

Lista 3 POO: Herança com **extends** em Java

Estas questões foram elaboradas para praticar herança, sobrescrita de métodos e aplicação de conceitos do mundo real.

Dentro do projeto, para cada questão, crie uma pasta separada, contendo o Main.java e o arquivo referente a resolução da questão, de forma que projeto seja composto por 10 pastas (uma para cada questão).

Questão 1: Sistema de Veículos

Implemente a classe **Veiculo** com os atributos **marca**, **modelo**, **ano** e **preco**, e os métodos:

- **calcularIpva()** - retorna 4% do preço para veículos comuns
- **exibirDetalhes()** - mostra todos os dados do veículo

a. Crie a classe **Moto** que herda de Veiculo e possui atributo **cilindrada**. Sobrescreva **calcularIpva()** para retornar 2% do preço.

b. Crie a classe **Caminhao** que herda de Veiculo e possui atributo **capacidadeCarga**. Sobrescreva **calcularIpva()** para retornar 1.5% do preço.

Questão 2: Sistema Bancário

Implemente a classe **ContaBancaria** com os atributos **numeroConta**, **saldo** e **titular**, e os métodos:

- **depositar(double valor)**
- **sacar(double valor)**
- **consultarSaldo()**

a. Crie a classe **ContaCorrente** que herda de ContaBancaria e possui atributo **limiteChequeEspecial**. Sobrescreva o método **sacar()** para permitir saques até o limite do cheque especial.

b. Crie a classe **ContaPoupanca** que herda de ContaBancaria e possui atributo **taxaRendimento**. Adicione o método **aplicarRendimento()**.

Questão 3: Sistema de Cursos

Implemente a classe **Curso** com os atributos **nome**, **cargaHoraria**, **preco** e **nivel** (Básico, Intermediário, Avançado), e os métodos:

- **calcularDesconto()** - retorna 10% do preço
- **exibirInformacoes()**

a. Crie a classe **CursoPresencial** que herda de Curso e possui atributo **local**. Sobrescreva **calcularDesconto()** para retornar 5% do preço.

b. Crie a classe **CursoOnline** que herda de Curso e possui atributo **plataforma**. Sobrescreva **calcularDesconto()** para retornar 15% do preço.

Questão 4: Sistema de Imóveis

Implemente a classe **Imovel** com os atributos **endereco**, **area**, **numeroQuartos** e **preco**, e os métodos:

- **calcularTaxaAdministracao()** - retorna 8% do preço
- **exibirCaracteristicas()**

a. Crie a classe **Apartamento** que herda de **Imovel** e possui atributo **andar** e **valorCondominio**. Sobrescreva **calcularTaxaAdministracao()** para retornar 6% do preço.

b. Crie a classe **Casa** que herda de **Imovel** e possui atributo **areaQuintal** e **temPiscina**. Sobrescreva **calcularTaxaAdministracao()** para retornar 10% do preço.

Questão 5: Sistema de Eletrônicos

Implemente a classe **ProdutoEletronico** com os atributos **marca**, **modelo**, **preco** e **voltagem**, e os métodos:

- **calcularGarantiaEstendida()** - retorna 15% do preço
- **exibirEspecificacoes()**

a. Crie a classe **Smartphone** que herda de **ProdutoEletronico** e possui atributos **tamanhoTela** e **capacidadeBateria**. Sobrescreva **calcularGarantiaEstendida()** para retornar 20% do preço.

b. Crie a classe **Notebook** que herda de **ProdutoEletronico** e possui atributos **processador** e **memoriaRAM**. Sobrescreva **calcularGarantiaEstendida()** para retornar 25% do preço.

Questão 6: Sistema de Entretenimento

Implemente a classe **Midia** com os atributos **titulo**, **anoLancamento**, **duracao** e **preco**, e os métodos:

- **calcularPrecoAluguel()** - retorna 10% do preço
- **exibirInfo()**

a. Crie a classe **Filme** que herda de **Midia** e possui atributos **diretor** e **genero**. Sobrescreva **calcularPrecoAluguel()** para retornar 15% do preço.

b. Crie a classe **Serie** que herda de **Midia** e possui atributos **numeroTemporadas** e **episodiosPorTemporada**. Sobrescreva **calcularPrecoAluguel()** para retornar 12% do preço.

Questão 7: Sistema de Esportes

Implemente a classe **Esporte** com os atributos **nome**, **numeroJogadores**, **duracaoPartida** e **popularidade**, e os métodos:

- **calcularCustoEquipamento()** - retorna valor base de R\$ 500,00
- **exibirRegras()**

a. Crie a classe **Futebol** que herda de **Esporte** e possui atributo **tamanhoCampo**. Sobrescreva **calcularCustoEquipamento()** para retornar R\$ 300,00.

b. Crie a classe **Basquete** que herda de **Esporte** e possui atributo **alturaCesta**. Sobrescreva **calcularCustoEquipamento()** para retornar R\$ 400,00.

Questão 8: Sistema de Restaurante

Implemente a classe **Prato** com os atributos **nome**, **preco**, **tempoPreparo** e **ingredientes** (lista), e os métodos:

- **calcularTaxaServico()** - retorna 10% do preço
- **exibirCardapio()**

a. Crie a classe **PratoVegano** que herda de Prato e possui atributo **certificacaoVegan**. Sobrescreva **calcularTaxaServico()** para retornar 5% do preço.

b. Crie a classe **PratoGourmet** que herda de Prato e possui atributo **chefResponsavel**. Sobrescreva **calcularTaxaServico()** para retornar 15% do preço.

Questão 9: Sistema de Bibliotecas

Implemente a classe **Livro** com os atributos **titulo**, **autor**, **anoPublicacao** e **preco**, e os métodos:

- **calcularMultaAtraso()** - retorna R\$ 2,00 por dia
- **exibirFichaCatalografica()**

a. Crie a classe **LivroDidatico** que herda de Livro e possui atributo **disciplina**. Sobrescreva **calcularMultaAtraso()** para retornar R\$ 1,00 por dia.

b. Crie a classe **LivroRaro** que herda de Livro e possui atributo **edicaoLimitada**. Sobrescreva **calcularMultaAtraso()** para retornar R\$ 5,00 por dia.

Questão 10: Sistema de Transporte

Implemente a classe **Transporte** com os atributos **capacidadePassageiros**, **velocidadeMaxima**, **consumo** e **preco**, e os métodos:

- **calcularCustoViagem(double distancia)** - retorna (distância/consumo) * 5.50
- **exibirDados()**

a. Crie a classe **Onibus** que herda de Transporte e possui atributo **numeroEixos**. Sobrescreva **calcularCustoViagem()** aplicando um desconto de 20%.

b. Crie a classe **Metro** que herda de Transporte e possui atributo **eletrico**. Sobrescreva **calcularCustoViagem()** aplicando um desconto de 30%.

Resolução de Questão Exemplo

Implemente a classe **Funcionario** com **nome**, **salario** e os métodos **addAumento(double valor)**, **ganhoAnual()** e **exibeDados()** - imprime os valores do funcionário.

a. crie a classe **Assistente**, que também é um funcionário, e que possui um número de matrícula (faça os métodos GET e SET). Sobrescreva o método **exibeDados()**.

b. sabendo que os **Assistentes Técnicos** possuem um bônus salarial e que os **Assistentes Administrativos** possuem um turno (dia ou noite) e um adicional noturno, crie as classes **Tecnico** e **Administrativo** e sobrescreva o método **ganhoAnual()** de ambas as classes (**Administrativo** e **Tecnico**).

```
// Classe base Funcionario
class Funcionario {
    protected String nome;
    protected double salario;

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

    public void addAumento(double valor) {
        this.salario += valor;
    }

    public double ganhoAnual() {
        return salario * 13; // 12 meses + 13º
    }

    public void exibeDados() {
        System.out.println("Nome: " + nome);
        System.out.println("Salário: R$ " + salario);
        System.out.println("Ganho Anual: R$ " + ganhoAnual());
    }

    // Getters e Setters
    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; }
}

// Classe Assistente
class Assistente extends Funcionario {
    private int numeroMatricula;

    public Assistente(String nome, double salario, int numeroMatricula) {
        super(nome, salario);
        this.numeroMatricula = numeroMatricula;
    }

    @Override
    public void exibeDados() {
        super.exibeDados();
        System.out.println("Matrícula: " + numeroMatricula);
        System.out.println("Cargo: Assistente");
    }

    // GET e SET para matrícula
    public int getNumeroMatricula() { return numeroMatricula; }
    public void setNumeroMatricula(int numeroMatricula) {
        this.numeroMatricula = numeroMatricula;
    }
}
```

```
// Classe Tecnico
class Tecnico extends Assistente {
    private double bonusSalarial;

    public Tecnico(String nome, double salario, int numeroMatricula, double
bonusSalarial) {
        super(nome, salario, numeroMatricula);
        this.bonusSalarial = bonusSalarial;
    }

    @Override
    public double ganhoAnual() {
        return (salario + bonusSalarial) * 13;
    }

    @Override
    public void exibeDados() {
        super.exibeDados();
        System.out.println("Bônus Salarial: R$ " + bonusSalarial);
        System.out.println("Ganho Anual com Bônus: R$ " + ganhoAnual());
    }
}

// Classe Administrativo
class Administrativo extends Assistente {
    private String turno; // "dia" ou "noite"
    private double adicionalNoturno;

    public Administrativo(String nome, double salario, int numeroMatricula,
String turno, double adicionalNoturno) {
        super(nome, salario, numeroMatricula);
        this.turno = turno;
        this.adicionalNoturno = adicionalNoturno;
    }

    @Override
    public double ganhoAnual() {
        double salarioBase = salario;
        if (turno.equalsIgnoreCase("noite")) {
            salarioBase += adicionalNoturno;
        }
        return salarioBase * 13;
    }

    @Override
    public void exibeDados() {
        super.exibeDados();
        System.out.println("Turno: " + turno);
        if (turno.equalsIgnoreCase("noite")) {
            System.out.println("Adicional Noturno: R$ " + adicionalNoturno);
        }
        System.out.println("Ganho Anual: R$ " + ganhoAnual());
    }
}
```

```
// Classe de teste
public class TesteFuncionarios {
    public static void main(String[] args) {
        System.out.println("=== FUNCIONÁRIO COMUM ===");
        Funcionario func1 = new Funcionario("João Silva", 3000);
        func1.exibeDados();

        System.out.println("\n=== ASSISTENTE ===");
        Assistente assistente = new Assistente("Maria Santos", 2500, 1001);
        assistente.exibeDados();

        System.out.println("\n=== TÉCNICO ===");
        Tecnico tecnico = new Tecnico("Pedro Alves", 2800, 1002, 500);
        tecnico.exibeDados();

        System.out.println("\n=== ADMINISTRATIVO - DIA ===");
        Administrativo adminDia = new Administrativo("Ana Costa", 2600, 1003,
"dia", 300);
        adminDia.exibeDados();

        System.out.println("\n=== ADMINISTRATIVO - NOITE ===");
        Administrativo adminNoite = new Administrativo("Carlos Lima", 2600, 1004,
"noite", 300);
        adminNoite.exibeDados();

        System.out.println("\n=== APLICANDO AUMENTOS ===");
        func1.addAumento(500);
        tecnico.addAumento(200);
        System.out.println("Novo salário do técnico: R$ " + tecnico.getSalario());
    }
}
```