



Java

Arrays Bidimensionais

METODOLOGIA

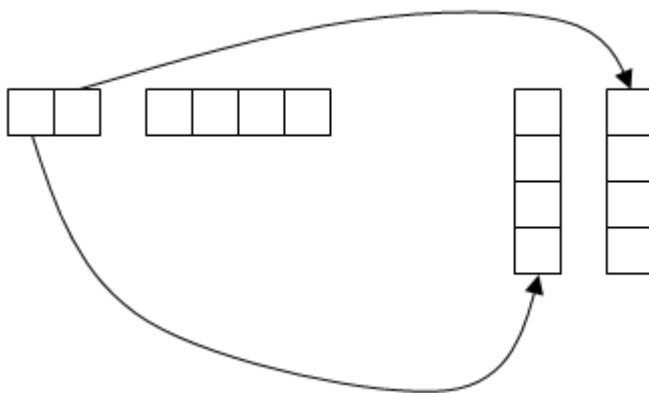
- › Interprete o documento calmamente e com atenção.
- › Acompanhe a execução do exercício no seu computador.
- › Não hesite em consultar o formador para o esclarecimento de qualquer questão.
- › Não prossiga para o ponto seguinte sem ter compreendido totalmente o ponto anterior.
- › Caso seja necessário, execute várias vezes o exercício até ter compreendido totalmente o processo.

CONTEÚDO PROGRAMÁTICO

1. [Arrays Bidimensionais](#)
2. [Ciclos For Encadeados](#)

1. Arrays Bidimensionais

Um array multidimensional não é mais do que um array unidimensional, em que cada posição do mesmo é outro array unidimensional. Vejamos o conceito graficamente:



	0	1	2	3	4	5	6	7	...
0									
1									
2									
3									
...									

Podemos considerar um array bidimensional como uma tabela. Para o manipularmos é apenas preciso referenciar os índices para a linha e para a coluna.

A sintaxe para criar arrays bidimensionais é em tudo semelhante à dos arrays unidimensionais.

```
<TIPO DE DADOS>[][] <NOME DO ARRAY> = new <TIPO DE DADOS>[n][m];
```

Se imaginarmos a estrutura necessária para guardar, por exemplo o tabuleiro de uma sopa de letras de dez por dez, a instrução para criar o mesmo seria a seguinte:

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

Pode assumir-se que o primeiro índice é o índice da linha e o segundo da coluna. Posto isto, para escrever o número 10 na primeira linha e terceira coluna, seria necessário apenas o seguinte código:

```
tabuleiro[1][3] = 10;
```

Para todos os acessos ao array bidimensional, quer para ler o valor como alterá-lo, é necessário especificar os dois índices.

i Apenas foi mencionada a criação de arrays bidimensionais mas não existe limite para o número de dimensões de um array. A sintaxe de criação de arrays com mais dimensões segue precisamente a mesma lógica. Cada par de chavetas retas especifica uma dimensão, o que significa que para criar um array de quatro dimensões a instrução correta seria:

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

2. Ciclos For Encadeados

Como foi visto anteriormente, os ciclos `for`, servem para repetir uma determinada ação várias vezes. Existem algumas situações em que isto ocorre de forma recorrente. Para preencher um array unidimensional usamos um ciclo `for`. No entanto, com um ciclo `for` é apenas possível percorrer uma linha ou uma coluna sendo necessária uma forma de percorrer as linhas e colunas restantes. Para isso, utilizamos um ciclo `for` dentro de outro, ou seja, para percorrermos esse mesmo array e procedermos à respetiva inicialização podemos fazer o seguinte:

```
for (int linha = 0; linha < 10; linha++){  
    for (int coluna = 0; coluna < 10; coluna++){  
        tabuleiro[linha][coluna] = 10;  
    }  
}
```

Isto significa que para cada linha são percorridas todas as colunas dessa mesma linha. A forma como os ciclos percorrem o array bidimensional é exemplificada na figura abaixo, através das setas.

0 1 2 3 4 5 6 7 8 9

