

Aula 04

Estrutura de dados básicas: Vetores e Matrizes

Rogério Pereira Junior

rogeriopereirajunior@gmail.com

30 de abril de 2024

Agenda

1 Vetores

2 Matrizes

Vetores

Vetores

- Variáveis são referências para um endereço de memória
- Eu prefiro lembrar o nome dia a 0xE6

| Endereço | 0x82 | 0xB4 | 0xE6 | 0X118 | 0X14A | | 0X49C | 0X4CE | 0X500 |
|----------|------|------|------|-------|-------|-----|-------|-------|-------|
| Conteúdo | | | 25 | | | ... | | 10 | |

int dia1;

/ reserva um espaço na memória */*

dia1 = 25;

/ atribuiu um valor a variável */*

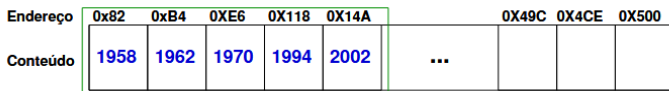
int dia2 = 10;

Questão

Desenvolva um programa que leia e armazene a idade de 5 pessoas. Após efetuar a leitura de todas as datas, informe a pessoa com menor idade.

Vetores

- Um vetor é uma estrutura de dados que armazena elementos de forma **contígua** na memória e fornece acesso a esses elementos por meio de um **índice**.
- O primeiro elemento é referenciado pelo índice 0, o segundo por 1, . .



```
int ano[5]; /* reservando 5 endereços de memória */
```

```
ano[0] = 1958;  
ano[1] = 1962;  
ano[2] = 1970;  
ano[3] = 1994;  
ano[4] = 2002;
```

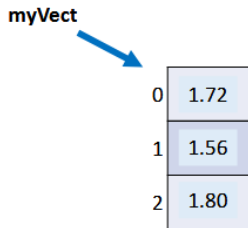
| índice | valor |
|--------|-------|
| 0 | 1958 |
| 1 | 1962 |
| 2 | 1970 |
| 3 | 1994 |
| 4 | 2002 |

Uma definição para vetores

Um vetor é uma coleção de variáveis de um mesmo tipo que é referenciado por um nome comum

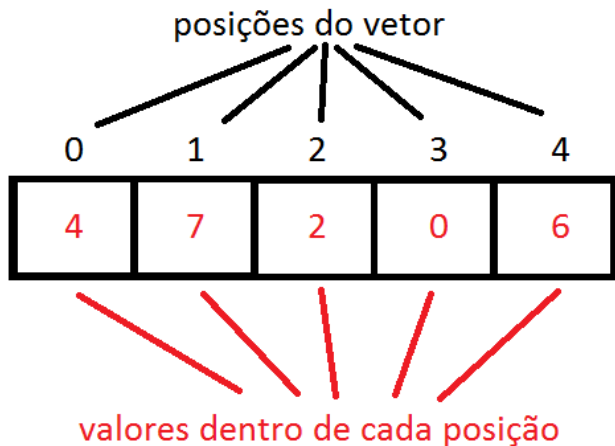
Vetores

- Em programação, "vetor" é o nome dado a arranjos unidimensionais
- Arranjo (array) é uma estrutura de dados:
 - Homogênea (dados do mesmo tipo)
 - Ordenada (elementos acessados por meio de posições)
 - Alocada de uma vez só, em um bloco contíguo de memória
- Vantagens:
 - Acesso imediato aos elementos pela sua posição
- Desvantagens:
 - Tamanho fixo
 - Dificuldade para se realizar inserções e deleções

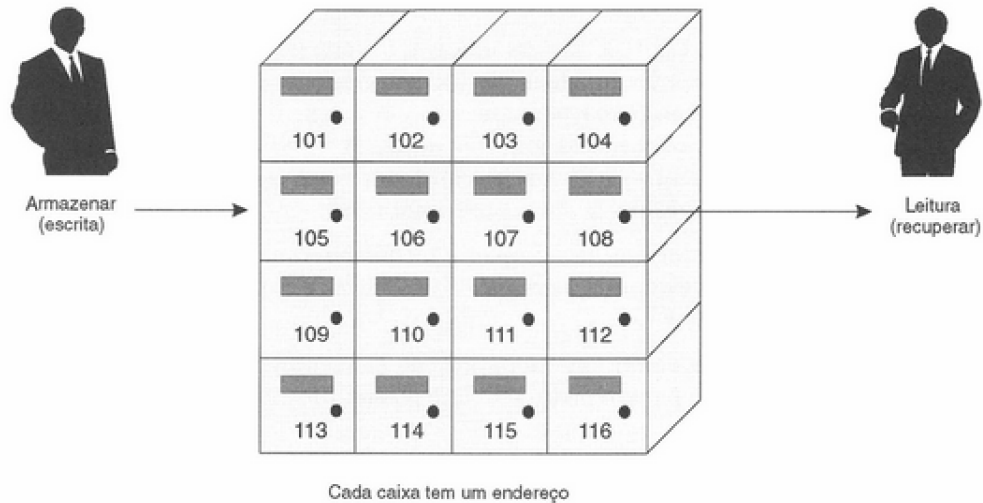


Vetores - Introdução

- Um vetor , ou arranjo (= array), é uma estrutura de dados que armazena uma sequência de objetos, todos do mesmo tipo, em posições consecutivas da memória RAM do computador.



Vetores - Introdução



Vetores - Iniciando um vetor

```
//Inicialização após a declaração:
int[] vetor;
vetor = new int[5];

int[] vetor = new int[5]; // Os elementos serão inicializados com 0 (valor padrão para int)

// Adicionando um valor em uma posição
vetor[0] = 10;
vetor[3] = 5;

//Acessando Elementos de um Array
int primeiroNumero = vetor[0]; // Acessa o primeiro elemento

// Acessando o tamanho do vetor
System.out.print(vetor.length)

// Inicialização direta:
int[] vetor = {1, 2, 3, 4, 5};

int tamanho = 5;
int[] outroVetor = new int[tamanho];
for (int i = 0; i < tamanho; i++) {
    outroVetor[i] = i + 1;
}

String[] nomes = {"Alice", "Bob", "Charlie"}; // Vetor de Strings
double[] precos = {19.99, 29.99, 39.99}; // Vetor de doubles

Modificando um elemento
precos[1] = 25.5;
```

Vetores - Percorrendo um vetor

- Precisamos de uma estrutura de repetição para **iterar** o vetor, percorrendo e indentificando elemento por elemento

```
// Criar e inicializar um vetor de inteiros
int[] vetor = {3, 7, 1, 5, 4};
// Imprimir elementos do vetor
for (int i = 0; i < vetor.length; i++) {
    System.out.print(vetor[i] + " ");
}
```

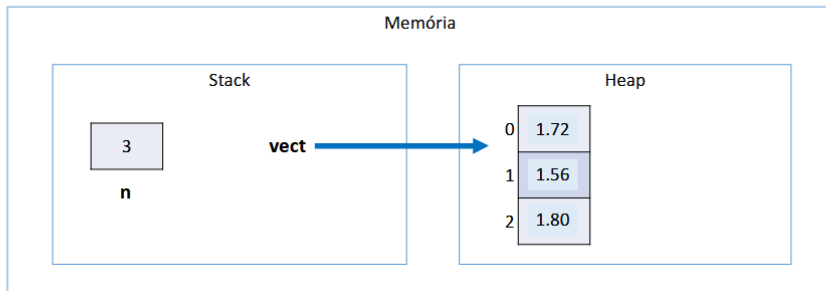
- Inserindo valores

```
// Solicitando ao usuário o tamanho do vetor
System.out.print("Digite a quantidade de nomes que deseja inserir: ");
int tamanho = sc.nextInt();
// Criando o vetor de strings
String[] nomes = new String[tamanho];
// Preenchendo o vetor com nomes fornecidos pelo usuário
for (int i = 0; i < tamanho; i++) {
    System.out.print("Digite o nome " + (i + 1) + ": ");
    nomes[i] = sc.next();
}
// Imprimindo os nomes do vetor
System.out.println("\nNomes no Vetor:");
for (int i = 0; i < tamanho; i++) {
    System.out.println("Nome " + (i + 1) + ": " + nomes[i]);
}
```

Exemplo - Vetores

Fazer um programa para ler um número inteiro N e a altura de N pessoas. Armazene as N alturas em um vetor. Em seguida, mostrar a altura média dessas pessoas.

| Input: | Output: |
|---------------------------|-----------------------|
| 3 1.72 1.56 1.80 | AVERAGE HEIGHT = 1.69 |



Vetores

Exercicio inicial

- 1 Faça um programa que leia um número inteiro positivo N e depois N números inteiros e armazene-os em um vetor. Em seguida, mostrar na tela todos os números negativos lidos.
- 2 Faça um programa que leia N números reais e armazene-os em um vetor. Em seguida: - Imprimir todos os elementos do vetor - Mostrar na tela a soma e a média dos elementos do vetor

Laço "for each"

Sintaxe opcional e simplificada para percorrer coleções

Sintaxe:

```
for (Tipo apelido : coleção) {  
    <comando 1>  
    <comando 2>  
}
```

Vetores - Usando o método split de Strings

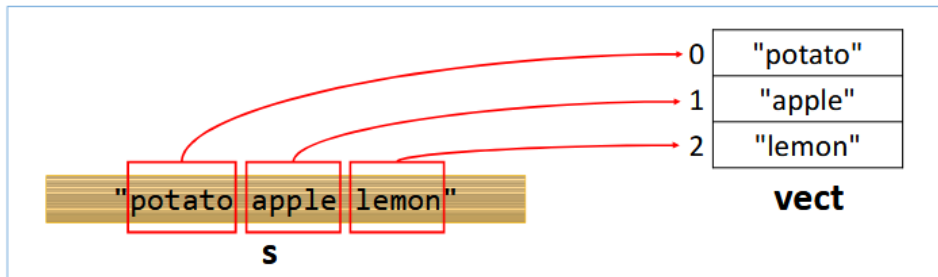
```
String s = "potato apple lemon";
```

```
String[] vect = s.split(" ");
```

```
String word1 = vect[0];
```

```
String word2 = vect[1];
```

```
String word3 = vect[2];
```



Vetores - Classe Arrays

```
// Ordena o array especificado em ordem crescente.
int[] arr = {3, 1, 4, 1, 5, 9, 2, 6, 5, 3};
Arrays.sort(arr);

//Compara dois arrays para verificar se são iguais.
int[] arr1 = {1, 2, 3};
int[] arr2 = {1, 2, 3};
boolean saoIguais = Arrays.equals(arr1, arr2);

// Preenche todo o array com o valor especificado.
int[] arr = new int[5];
Arrays.fill(arr, 42);

//Realiza uma pesquisa binária no array para encontrar a posição do elemento especificado.
int[] arr = {1, 2, 3, 4, 5};
int indice = Arrays.binarySearch(arr, 3);

// Copia a parte especificada do array original para um novo array com o comprimento especificado.
int[] arr = {1, 2, 3, 4, 5};
int[] copia = Arrays.copyOf(arr, 3);

Integer[] array = {4, 2, 7, 1, 9, 3, 6};
// Ordena em ordem decrescente usando Comparator.reverseOrder()
Arrays.sort(array, Comparator.reverseOrder());


// Exibe o array ordenado em ordem decrescente
System.out.println("Array ordenado em ordem decrescente: " + Arrays.toString(array));
```

Matrizes

Matrizes

- Em programação, "matriz" é o nome dado a arranjos bidimensionais
 - Atenção: "vetor de vetores"
- Arranjo (array) é uma estrutura de dados:
 - Homogênea (dados do mesmo tipo)
 - Ordenada (elementos acessados por meio de posições)
 - Alocada de uma vez só, em um bloco contíguo de memória
- Vantagens:
 - Acesso imediato aos elementos pela sua posição
- Desvantagens:
 - Tamanho fixo
 - Dificuldade para se realizar inserções e deleções

myMat



| | 0 | 1 | 2 | 3 |
|---|------|------|------|------|
| 0 | 3.5 | 17.0 | 12.3 | 8.2 |
| 1 | 4.1 | 6.2 | 7.5 | 2.9 |
| 2 | 11.0 | 9.5 | 14.8 | 21.7 |