

Programação I

Vetores e Matrizes

Samuel da Silva Feitosa

Aula 5

Introdução

- Nesta aula vamos estudar os conceitos a respeito dos tipos de dados homogêneos, **vetores** e **matrizes**.
 - Também são conhecidos como **arrays**, que podem ter uma ou mais dimensões.
 - Seus elementos são como variáveis simples do seu tipo, que permite armazenar ou recuperar valores.

Vetores

Vetores

- Os elementos são organizados em uma única dimensão.
 - Cada elemento é acessível por meio de um índice relacionado à sua posição.
 - O índice 0 (zero) permite acessar o primeiro elemento, o índice 1 (um) acessa o segundo, e assim por diante.
- Declaração de um vetor:
<Tipo> identificador[];

Alocação de Vetores

- Vetores são objetos em Java.
 - Devem ser alocados em tempo de execução, quando seu número de elementos é indicado a partir de uma variável ou número inteiro.

identificador = new <Tipo>[tamanho];

- Declaração e alocação podem ser indicadas juntas:

<Tipo> identificador[] = new <Tipo>[tamanho];

Exemplo de uso de Vetor

- Declaração e alocação de um vetor, seu preenchimento com valores dados pelo usuário e exibição dos seus elementos.

```
public static void main(String[] args) {  
    int v[] = new int[5]; // declaração e alocação  
    Scanner sc = new Scanner(System.in);  
    for (int i = 0; i < 5; i++) {  
        System.out.println("Informe v[" + i + "]: ");  
        v[i] = sc.nextInt();  
    }  
    for (int i = 0; i < v.length; i++) {  
        System.out.println("v[" + i + "] = " + v[i]);  
    }  
    sc.close();  
}
```

Mais informações sobre Vetores

- Todo vetor em Java é automaticamente preenchido com valores nulos.
 - Elementos recebem **0 (zero)** quando vetor for do tipo numérico, **false** quando for booleano e **null** quando for de tipo objeto.
- Também é possível inicializar o vetor em sua declaração.

```
double vetor[] = { 2.3, -5.2, 15, 0, 0.7 };
```

Operações com Vetores

- A biblioteca do Java possui uma classe **Arrays** que oferece facilidades para realizar operações de preenchimento, comparação, exibição e cópias de vetores/matrizes.
 - Método **fill**: usado para preencher um vetor.
 - Método **equals**: compara dois vetores.
 - Método **toString**: constrói uma String com os valores presentes no vetor.
 - Métodos **copyOf** e **copyOfRange**: efetuam cópias totais ou parciais dos vetores.
 - Método **sort**: realiza a ordenação do vetor.

Exemplos de Operações

- Testa diversas operações sobre vetores.

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    System.out.println("Informe o número de elementos: ");  
    int tam = sc.nextInt();  
    System.out.println("Preencher com qual valor? ");  
    double val = sc.nextDouble();  
    double vetor[] = new double[tam];  
    Arrays.fill(vetor, val);  
    System.out.println("Conteúdo: " + Arrays.toString(vetor));  
    double copia[] = Arrays.copyOf(vetor, vetor.length);  
    System.out.println("Cópia: " + Arrays.toString(copia));  
    System.out.println("São iguais? " + Arrays.equals(vetor, copia));  
    sc.close();  
}
```

Matrizes

Matrizes

- Enquanto nos vetores os elementos são organizados em **uma única dimensão**, nas matrizes, eles podem ser distribuídos em **várias dimensões distintas**.
 - A declaração e a alocação podem ser indicadas juntas, utilizando-se um par de colchetes para cada dimensão.

<Tipo> identificador = new <Tipo>[dim_1][dim_2]...[dim_n];

Exemplo de uso de Matriz

- Lê e mostra informações de uma matriz.

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    double m[][] = new double[3][4];  
  
    for (int i = 0; i < 3; i++) {  
        for (int j = 0; j < 4; j++) {  
            System.out.println("Informe m[" + i + "][" + j + "]: ");  
            m[i][j] = sc.nextDouble();  
        }  
    }  
  
    for (int i = 0; i < 3; i++) {  
        for (int j = 0; j < 4; j++) {  
            System.out.print(m[i][j] + "\t");  
        }  
        System.out.println();  
    }  
}
```

Exemplo de inicialização de Matrizes

- Inicializa e apresenta o conteúdo.

```
public static void main(String[] args) {  
    int m1[][] = { {1, 2, 3},  
                   {4, 5, 6} };  
  
    int m2[][] = { {1, 2},  
                   {3, 4},  
                   {5, 6} };  
  
    System.out.println("Matriz 1");  
    for (int i = 0; i < 2; i++) {  
        for (int j = 0; j < 3; j++) {  
            System.out.print(m1[i][j] + "\t");  
        }  
        System.out.println();  
    }  
  
    System.out.println("Matriz 2");  
    for (int i = 0; i < 3; i++) {  
        for (int j = 0; j < 2; j++) {  
            System.out.print(m2[i][j] + "\t");  
        }  
        System.out.println();  
    }  
}
```

res

Considerações Finais

- Nesta aula revisamos os conceitos de **Vetores** e **Matrizes**, e sua aplicação na linguagem Java.
- Também vimos algumas funcionalidades presentes na biblioteca padrão de Java para manipular **Arrays**.
 - Preenchimento, comparação, ordenação, etc.
- Estas estruturas de dados são bastante usadas na prática do dia-a-dia na computação.

Exercícios

- Implementar os exercícios da lista na plataforma *beecrowd*:
 - Lista 3 - Vetores e Matrizes.
- Instruções de acesso à turma no SIGAA.