

Exercicios - Aula 4

Lista Exercicio 4

Exercicio 1

Classe Aluno

```
package exercices.aula4.aluno;

public class Aluno {

    private String ra;
    private String nome;
    private String curso;

    public Aluno(){}

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

    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 getCurso() {
        return curso;
    }

    public void setCurso(String curso) {
        this.curso = curso;
    }

    @Override
    public String toString() {
```

```
        return "ra: " + ra + "\n" +  
               "nome: " + nome + "\n" +  
               "curso: " + curso + "\n";  
    }  
}
```

Classe Aluno de Graduação

```
package exercices.aula4.aluno;  
  
import exercices.aula4.aluno.Aluno;  
  
public class AlunoGraduacao extends Aluno {  
  
    private String anoConclusaoEnsinoMedio;  
  
    public AlunoGraduacao() {}  
  
    public AlunoGraduacao(String ra, String nome, String curso, String  
anoConclusaoEnsinoMedio) {  
        super(ra, nome, curso);  
        this.anoConclusaoEnsinoMedio = anoConclusaoEnsinoMedio;  
    }  
  
    public String getAnoConclusaoEnsinoMedio() {  
        return anoConclusaoEnsinoMedio;  
    }  
  
    public void setAnoConclusaoEnsinoMedio(String anoConclusaoEnsinoMedio)  
{  
        this.anoConclusaoEnsinoMedio = anoConclusaoEnsinoMedio;  
    }  
  
    @Override  
    public String toString() {  
        return "RA: " + this.getRa() + "\n Nome: " + this.getNome() + "\n  
Curso: " + this.getCurso() + "\n Ano de conclusao ensino medio: " +  
anoConclusaoEnsinoMedio + "\n";  
    }  
}
```

Classe Aluno de Pós-Graduação

```
package exercices.aula4.aluno;  
  
import exercices.aula4.aluno.Aluno;
```

```
public class AlunoPosGraduacao extends Aluno {

    private String anoConclusaoGraduacao;

    public AlunoPosGraduacao() {}

    public AlunoPosGraduacao(String ra, String nome, String curso, String
anoConclusaoGraduacao) {
        super(ra, nome, curso);
        this.anoConclusaoGraduacao = anoConclusaoGraduacao;
    }

    public String getAnoConclusaoGraduacao() {
        return anoConclusaoGraduacao;
    }

    public void setAnoConclusaoGraduacao(String anoConclusaoGraduacao) {
        this.anoConclusaoGraduacao = anoConclusaoGraduacao;
    }

    @Override
    public String toString() {
        return "RA: " + this.getRa() + "\n Nome: " + this.getNome() + "\n
Curso: " + this.getCurso() + "\n Ano de conclusao da graduacao: " +
anoConclusaoGraduacao + "\n";
    }
}
```

Classe Principal

```
package exercices.aula4.aluno;

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

public class App {

    public static void main(String[] args) {

        List<Aluno> list = new ArrayList<Aluno>();
        int op;
        op = Integer.parseInt(JOptionPane.showInputDialog("Digite uma
opcao: \n 1 - Inserir aluno \n 2 - Listar alunos \n 0 - Sair"));

        while(op != 0) {
            switch (op) {
                case 1:
```

```
        list.add(inserirAluno());
        break;
    case 2:
        listarAlunos(list);
        break;
    default:
        JOptionPane.showMessageDialog(new JFrame(), "Opcao
invalida!!!", "Error",
                                JOptionPane.ERROR_MESSAGE);
        break;
    }

    op = Integer.parseInt(JOptionPane.showInputDialog("Digite uma
opcao: \n 1 - Inserir aluno pessoa aluno\n 2 - Listar alunos \n 0 -
Sair"));
}

}

public static Aluno inserirAluno() {
    int op;

    op = Integer.parseInt(JOptionPane.showInputDialog("Digite o tipo de
aluno q deseja inserir: \n 1 - Graduacao \n 2 - Pos-graduacao"));

    if(op == 1) {
        return criarAlunoGraduacao();
    }

    return criarAlunoPosGraduacao();
}

public static AlunoGraduacao criarAlunoGraduacao() {
    AlunoGraduacao aluno = new AlunoGraduacao();

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

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

    aluno.setCurso(JOptionPane.showInputDialog("Digite o Curso do
aluno"));

    aluno.setAnoConclusaoEnsinoMedio(JOptionPane.showInputDialog("Digite o ano
de conclusao do ensino medio: "));

    return aluno;
}

public static AlunoPosGraduacao criarAlunoPosGraduacao() {
    AlunoPosGraduacao aluno = new AlunoPosGraduacao();
```

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

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

        aluno.setCurso(JOptionPane.showInputDialog("Digite o Curso do aluno"));

        aluno.setAnoConclusaoGraduacao(JOptionPane.showInputDialog("Digite o ano de conclusao da Graduacao: "));

        return aluno;
    }

    public static void listarAlunos(List<Aluno> alunos) {
        String ra;

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

        List<Aluno> aluno = alunos.stream().filter(p ->
p.getRa().equals(ra)).collect(Collectors.toList());

        JOptionPane.showMessageDialog(new JFrame(),
aluno.get(0).toString(), "Info",
JOptionPane.INFORMATION_MESSAGE);
    }
}
```

Exercicio 2

- Encapsulamento

No exercício anterior foi utilizado encapsulamento, pois os atributos das classes foram definidos como privados só podendo ser acessador da própria classe ou com método publico.

```
private String ra;
private String nome;
private String curso;
```

- Herança

No exercício anterior foi usado herança para reaproveitar os atributos da classe aluno, ocorrendo uma generalização e especialização nas classes para ter um reaproveitamento de código.

```
public class AlunoGraduacao extends Aluno
```

- Polimorfismo

No exercício anterior foi usado Polimorfismo Ad Hoc de Sobrecarga, pois as classes especialistas sobrescreveram o método toString para poder retornar os dados das datas de conclusão, porem fazendo referência a classe principal.

```
@Override
public String toString() {
    return "RA: " + this.getRa() + "\n Nome: " + this.getNome() + "\n
Curso: " + this.getCurso() + "\n Ano de conclusao ensino medio: " +
anoConclusaoEnsinoMedio + "\n";
}
```

Exercicio 3

Classe Funcionario

```
package exercices.aula4.funcionario;

public class Funcionario {

    private String matricula;
    private String nome;
    private double salario;

    public Funcionario(){}

    public Funcionario(String matricula, String nome, double salario) {
        this.matricula = matricula;
        this.nome = nome;
        this.salario = salario;
    }

    public String getMatricula() {
        return matricula;
    }

    public void setMatricula(String matricula) {
        this.matricula = matricula;
    }

    public String getNome() {
        return nome;
    }

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

    public double getSalario() {
```

```
        return salario;
    }

    public void setSalario(double salario) {
        this.salario = salario;
    }

    @Override
    public String toString() {
        return "Funcionario: \n" +
            "matricula: " + matricula + "\n" +
            "nome: " + nome + "\n" +
            "salario: " + getSalario() + "\n";
    }
}
```

Classe Consultor

```
package exercices.aula4.funcionario;

public class Consultor extends Funcionario {

    @Override
    public double getSalario() {
        return super.getSalario() * 1.1;
    }

    public double getSalario(float porcentagem) {
        return super.getSalario() * porcentagem;
    }
}
```

Classe Principal

```
package exercices.aula4.funcionario;

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

public class App {

    public static void main(String[] args) {

        List<Funcionario> list = new ArrayList<Funcionario>();
        int op;
        op = Integer.parseInt(JOptionPane.showInputDialog("Digite uma
opcao: \n 1 - Inserir funcionario \n 2 - Listar funcionario \n 0 - Sair"));
    }
}
```

```
        while(op != 0) {
            switch (op) {
                case 1:
                    list.add(inserirFuncionario());
                    break;
                case 2:
                    listarFuncionarios(list);
                    break;
                default:
                    JOptionPane.showMessageDialog(new JFrame(), "Opcao
invalida!!!", "Error",
                                JOptionPane.ERROR_MESSAGE);
                    break;
            }

            op = Integer.parseInt(JOptionPane.showInputDialog("Digite uma
opcao: \n 1 - Inserir funcionario\n 2 - Listar funcionario \n 0 - Sair"));
        }

    }

    public static Funcionario inserirFuncionario() {
        int op;

        op = Integer.parseInt(JOptionPane.showInputDialog("Digite o tipo de
funcionario q deseja inserir: \n 1 - Funcionario \n 2 - Consultor"));

        if(op == 1) {
            return criarFuncionario();
        }

        return criarConsultor();
    }

    public static Funcionario criarFuncionario() {
        Funcionario funcionario = new Funcionario();

        funcionario.setMatricula(JOptionPane.showInputDialog("Digite a
matricula do funcionario "));

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

        funcionario.setSalario(Double.parseDouble(JOptionPane.showInputDialog("Digi
te o salario do funcionario: ")));

        return funcionario;
    }

    public static Consultor criarConsultor() {
        Consultor consultor = new Consultor();
```



```
        consultor.setMatricula(JOptionPane.showInputDialog("Digite a
matricula do funcionario "));

        consultor.setNome(JOptionPane.showInputDialog("Digite o nome do
funcionario: "));

consultor.setSalario(Double.parseDouble(JOptionPane.showInputDialog("Digite
o salario do funcionario: ")));

        return consultor;
    }

    public static void listarFuncionarios(List<Funcionario> funcionarios) {
        String matricula;

        matricula = JOptionPane.showInputDialog("Digite a matricula do
funcionario: ");

        List<Funcionario> funcionario = funcionarios.stream().filter(p ->
p.getMatricula().equals(matricula)).collect(Collectors.toList());

        JOptionPane.showMessageDialog(new JFrame(),
funcionario.get(0).toString(), "Info",
JOptionPane.INFORMATION_MESSAGE);
    }
}
```