Exercício 3: Sistema de Reservas de Hotel

Descrição do Estudo de Caso

Um hotel deseja implementar um sistema para gerenciar suas reservas. O sistema deve permitir ao recepcionista:

- **Registrar** novas reservas.
- **Cancelar** reservas existentes.
- **Consultar** o status de um quarto específico.

Você foi contratado para desenvolver uma aplicação em TypeScript que atenda a essas necessidades.

Tarefas

1. **Crie uma classe `Reserva`** que possua:
    - Propriedades:
        - `numeroQuarto` (number): número do quarto.
        - `nomeHospede` (string): nome do hóspede.
        - `dataEntrada` (Date): data de entrada.
        - `dataSaida` (Date): data de saída.
    - Construtor que inicializa todas as propriedades acima.
2. **Crie uma classe `Hotel`** que gerencie as reservas:
    - Propriedade privada:
        - `reservas` (array de `Reserva`): lista de reservas ativas.
    - Métodos:
        - `registrarReserva(reserva: Reserva): void` - Adiciona uma nova reserva.
        - `cancelarReserva(numeroQuarto: number): void` - Remove a reserva do quarto especificado.
        - `consultarStatusQuarto(numeroQuarto: number): string` - Retorna "Reservado" ou "Disponível" para o quarto especificado.
3. **Implemente funções** para testar o sistema:
    - **Função para registrar reservas**: Crie pelo menos duas instâncias de `Reserva` e registre-as no hotel usando o método `registrarReserva`.
    - **Função para cancelar reserva**: Cancele a reserva de um dos quartos.
    - **Função para consultar status**: Consulte o status de um quarto e imprima o resultado no console.

Requisitos Técnicos

- Utilize os conceitos de **classe** e **função** em TypeScript.
- Use modificadores de acesso apropriados (`public`, `private`) para encapsulamento.
- Tipifique todas as variáveis e retornos de funções explicitamente.
- Trate possíveis erros, como tentativas de cancelar ou consultar reservas inexistentes.

Objetivos de Aprendizado

- Praticar a definição e utilização de **classes** em TypeScript.
- Implementar e invocar **funções** para manipular objetos e classes.
- Aplicar conceitos de encapsulamento e tipagem estática.
- Simular um cenário real de gerenciamento de reservas em uma aplicação.

Entrega

- Código-fonte em TypeScript com a implementação das classes e funções solicitadas.
- Comentários no código explicando o funcionamento de cada parte.
- Um arquivo README explicando como executar o programa e os testes realizados.