



Disciplina: Programação Orientada a Objetos com Java
Professor: Niltemberg Carvalho

Trabalho Prático: Agenda de Contatos em Java com Programação Orientada a Objetos

1. Objetivo Geral

Desenvolver uma aplicação de agenda de contatos em Java, utilizando os fundamentos da Programação Orientada a Objetos: classes, atributos, métodos, construtores, herança, polimorfismo e tratamento de exceções.

2. Requisitos da Aplicação

A aplicação deve permitir:

- Adicionar contato com os campos: nome completo, telefone(s), e-mail(s), endereço
- Listar todos os contatos
- Buscar contato por nome ou e-mail
- Editar dados de um contato
- Remover contato
- Armazenar os contatos em memória com uso de ArrayList
- Aplicar tratamento de exceções

3. Conceitos de POO a serem aplicados

Classe	Atributos	Métodos esperados	Conceitos aplicados
Contato	nome, telefones, emails, endereço	get/set, adicionarTelefone(), adicionarEmail(), toString()	Classe base, atributos privados, construtores, encapsulamento
Endereco	rua, cidade, estado, cep	get/set, toString()	Classe simples, composição, encapsulamento
ContatoPessoal	herda de Contato, dataAniversario	toString() sobrescrito	Herança, sobrescrita, polimorfismo
ContatoProfissional	herda de Contato, empresa, cargo	toString() sobrescrito	Herança, sobrescrita, polimorfismo
Agenda	contatos: List<Contato>	adicionarContato(), removerContato(), buscarPorNome(), listarTodos(), editarContato()	Agregação de objetos, métodos de manipulação de lista
Main (menu)	—	main(), chamadas com Scanner e Agenda	Execução da aplicação, controle de fluxo

4. Métodos recomendados por classe

- Contato: adicionarTelefone(String), adicionarEmail(String), toString()
- Agenda: adicionarContato(Contato), removerContato(String), buscarPorNome(String), listarTodos(), editarContato(String)

5. Conceitos de POO aplicados na prática

- Classe: Contato, Endereco, Agenda
- Atributo: nome, telefones, emails etc.
- Método: adicionarContato(), removerContato(), toString()
- Construtor: inicialização de contatos
- Encapsulamento: atributos privados com getters/setters
- Herança: ContatoPessoal e ContatoProfissional herdam de Contato
- Polimorfismo: sobrescrita de toString()
- Tratamento de exceções: entradas inválidas, campos obrigatórios

6. Entrega Esperada

Arquivos a serem entregues:

- Código-fonte organizado em pacotes Java (Subir no GitHub e Enviar o Link)
- Video explicativo de no máximo 5min sobre os conceitos aplicados em java (Subir no Youtube e enviar o link)

Critérios de avaliação:

- Uso correto dos conceitos de POO – 1,5
- Organização e clareza do código – 0,5
- Funcionamento completo da aplicação – 1,0

7. Cronograma Sugerido

- Entrega final (Código + Video) – 23/06/25