## Estacionamento

- + private String id: identificador único de estacionamento
- + private String nome: Nome do estacionamento.
- + private String endereco: Endereço do estacionamento.
- + private List<Vaga> listaDeVagas: Lista de vagas disponíveis no estacionamento.
- public void registrarEntradaVeiculo(Veiculo veiculo): Registra a entrada de um veículo em uma vaga.
- public void registrarSaidaVeiculo(Veiculo veiculo): Registra a saída de um veículo e libera a vaga.

Cobrança

cobrado por minuto (R\$ 4 a cada 15 minutos.

+ private double valorMaximo: Valor máximo

tempoOcupacao): Calcula o valor da cobrança

com base no tempo de ocupação da vaga,

+ private double valorPorMinuto: Valor

ou seja, R\$ 0,267/minuto).

- public double calcularValor(int

respeitando o limite de R\$ 50.

de cobrança (R\$ 50).

## **▼** 1..\* Vaga + private String id: Identificação da vaga (e.g., Veículo "Y08"). + private boolean estaOcupada: Indica se a vaga está ocupada. + private String placa: Placa do veículo. + private Veiculo veiculoAtual: Veículo + private String modelo: Modelo do veículo. atualmente ocupando a vaga (null se a vaga + private Cliente cliente: Cliente ao qual o estiver livre). veículo está associado. - public void associarCliente(Cliente cliente): - public void ocuparVaga(Veiculo veiculo): Associa o veículo a um cliente. Marca a vaga como ocupada e associa o veículo à vaga. 1..\* - public void liberarVaga(): Libera a vaga e remove a associação com o veículo. **♥** 1 Cliente

+ private String id: Identificador único do cliente.
+ private String nome: Nome do cliente (pode

+ private List</br>
Veiculo> listaDeVeiculos: Lista de veículos cadastrados no nome do cliente.

 public void cadastrarVeiculo (Veiculo veiculo): Associa um novo veiculo ao cliente.
 public void removerVeiculo (Veiculo veiculo): Remove um veiculo cadastrado do cliente.

ser "anônimo").