

Vetores

Em Java, um vetor é uma estrutura de dados que armazena uma coleção de elementos do mesmo tipo. É semelhante a uma lista de compras, onde cada item é uma entrada do vetor.

Por exemplo, se quisermos armazenar as notas de um aluno em uma disciplina, poderíamos criar um vetor de notas do tipo `double` da seguinte forma:

```
double[] notas = {8.5, 9.0, 7.5, 6.0, 8.0};
```

Nesse caso, o vetor `notas` contém cinco elementos do tipo `double`.

Para acessar um elemento específico do vetor, podemos usar um índice, que começa em 0 para o primeiro elemento e aumenta em 1 para cada elemento subsequente. Por exemplo, para acessar a primeira nota do vetor `notas`, podemos fazer o seguinte:

```
double primeiraNota = notas[0];
```

Também podemos alterar o valor de um elemento do vetor atribuindo um novo valor a um índice específico. Por exemplo, para mudar a segunda nota para 9.5, podemos fazer o seguinte:

```
notas[1] = 9.5;
```

Propriedades do vetor:

1. **Tamanho fixo:** Quando um vetor é criado, seu tamanho é fixado e não pode ser alterado durante a execução do programa. Para adicionar ou remover elementos de um vetor, é necessário criar um novo vetor com um tamanho diferente e copiar os elementos do vetor original para o novo vetor.
2. **Tipo de dados:** Todos os elementos de um vetor devem ter o mesmo tipo de dados. Por exemplo, um vetor de inteiros só pode armazenar valores inteiros.
3. **Índice baseado em zero:** O índice de um vetor começa em 0 para o primeiro elemento e aumenta em 1 para cada elemento subsequente. O último elemento do vetor tem um índice igual ao tamanho do vetor menos um.
4. **Acesso direto aos elementos:** É possível acessar diretamente qualquer elemento de um vetor usando um índice. Isso permite a leitura e a gravação dos elementos do vetor.

5. **Alocação contígua de memória:** Os elementos de um vetor são armazenados em uma região contígua de memória, o que permite o acesso rápido aos elementos usando aritmética de ponteiros.
6. **Propriedade length:** Cada vetor tem uma propriedade chamada length, que retorna o número de elementos do vetor. Por exemplo, se um vetor tem 5 elementos, sua propriedade length é igual a 5.

Exemplo do uso da propriedade length:

```
int[] numeros = {1, 2, 3, 4, 5};  
System.out.println("O tamanho do vetor é " + numeros.length);
```

```
0 tamanho do vetor é 5
```

Declarar vs inicializar:

Em Java, podemos declarar um vetor sem inicializá-lo ou declarar e inicializar um vetor em uma única linha. A diferença entre essas duas abordagens é que, quando declaramos um vetor sem inicializá-lo, seu valor padrão é `null` para um vetor de objetos e `0` ou `false` para um vetor de tipos primitivos. Quando inicializamos um vetor em uma única linha, seus elementos são definidos na própria declaração.

Declaração de um vetor sem inicialização:

```
int[] numeros; // declara um vetor de inteiros
```

Inicialização de um vetor:

```
int[] numeros = {1, 2, 3, 4, 5}; // inicializa um vetor de inteiros com 5  
elementos
```

Outra forma de inicializar um vetor é usando a palavra-chave `new` seguida do tipo do vetor e do tamanho desejado entre colchetes. Por exemplo:

```
int[] numeros = new int[5]; // inicializa um vetor de inteiros com 5  
elementos, todos com valor padrão de 0
```

Podemos atribuir valores aos elementos do vetor depois de inicializá-lo, usando seu índice. Por exemplo:

```
numeros[0] = 10; // atribui o valor 10 ao primeiro elemento do vetor  
numeros[1] = 20; // atribui o valor 20 ao segundo elemento do vetor  
// ... e assim por diante
```

Note que, quando declaramos um vetor sem inicializá-lo, devemos inicializá-lo antes de usá-lo em outras partes do código. Se tentarmos acessar um vetor não inicializado, receberemos um erro em tempo de execução.

Exercícios:

1. Escreva um programa Java que inicialize um vetor de inteiros com 5 elementos e imprima o valor de cada elemento na tela.

```
int[] numeros = {1, 2, 3, 4, 5};
for (int i = 0; i < numeros.length; i++) {
    System.out.println("O valor do elemento " + i + " é " + numeros[i]);
}
```

2. Escreva um programa Java que calcule a soma de todos os elementos de um vetor de inteiros.

```
int[] numeros = {1, 2, 3, 4, 5};
int soma = 0;
for (int i = 0; i < numeros.length; i++) {
    soma += numeros[i];
}
System.out.println("A soma dos elementos do vetor é " + soma);
```

3. Escreva um programa Java que encontre o maior elemento em um vetor de inteiros.

```
int[] numeros = {5, 2, 10, 1, 8};
int maior = numeros[0];
for (int i = 1; i < numeros.length; i++) {
    if (numeros[i] > maior) {
        maior = numeros[i];
    }
}
System.out.println("O maior elemento do vetor é " + maior);
```

4. Escreva um programa Java que verifique se um elemento está presente em um vetor de inteiros.

```
int[] numeros = {1, 2, 3, 4, 5};
int elemento = 3;
boolean presente = false;
for (int i = 0; i < numeros.length; i++) {
    if (numeros[i] == elemento) {
        presente = true;
        break;
    }
}
```

```
if (presente) {  
    System.out.println("O elemento " + elemento + " está presente no  
vetor.");  
} else {  
    System.out.println("O elemento " + elemento + " não está presente no  
vetor.");  
}
```

5. Desafio:

Escreva um programa Java que leia dois vetores de inteiros, com o mesmo tamanho, e verifique se eles são iguais, ou seja, se cada elemento do primeiro vetor é igual ao elemento correspondente do segundo vetor. Se os vetores forem iguais, o programa deve imprimir "Os vetores são iguais". Caso contrário, o programa deve imprimir "Os vetores são diferentes".

Thiago André Cardoso Silva

twitter/instagram: [@programador_who](#)