



Vetores e ArrayList em Java

Prof. Esp. Pedro Miho

Tópicos Abordados em Aula

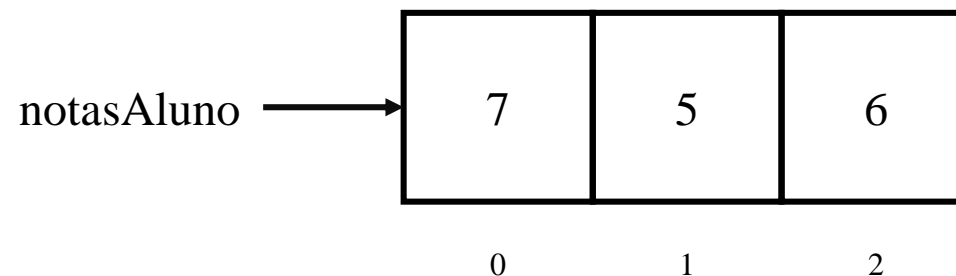
Nesta aula, você aprenderá a usar vetores e ArrayLists em Java, entendendo suas diferenças, aplicações e vantagens. Vamos explorar como declará-los, manipulá-los e aplicá-los em exemplos práticos!

- Introdução aos Vetores e ArrayList
- Vetores em Java
- ArrayList em Java
- Comparação entre Vetores e ArrayList

Introdução aos Vetores

Um vetor é uma estrutura de dados utilizada para armazenar uma coleção de elementos do mesmo tipo, onde cada elemento é identificado por um índice numérico que começa a partir do índice 0.

- **Tamanho fixo:** Deve ser definido no momento da declaração e não pode ser alterado.
- **Acesso rápido:** Permite acessar qualquer elemento diretamente pelo índice.
- **Tipo homogêneo:** Todos os elementos devem ser do mesmo tipo (int, double, String, etc.).



Sintaxe Vetores

Primeiro informamos a quantidade de itens dentro de um vetor.

```
double[] notasAluno = new double[3];
```

Diagrama de anotação para a declaração de vetor:

- `double`: Tipo de dado do vetor
- `notasAluno`: Nome Vetor
- `new double`: Tipo de dado do vetor
- `[3]`: Quantidade de itens no vetor

Atribuindo valores a esse vetor.

```
notasAluno[0] = 7;  
notasAluno[1] = 5;  
notasAluno[2] = 6;
```

Diagrama de anotação para a atribuição de valores:

- `notasAluno`: Nome Vetor
- `[0]`, `[1]`, `[2]`: Posição do item no vetor
- `7`, `5`, `6`: Valor atribuído a posição desejada

Sintaxe Vetores

Atribuindo os valores usando o for.

```
for (int i = 0; i < notasAluno.length; i++) {  
    System.out.print("Digite a " + (i + 1) + "ª nota do aluno: ");  
    notasAluno[i] = sc.nextDouble();  
}
```

Exibindo as informações salvas no vetor usando o for

```
for (int i = 0; i < notasAluno.length; i++) {  
    notasAluno[i]  
}
```

Sintaxe Vetores

Percorrendo um Vetor usando o For Each

```
for (tipoVetor variavelTemporaria : nomeVetor) {  
    System.out.println(variavelTemporaria);  
}
```

```
for (Double obj : notasAluno) {  
    System.out.println(obj);  
}
```

Introdução a Listas

As listas em Java fazem parte da Java Collections Framework e são uma forma flexível e eficiente de gerenciar dados. Ao contrário dos arrays, elas têm tamanho dinâmico, permitindo crescer ou encolher conforme necessário. A interface principal é `java.util.List`, implementada por classes como `ArrayList`, `LinkedList`, e `Vector`.

- Indexadas: Os elementos são armazenados em uma ordem específica e podem ser acessados por seus índices, como em arrays.
- Dinamismo: Diferentemente de arrays, o tamanho das listas pode ser alterado dinamicamente.
- Tipos Genéricos: Suportam o uso de tipos genéricos para evitar problemas de tipo.
- Operações Variadas: Incluem métodos úteis como adição, remoção, busca, ordenação, etc.

Tipos Primitivos e Listas em Java

Em Java, tipos primitivos (int, double, char) são valores simples armazenados diretamente na memória e não possuem métodos. Já as listas (List) da Java Collections Framework lidam apenas com **objetos** para aproveitar métodos e flexibilidade dos tipos genéricos. Por isso, tipos primitivos não podem ser usados diretamente em listas.

```
List <int> lista = new ArrayList<>();
```



```
List <Integer> lista = new ArrayList<>();
```



Boxing e Unboxing

O Boxing é o processo de converter um tipo primitivo (como int, double, char) em seu correspondente tipo objeto (como Integer, Double, Character).

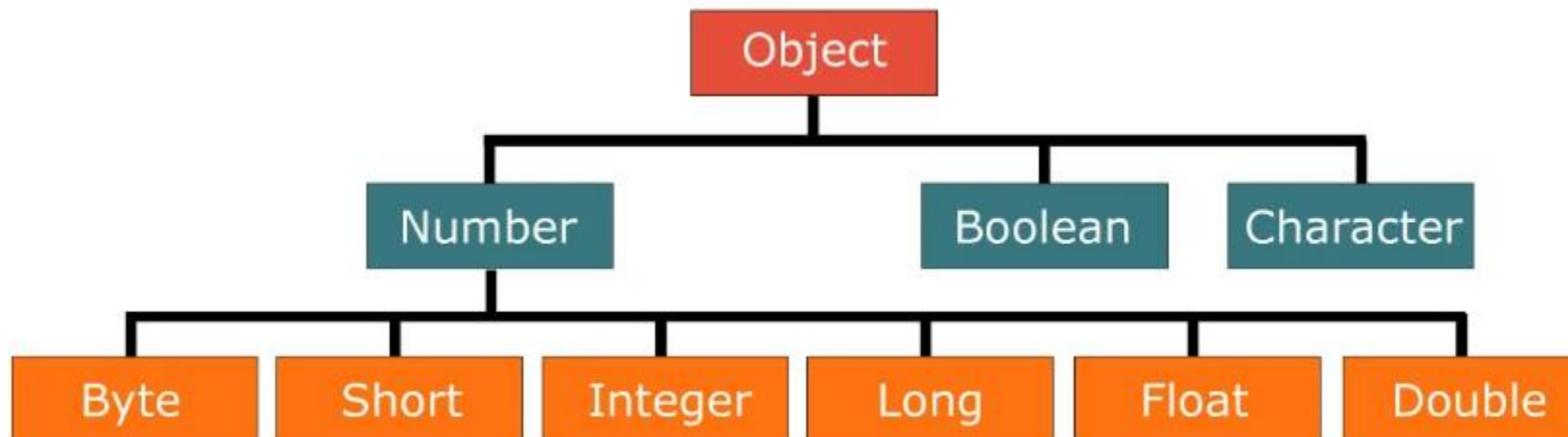
```
int numeroPrimitivo = 10;  
Integer numeroObjeto = numeroPrimitivo;
```

O Unboxing é o processo inverso do boxing, ou seja, converter um objeto wrapper de volta para o tipo primitivo correspondente.

```
Integer numeroBoxed = 10;  
int numero = numeroBoxed;
```

Wrapper Classes

As Wrapper Classes são classes que encapsulam tipos primitivos em Java, permitindo que eles sejam tratados como objetos. Isso é útil porque muitas APIs e coleções da linguagem Java, como as da Java Collections Framework, trabalham apenas com objetos, e não diretamente com tipos primitivos.



Sintaxe List em Java

Criando uma lista em Java

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

```
    List <String> lista = new ArrayList<>();
```

↓

Interface que define o comportamento de uma lista

↑

Tipo do dado da lista

↓

Aqui ocorre a instanciação do objeto da classe ArrayList

Sintaxe List em Java

Adicionando itens a Lista

```
lista.add("Pedro");  
lista.add("Livia");  
lista.add("Roberto");  
lista.add("Lucas");
```

Adicionando itens a Lista na posição desejada

```
lista.add(2, "Julia");  
lista.add(3, "Laura");
```

Sintaxe List em Java

Verificar o tamanho da lista

```
lista.size();
```

Removendo itens da lista pelo valor

```
lista.remove("Pedro");  
lista.remove("Roberto");
```

Removendo itens da lista pela posição

```
lista.remove(2);  
lista.remove(3);
```

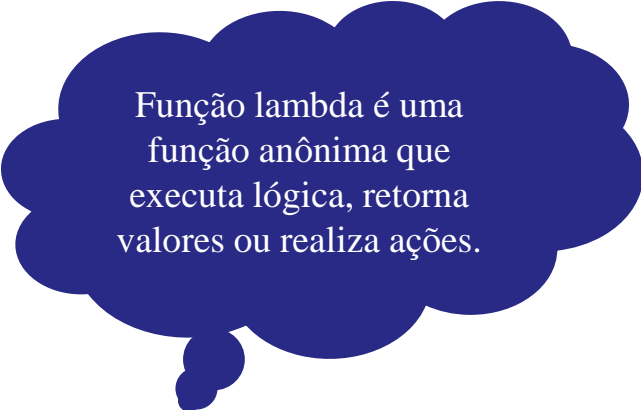
Sintaxe List em Java

Removendo itens do tipo texto itens a partir de um Predicado

```
lista.removeIf(itens -> itens.charAt(0) == 'L');
```

Removendo itens do tipo numérico itens a partir de um Predicado

```
numeros.removeIf(x -> x < 10 );
```



Função lambda é uma função anônima que executa lógica, retorna valores ou realiza ações.

Sintaxe List em Java

Descobrimo a posição de uma elemento

```
lista.indexOf("Julia");
```

Ordenando a lista em ordem crescente

```
Collections.sort(lista);
```

Desafio 01: Criação de uma Lista com Nomes e Filtragem Personalizada

Crie uma lista de nomes e filtre os que começam com uma letra específica. Depois, exiba quantos nomes foram encontrados.

Exemplo:

Nomes encontrados que começam com "A": [Ana, Alice]

Total de nomes: 2

Desafio

Desafio 02: Soma e Média de Números

Crie uma lista de números inteiros. Calcule a soma e a média de todos os números na lista.

Exemplo:

Lista de números: [10, 20, 30, 40]

Soma: 100

Média: 25.0

Desafio

Desafio 03: Verificar Elementos Duplicados

Crie uma lista de números e descubra quais números são pares. Crie uma nova lista somente com os números pares.

Exemplo:

Lista de números: [10, 20, 5, 6, 9, 5]

Números duplicados: [10, 20, 6]

Desafio 04: Remove Palavras que Contêm um Caracter Específico

Crie uma lista de palavras e remova todas que contêm uma letra específica fornecida pelo usuário.

Exemplo:

Palavras originais: [Banana, Maçã, Abacate, Manga]

Removendo palavras que contêm "a": []

Desafio 03: Palavras de Comprimento Par ou Ímpar

Solicite que o usuário insira várias palavras. Ordene-as pelo comprimento (do menor para o maior) e exiba o resultado.

Exemplo:

Palavras inseridas: [Java, Python, App, Aplicação]

Palavras com comprimento par: [Python, Aplicação]

Palavras com comprimento ímpar: [Java, App]