

## Roteiro 7

Neste roteiro trabalharemos com **Vetores de Objetos** em Java. Já conhecemos vetores e eles nos permitem armazenar e manipular um conjunto de dados de um mesmo tipo. Neste caso iremos armazenar não somente dados primitivos, mas objetos. Eles são úteis quando precisamos lidar com coleções de elementos (objetos), como por exemplo, armazenar uma lista de alunos, ou uma lista de produtos, ou qualquer outro tipo de entidade.

**Os vetores podem ser :**

- **Estáticos** – Possuem um tamanho fixo e definido.
  - o **Primitivos** – int, double, float, char

```
int[] vetor1 = new int[5];
double[] vetor2 = new double[10];
```

- o **Objetos** – Objetos do próprio Java ou Objetos criados por nós.

```
String[] listaNomes = new String[15];
Loja[] listaLojas = new Loja[8];
Aluno[] listaAlunos = new Aluno[10];
```

- **Dinâmicos** – Não possuem um tamanho definido e expandimos a memória conforme a necessidade. Este tipo de estrutura de dados armazena **apenas Objetos**.
  - o **ArrayList** – Existem várias estruturas para armazenamento dinâmico de objetos em Java, mas abordaremos o mais simples que é o ArrayList;

```
ArrayList<String> listaNomes = new ArrayList<String>();
ArrayList<Loja> listaLojas = new ArrayList<Loja>();
ArrayList<Aluno> listaAlunos = new ArrayList<Aluno>();
```

### Parte 1 (roteiro7.parte1) – Vetores Estáticos de Objetos

1 – Crie o pacote **roteiro7.parte1** com as classes **Principal** e **Aluno**. **OBS.:** Iremos utilizar a mesma classe **Aluno** do roteiro4.parte4, conforme diagrama abaixo

| Aluno<br>(from parte4)  |
|---|
| -matricula: int<br>-nome: String<br>-curso: String<br>-anoIngresso: int<br>-qtdeDisciplina: int<br>-situacao: String  |
| «constructor»~Aluno(matricula: int, nome: String, curso: String, anoIngresso: int)<br>+getMatricula(): int<br>+setMatricula(matricula: int): void<br>+getNome(): String<br>+setNome(nome: String): void<br>+getAnoIngresso(): int<br>+setAnoIngresso(anoIngresso: int): void<br>+getCurso(): String<br>+setCurso(curso: String): void<br>+getQtdeDisciplina(): int<br>+setQtdeDisciplina(qtdeDisciplina: int): void<br>+getSituacao(): String<br>+setSituacao(situacao: String): void |

2 - Agora na classe principal, crie um vetor estático para armazenar os objetos da classe Aluno conforme o código abaixo. Faça os devidos testes.

```
public class Principal {

    public static void main(String[] args) {

        //Criando um vetor estático de objetos Aluno
        Aluno[] listaAlunos = new Aluno[3];

        //Instanciando os objetos do tipo Aluno
        Aluno aluno01 = new Aluno(111, "Jose", "Sistema de Informação", 2019);
        Aluno aluno02 = new Aluno(222, "Maria", "Engenharia Civil", 2020);
        Aluno aluno03 = new Aluno(333, "Carlos", "Química", 2021);

        //Armazenando os objetos no vetor de alunos
        listaAlunos[0] = aluno01;
        listaAlunos[1] = aluno02;
        listaAlunos[2] = aluno03;

        System.out.println("Listando os Alunos : ");
        System.out.println("*****");
        for (int i = 0; i < 3; i++) {
            System.out.println("Matricula : " + listaAlunos[i].getMatricula());
            System.out.println("Nome : " + listaAlunos[i].getNome());
            System.out.println("Curso : " + listaAlunos[i].getCurso());
            System.out.println("Ano Ingresso : " + listaAlunos[i].getAnoIngresso());
            System.out.println("*****");
        }

    }

}
```

3 – Modifique a classe Principal para que o input seja feito pelo usuário em um laço de repetição. **OBS.:** Utilize o mesmo vetor estático `listaAlunos`

4 – Que tipo de problemas ou limitações você percebe ao utilizar um vetor estático de objetos ?

## Parte 2 (roteiro7.parte2) – Vetores Dinâmicos de Objetos (ArrayList)

O **ArrayList** é uma classe em Java que implementa a interface **List**. Ele é uma implementação dinâmica de um **array**, o que significa que pode crescer ou encolher automaticamente à medida que novos elementos são adicionados ou removidos.

Para usar um **ArrayList**, você precisa **importar** a classe **ArrayList** e, em seguida, criar uma instância dessa classe:

```
1  import java.util.ArrayList;|
2
3  public class Principal {
4      Run | Debug
5      public static void main(String[] args) {
6          // Declaração e inicialização de um ArrayList de Strings
7          ArrayList<String> nomes = new ArrayList<>();
8      }
```

Por ser uma classe, o **ArrayList** possui métodos que facilitam a manipulação de um array dinâmico, abaixo os métodos mais utilizados.

### Adicionar Elementos:

- **.add(E elemento)**, Adiciona um elemento ao final do **ArrayList**.
- **.add(int índice, E elemento)**, insere um elemento em um índice específico no **ArrayList**.

### Acessar Elementos:

- **.get(int índice)**, retorna o elemento no índice especificado.

### Remover Elementos:

- **.remove(int índice)**, remove o elemento no índice especificado.

### Tamanho e Verificação:

- **.size()**, retorna o número de elementos no **ArrayList**.
- **.isEmpty()**, verifica se o **ArrayList** está vazio.

1 – Crie o pacote **roteiro7.parte2** com a cópia das classes **Principal** e **Aluno** implementados na parte1.

2 – Faça as adaptações necessárias para transformar o vetor de objetos estáticos em um vetor de objetos dinâmico. Utilize como exemplo o pequeno trecho de código abaixo para fazer as adaptações necessárias no seu código.

```
import java.util.ArrayList;

Aluno[] listaAlunos = new Aluno[3];
ArrayList<Aluno> listaAlunos = new ArrayList<Aluno>();

Aluno aluno01 = new Aluno(111, "Jose", "Sistema de Informação", 2019);
Aluno aluno02 = new Aluno(222, "Maria", "Engenharia Civil", 2020);
Aluno aluno03 = new Aluno(333, "Carlos", "Quimica", 2021);

listaAlunos.add(aluno01);
listaAlunos.add(aluno02);
listaAlunos.add(aluno03);
```

### Parte 3 (roteiro7.parte3) – Exercício

1 – Crie o pacote **roteiro7.parte3** com a cópia das classes **Principal** e **Aluno** implementados na parte2.

2 – Vamos evoluir o projeto. Acrescente um novo atributo na classe **Aluno** chamado **listaDisciplinas**. Este atributo deve ser um **ArrayList** de **String**.

```
import java.util.ArrayList;

public class Aluno {

    private int matricula;
    private String nome;
    private String curso;
    private int anoIngresso;
    private int qtdeDisciplina;
    private String situacao;
    private ArrayList<String> listaDisciplinas;

    Aluno(int matricula, String nome, String curso, int anoIngresso){
        this.matricula = matricula;
        this.nome = nome;
        this.curso = curso;
        this.anoIngresso = anoIngresso;
        this.listaDisciplinas = new ArrayList<>();
    }

    {Gets e Sets}
}
```

3 – Crie um método na classe **aluno** para que seja possível adicionar uma disciplina no **ArrayList** **listaDisciplinas**.

4 – Crie um método que retorne a lista de disciplinas que o aluno está matriculado.

5 – Faça as adaptações necessárias na classe **Principal** para que seja possível criar um aluno e adicionar as disciplinas deste aluno.

6 – Avalie se ainda é necessário o atributo **qtdeDisciplina**. Se não for necessário, remova o atributo com todos os ajustes necessários.

**OBS.:** Independente das mudanças feitas no item 6, apresente na classe **Principal** a quantidade de disciplinas que o aluno foi matriculado.