dimensões quisermos.

### Declaração:

```
// Um array de duas dimensões.
```

```
int[][] tabela;
```

```
// Um array de três dimensões.
```

```
int[][][] cubo[];
```

```
// Um array de quatro dimensões.
```

```
int[] [][]hipercubo[];
```

```
// Perceba que as dimensões podem ser definidas do lado
```

```
// esquerdo ou direito da variável.
```

[COPIAR CÓDIGO](#)

### Inicialização:

```
// Inicializando a primeira dimensão com 10 e a segunda com 15
```

```
tabela = new int[10][15];
```

```
// Inicializando a primeira dimensão com 10 e deixando as  
outras
```

```
// para serem iniciadas depois
```

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

```
// Inicializando com valores
```

```
int[][] teste = new int[][]{{1,2,3},{3,2,1},{1,1,1}};
```

[COPIAR CÓDIGO](#)

Acesso:

```
// Acessando a posição (0,1)
```

```
System.out.println(tabela[0][1]);
```

[COPIAR CÓDIGO](#)

Podemos criar um array que não precisa ser "quadrado", ele pode ter tamanhos estranhos:

```
int[][] estranha = new int[2][];
```

```
estranha[0] = new int[20];
```

```
estranha[1] = new int[10];
```

```
for(int i=0;i<estranha.length;i++) {
```

```
    System.out.println(estranha[i].length); // imprime 20 e 10
```

```
}
```

[COPIAR CÓDIGO](#)