

Avaliação de programação orientada a objetos em Java – Imobiliária

Projeto Final M3

Parte 1 (2 pontos)

Implemente as seguintes novas funcionalidades no sistema:

- Listagem de histórico de locações de um imóvel (escolhido pelo usuário);
- Listagem de histórico de venda de um imóvel (escolhido pelo usuário);
- Manter o histórico dos pagamentos mensais realizados em uma locação.
- Detalhar uma locação, apresentando o total pago e se existem prestações atrasadas.

Parte 2 (5 pontos)

Utilizando Tratamento de exceções, modifique o projeto da M2 para validar as seguintes regras de negócio:

1. Não é possível cadastrar 2 imóveis com o mesmo número de matrícula.
2. Não é possível colocar um mesmo imóvel para vender se o mesmo já tiver uma venda ativa.
3. Não é possível colocar um mesmo imóvel para alugar se o mesmo já tiver locação ativa.
4. Não é possível cadastrar clientes com o mesmo CPF.
5. Não é possível cancelar uma venda já finalizada.
6. Não é possível cancelar uma locação ativa (somente as que estiverem aguardando a efetivação da locação).
7. Não é possível excluir um imóvel que já esteja vinculado a uma venda ou locação.
8. Não é possível excluir um cliente que já esteja vinculado a uma venda ou locação.

Atenção!

Caso aconteça alguma dessas exceções, o sistema deverá notificar o usuário de forma amigável (um caixa de alerta, por exemplo) e não deve perder o estado atual da aplicação.

Lembre-se de definir o melhor lugar para realizar o tratamento da exceção, não se deve misturar classes de domínio com interação com o usuário (interface com o usuário).

Parte 3 (3 pontos)

Modifique o projeto substituindo a utilização de vetores por implementações da API Collections. Considere os seguintes requisitos na hora de utilizar as implementações:

1. Deve ser possível listar os imóveis ordenados por número de matrícula.
2. Deve ser possível listar as vendas de um imóvel ordenados pela Data (do mais novo para o mais antigo).
3. Sempre defina a variável de uma coleção pela sua Interface (List, Set, Map) e depois escolha a melhor implementação conforme os requisitos do sistema.

Relembrando o Cenário do Projeto

Modelagem e implementação das classes de domínio

Você precisará de uma classe para representar a imobiliária. Em uma imobiliária haverá vários imóveis, e você deve pensar em como resolver este problema no nível da programação (como armazenar vários imóveis na memória).

Os imóveis podem ser dos seguintes tipos: casa, apartamento, terreno e sala comercial. Pense nas características (atributos) que cada tipo de imóvel terá.

Deve ser possível cadastrar um novo imóvel. Deve ser possível colocar esse imóvel para venda ou locação (ou ambos). Defina as características e comportamentos necessários para a venda de um imóvel e as características e comportamentos necessários para a locação.

Todo os imóveis devem ter um método para exibir os seus detalhes. Este método retornará apenas uma String contendo os dados do imóvel. Entretanto, os detalhes de um sítio são diferentes dos detalhes de um apartamento, que podem ser diferentes de um apartamento, etc. Sendo assim, existe a necessidade de que cada imóvel possua uma implementação diferente do método que exibe os dados do imóvel. Ou seja, a classe que representa um apartamento exibe os dados de um apartamento, já a classe que representa um terreno exibe os dados de um terreno, e assim por diante. Pense em uma solução onde um método com o mesmo nome seja usado por todas as classes que representam os imóveis, mas que em cada caso (cada tipo de imóvel) este método tenha uma implementação diferente e adequada para cada situação.

A classe que representará a imobiliária deverá ser capaz de exibir os detalhes de todos os imóveis que estão cadastrados. Atenção neste ponto do exercício: a classe imobiliária não precisa saber com que tipo de imóvel ela está lidando no momento de exibir os detalhes de um imóvel, ela deve saber apenas que se trata de um imóvel. A classe que representa a imobiliária vai exibir os detalhes de todos os imóveis invocando um método que deve existir em todos os imóveis, independentemente do tipo (apartamento, casa, terreno, etc.). Cada imóvel retorna uma String contendo a sua descrição. A classe que representa a imobiliária pega esta String e exibe-a ao usuário. Veja que esta solução garante que se você criar um outro tipo de imóvel você poderá adicioná-lo na imobiliária e exibir seus detalhes sem alterar o código.

A classe que representará a imobiliária deverá ser capaz de finalizar a venda de um imóvel colocado para venda, informando a data da venda, o comprador e o valor pago. A classe que representará a imobiliária deverá ser capaz de iniciar a locação de um imóvel colocado para locação, informando o inquilino, o período de locação e o valor mensal.

Não é possível colocar um mesmo imóvel para vender se ele já estiver associado a outra venda em aberto e não é possível colocar um imóvel para alugar se o mesmo já pertence a uma locação ativa.