Hotel – Parte 1

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

Trata-se de um sistema de acompanhamento dos hóspedes de um hotel durante a estadia.

1. Crie uma classe de nome Funcionario para representar o responsável pela estadia e acompanhamento ao hóspede. Defina os seguintes atributos:
   - Nome.
   - E-mail.
   - Salário

2. Crie uma classe de nome Hospede. Defina os seguintes atributos:
   - Nome.
   - E-mail.
   - Data de nascimento.

3. Crie uma classe de nome Aposento. Defina os seguintes atributos:
   - Descrição.
   - Número.
   - Valor da diária.

4. Crie uma classe de nome Consumo para representar um consumo efetuado pelo hóspede. Defina os seguintes atributos:
   - Descrição.
   - Valor.

5. No momento da instanciação de objetos das classes criadas até o momento, todos os seus respectivos atributos devem ser inicializados obrigatoriamente.

6. Codificar um método que seja capaz de fornecer uma descrição de uma determinada instância para todas as classes criadas anteriormente. 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.

   Exemplo para a classe Hospede:
   Fulano de Tal da Silva – email: fulano@email.com

7. Faça os devidos testes.

8. **APRESENTAÇÃO DO SISTEMA PARA AVALIAÇÃO.**

Hotel – Parte 2

Conceito abordado: vetor, composição, modificador static.

1. Crie uma classe de nome Hospedagem para representar o registro de uma hospedagem no hotel. Defina os seguintes atributos:
   - Funcionário responsável pela estadia do hóspede.
   - Hóspede.
   - Aposento.
   - Data da entrada.
   - Data da saída.
   - Lista do consumo do hóspede. Esse atributo deve ser um vetor de objetos da classe Consumo. Dimensione a quantidade de elementos no momento da instanciação.

2. Para a classe Hospedagem implemente um construtor para inicializar todos os seus atributos e sobrescreva o método toString.

3. Adicione um atributo na classe Hospedagem capaz de armazenar um número identificador. Deve ser um número inteiro gerado automaticamente pelo sistema, único, sequencial e iniciando em 1. Quando houver mudança de ano este número sequencial deve ser reiniciado em 1, exemplo: 2019-1, 2019-2, 2019-3,... , 2020-1, 2020-2, ...

4. Adicione um atributo na classe Consumo capaz de armazenar o número identificador. Seu conteúdo deve ser gerado automaticamente, sugestão: CONS1,CONS2, CONS3, ...

5. Adicione um atributo na classe Aposento capaz de armazenar o número identificador. Seu conteúdo deve ser gerado automaticamente, sugestão: AP1, AP2, AP3, ...

6. Disponibilize na classe Hospedagem um método responsável por retornar, para um determinado hóspede a valor da conta, levando em consideração todo o seu consumo e as diárias utilizadas. Use for-each para percorrer o vetor.

7. Faça as devidos testes.

8. **APRESENTAÇÃO DO SISTEMA PARA AVALIAÇÃO.**

Hotel – 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 no atributo lista de consumo da classe Hospedagem.

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

   Use um mecanismo para garantir que a lista de consumo seja composta **exclusivamente** de objetos da classe Consumo.

2. Codificar métodos na classe Hospedagem que permitam adicionar, remover e verificar a existência de determinado consumo na lista de consumo.

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.**

Hotel – Parte 4

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

1. Você notou a existência de duplicidade de código nas classes Funcionario e Hospede? Implemente uma solução para esse problema. Continue a leitura deste documento e apresente o diagrama de classes ao professor antes de qualquer codificação.

2. A aplicação deverá prever uma forma de tratamento específica para os funcionários e hóspedes, dentre outros que possam ser incorporados ao sistema no futuro. Por enquanto considere essa funcionalidade apenas para os funcionários e hóspedes. A forma de tratamento deve levar em conta o sexo da pessoa. Algumas sugestões:

   - Para hóspedes: Prezado Hóspede Senhor Silva, Prezada Hóspede Senhora Oliveira.

   - Para funcionários: Senhor José Silva, Senhora Joana Oliveira.

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 IdentificadorDePessoas que seja capaz de identificar os diferentes tipos de pessoas. Por exemplo: para uma determinada pessoa a funcionalidade proposta deve identificá-la como “Funcionário” ou “Hóspede”, dentre outros tipos de pessoas que podem ser incorporadas 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.**

Hotel – Parte 5

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

1. Vamos incorporar uma restrição no sistema. Elabore uma solução que permita a ocorrência (e o devido tratamento) de uma exceção para o caso de se registrar data de saída do hóspede inferior à data de entrada.

2. Disponibilizar os seguintes relatórios:

   a) A relação dos itens consumidos pelo hóspede em ordem decrescente pela descrição do consumo.
   b) A relação dos itens consumidos pelo hóspede em ordem crescente pela descrição do consumo.
   c) A relação dos itens consumidos pelo hóspede em ordem crescente pelo valor do consumo.
   d) Recibo ao hóspede com todos os dados da hospedagem, consumo, diárias, datas, etc.

3. Faça os testes necessários.

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

Hotel – Parte 6

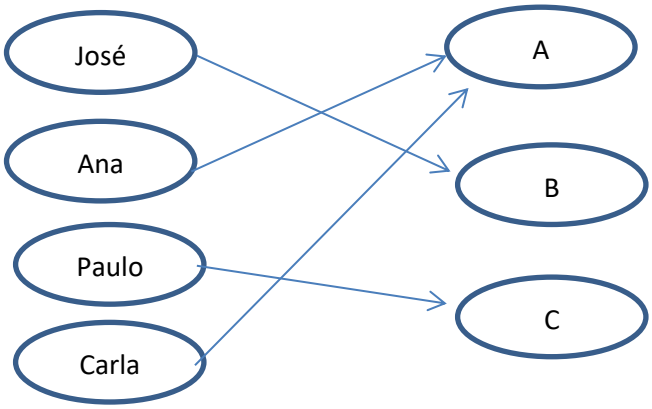
Conceito abordado: HashMap.

1. Criar um ranking de hóspede onde cada hóspede é único e estará vinculado a um valor acumulado dos valores pagos, portanto o ranking é acumulativo. Só poderá fazer parte do ranking, hóspedes com no mínimo uma diária registrada.

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

   Essa estrutura deve associar cada nome de hóspede a uma categoria. Categoria A para hóspedes com total de compras acima de R$30.000,00. Categoria B para hóspedes com total de compras acima de R$ 10.000,00 e abaixo de R$30.000,00. Categoria C para hóspedes com total de compras abaixo de R$ 10.000,00.

   Exemplo:



2. Disponibilize as seguintes operações a serem realizadas com o ranking de hóspedes:

   a) Listar todos os hóspedes e seus totais de gastos no hotel.
   b) Consultar a valor total que um hóspede gastou no hotel.
   c) Consultar a categoria de um hóspede.

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

4. Faça os testes necessários.

5. **APRESENTAÇÃO DO SISTEMA PARA AVALIAÇÃO.**

Hotel – Parte 7

Conceito abordado: Generics.

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

   Proponha e implemente uma solução capaz de generalizar o tratamento a qualquer tipo de lista, seja ela de consumo, de hóspedes, funcionários 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.**