Array List

Professor: Henrique Delegrego

Array List

Introdução

- Estrutura de dados semelhante a um vetor, mas com tamanho dinâmico
- Pode crescer ou diminuir conforme elementos são adicionados ou removidos
- Armazena somente objetos
- Caso precise armazenar números, use o tipo **Integer** ou **Double**
- Por questões de integridade de código, criaremos uma **List** e instanciaremos como tipo **ArrayList**

Array List

```
public static void main(String[] args) {

    List<String> listaNomes = new ArrayList<>();

    listaNomes.add("Sérgio"); // Adiciona elementos na lista
    listaNomes.add("Bruna");
    listaNomes.add("Aline");
    listaNomes.add(1, "Cláudio"); // Adiciona em um índice específico da lista
    listaNomes.remove("Sérgio"); // Remove o elemento "Sérgio"
    listaNomes.remove(1); // Remove o elemento que está no índice 1
    listaNomes.contains("Bruna"); // Retorna true se existe o elemento bruna
    System.out.println(listaNomes.get(1) + " está no índice 1");
    System.out.println("O índice de Cláudio é " + listaNomes.indexOf("Cláudio"));
    boolean b = listaNomes.get(1).equals("Bruna"); // Retorna true se o elemento no índice 1 for Bruna
    listaNomes.sort(null); // Ordena a lista em ordem crescente
    listaNomes.sort(Collections.reverseOrder()); // Ordena a lista em ordem decrescente
    listaNomes.clear(); // Apaga todos os elementos da lista
    System.out.println(listaNomes); // Imprime a lista (não precisa de for)

    for (int i = 0; i < listaNomes.size(); i++) {
        System.out.println(listaNomes.get(i)); // Imprime os elementos por índice, um por um
    }
```

Agora:

- Façam os exercícios de Array List

Array List

Em orientação a objetos

- Array Lists são comuns para representar relações entre classes
- Vamos pensar em um exemplo de um aluno:
  - A classe Aluno representa um estudante, em que cada objeto tem valores diferentes
  - A classe SalaDeAula é o coletâneo de alunos, agrupando vários alunos em uma lista
  - Essa abordagem exemplifica tanto o conceito de **associação** quanto a utilização de uma ArrayList, que facilita o gerenciamento dos objetos

Array List

Em Java

```java
public class Aluno {
    private String nome;
    private int idade;
    private double media;

    public Aluno(String nome, int idade, double media) {
        setNome(nome);
        setIdade(idade);
        setMedia(media);
    }
```

```java
public class SalaDeAula {

    // Atributos de SalaDeAula

    private List<Aluno> listaAlunos; // Lista de alunos

    // Construtor
    public SalaDeAula() {
        listaAlunos = new ArrayList<>();
    }

    // Métodos
```

- Será em SalaDeAula que os métodos que envolvem todos os Alunos serão implementados

Array List

For Iterable

- Também conhecido como **for each**, simplifica a iteração sobre coleções, eliminando a necessidade de um contador explícito
- Tem 2 seções:
  - Criação da variável
  - Qual será a lista a ser iterada
- A variável (nesse caso “a”) será o objeto atual da iteração

```
// Exibir todos os alunos da sala
public void exibirAlunos() {
    for (Aluno a : listaAlunos) {
        System.out.println(a);
    }
}
```

Agora:

- Façam os exercícios de Relacionamento de Objetos