



PECS | Programa de Pós-Graduação em
Engenharia de Computação e Sistemas

ALGORITMOS E ESTRUTURAS DE DADOS

Disciplina: ALGORITMOS E ESTRUTURAS DE DADOS

Carga Horária: 60h

Professor: **Dr. Reinaldo**

Introdução aos Vetores em Java

Os vetores são estruturas de dados fundamentais em Java, permitindo o armazenamento e manipulação de múltiplos elementos do mesmo tipo em uma única variável. Eles oferecem acesso rápido aos dados através de índices numéricos, facilitando a resolução de problemas complexos.



Declaração de Vetores

Sintaxe:

Para declarar um vetor em Java, você usa a seguinte sintaxe: `tipo_de_dados[] nome_do_vetor;`. Por exemplo: `int[] numeros;` ou `String[] nomes;`.

Tamanho do Vetor

Ao declarar um vetor, você pode opcionalmente especificar o seu tamanho, como `int[5] numeros;`, que cria um vetor com espaço para 5 elementos.

Sintaxe:

Para declarar um vetor em Java, você usa a seguinte sintaxe: `tipo_de_dados[] nome_do_vetor;`. Por exemplo: `int[] numeros;` ou `String[] nomes;`.

Vetores

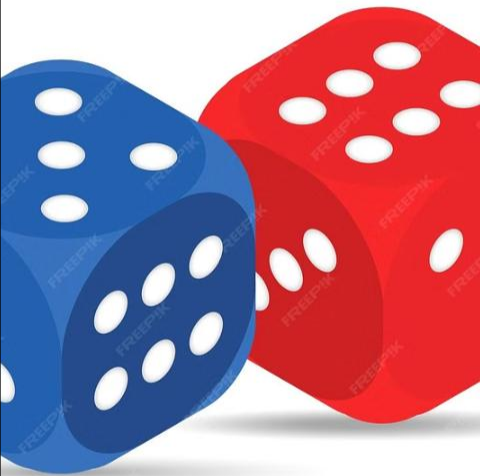
Em Java, os vetores podem ser inicializados de várias maneiras. A forma mais comum é declarar o tamanho do vetor e atribuir valores aos seus elementos individualmente. Também é possível inicializar um vetor com valores pré-definidos, usando chaves { } para delimitar os elementos.

Inicialização manual: `int[] numeros = new int[5]; numeros[0] = 10; numeros[1] = 20; numeros[2] = 30;`

Inicialização com valores pré-definidos: `String[] nomes = {"João", "Maria", "Pedro", "Ana"};`

Inicialização com loop for: `int[] valores = new int[10]; for (int i = 0; i < 10; i++) { valores[i] = i * 2; }`

Acessando Elementos de Vetores



1

Índices

Em vetores, cada elemento é acessado por meio de um índice, que é um número inteiro que representa a posição do elemento no vetor, começando do zero.

2

Sintaxe de Acesso

Para acessar um elemento de um vetor, usa-se a seguinte sintaxe: `nomeDoVetor[índice]`, onde `nomeDoVetor` é o identificador do vetor e `índice` é o número inteiro que representa a posição do elemento.

3

Limites de Acesso

O acesso a elementos de um vetor deve respeitar os limites do vetor, ou seja, o índice deve estar entre 0 e o tamanho do vetor menos 1. Caso contrário, ocorrerá um erro de execução.

Tamanho de Vetores

Um dos aspectos fundamentais ao trabalhar com vetores em Java é compreender o seu tamanho. Cada vetor possui uma dimensão fixa, determinada no momento de sua declaração e inicialização. Essa informação é essencial para acessar e manipular corretamente os elementos armazenados no vetor.

100

Elementos

O tamanho de um vetor representa a quantidade máxima de elementos que ele pode armazenar.

0

Índices

Os índices de um vetor em Java sempre variam de 0 a (tamanho-1), permitindo acessar todos os elementos de forma eficiente.

Conhecer o tamanho do vetor é crucial para realizar operações como iteração, busca, inserção e remoção de elementos de forma segura e confiável. Essa informação deve ser sempre verificada e considerada ao programar com vetores.

Operações com Vetores

Adição e Subtração

É possível realizar operações de adição e subtração entre vetores de mesmo tamanho, elemento a elemento. O resultado é um novo vetor com as operações aplicadas.

Multiplicação por Escalar

Multiplicar um vetor por um número (escalar) tem o efeito de aumentar ou diminuir todos os seus elementos proporcionalmente.

Produto Escalar

O produto escalar entre dois vetores é obtido multiplicando-se cada elemento de um vetor pelo elemento correspondente do outro vetor e somando todos os resultados.

Produto Vetorial

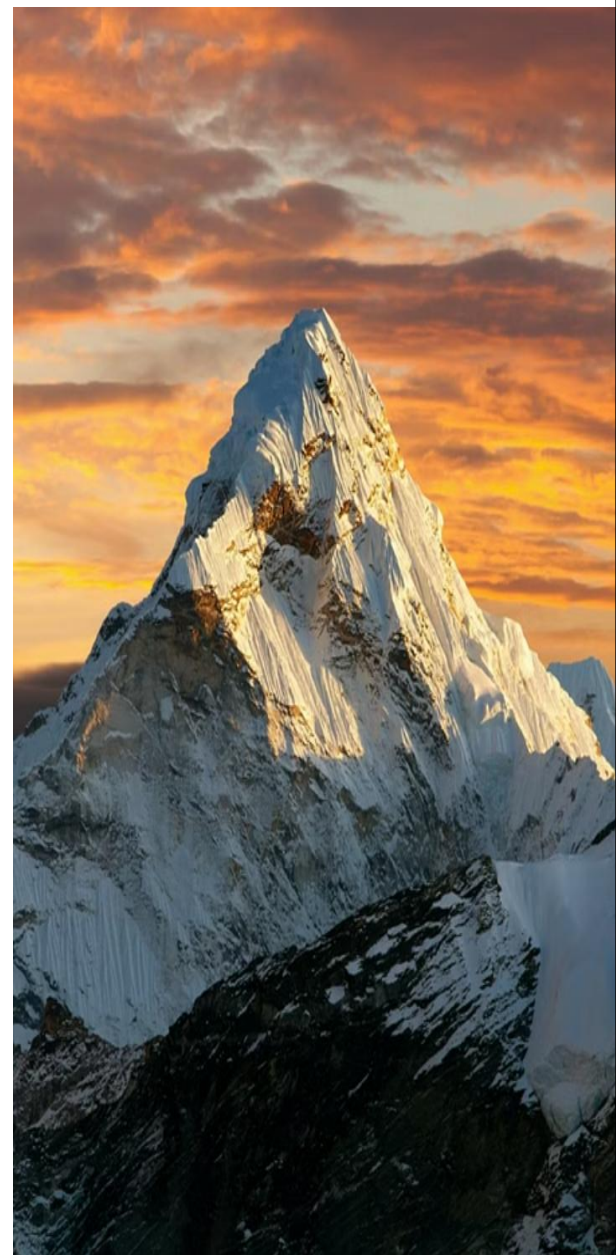
O produto vetorial entre dois vetores é um novo vetor perpendicular aos vetores originais. Essa operação é útil em cálculos de física e engenharia.

Ordenação de Vetores

A ordenação de vetores é uma técnica fundamental em programação, permitindo organizar os elementos de um vetor em uma sequência específica, como ordem crescente ou decrescente. Isso é especialmente útil quando precisamos realizar pesquisas ou manipulações eficientes em grandes conjuntos de dados.

Existem diversos algoritmos de ordenação disponíveis em Java, como **Bubble Sort**, **Insertion Sort**, **Selection Sort**, **Merge Sort** e **Quick Sort**. Cada algoritmo possui suas próprias vantagens e desvantagens em termos de complexidade de tempo e espaço.

A escolha do algoritmo de ordenação mais adequado depende de fatores como o tamanho do vetor, a distribuição dos dados e os requisitos de desempenho da aplicação. Por exemplo, o Quick Sort é geralmente considerado um dos algoritmos mais eficientes para ordenação de grandes vetores.



Operações Matemáticas

Operadores Aritméticos

Java suporta os operadores aritméticos básicos, como adição (+), subtração (-), multiplicação (*), divisão (/) e módulo (%).

Precedência de Operações

A ordem de execução dos operadores matemáticos segue as regras de precedência, como parênteses, exponenciação, multiplicação/divisão e adição/subtração.

Expressões Complexas

Combinando múltiplos operadores, é possível criar expressões matemáticas complexas que realizam cálculos avançados.