**Aula XV - Desenvolvimento Web III**

**Exercícios de fixação de Array**

**Classe ArrayList**

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

***Fontes:***

[*https://distancia.qi.edu.br/mod/book/view.php?id=18698*](https://distancia.qi.edu.br/mod/book/view.php?id=18698)

[*https://www.devmedia.com.br/explorando-a-classe-arraylist-no-java/24298*](https://www.devmedia.com.br/explorando-a-classe-arraylist-no-java/24298)

[*https://www.w3schools.com/java/java\_arraylist.asp*](https://www.w3schools.com/java/java_arraylist.asp)

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

[*https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html*](https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html)

[*http://www.javaprogressivo.net*](http://www.javaprogressivo.net/)

[*https://www.devmedia.com.br/explorando-a-classe-arraylist-no-java/24298*](https://www.devmedia.com.br/explorando-a-classe-arraylist-no-java/24298)

[*https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html*](https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html)

[*https://www.w3schools.com/java/java\_arraylist.asp*](https://www.w3schools.com/java/java_arraylist.asp)

[*https://www.devmedia.com.br/java-collections-como-utilizar-collections/18450*](https://www.devmedia.com.br/java-collections-como-utilizar-collections/18450)

[*https://www.devmedia.com.br/visao-geral-da-interface-collection-em-java/25822*](https://www.devmedia.com.br/visao-geral-da-interface-collection-em-java/25822)

[*https://docs.oracle.com/javase/8/docs/api/java/util/Collection.html*](https://docs.oracle.com/javase/8/docs/api/java/util/Collection.html)

[*https://www.javatpoint.com/collections-in-java*](https://www.javatpoint.com/collections-in-java)

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

**Correção do exercício abaixo:**

1. Crie um programa que leia um conjunto de notas, cuja quantidade seja determinada pelo usuário. Calcule a média de todas elas. Exiba o conjunto das notas maiores do que a média calculada. Em seguida, de forma agrupada, exiba o outro conjunto de notas (menores do que a média).

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

**Exercícios de Fixação:**

1. Crie um vetor capaz de armazenar 50 números inteiros. Em seguida faça o seu preenchimento automático com os números de 101 a 150, ou seja, na posição número 0 ponha 101, na posição 1 ponha o número 102, e assim sucessivamente. Em seguida exiba os valores deste vetor.
2. Faça um programa de consulta que leia nomes de pessoas, sendo a quantidade determinada pelo usuário. Logo após a entrada pergunte ao usuário o número do nome que ele gostaria de consultar. Após sua resposta, exiba o nome e o telefone. Chame atenção do usuário em caso de uma consulta inválida, ou seja, com números menores ou iguais a zero, ou maiores do que a quantidade cadastrada.
3. Crie um programa em Java que solicite quantas pessoas deseja cadastrar, leia este conjunto de nomes de pessoas. Exiba-os em ordem alfabética e crescente.

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

**Classe ArrayList**

**Um ArrayList é uma coleção dinâmica capaz de armazenar um número indeterminado de objetos.**

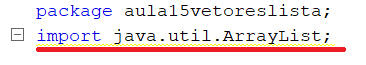
A **classe ArrayList** permite criar um objeto que é capaz de armazenar e gerenciar uma coleção de outros objetos. **Um objeto da classe ArrayList é semelhante à estrutura array, porém os arrays são estáticos, nos quais temos que determinar um número de elementos que serão armazenados e esse número de elementos permanece até o final do programa, já a coleção ArrayList é dinâmica,** **isso significa que a mesma não precisa ter um número determinado para armazenamento e sua estrutura se adapta com cada objeto inserido.**

**Vantagens em utilizar a classe ArrayList:**

Vantagens em relação à estrutura array:

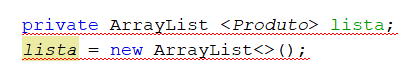
* **Podemos guardar um conjunto de dados especificando o tipo de objeto;**
* **É dinâmico: não possui tamanho definido;**
* **Já possui métodos para facilitar o gerenciamento da coleção – inserção, exclusão, classificação e assim por diante.**

**Importação da Classe:**

**Declaração de um ArrayList:**

|  |
| --- |
| **visibilidade**  **ArrayList <TipoObjeto> nomeObjeto;**  **nomeObjeto = new ArrayList<>();**  **ou visibilidade**  **ArrayList <TipoObjeto> nomeObjeto = new ArrayList<>();** |

**Exemplo:**



**Tipos de Objetos**

Podemos armazenar objetos do tipo números inteiros, números reais, textos,

valores booleanos e objetos de classes criadas no sistema. Vejamos um exemplo de cada.

|  |  |
| --- | --- |
| **Tipo de Objeto** | **Sintaxe** |
| Inteiro |  |
| String |  |
| Byte |  |
| Boolean |  |
| Classes |  |

**Dica:**

|  |
| --- |
| Obs.:  Dependendo da versão do seu JDK, pode haver necessidade de repetir o tipo de dado.  Exemplo:  private ArrayList <Funcionario>listaDeFuncionarios;  listaDeFuncionarios = new ArrayList<Funcionario>(); |

**Métodos da classe ArrayList:**

|  |  |
| --- | --- |
| **Método** | **Função** |
| **add(elemento)** | Insere um elemento no ArrayList |
| **clear()** | Limpa toda a lista do ArrayList |
| **isEmpty()** | Verifica se a lista está vazia (retorna um boolean) |
| **size()** | Retorna a quantidade do ArrayList |
| **get(índice)** | Retorna o elemento que está armazenado no índice especificado |
| **remove(índice)** | Remove o elemento contido no índice especificado |
| **remove(objeto)** | Remove o objeto especificado |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| Branco | Vermelho | Verde | Azul | Preto |

**Interface Collection:** A coleção em Java é uma estrutura que fornece uma arquitetura para armazenar e manipular o grupo de objetos.

Iterator() e sort()