**Aula XVI - Desenvolvimento Web III**

**Pesquisa no ArrayList**

**Estudo de Caso**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Fontes:*

[*https://blog.education-ecosystem.com/coisas-que-voce-deveria-saber-sobre-java-arraylist/*](https://blog.education-ecosystem.com/coisas-que-voce-deveria-saber-sobre-java-arraylist/)

*Livro de Lógica de Programação - Curso Técnico em Informática - QI Faculdade & Escola Técnica*

[*https://distancia.qi.edu.br/course/view.php?id=487&section=1*](https://distancia.qi.edu.br/course/view.php?id=487&section=1)

[*https://docs.oracle.com/javase/7/docs/api/*](https://docs.oracle.com/javase/7/docs/api/)

<https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Pesquisando e buscando em um ArrayList**

Quando trabalhamos com coleções de objetos é muito comum termos a necessidade de buscar algo dentro da coleção, essa busca pode ser específica (procurando um único elemento) ou abrangente (procurando um subconjunto de elementos).

**Busca Específica**

Uma busca específica é quando o critério utilizado para a busca é exclusivo para cada objeto. Exemplo: busca por RG, CPF, placa do veículo, nome completo, e assim por diante. Neste tipo de busca teremos apenas um retorno, um objeto como resultado, afinal não temos dois clientes com o mesmo RG, ou dois carros com a mesma placa, isto é algo específico de cada objeto. Nas buscas específicas o resultado será apenas um objeto. Portanto, neste tipo de busca podemos ter nenhum ou um único objeto como retorno.

**Busca Abrangente**

Uma busca abrangente é quando o critério da busca pode resultar em vários objetos. Exemplo: buscar todas as pessoas do sexo feminino, todos os alunos de lógica em EaD, os clientes aniversariantes do dia. Na busca abrangente teremos como retorno uma lista de objetos. Neste tipo de busca podemos ter nenhum, um ou vários objetos como resultado.

**Procedimento de Busca**

Para programar uma busca, precisamos de um laço de repetição para percorrer cada elemento (cadastro) da lista, analisando os dados que atendem ao critério da busca.

**Exemplos de Buscas**

**Exemplo de busca específica**

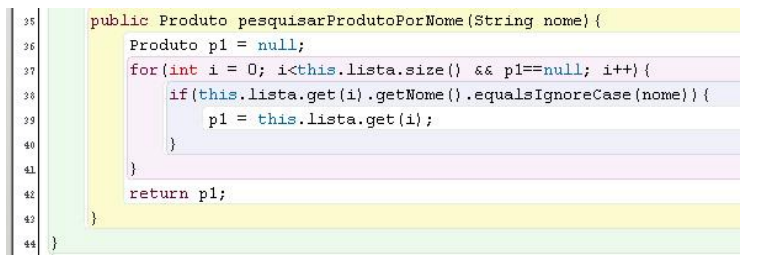
Vamos pensar na nossa lista de compras, queremos encontrar um produto específico, imagine que na lista há apenas um nome para cada produto, neste caso não teremos dois produtos iguais, certo? Portanto faremos uma busca pelo nome do produto.

No diagrama, acrescentaríamos o método da seguinte forma:

|  |
| --- |
| **+pesquisarProdutoPorNome(nome:String):Produto** |

Abaixo temos a sintaxe do método de pesquisa. Repare que na linha 37 temos um laço de repetição no qual a variável i representa o índice que começa em 0 e vai até o final da lista ou até encontrar o produto desejado.

Na linha 38 temos uma condicional comparando o nome do produto que está na lista com o nome que o usuário pediu para localizar. Observe que utilizamos o método “equalsIgnoreCase”. É um método da classe String e serve para comparar uma String com outra, sem diferenciar maiúsculas de minúsculas.



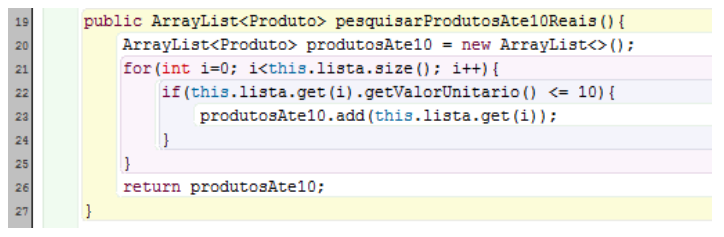
Este primeiro exemplo é uma busca específica, pois especificamos um item que será comum a apenas um objeto que está na lista, portanto o retorno será apenas um elemento ou nenhum, caso não localize o produto com aquele nome.

**Exemplo de busca abrangente**

Novamente com o nosso projeto lista de compras iremos pesquisar agora os produtos que tiverem um valor unitário de até R$ 10,00. Pense, neste caso, podemos ter nenhum, um ou vários produtos que custam até R$ 10,00.

|  |
| --- |
| **+pesquisarProdutosAte10Reais():ArrayList<Produto>** |

Neste método não precisaremos de argumentos, pois o usuário não precisará informar nenhum dado para a pesquisa.



Observe a figura acima, note que no método pesquisarProdutosAte10Reais não temos argumento e será um método retornando uma lista de produtos. Isso porque podemos ter vários produtos custando até 10 reais. Na linha 21 temos um laço de repetição no qual irá percorrer toda a lista de elemento a elemento até o final da mesma. Na linha 22 temos uma condicional para testar se e o elemento da lista possui o valorUnitario menor ou igual a 10 reais, se tiver, irá armazenar o mesmo em uma outra lista criada para guardar os elementos que possuírem valores até 10 reais, esta lista será retornada ao final do método.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Exercícios ArrayList:**

1. Criar um programa em Java que consiste em uma lista de compras:
   1. Adicionar itens na lista de compras;
   2. Consultar itens pelo seu índice;
   3. Alterar itens da Lista de compras;
   4. Exibir o tamanho da lista de compras;
   5. Remover itens da lista de compras;
   6. Classificar em ordem alfabética
   7. Limpar a lista de compras
   8. Obs: Criar uma interface que exibe o que está acontecendo com o programa
2. Criar um programa em Java que consistem em:
   1. Listar números de 1 a 50;
   2. Exibir a lista na horizontal e vertical;
   3. Exibir o tamanho da lista;
   4. Somar os valores da lista;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Exemplo Completo**

Vamos pensar em uma fila de atendimento bancário, nesta fila cada pessoa que chega específica seu **nome, CPF** e seu **interesse** **(Os interesses podem ser: 1-pagamento de conta, 2-Recebimento de salário, 3-Outros)**.

**Vamos desenvolver um sistema para esta fila de atendimento, seu sistema deve permitir adicionar pessoas na fila de atendimento, atender uma pessoa (chame pelo nome e exclua a mesma automaticamente da fila), listar todas as pessoas que estão na fila, ver a quantidade de pessoas que estão na fila, pesquisar uma pessoa pelo CPF, listar as pessoas de acordo com interesse informado no sistema (exemplo: pessoas que desejam efetuar pagamento, recebimento ou outro interesse). No final do expediente tenha uma forma de excluir todas as pessoas da fila. Temos, assim, um objeto Pessoa e um objeto FilaDeAtendimento, de modo que uma fila de atendimento contém uma coleção de objetos da classe Pessoa.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  |  |  |  |  |  |  |  |