

### 1. O que é uma matriz?

Uma matriz em programação é uma estrutura de dados que armazena elementos em linhas e colunas, formando uma tabela bidimensional. Em Java, podemos declarar e inicializar uma matriz para armazenar diferentes tipos de dados, como inteiros, strings, ou objetos.

Exemplo: Declaração de uma matriz de inteiros

```
int[][] matriz = new int[3][3];
```

### 2. Inicializando uma matriz

Existem várias maneiras de inicializar uma matriz em Java. Vamos começar inicializando com valores padrão.

Exemplo 1: Inicialização manual

```
int[][] matriz = {  
    {1, 2, 3},  
    {4, 5, 6},  
    {7, 8, 9}  
};
```

### 3. Acessando elementos da matriz

Para acessar um elemento em uma matriz, usamos a posição (linha, coluna) correspondente. Lembre-se que em Java, os índices começam em 0.

Exemplo: Acessar e imprimir um elemento

```
System.out.println("Elemento da primeira linha e segunda coluna: " +  
matriz[0][1]);
```

#### 4. Percorrendo uma matriz

Para percorrer todos os elementos de uma matriz, usamos loops aninhados.

Exemplo: Percorrer uma matriz 3x3

```
for (int i = 0; i < matriz.length; i++) {  
    for (int j = 0; j < matriz[0].length; j++) {  
        System.out.print(matriz[i][j] + " ");  
    }  
    System.out.println(); // Pula para a próxima linha  
}
```

#### 5. Exercícios práticos

Exercício 1: Criar e preencher uma matriz

```
import java.util.Scanner;  
  
public class MatrizExemplo {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        int[][] matriz = new int[3][3];  
  
        // Preencher a matriz com valores do usuário  
        for (int i = 0; i < 3; i++) {  
            for (int j = 0; j < 3; j++) {  
                System.out.print("Digite o valor para a posição [" + i + "][" + j +  
": ");  
                matriz[i][j] = scanner.nextInt();  
            }  
        }  
  
        // Exibir a matriz  
        System.out.println("Matriz preenchida:");  
        for (int i = 0; i < 3; i++) {
```

```

        for (int j = 0; j < 3; j++) {
            System.out.print(matriz[i][j] + " ");
        }
        System.out.println();
    }
}

```

Exercício 2: Matriz identidade

```

int[][] matrizIdentidade = new int[3][3];

for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        if (i == j) {
            matrizIdentidade[i][j] = 1; // Diagonal principal
        } else {
            matrizIdentidade[i][j] = 0;
        }
    }
}

```

## 6. Operações com Matrizes

Vamos demonstrar algumas operações com matrizes, como somar duas matrizes.

Exemplo: Soma de duas matrizes 2x2

```

int[][] matrizA = {{1, 2}, {3, 4}};
int[][] matrizB = {{5, 6}, {7, 8}};
int[][] matrizSoma = new int[2][2];

for (int i = 0; i < matrizSoma.length; i++) {
    for (int j = 0; j < matrizSoma.length; j++) {
        matrizSoma[i][j] = matrizA[i][j] + matrizB[i][j];
    }
}

```

```

}

// Exibir a matriz resultante
for (int i = 0; i < matrizSoma.length; i++) {
    for (int j = 0; j < matrizSoma[0].length; j++) {
        System.out.print(matrizSoma[i][j] + " ");
    }
    System.out.println();
}

```

Exemplo de um método em Java que retorna uma matriz. No exemplo, criamos um método chamado `gerarMatriz`, que recebe dois parâmetros (o número de linhas e o número de colunas) e preenche a matriz com valores incrementais, retornando-a ao final.

```

public static int[][] gerarMatriz(int linhas, int colunas) {
    int[][] matriz = new int[linhas][colunas];
    int valor = 1;

    // Preenchendo a matriz com valores incrementais
    for (int i = 0; i < linhas; i++) {
        for (int j = 0; j < colunas; j++) {
            matriz[i][j] = valor++;
        }
    }
    // Retorna a matriz preenchida
    return matriz;
}

```

## Estudo de caso

- **Requisitos:**

- Dois jogadores humanos devem poder interagir com o jogo.
- O jogo deve apresentar um tabuleiro 3x3.
- Os jogadores devem alternar seus turnos para marcar uma casa vazia com seu respectivo símbolo (X ou O).
- O jogo deve verificar a cada turno se houve um vencedor (três símbolos iguais em linha, coluna ou diagonal).
- O jogo deve verificar se houve um empate (todas as casas preenchidas sem um vencedor).
- O jogo deve exibir o resultado (vencedor ou empate).