

Alunos: Jataiza De Arruda Barboza
Aluno HUGO BENICIO CONRADO

RA:320231856
RA:324125544

Atividade Avaliativa 2 - Sistema de Hospedagem

Neste código você vai encontrar os principais conceitos da linguagem orientada objeto começamos utilizando o conceito de abstração

Quais seria esse conceito dentro do código?

No nosso código foi utilizada a abstração aplicada principalmente nas classes **Quarto**, **Cliente** e **Residencia** pois eles representam objetos do mundo real.

```
package quarto;

public abstract class Quarto {
    protected double valorDiaria;

    public Quarto(double valorDiaria) {
        this.valorDiaria = valorDiaria;
    }

    public abstract double calcularDiaria();
}
```

```
package cliente;

public class Cliente {
    private String nome;
    private String cpf;
    private String endereco;
    private String telefone;
    private String email;

    public Cliente(String nome, String cpf, String endereco, String telefone, String email) {
        this.nome = nome;
        this.cpf = cpf;
        this.endereco = endereco;
        this.telefone = telefone;
        this.email = email;
    }

    public String getNome() {
        return nome;
    }

    public String getCpf() {
        return cpf;
    }
}
```

```

public class Aluguel {
    private Cliente cliente;
    private Quarto quarto;
    private LocalDateTime dataEntrada;
    private LocalDateTime dataSaida;
    private long numeroDiarias;

    public Aluguel(Cliente cliente, Quarto quarto, LocalDateTime dataEntrada, LocalDateTime dataSaida) {
        this.cliente = cliente;
        this.quarto = quarto;
        this.dataEntrada = dataEntrada;
        this.dataSaida = dataSaida;
        this.numeroDiarias = calcularNumeroDiarias();
    }
}

```

```

public class Residencia {
    private String endereco;
    private String numero;
    private String bairro;
    private String cep;
    private String telefone;
    private String email;
    private Quarto[] quartos;

    public Residencia(String endereco, String numero, String bairro, String cep, String telefone, String email) {
        this.endereco = endereco;
        this.numero = numero;
        this.bairro = bairro;
        this.cep = cep;
        this.telefone = telefone;
        this.email = email;
        this.quartos = quartos;
    }
}

```

Encapsulamento é a prática de esconder os detalhes internos de uma classe e expor apenas os métodos necessários para interagir com o objeto.
Este conceito foi aplicado no **Atributos privados e getters/setters**

```

public String getNome() {
    return nome;
}

public String getCpf() {
    return cpf;
}

public String getEndereco() {
    return endereco;
}

```

A herança permite que uma classe herde os atributos e métodos de uma classe, você pode verificar essa herança no **Quarto Solteiro e Quarto Casal que herdam de Quarto** para especializar comportamentos e atributos.

```

public class QuartoSolteiro extends Quarto {
    private int numeroDeCamas;

    public QuartoSolteiro(double valorDiaria, int numeroDeCamas) {
        super(valorDiaria);
        this.numeroDeCamas = numeroDeCamas;
    }

    @Override
    public double calcularDiaria() {
        return valorDiaria + (numeroDeCamas - 1) * 10; // Adiciona 10 reais por cama extra
    }
}

```

No código fornecido, o polimorfismo é aplicado na classe Quarto, que tem dois tipos de quartos (solteiro e casal). Embora as classes QuartoSolteiro e QuartoCasal não estejam explicitamente criadas, poderia-se facilmente criar essas subclasses, onde ambos compartilham o método calcularValorDiaria, mas com comportamentos específicos para cada tipo de quarto.

Por exemplo, se existirem subtipos de Quarto (como QuartoSolteiro e Quarto Casal)

```
public class QuartoSolteiro extends Quarto {  
    private int numeroDeCamas;  
  
    public QuartoSolteiro(double valorDiaria, int numeroDeCamas) {  
        super(valorDiaria);  
        this.numeroDeCamas = numeroDeCamas;  
    }  
  
    @Override  
    public double calcularDiaria() {  
        return valorDiaria + (numeroDeCamas - 1) * 10; // Adiciona 10 reais por cama extra  
    }  
}
```

Composição é usada, onde uma Residência possui uma lista de objetos Quarto