

Atividade - Imobiliária

Assuntos abordados:

- Herança
- Classes / métodos abstratos
- Exception
- Ordenação

Com base no cenário a seguir, faça o que é pedido:

Imobiliária deseja informatizar o cadastro dos seus imóveis.

Os imóveis são caracterizados por: código (número inteiro), endereço (texto), bairro (texto) e valor (número fracionário). A imobiliária trabalha com apenas dois tipos de imóvel: apartamentos e casas.

Sobre os apartamentos é necessário saber também a área de fração ideal (número fracionário) e a área privativa (número fracionário), e sobre as casas a área de terreno (número fracionário) e a área construída (número fracionário);

É importante observar que o valor do imóvel deve ser sempre maior que zero.

1. [0,5] Crie uma exceção para tratar o caso de fornecimento de valor do imóvel menor ou igual a zero no cadastro de imóveis. A correta seleção da superclasse desta exceção (Exception, RuntimeException e Error) faz parte da avaliação. Fica como sugestão para o nome da exceção: ValorImovelInvalidoException.
2. [4,0] Defina uma hierarquia e crie as classes necessárias ao cadastro de imóveis, observando as seguintes recomendações:
 - 2.1. Lembre-se de que apenas apartamentos e casas podem ser cadastrados, ou seja, não deve ser possível instanciar um imóvel que não seja de um desses tipos;
 - 2.2. As classes devem ter construtores parametrizados com todos os seus atributos para facilitar a instanciação das mesmas;
 - 2.3. As classes devem ter métodos getters e setters para viabilizar alteração em seus atributos;
 - 2.4. As classes devem ter implementação do método toString para facilitar a listagem de dados;

- 2.5. O(s) método(s) que definir(em) o atributo valor do imóvel deve fazer o adequado tratamento de que o valor do imóvel deve ser maior que zero, utilizando adequadamente a exceção `ValorImovelInvalidoException`;
- 2.6. Tanto casas quanto apartamentos devem ter um método para cálculo do valor do imposto, com as seguintes características:
- Apartamentos: $130 * \text{área privativa} + 40 * \text{área de fração ideal}$;
 - Casas: $220 * \text{área construída} + 5 * \text{área de terreno}$;
3. [1,0] Crie uma classe `ImovelDAO`, com os métodos estáticos necessários ao atendimento das demandas da camada de apresentação (TUI). Você deve ter uma única lista, que deverá armazenar tanto casas quanto apartamentos.
4. Crie uma classe `ImovelTUI`:
- 4.1. Métodos estáticos:
- a) [0,5] `cadastrarCasa`: este método deve obter os dados necessário para instanciar uma Casa e utilizar o método adequado do `ImovelDAO` para persistir esta instância;
 - b) [0,5] `cadastrarApartamento`: este método deve obter os dados necessário para instanciar um Apartamento e utilizar o método adequado do `ImovelDAO` para persistir esta instância;
 - c) [1,5] `exibirImoveisPorCodigo`: este método deve listar os imóveis por ordem crescente de código. Esta deve ser a ordenação padrão para imóveis (a implementação do critério de ordenação deve ser feita na classe de Imovel). Sua listagem deve apresentar: código, endereço e bairro. A ordenação deve ocorrer no DAO;
 - d) [1,5] `exibirImoveisPorBairroValor`: este método deve listar os imóveis, por ordem crescente de Bairro e por ordem crescente de valor dentro de cada Bairro. Sua listagem deve apresentar: código, endereço, valor do imóvel e valor do imposto. A ordenação deve ocorrer no DAO;
 - e) [0,5] `executarMenu`: um menu para acesso às 4 funcionalidades solicitadas.

Atenção:

1. Nessa implementação, a validação (valor do imóvel) deve ficar na classe de domínio e a classe TUI deve acessar diretamente a classe de DAO, NÃO devendo ser implementada uma classe de negócio.

Arquitetura definida para a atividade:

