Arrays

FUNDAMENTOS DE PROGRAMAÇÃO

Aulas finais de java! Presente de Despedida.☺

APRESENTAÇÃO DO CONTEÚDO

O que iremos ver:

- Arrays(introdução)
- Diferença entre váriaveis simple vs váriaveis compostas
- Execícios propostos

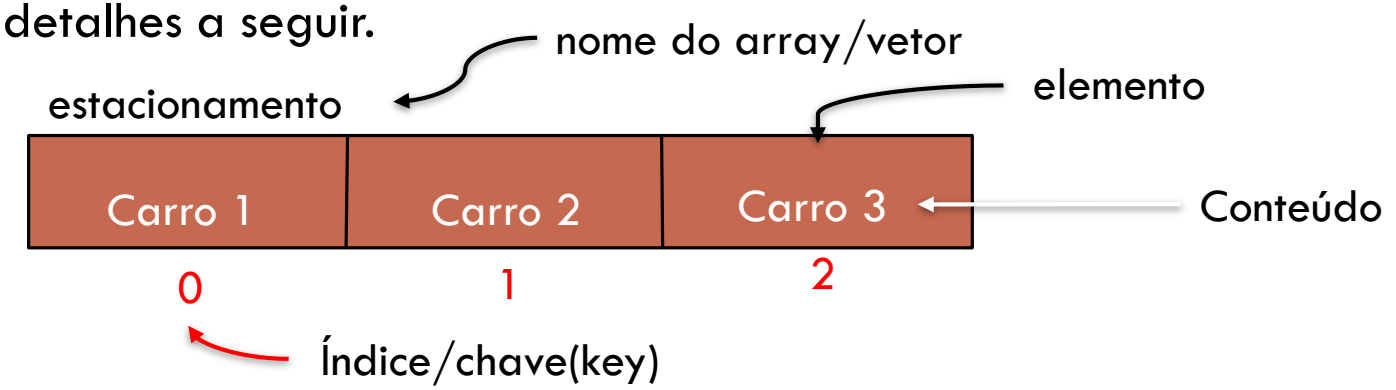
ARRAY(INTRODUÇÃO)

Nessa aula iremos falar sobre as váriaveis compostas, mas antes disso iremos ver a diferença entre variáveis simples e variáveis compostas:

- Variáveis Simples são aquelas que somente armazenam um valor por vez.

- Variáveis Compostas são aquelas que devem ser capazes de armazenar vários valores em uma mesma estrutura e é importante fazer menção que no java essas estruturas compostas apenas são do mesmo tipo, tendo acesso assim ao seu valor por indíces, mas veremos com mais detalhes a seguir.

Exemplo:



ARRAY(DECLARAÇÃO)

Dessa forma, criamos um espaço na mémoria, na verdade uma sequência de espaços que podemos guardar vários valores do mesmo tipo. Então, um array(vetor)/variavel composta é uma variável que tem vários elementos, cada elemento é composto pelo seu valor e por uma chave de identificação, o tamanho dos arrays declara-se num primeiro momento e não pode ser trocado.

Sintaxe:

```
<tipo_dado> <nome_array>[ ];
<nome_array> = new <tipo_dado>[<tamanho>];
```

Exemplo de declaração:

```
String estacionamento[ ];
estacionamento = new String[3];
```

ARRAY(ACESSO)

Os arrays são numerados a partir de ***zero***, que é o primeiro elemento, até o ***tamanho-1*** que é o último elemento. Isto é, se temos um array de dez elementos, o primeiro elemento sería o ***zero*** (***0***) e o último elemento sería o ***nove*** (***9***).

Para acessar um elemento especifico utilizamos os parentesis rectos da forma seguinte:

```
<nome_array>[<índice>];
```

Entendemos por acesso, tanto a leitura como a escrita de valores no array.

Exemplo:

```
estacionamento[0]; // Carro 1
```

EXEMPLO

Para aceder ao terceiro elemento o faríamos do seguinte jeito:

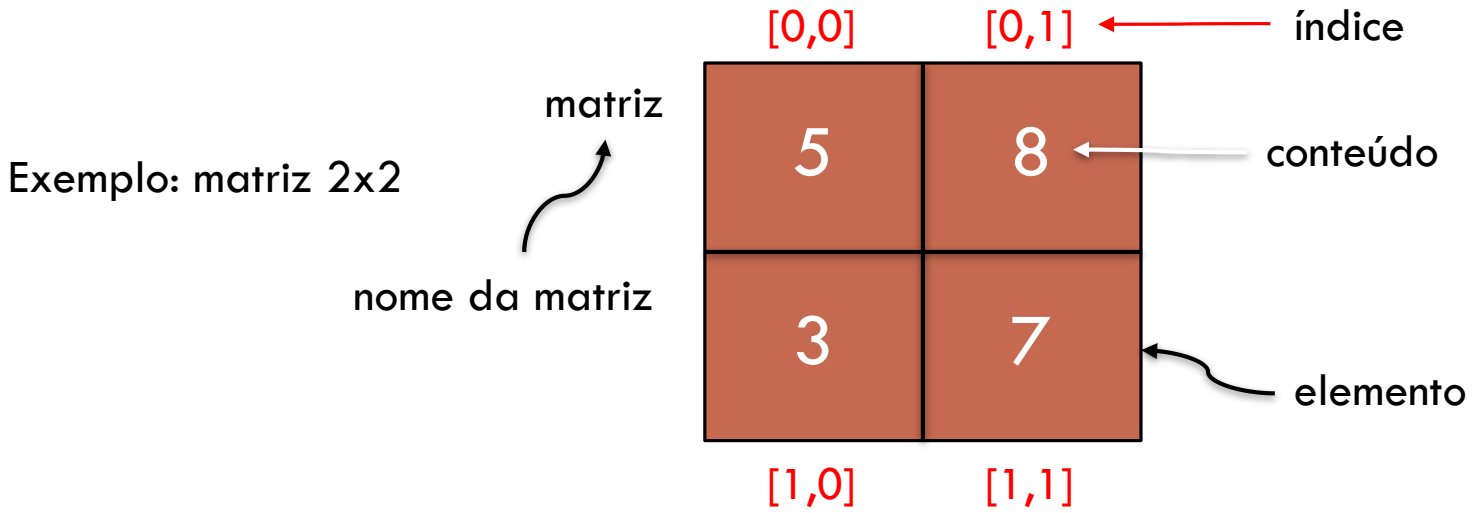
```
// Leitura do valor.
String x = estacionamento[2];


// Atribuição ou escrita de um valor:
estacionamento[2] = “Carro 4”;
```

ARRAYS MULTIDIMENSIONAIS

Podemos declarar arrays com varios subíndices, podendo ter arrays de dois niveis (matrizes), arrays de três niveis (cubo), etc

Entretanto, a partir do terceiro nível perdemos a perspectiva geométrica.



ARRAYS MULTIDIMENSIONAIS

Para declarar e inicializar um array de varios subíndices, faremos da seguinte forma:

```
<tipo_dado> <nome_array>[ ][ ];
<nome_array> = new <tipo_dado>[<tamanho>][<tamanho>];
```

Deste jeito podemos declarar uma matriz de 2x2 da seguinte forma:

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

ARRAYS MULTIDIMENSIONAIS(ACESSO)

O acesso é feito da mesma forma que anteriormente:

```
// Para ter acesso a um elemento dentro da matriz, fazemos da seguinte maneira:
int x = matriz[1][1];

// Para atribuir um valor
matriz[1][1] = 2;
```

ARRAYS MULTIDIMENSIONAIS - CONTINUAÇÃO

```
int matriz[][];
matriz = new int[4][4];

for (int x=0; x < matrix.length; x++) {
     for (int y=0; y < matriz[x].length; y++) {
         System.out.println (matriz[x][y]);
     }
}
```

ARRAYS - INICIALIZAÇÃO

```
<tipo_dado> array[] = {elemento1,elemento2,...,elementoN};

// Temos um array de 5 elementos.
char array[ ] = {'a','b','c','d','e'};

// Temos um array de 4x4 elementos.
int array[ ][ ] = {{1,2,3,4}, {5,6,7,8}};
```

FIM

Feito por:

- Anselmo Nhamage