

Douglas Masuzzo / Derick de
Paula / Otávio Augusto /
Wallyson dos Santos / Renan
Garbo - 2º Ciclo - Análise e
Desenvolvimento de Sistemas
(Tarde

PESQUISA – ARRAY LIST

1. INTRODUÇÃO

Sob contexto da Linguagem Java, o ArrayList representa uma estrutura de dados dinâmicos que permitem o armazenamento e a manipulação de elementos com eficiência e versatilidade. Por pertencer ao pacote Java.Util, a técnica é utilizada devido à sua capacidade de redimensionamento automático e à facilidade de uso em comparação aos Arrays Convencionais.

2. CONCEITO

Diferente dos Arrays tradicionais, que possuem valores fixos, os ArrayList oferecem flexibilidade, permitindo que tais elementos sejam adicionados ou removidos de acordo com a necessidade de uma nova estrutura. Caracterizando a técnica como uma opção preferencial em relação aos números de elementos, que serão manipulados e suas variações durante a execução do programa.

Além disto, o estudo fornece diferentes métodos para aplicação de elementos, como: add(), remove(), get(), set(), size(), etc. Dentre outros recursos que interpretam ações coordenadas.

A sua sintaxe pode ser interpretada como:

```
import java.util.ArrayList; // Importando uma classe ArrayList

ArrayList<String> produtos = new ArrayList<String>(); // Criando um objeto em ArrayList
```

3. MÉTODOS

Os métodos de recursos podem ser exemplificados de acordo com a abordagem exigida:

3.1 - Add() : Para adicionar elementos em uma lista, utiliza-se do comando:

```
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        ArrayList<String> cars = new ArrayList<String>();
        cars.add("Volvo"); // Adicionando um novo elemento
        cars.add("BMW"); // Adicionando um novo elemento
        cars.add("Ford"); // Adicionando um novo elemento
        cars.add("Mazda"); // Adicionando um novo elemento
        System.out.println(cars);
    }
}
```

3.2 - Get () : Após inserir um novo elemento através do add(), é possível acessar este elemento (diretamente ao seu Index) utilizando o comando:

```
cars.get(); // Acessando um elemento ( pelo index )
```

3.3 - Remove (): Para deletar ou excluir um elemento, utiliza o comando:

```
cars.remove(); // Remove um elemento
```

3.4 - Clear (): Ao remover todos os elementos na lista, aplique o comando:

```
cars.clear(); // Limpa todos os elementos anteriores
```

3.5 - Size (): Para verificar todos os elementos da lista, utilize:

```
cars.size(); // Analisa o espaço existente dos elementos
```

4. CONCLUSÃO

Com base na análise e aplicação de prática apresentada, conclui-se que o ArrayList é uma estrutura de dados fundamental para o desenvolvimento em Java, oferecendo recursos que ampliam significativamente a flexibilidade no armazenamento de manipulação de dados. Portanto, o domínio do uso do estudo é fundamental para aprofundar os conhecimentos na linguagem JAVA e desenvolver aplicações com maior controle sobre estruturas de dados dinâmicas.

```
package Projeto;
import java.util.ArrayList;
import java.util.Scanner;

public class A1 {
    public int id;
    public String name;
    public String especify;
    public double valor;

    static ArrayList<A1> lista = new ArrayList<>();
    static Scanner input = new Scanner(System.in);

    public A1(int i, String n, String e, double v) {
        id = i;
        name = n;
        especify = e;
        valor = v;
    }

    @Override
    public String toString() {
        return "ID: " + id + "\nNome: " + name + "\nTipo: " +
        especify + "\nR$" + valor;
    }

    public static void criar() {
        System.out.print("ID: ");
        int i = input.nextInt(); input.nextLine();
        System.out.print("Nome: ");
        String n = input.nextLine();
        System.out.print("Especificação: ");
        String e = input.nextLine();
        System.out.print("Valor: ");
        double v = input.nextDouble();

        lista.add(new A1(i, n, e, v));
        System.out.println("Produto criado.");
    }
}
```

```
public static void listar() {
    if (lista.isEmpty()) System.out.println("Nenhum
produto.");
    else for (A1 p : lista) System.out.println(p);
}

public static void editar() {
    System.out.print("ID do produto a editar: ");
    int busca = input.nextInt(); input.nextLine();
    for (A1 p : lista) {
        if (p.id == busca) {
            System.out.print("Novo nome: ");
            p.name = input.nextLine();
            System.out.print("Nova especificação: ");
            p.especific = input.nextLine();
            System.out.print("Novo valor: ");
            p.valor = input.nextDouble();
            System.out.println("Produto atualizado.");
            return;
        }
    }
    System.out.println("Produto não encontrado.");
}

public static void apagar() {
    System.out.print("ID do produto a apagar: ");
    int busca = input.nextInt();
    boolean removido = lista.removeIf(p -> p.id ==
busca);
    System.out.println(removido ? "Produto removido." :
"Produto não encontrado.");
}

public static void buscar() {
    System.out.print("ID do produto: ");
    int busca = input.nextInt();
    for (A1 p : lista) {
        if (p.id == busca) {
            System.out.println(p);
            return;
        }
    }
    System.out.println("Produto não encontrado.");
}

public static void total() {
    double soma = 0;
    for (A1 p : lista) soma += p.valor;
```

```
        System.out.println("Valor total: R$" + soma);
    }
}
```

```
--- MENU ---
1. Criar produto
2. Listar produtos
3. Editar produto
4. Apagar produto
5. Buscar por ID
6. Calcular valor total
7. Sair
Escolha: 1
ID: 5510
Nome: abc
Especificação: espe
Valor: 0,00
Produto criado.

--- MENU ---
1. Criar produto
2. Listar produtos
3. Editar produto
4. Apagar produto
5. Buscar por ID
6. Calcular valor total
7. Sair
Escolha: 5
ID do produto: 5510
ID: 5510
Nome: abc
Tipo: espe
R$0.0
```

5. REFERÊNCIAS

BIBLIOGRÁFICAS

W3SHCOOLS. Java ArrayList. Disponível em:
https://www.w3schools.com/java/java_arraylist.asp. Acesso em: 18 mai. 2025.