Trabalho de Programação Orientada a Objeto

1. Criar uma classe abstrata denominada de Pessoa, com os seguintes atributos e métodos:
   Atributos: String nome; String cpf e String endereço
   Métodos: Construtor, get/set e void mostraDados() para mostrar os atributos da classe

2. Criar uma classe Locatario como uma subclasse da classe Pessoa.

3. Criar uma classe Corretor como uma subclasse da classe Pessoa. A classe possui os seguintes atributos e métodos:
   Atributo: double valor comissãoRecebida.
   Método: Construtor, get/set para o atributo.

4. Criar uma classe Proprietario que herda da classe Pessoa. A classe Proprietario possui os seguintes atributos e métodos:
   Atributos: um vetor do tipo Imovel denominado de imóveis. Declarar o vetor com 30 posições.
   Métodos:
   . void incluiImovel (Imovel imo) – adicionar o parâmetro "imo" em uma posição livre do vetor. Não é permitido ter no vetor mais de um imóvel com o mesmo idImovel.
   . void excluiImóvel (int id) – exclui do vetor, o imóvel com o atributo idImovel igual ao parâmetro id.
   .void listarImovei( ) – mostra todos os imóveis cadastrados no vetor
   Imovel buscaImovel(int id) – este método percorre o vetor imóveis até encontrar um imóvel que tenha o atributo idImovel igual ao parâmetro id. Se encontrar o método deve retornar o imóvel, Se não encontrar, retorna null.
   .void mostraDados() – mostrar todos os atributos da classe.

5. Criar uma classe denominada de ParticipacaoProprietario com os seguintes atributos e métodos:
   Atributos:
   Proprietario proprietario;
   double percentual
   Métodos: Construtor, get/set e void mostraDados() para mostrar os atributos da classe

6. Criar uma classe abstrata denominada de Imovel com os seguintes atributos e métodos:

Atributos: int idImovel, String endereco; double valorBaseLocacao, double valorBaseVenda, double iptu;
Um vetor do tipo PartcipacaoProprietario. Declarar o vetor com 10 posições.
Métodos:
Construtor, get/set para os atributos
. void incluiParticipacaoProprietario (Proprietario prop, double percentual) – instancia um objeto do tipo ParticipacaoProprietario e o adiciona em uma posição livre do vetor.
. void excluiParticipacaoProprietario (Proprietario prop) – exclui do vetor, a participação do proprietário do imóvel.
.void listarParticipacaoProprietario ( ) – mostra todos os proprietários que possuem participação
.void mostraDados() – mostrar todos os atributos da classe.
. double calcularTaxaAdministracao() – método abstrato

7. Criar a classe ImovelResidencial como subclasse da classe Imovel. Implementar o método calcularTaxaAdministracao(), que retorna 5% do valor do atributo valorBaseLocacao.
8. Criar a classe ImovelComercial como subclasse da classe Imovel. Implementar o método calcularTaxaAdministracao(), que retorna 6% do valor do atributo valorBaseLocacao.
9. Criar a classe ImovelMisto como subclasse da classe Imovel. Implementar o método calcularTaxaAdministracao(), que retorna 7% do valor do atributo valorBaseLocacao.
10. Criar a classe ImovelIndustrial como subclasse da classe Imovel. Implementar o método calcularTaxaAdministracao(), que retorna 8% do valor do atributo valorBaseLocacao.

11. Criar a classe abstrata Contrato com os seguintes atributos e métodos:
    Atributos: Imovel imovel; String dataInicio; double valorPrincipal.
    Métodos:
    construtor, get/set para os atributos, void mostraDados() para mostrar os atributos da classe
    void processarMensalidade() – método abstrato

12. Criar uma classe ContratoLocacao como uma subclasse da classe Contrato. A classe possui os seguintes atributos e métodos:

    Atributos: Locatario locatario; boolean encerrado; DevedorServico devserv, double multaAtraso; double multaRescisao; boolean atraso; boolean vistoriaAprovada;

    Métodos:
    construtor, get/set para os atributos.
    void registrarAtraso() – este método atribui o valor true para atraso e invoca o método adicionar devedor do objeto DevedorService passando como parâmetro o locatário e o valorPrincipal acrescentado da multa por atraso.
    void encerrarComVistoria(boolean vistoriaAprovada) – atribui o parâmetro "vistoriaAprovada" para o atributo vistoriaAprovada e "true" para encerrado. Se o valor da "vistoriaAprovada" for igual a true, mostrar a mensagem "Vistoria aprovada. Contrato encerrado", caso contrário, mostrar a mensagem "Vistoria reprovada! Inquilino deve realizar os reparos."
    void processarMensalidadel() – se o valor do atributo encerrado for igual a true, o método termina. O valor do aluguel é calculado, O valor do aluguel é a soma do valor principal com o valor do iptu. O método de obter a taxa de administração e decrementar a taxa do valor principal. O método também deve mostrar o valor de participação no aluguel de cada proprietário.

13. Criar uma classe ContratoVenda como uma subclasse da classe Contrato. A classe possui os seguintes atributos e métodos:

    Atributos: Pessoa comprador; Corretor corretor; double percentualImobiliaria; double percentualCorretor;
    Métodos:
    construtor, get/set para os atributos
    void processarMensalidadel() – o método obtém o valor da comissão do corretor e da imobiliária. O método calcula o valor líquido da venda (valor principal – valor comissão corretor – valor comissão imobiliária. O método adiciona a comissão para o corretor. O sistema calcula o valor de participação na venda de cada proprietário.

14. Criar a classe Devedor com os seguinte atributos e métodos:
    Atributos:
    Locatario locatário; double debito
    Métodos: Construtor, get/set e void mostraDados() para mostrar os atributos da classe

15. Criar a classe DevedorServico com os seguinte atributos e métodos:
    Atributos: vetor do tipo Devedor
    Metodos: incluir itens no vetor, excluir itens do vetor, mostrar o devedores, retornar um devedor específico.

16. Criar a classe Imobiliaria com os seguintes atributos e métodos:
    Atributos: String nome; vetor do tipo Pessoa, vetor do tipo Imovel, vetor do tipo Contrato
    Métodos: incluir itens nos vetores; retirar itens dos vetores, buscar objetos específicos nos vetores, mostra valores dos vetores.
    void processarMensalidades() – o método deve calcular a mensalidade (locação e venda) de cada contrato cadastrado.