Leilão – Parte 1

Conceitos abordados: classes, objetos, método main, sobrescrita de métodos, método construtor e método comum.

Trata-se de um sistema responsável por gerenciar as principais atividades em um leilão. Os lotes a serem leiloados são disponibilizados para visita dos clientes. Após as visitas, os clientes farão lances para tentar comprar o lote de interesse, vence aquele com maior lance. Os detalhes da implementação serão informados a seguir.

1. Crie uma classe para representar um cliente da empresa de leilão. É importante que empresa saiba o nome, endereço completo, telefone e e-mail.

2. Crie uma classe para representar o lote que será leiloado. Um lote é composto por automóveis que possuem as seguintes características: valor inicial (é o valor do lote ao iniciar o leilão), incremento mínimo (é o valor mínimo de incremento entre os lances a serem ofertados), descrição resumida do lote, situação do lote (arrematado ou não), quilometragem do carro, marca, modelo e ano de fabricação.

3. No momento da instanciação de objetos das classes acima, todos os seus respectivos atributos devem ser inicializados obrigatoriamente.

4. Codificar um método capaz de fornecer uma descrição de uma determinada instância das classes do sistema. O método deve possuir a seguinte assinatura:

   ```
   public String toString();
   ```

   O método deve retornar uma string em um formato adequado de acordo com a classe a que o objeto pertence. Veja alguns exemplos abaixo.

   Sr(a) Fulano da Silva Teixeira.
   (fulanosilva@email.com)

5. Faça os devidos testes.

6. **APRESENTAÇÃO DO SISTEMA PARA AVALIAÇÃO.**

Leilão – Parte 2

Conceito abordado: vetor, composição, modificador static.

1. Crie uma classe para representar os lances que os clientes ofertarão por um determinado lote. É importante as seguintes informações referente a um lance: cliente, lote e o valor do lance.

2. Adicione um atributo na classe Cliente capaz de armazenar seu código identificador. Deve ser gerado automaticamente, único, sequencial e iniciando em 1. Para o 1º cliente cadastrado use C1, para o 2º use C2 e assim sucessivamente.

3. Adicione um atributo na classe Lote capaz de armazenar seu código identificador. Deve ser gerado automaticamente, único, sequencial e iniciando em 1. Para o 1º lote cadastrado use L1, para o 2º use L2 e assim sucessivamente.

4. Adicione um atributo na classe Lance capaz de armazenar seu código identificador. Deve ser formado pelo código identificador do lote, seguido de um hífen, seguido de um numero inteiro gerado automaticamente, único, sequencial e iniciando em 1.

5. Crie uma classe para representar um pregão. Um pregão reúne vários lotes a serem ofertados em uma determinada data. Na data determinada do pregão, os clientes oferecem lances para um determinado lote. Vence o cliente com maior lance.

   É importante que essa classe armazene a data do pregão, a situação do pregão (se está aberto, pode receber lances, ou fechado) e uma lista de lances que serão ofertados pelos clientes.

   Sugiro usar um vetor para armazenar a lista de lances. No momento da instanciação de objetos dessa classe dimensione a quantidade de elementos do vetor de lances.

   Disponibilize nesta classe um método responsável por emitir um relatório que contemple as seguintes informações: Dados do pregão, lista dos respectivos lances e para cada lance seus respectivos dados.

   Observação: use for-each para percorre o vetor.

6. Faça as devidos testes.

7. **APRESENTAÇÃO DO SISTEMA PARA AVALIAÇÃO.**

Leilão – Parte 3

Conceito abordado: Coleções (ArrayList), encapsulamento.

1. Apesar de nosso sistema já ter melhorado bastante ainda podemos avançar um pouco mais. Usaremos ArrayList em substituição ao vetor usado na classe que representa um pregão.

   Essa alteração proporcionará muitas facilidades no manuseio (percurso, inserção, remoção e ordenação) da lista de lances. Se continuássemos usando vetor teríamos muito trabalho com a implementação destas operações.

   Use um mecanismo para garantir que a lista de lances seja composta **exclusivamente** de objetos da classe que representa o lance.

2. Codificar métodos nessa classe (pregão) que permitam registrar e remover lances.

3. Encapsule todas as classes do sistema.

4. Implemente um mecanismo que evite a quebra de encapsulamento nas classes do sistema.

5. Faça os testes necessários.

6. **APRESENTAÇÃO DO SISTEMA PARA AVALIAÇÃO.**

Leilão – Parte 4

Conceito abordado: Herança, polimorfismo, dynamic binding, static binding

1. Atualmente o sistema só reconhece um tipo de lote: automóvel. Faça as devidas alterações de forma a diferenciar dois tipos de lotes: automóvel e imóvel. O sistema deve se preparar para incorporar, sem dificuldades, novos tipos de lotes no futuro como por exemplo equipamentos, material de construção, dentre outros.

   Os imóveis apresentam algumas particularidades em relação aos automóveis, que são o endereço e valor de avaliação feita por um corretor contratado pela empresa de leilão.

2. Perceba que já existem dados referentes a endereço na classe Cliente. Elabore uma solução de forma a evitar duplicidade em relação ao endereço do imóvel.

3. Apresente o diagrama de classe ao professor antes de continuar. Use lápis e papel!

4. Agora chegou a hora de codificar as propostas apresentadas.

5. Crie uma classe de nome IdentificadorDeLotes que seja capaz de identificar os diferentes tipos de lotes. Por exemplo: para um determinado lote a funcionalidade proposta deve identificá-lo como “Lote é composto apenas de automóveis” ou “Lote é composto apenas de imóveis”, dentre outros tipos de lotes que podem ser incorporados ao sistema no futuro. Essa classe deve disponibilizar suas funcionalidades sem a necessidade de instanciar objetos.

6. Faça os testes necessários.

7. **APRESENTAÇÃO DO SISTEMA PARA AVALIAÇÃO.**

Leilão – Parte 5

Conceito abordado: Exceções, ordenações de listas, interface e interface Comparator.

1. Vamos incorporar algumas restrições ao sistema. Elabore uma solução que permita a ocorrência de uma exceção para o caso de se registrar lances em um pregão fechado.

2. Disponibilizar uma listagem de lances em ordem crescente por valor do lance.

3. Disponibilizar uma listagem de lances em ordem decrescente por valor do lance.

4. Disponibilizar uma listagem de lances em ordem alfabética por nome do cliente.

5. Crie uma classe para representar um leilão. Ela dever ser responsável pelas seguintes funcionalidades:

   a) Abrir e fechar o pregão.

   b) No caso do pregão aberto, obter os lances de um cliente para um lote enquanto o intervalo entre lances for inferior a 10 segundos. Para cada lance realizado a contagem de tempo deve ser reiniciada. Caso o intervalo entre lances for superior a 10 segundos o pregão deve ser fechado e status do lote atualizado (arrematado ou não). Ao fim dos 10 segundos se houve lance anterior o lote deve ser considerado arrematado, caso contrário, considere o lote NÃO arrematado.

   c) Listar todos os lances de um pregão, agrupado por lote. Não se esqueça de informar o status de cada lote

6. Faça os testes necessários.

7. **APRESENTAÇÃO DO SISTEMA PARA AVALIAÇÃO.**

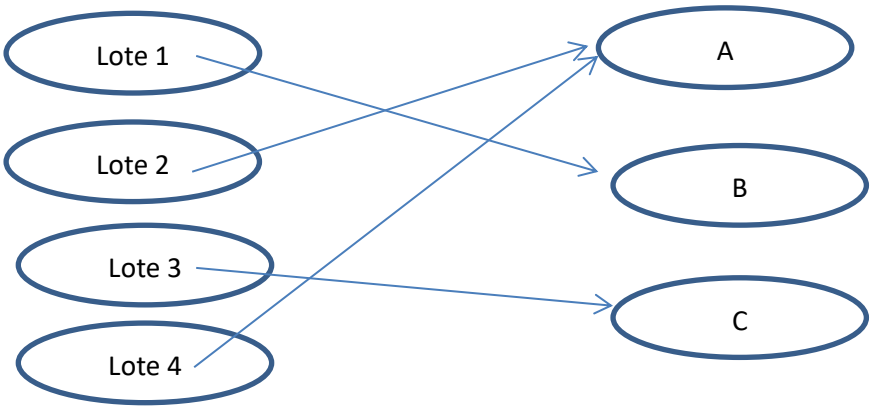
Leilão – Parte 6

Conceito abordado: HashMap.

1. Elabore uma estrutura para classificar os lotes em categorias. Trata-se de um ranking, onde cada lote é único e estará vinculado ao valor do lance dado em seu arremate. O ranking é acumulativo e só poderá fazer parte lotes com no mínimo um lance.

   Essa estrutura deve associar cada lote a uma categoria da seguinte forma:
   - Categoria A para lotes com lance acima de R$100.000,00.
   - Categoria B para lotes com lance acima de R$ 50.000,00 e abaixo de R$100.000,00.
   - Categoria C para lotes com lance abaixo de R$ 50.000,00.

   Exemplo:



   Operações a serem realizadas com o ranking de clientes:

   a) Listar todos os lotes e suas respectivas categorias.
   b) Consultar a categoria de um lote.

   Crie um mecanismo capaz de atualizar essa estrutura sempre que desejado.

   Dica: pesquise sobre HashMap. Assista ao vídeo disponibilizado que trata de um exemplo de uso de HashMap.

1. Faça os testes necessários.

2. **APRESENTAÇÃO DO SISTEMA PARA AVALIAÇÃO.**

Leilão – Parte 7

Conceito abordado: Generics.

1. Existe um atributo, no sistema, que armazena uma lista de lances, certo? Trata-se de um ArrayList configurado para aceitar apenas objetos da classe Lance. Você codificou diversos métodos (listar, adicionar, remover, etc) para manipular essa lista, está lembrado?

   Agora imagine se precisássemos gerenciar outras listas em nosso sistema, além dessa, por exemplo: lista de clientes, dentre outras. Haverá duplicidade em nosso código!

   Proponha e implemente uma solução capaz de generalizar o tratamento a qualquer tipo de lista, seja ela de lances, clientes, ou outra qualquer que venha a ser implementada no futuro.

2. Pesquise sobre o recurso do Java chamado Generics. Assista ao vídeo disponibilizado que trata de um exemplo de uso de Generics.

3. Faça os testes necessários.

4. **APRESENTAÇÃO DO SISTEMA PARA AVALIAÇÃO.**