

Exercicios - Aula 3

Lista Exercicio 3

Exercicio 1

Classe Aluno

```
package exercices.aula3.curso;

public class Aluno {

    private String ra;
    private String nome;

    public Aluno() {}

    public Aluno(String ra, String nome) {
        this.ra = ra;
        this.nome = nome;
    }

    public String getRa() {
        return ra;
    }

    public void setRa(String ra) {
        this.ra = ra;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String imprimir() {
        return "RA: " + this.ra + "\n nome: " + this.nome + "\n";
    }
}
```

Classe Curso

```
package exercices.aula3.curso;

import exercices.aula3.curso.Aluno;
```

```
import java.util.ArrayList;
import java.util.List;

public class Curso {

    private int codigo;
    private String nome;
    private int cargaHoraria;
    private List<Aluno> alunos = new ArrayList<Aluno>();

    public Curso() {}

    public Curso(int codigo, String nome, int cargaHoraria) {
        this.codigo = codigo;
        this.nome = nome;
        this.cargaHoraria = cargaHoraria;
    }

    public int getCodigo() {
        return codigo;
    }

    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public int getCargaHoraria() {
        return cargaHoraria;
    }

    public void setCargaHoraria(int cargaHoraria) {
        this.cargaHoraria = cargaHoraria;
    }

    public void inserirAluno(Aluno aluno) {
        this.alunos.add(aluno);
    }

    public void removerAluno(String ra) {
        this.alunos.removeIf(aluno -> aluno.getRa().equals(ra));
    }

    public String imprimir() {
        return "Codigo: " + this.codigo +
            "\nNome do curso: " + this.nome +
```

```
        "\nCarga Horaria: " + this.cargaHoraria;
    }

    public String imprimirCompleto() {
        String response = "CURSO: \n" + this.imprimir();

        response += "\n ALUNOS: \n \n";

        for (Aluno aluno : this.alunos) {
            response += aluno.imprimir();
        }

        return response;
    }
}
```

Classe Principal

```
package exercices.aula3.curso;

import javax.swing.*;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
import java.util.stream.Collectors;

public class App {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        List<Curso> list = new ArrayList<Curso>();
        int op;

        op = Integer.parseInt(JOptionPane.showInputDialog("Digite uma
opcao: \n 1 - Criar Curso \n 2 - Criar Aluno \n 3 - Remover Aluno \n 4 -
Mostrar todos os alunos \n 5 - Mostrar todos os alunos \n 0 - Sair"));

        do {
            switch (op) {
                case 1:
                    list.add(criarCurso());
                    break;
                case 2:
                    list = criarAlunoAddCurso(list);
                    break;
                case 3:
                    list = removerAlunoCurso(list);
                    break;
                case 4:
                    imprimirCursos(list);
            }
        } while (op != 0);
    }
}
```

```
                break;
            case 5:
                imprimirAlunosCurso(list);
                break;
            default:
                JOptionPane.showMessageDialog(new JFrame(), "Opcao
invalida!!!", "Error",
                JOptionPane.ERROR_MESSAGE);
        }

        op = Integer.parseInt(JOptionPane.showInputDialog("Digite uma
opcao: \n 1 - Criar Curso \n 2 - Criar Aluno \n 3 - Remover Aluno \n 4 -
Mostrar todos os alunos \n 5 - Mostrar todos os alunos \n 0 - Sair"));
    } while(op != 0);

    System.out.println("Bye!!!");

}

static Curso criarCurso() {
    Curso c = new Curso();

    c.setCodigo(Integer.parseInt(JOptionPane.showInputDialog("Digite o
codigo do curso: ")));

    c.setNome(JOptionPane.showInputDialog("Digite o nome do curso: "));

    c.setCargaHoraria(Integer.parseInt(JOptionPane.showInputDialog("Digite a
carga horaria do curso: ")));

    return c;
}

static List<Curso> criarAlunoAddCurso(List<Curso> cursos) {
    Aluno aluno;
    int codigo;

    codigo = Integer.parseInt(JOptionPane.showInputDialog("Digite o
codigo do curso: "));

    aluno = criarAluno();

    List<Curso> curso = cursos.stream().filter(c -> c.getCodigo() ==
codigo).collect(Collectors.toList());

    cursos.removeIf(c -> c.getCodigo() == codigo);

    curso.get(0).inserirAluno(aluno);

    cursos.add(curso.get(0));

    return cursos;
}
```

```
}

static List<Curso> removerAlunoCurso(List<Curso> cursos) {
    int codigo;
    String ra;

    codigo = Integer.parseInt(JOptionPane.showInputDialog("Digite o
codigo do curso: "));

    ra = JOptionPane.showInputDialog("Digite o RA do aluno: ");

    List<Curso> curso = cursos.stream().filter(c -> c.getCodigo() ==
codigo).collect(Collectors.toList());

    cursos.removeIf(c -> c.getCodigo() == codigo);

    curso.get(0).removerAluno(ra);

    cursos.add(curso.get(0));

    return cursos;
}

static Aluno criarAluno() {
    Aluno a = new Aluno();

    a.setRa(JOptionPane.showInputDialog("Digite o Ra do aluno: "));

    a.setNome(JOptionPane.showInputDialog("Digite o nome do aluno: "));

    return a;
}

static void imprimirCursos(List<Curso> cursos) {
    String response = "\n";

    for(Curso curso: cursos) {
        response += curso.imprimir() + "\n";
    }

    JOptionPane.showMessageDialog(new JFrame(), response, "Info",
JOptionPane.INFORMATION_MESSAGE);
}

static void imprimirAlunosCurso(List<Curso> cursos) {
    int codigo;
    String response = "\n";

    codigo = Integer.parseInt(JOptionPane.showInputDialog("Digite o
codigo do curso: "));

    List<Curso> curso = cursos.stream().filter(c -> c.getCodigo() ==
codigo).collect(Collectors.toList());
```

```
        response += curso.get(0).imprimirCompleto();

        JOptionPane.showMessageDialog(new JFrame(), response, "Info",
            JOptionPane.INFORMATION_MESSAGE);
    }

}
```

Exercicio 2

Classe Automovel

```
package exercices.aula3.veiculo;

public class Automovel {

    private String marca;
    private String modelo;

    public Automovel() {}

    public Automovel(String marca, String modelo) {
        this.marca = marca;
        this.modelo = modelo;
    }

    public String getMarca() {
        return marca;
    }

    public void setMarca(String marca) {
        this.marca = marca;
    }

    public String getModelo() {
        return modelo;
    }

    public void setModelo(String modelo) {
        this.modelo = modelo;
    }

    public String imprimir() {
        return "\n Marca: " + this.marca + "\n" +
            "Modelo: " + this.modelo + "\n";
    }

}
```

Classe Pessoa

```
package exercices.aula3.veiculo;

import exercices.aula3.curso.Aluno;

import java.util.ArrayList;
import java.util.List;

public class Pessoa {

    private int codigo;
    private String nome;
    private List<Automovel> automoveis = new ArrayList<Automovel>();

    public Pessoa() {}

    public Pessoa(int codigo, String nome) {
        this.codigo = codigo;
        this.nome = nome;
    }

    public int getCodigo() {
        return codigo;
    }

    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public void inserirAutomovel(Automovel automovel) {
        this.automoveis.add(automovel);
    }

    public void excluirAutomovel(String modelo) {
        this.automoveis.removeIf(aut -> aut.getModelo().equals(modelo));
    }

    public List<Automovel> getAutomoveis() {
        return this.automoveis;
    }

    public String imprimir() {
        return "\nCodigo: " + this.codigo +
```

```
        "\nNome: " + this.nome;
    }

    public String imprimirCompleto() {
        String response = "\nPessoa: \n" + this.imprimir();

        response += "\n Carros: \n \n";

        for (Automovel automovel : this.automoveis) {
            response += automovel.imprimir();
        }

        return response;
    }
}
```

Classe principal

```
package exercices.aula3.veiculo;

import javax.swing.*;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
import java.util.stream.Collectors;

public class App {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        List<Pessoa> list = new ArrayList<Pessoa>();
        int op;

        op = Integer.parseInt(JOptionPane.showInputDialog("Digite uma
opcao: \n 1 - Criar Pessoa \n 2 - Criar Automovel \n 3 - Transferir
automovel \n 4 - Mostrar todas as pessoas \n 5 - Mostrar automoveis da
pessoa \n 0 - Sair"));

        do {
            switch (op) {
                case 1:
                    list.add(criarPessoa());
                    break;
                case 2:
                    list = criarAutomovelAddPessoa(list);
                    break;
                case 3:
                    list = transferirAutomovel(list);
                    break;
                case 4:
```



```
        imprimirPessoas(list);
        break;
    case 5:
        imprimirPessoaCarros(list);
        break;
    default:
        JOptionPane.showMessageDialog(new JFrame(), "Opcao
invalida!!!", "Error",
                                   JOptionPane.ERROR_MESSAGE);
    }

    op = Integer.parseInt(JOptionPane.showInputDialog("Digite uma
opcao: \n 1 - Criar Pessoa \n 2 - Criar Automovel \n 3 - Transferir
automovel \n 4 - Mostrar todos as pessoas \n 5 - Mostrar automoveis da
pessoa \n 0 - Sair"));
    } while(op != 0);

    System.out.println("Bye!!!");

}

static Pessoa criarPessoa() {
    Pessoa p = new Pessoa();

    p.setCodigo(Integer.parseInt(JOptionPane.showInputDialog("Digite o
codigo da pessoa: ")));

    p.setNome(JOptionPane.showInputDialog("Digite o nome da pessoa:
"));

    return p;
}

static List<Pessoa> criarAutomovelAddPessoa(List<Pessoa> pessoas) {
    Automovel automovel;
    int codigo;

    codigo = Integer.parseInt(JOptionPane.showInputDialog("Digite o
codigo da pessoa: "));

    automovel = criarAutomovel();

    List<Pessoa> pessoa = pessoas.stream().filter(p -> p.getCodigo() ==
codigo).collect(Collectors.toList());

    pessoas.removeIf(p -> p.getCodigo() == codigo);

    pessoa.get(0).inserirAutomovel(automovel);

    pessoas.add(pessoa.get(0));

    return pessoas;
}
```

```
static Automovel criarAutomovel() {
    Automovel a = new Automovel();

    a.setMarca(JOptionPane.showInputDialog("Digite o marca do
automovel: "));

    a.setModelo(JOptionPane.showInputDialog("Digite o modelo do
automovel: "));

    return a;
}

static List<Pessoa> transferirAutomovel(List<Pessoa> pessoas) {
    int codigoPessoaOrigem;
    int codigoPessoaDestino;
    String modelo;
    List<Pessoa> pessoaOrigem, pessoaDestino;
    List<Automovel> automovelTranferir;

    codigoPessoaOrigem =
Integer.parseInt(JOptionPane.showInputDialog("Digite o codigo da pessoa que
deseja transefeir o veiculo: "));

    modelo = JOptionPane.showInputDialog("Digite o modelo do automovel
que deseja transferir: ");

    codigoPessoaDestino =
Integer.parseInt(JOptionPane.showInputDialog("Digite o codigo da pessoa que
deseja receber o veiculo: "));

    pessoaOrigem = pessoas.stream().filter(p -> p.getCodigo() ==
codigoPessoaOrigem).collect(Collectors.toList());

    pessoaDestino = pessoas.stream().filter(p -> p.getCodigo() ==
codigoPessoaDestino).collect(Collectors.toList());

    automovelTranferir =
pessoaOrigem.get(0).getAutomoveis().stream().filter(aut ->
aut.getModelo().equals(modelo)).collect(Collectors.toList());

    pessoaOrigem.get(0).excluirAutomovel(modelo);

    pessoaDestino.get(0).inserirAutomovel(automovelTranferir.get(0));

    pessoas.removeIf(p -> p.getCodigo() == codigoPessoaOrigem);
    pessoas.removeIf(p -> p.getCodigo() == codigoPessoaDestino);

    pessoas.add(pessoaOrigem.get(0));
    pessoas.add(pessoaDestino.get(0));

    return pessoas;
}
```

```
static void imprimirPessoas(List<Pessoa> pessoas) {
    String response = "\n";

    for(Pessoa pessoa: pessoas) {
        response += pessoa.imprimir() + "\n";
    }

    JOptionPane.showMessageDialog(new JFrame(), response, "Info",
        JOptionPane.INFORMATION_MESSAGE);
}

static void imprimirPessoaCarros(List<Pessoa> pessoas) {
    int codigo;
    String response = "\n";

    codigo = Integer.parseInt(JOptionPane.showInputDialog("Digite o
codigo da pessoa: "));

    List<Pessoa> pessoa = pessoas.stream().filter(p -> p.getCodigo() ==
codigo).collect(Collectors.toList());

    response += pessoa.get(0).imprimirCompleto();

    JOptionPane.showMessageDialog(new JFrame(), response, "Info",
        JOptionPane.INFORMATION_MESSAGE);
}
}
```

Gabriel D. Padua