

# Linguagem de Programação

## VETORES DINÂMICOS

Prof. Me. Humberto Zanetti



# Vetores dinâmicos

- A classe ArrayList
- Operações e manipulação
- Iterando vetores dinâmicos

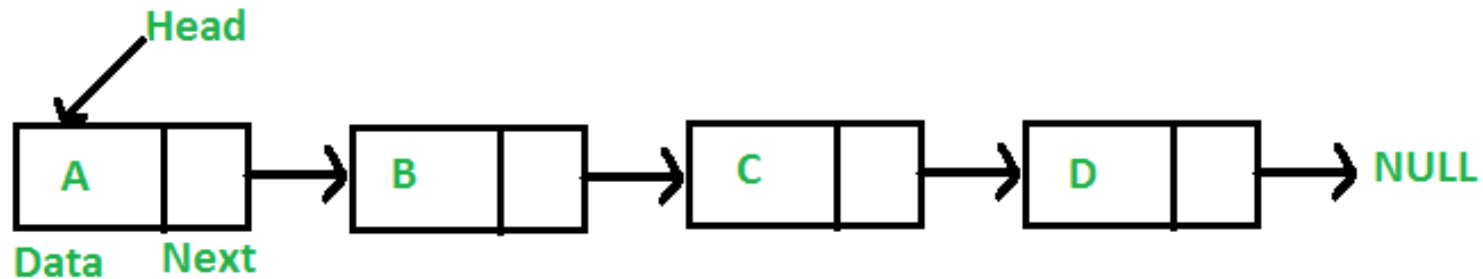
# Vetores dinâmicos

- Em muitas situações, temos manipular uma quantidade de dados a qual não sabemos a priori
- A solução mais prática seria manipular um vetor que a princípio não teria um tamanho definido.
- Para isso, em Java, temos a classe **ArrayList**!
  - Essas estruturas possuem um comportamento similar à listas.
- O primeiro passo temos que importar:

```
import java.util.ArrayList;
```

# Listas!

- Listas são estruturas de dados organizados de forma que cada elemento estão diretamente ligados e podemos organizar todos os elementos, caso seja inserido ou deletado algum elemento.



- Essa estrutura será estudado com maiores detalhes na disciplina de **Estrutura de Dados**, por isso trataremos de forma prática e superficial.

# Tipos de dados

- Para a definição de tipos de dados em vetores em lista, temos que usar as classes de tipo. Entre elas temos:

**String:** para cadeias de caracteres

**Integer:** para números inteiros

**Double:** para números reais

etc...

# Declarando um ArrayList

- Podemos criar um ArrayList de 2 maneiras, coma declaração explícita no operador “diamond” (<>)

```
ArrayList<Integer> numeros = new ArrayList<Integer>();
```

OU

```
ArrayList<Integer> numeros = new ArrayList<>();
```

# Funções de manipulação

- Função **add()**
  - Como os vetores dinâmicos podem aumentar e reduzir conforme o uso, é necessário uma função para inserir novos dados assim que necessário.

```
ArrayList<Integer> numeros = new ArrayList<Integer>();  
numeros.add(10);  
numeros.add(2);  
numeros.add(100);
```

# Funções de manipulação

- Função **get()**

- A função get serve para selecionar o elemento na posição do vetor.

```
ArrayList<Integer> numeros = new ArrayList<Integer>();  
numeros.add(10);  
numeros.add(2);  
numeros.add(100);
```

```
System.out.println(numeros.get(0));
```



# Funções de manipulação

- Função **remove()**

- Remove um elemento dentro do vetor. Uma vez que o elemento é removido, a lista toda é reorganizada.

```
ArrayList<Integer> numeros = new ArrayList<>();  
numeros.add(10);  
numeros.add(2);  
numeros.add(100);  
System.out.println("1.o elemento: "+numeros.get(0));  
numeros.remove(0);  
System.out.println("Novo 1.o elemento: "+numeros.get(0));
```

# Funções de manipulação

- Função **size()**
  - Retorna o tamanho do vetor dinâmico, ou seja, a quantidade de elementos existente no vetor.

```
ArrayList<Integer> numeros = new ArrayList<>();  
numeros.add(10);  
numeros.add(2);  
numeros.add(100);  
System.out.println("Tamanho do vetor: "+numeros.size());  
numeros.remove(0);  
System.out.println("Tamanho do vetor: "+numeros.size());
```

# Exemplo prático

- Faça um programa em que o usuário digita um número indefinido de números reais (pergunte se o usuário deseja inserir mais um número ou encerrar). Após encerrar a entrada de dados, deverá ser exibido em tela:
  - A quantidade de elementos digitados
  - A média dos valores inseridos
  - O maior valor existente dentro do vetor
  - Mostrar todos os elementos que foram inseridos

# Funções de manipulação

- Função **contains()**
  - Verifica se o valor é existente dentro

```
ArrayList<Integer> numeros = new ArrayList<>();  
numeros.add(10);  
numeros.add(2);  
numeros.add(100);
```

```
System.out.println(numeros.contains(100));//retorna true
```

# Funções de manipulação

- Função **indexOf()** e **lastIndexOf()**
  - Retornam a primeira ou a última posição que possui o elemento desejado. Caso não haja o elemento, retorna -1

```
ArrayList<String> nomes = new ArrayList<>();  
nomes.add("anakin");  
nomes.add("luke");  
nomes.add("leia");  
nomes.add("luke");
```

```
System.out.println(nomes.indexOf("luke")); //retorna 1  
System.out.println(nomes.lastIndexOf("luke")); //retorna 3  
System.out.println(nomes.indexOf("han")); //retorna -1
```

# Para saber mais...

- Há muito mais possibilidades de uso de vetores dinâmicos ou listas em Java, mas dependem diretamente de um conhecimento prévio em Orientação a Objetos.
- Para conhecer mais sobre a classe ArrayList ver:
  - <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>

# Exercício prático

- Faça um programa onde o usuário digite um número indeterminado de nomes e um telefone, como se fosse uma agenda de contatos simples. Usar 2 vetores dinâmicos diferentes para armazenar os dados (um para nome e outro para telefone). O programa deverá dar a opção de incluir, deletar ou buscar um nome na agenda. Caso busque o nome, mostrar o telefone correspondente.

# Dúvidas!?

---

[humberto.zanetti@fatec.sp.gov.br](mailto:humberto.zanetti@fatec.sp.gov.br)