

# PROGRAMAÇÃO MODULAR

ALUNO: GUILHERME GOMES DE BRITES

## 02 - Exercício sobre especialização e polimorfismo

QUESTÃO 01:

CLASSE IMOVEL:

```
package exercicios.exercicioPolimorfismo;

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

abstract class Imovel {
    protected double valorDeVenda;
    protected String endereco;
    protected int anoDeConstrucao;
    protected String proprietario;
    protected List<Beneficio> beneficios = new ArrayList<>();

    public Imovel(double valorDeVenda, String endereco, int
anoDeConstrucao, String proprietario) {
        this.valorDeVenda = valorDeVenda;
        this.endereco = endereco;
        this.anoDeConstrucao = anoDeConstrucao;
        this.proprietario = proprietario;
    }

    public abstract double calcularAluguel();

    public int getAnoDeConstrucao() {
        return anoDeConstrucao;
    }

    public void adicionarBeneficio(Beneficio beneficio) {
        beneficios.add(beneficio);
    }

    public String getProprietario() {
        return proprietario;
    }

    public String getEndereco() {
        return endereco;
    }
}
```

#### CLASSE BENEFICIO:

```
package exercicios.exercicioPolimorfismo;

public class Beneficio {
    private String descricao;
    private double valor;

    public Beneficio(String descricao, double valor) {
        this.descricao = descricao;
        this.valor = valor;
    }

    public double getValor() {
        return valor;
    }
}
```

#### CLASSE CASA:

```
package exercicios.exercicioPolimorfismo;

public class Casa extends Imovel {

    public Casa(double valorDeVenda, String endereco, int
anoConstrucao, String proprietario) {
        super(valorDeVenda, endereco, anoConstrucao, proprietario);
    }

    @Override
    public double calcularAluguel() {
        double aluguel = valorDeVenda * 0.005;
        int anos = 2025 - anoDeConstrucao;
        double desconto = Math.min(((double) anos / 5) * 0.10,
0.30);
        aluguel *= (1 - desconto);

        for (Beneficio beneficio : beneficios) {
            aluguel += beneficio.getValor();
        }
        return aluguel;
    }
}
```

#### CLASSE APARTAMENTO:

```
package exercicios.exercicioPolimorfismo;
```

```

public class Apartamento extends Imovel {
    private double valorCondominio;

    public Apartamento(double valorDeVenda, String endereco, int
anoConstrucao, String proprietario, double valorCondominio) {
        super(valorDeVenda, endereco, anoConstrucao, proprietario);
        this.valorCondominio = valorCondominio;
    }

    @Override
    public double calcularAluguel() {
        double aluguel = valorDeVenda * 0.004;
        int anos = 2025 - anoDeConstrucao;
        double desconto = Math.min(((double) anos / 5) * 0.05,
0.30);
        aluguel *= (1 - desconto);

        for (Beneficio beneficio : beneficios) {
            aluguel += beneficio.getValor();
        }

        aluguel += valorCondominio;
        return aluguel;
    }

    public double getValorCondominio() {
        return valorCondominio;
    }
}

```

#### CLASSE IMOBILIARIA:

```

package exercicios.exercicioPolimorfismo;

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

public class Imobiliaria {
    private List<Imovel> imoveis = new ArrayList<>();

    public void adicionarImovel(Imovel imovel) {
        imoveis.add(imovel);
    }

    public List<Imovel> buscarPorAno(int ano) {
        List<Imovel> encontrados = new ArrayList<>();
        for (Imovel imovel : imoveis) {
            if (imovel.getAnoDeConstrucao() == ano) {
                encontrados.add(imovel);
            }
        }
        return encontrados;
    }
}

```

```

    }
}
return encontrados;
}

public double calcularValorRecebido(Imovel imovel) {
    double aluguelSemCondominio = imovel instanceof Apartamento
        ? imovel.calcularAluguel() - ((Apartamento)
imovel).getValorCondominio()
        : imovel.calcularAluguel();
    return aluguelSemCondominio * 0.12;
}

public void mostrarResumoProprietario(String proprietario) {
    double valorBruto = 0;
    double valorLiquido = 0;
    for (Imovel imovel : imoveis) {
        if (imovel.getProprietario().equals(proprietario)) {
            double aluguel = imovel.calcularAluguel();
            double recebidoImobiliaria =
calcularValorRecebido(imovel);
            double aluguelSemCondominio = imovel instanceof
Apartamento
                ? aluguel - ((Apartamento)
imovel).getValorCondominio()
                : aluguel;
            valorBruto += aluguel;
            valorLiquido += aluguelSemCondominio -
recebidoImobiliaria;
        }
    }
    System.out.println("Proprietário: " + proprietario);
    System.out.println("Valor Bruto: R$ " + valorBruto);
    System.out.println("Valor Líquido: R$ " + valorLiquido);
}
}

```

CLASSE MAIN PARA TESTAR FUNCIONAMENTO:

```

package exercicios.exercicioPolimorfismo;

import java.util.List;

public class Main {
    public static void main(String[] args) {
        Imobiliaria imobiliaria = new Imobiliaria();

        // Criando Casa
    }
}

```

```

        Casa casa = new Casa(500000, "Rua A, 123", 2010, "João");
        casa.adicionarBeneficio(new Beneficio("Piscina", 500));
        casa.adicionarBeneficio(new Beneficio("Área de lazer",
300));

        // Criando Apartamento
        Apartamento ap = new Apartamento(300000, "Av B, 456", 2015,
"Maria", 400);
        ap.adicionarBeneficio(new Beneficio("Elevador", 200));

        // Adicionando imóveis à imobiliária
        imobiliaria.adicionarImovel(casa);
        imobiliaria.adicionarImovel(ap);

        // Mostrando resumo dos proprietários
        imobiliaria.mostrarResumoProprietario("João");
        imobiliaria.mostrarResumoProprietario("Maria");

        // Buscar imóveis pelo ano de construção
        List<Imovel> imoveis2015 = imobiliaria.buscarPorAno(2015);
        for (Imovel imovel : imoveis2015) {
            System.out.println("Imóvel encontrado, ano de
construção(2015): " + imovel.getEndereco() +
                ", Proprietário: " + imovel.getProprietario() +
                ", Aluguel: R$ " + imovel.calcularAluguel());
        }

        // Aqui começa a parte para testar o valor recebido pela
imobiliária
        System.out.println("\n-- Valores a receber pela
Imobiliária ---");
        for (Imovel imovel : List.of(casa, ap)) {
            double valorRecebido =
imobiliaria.calcularValorRecebido(imovel);
            System.out.println("Imóvel: " + imovel.getEndereco());
            System.out.println("Valor recebido pela imobiliária: R$
" + String.format("%.2f", valorRecebido));
            System.out.println();
        }
    }
}

```