

Módulo 4

Array

Programação Orientada a Objetos I

Java

(Rone Ilídio)

Array

- O *array* é um grupo de posições contíguas que possuem o mesmo nome e o mesmo tipo. Para referir-se a uma localização ou elemento particular no *array*, especificamos o nome do *array* e o número da posição (índice ou subscrito) do elemento particular do *array*.
- O índice ou subscrito deve ser um inteiro ou uma expressão inteira.
- A primeira posição é sempre 0.

Array

- Todos os *arrays* em Java conhecem seu próprio comprimento, informado da seguinte forma: `nome_array.length`

Array - Declaração

- Para declarar um *array* de inteiros de 12 posições:

```
int c[] = new int[12];
```

- O operador *new* é utilizado para alocar espaço na memória.

- Esta operação é similar à anterior:

```
int c[];
```

```
c = new int[12];
```

Obs: quando se aloca um *array* seu elemento recebem, por *default*, 0 para variáveis numéricas, *false* para *boolean* ou *null* para qualquer tipo não primitivo.

Array – Iniciando Valores

- Para iniciar um vetor com valores utiliza-se a sintaxe:

```
int n[] = { 10,20,30,40,50};
```

- O resultado desta declaração é um *array* de 5 posições (0 a 4)

Array – Acessando Elementos

- Para acessar um elemento de um *array* basta escrever o nome do *array* e a posição do elemento. Ex:

```
int n[] = { 10,20,30,40,50};
```

```
int x = n[3];
```

A variável x terá valor igual a 40.

```
import javax.swing.*;

public class vetor{

    public static void main(String[] args){
        int vet[] = new int[5];
        String entrada;
        for (int i=0; i<=4 ;i++ ){
            entrada = JOptionPane.showInputDialog("Informe um
                                                    valor.");

            vet[i] = Integer.parseInt(entrada);
        }
        for (int i=0; i<=4 ;i++ ){
            System.out.println( i + " - " + vet[i] + "\n");
        }
    }
}
```

Array – Multidimensionais

- Também são chamados de matrizes.
- O Java não aceita diretamente *array* multidimensionais.
- Porém, aceita *arrays* formados por elementos que também são *arrays*.
- Exemplo, a declaração gera um array multidimensional com 3 linhas e 4 colunas:

```
int b[][];
```

```
b = new int[3][4];
```



```
import javax.swing.*;

public class matriz{

    public static void main(String[] args){
        int matriz[ ][ ] = new int[3][2];
        String entrada;
        for (int i=0; i<=2 ;i++ )
            for (int u=0; u<=1; u++)
                entrada = JOptionPane.showInputDialog("Informe um valor.");
                matriz[i][u] = Integer.parseInt(entrada);

        for (int i=0; i<=2 ;i++ )
            for (int u=0; u<=1; u++)
                System.out.println( i + " " + u + " - " + matriz[i][u] + "\n");

    }

}
```

Exercícios

- Crie um programa onde o usuário informa todos os elementos de um vetor (10 posições) e obtenha como resultado qual é o maior elemento.
- Crie aplicativo com um vetor onde seus elementos são preenchidos na sua declaração. Esse aplicativo deve receber do usuário um número e verificar se tal número existe ou não no vetor
- Crie *applet* que receba do usuário uma seqüência de números. Tal applet deverá imprimir essa seqüência de números ordenada.

Exercícios

- Crie um aplicativo que contenha uma matriz 3x3, de inteiros. Essa matriz deve ser inicializada com valores vindos do usuário. Ao final essa matriz deverá ser impressa na tela.
- Crie um aplicativo que preencha uma matriz 4x4 de inteiros com números randômicos variando entre 0 e 100. Ao final deve ser impressa a soma da diagonal principal desta matriz.

Array como parâmetro de métodos

- Quando um *array* é passado como parâmetro para um método, essa passagem é sempre feita por referência
- Na declaração do método não é necessário especificar o tamanho do vetor

```
import javax.swing.*;

public class array_metodo extends JApplet{

    public void init(){

        int vet_teste[] = {3,34,3,87,3,4};

        dobraArray(vet_teste);

        String saida = "";

        for(int i=0; i<vet_teste.length; i++)

            saida = saida + " " + vet_teste[i];

        JOptionPane.showMessageDialog(null,saida);

    }

    public void dobraArray(int vetor[]){

        for(int i=0; i<vetor.length; i++)

            vetor[i] = vetor[i] * 2;

    }

}
```