|  |  |
| --- | --- |
|  | **UNIVERSIDADE FEDERAL DE UBERLÂNDIA**  Disciplina: **Programação Orientada a Objetos 1**  Curso: **Ciência da Computação**  Professora: **Elaine Ribeiro Faria** |

**Trabalho Final da Disciplina - Parte 1**

Tema: Implementação dos requisitos em Java

Instruções:

1. Faça a codificação seguindo o enunciado usando a linguagem Java.
2. Crie um arquivo txt chamado IntegrantesGrupo.txt contendo:

Nome e nro de matrícula dos integrantes do grupo

1. Envie pelo Microsoft Teams um arquivo. Zip contendo o código fonte (arquivos. Java do seu projeto) e o arquivo .txt

Data da entrega: 07/03/24 às 23:59

Avaliativa: **SIM - 15 pontos**

Perguntas:

1. Implementar usando a linguagem Java o sistema descrito a partir do conjunto de requisitos a seguir.

Domínio do problema: Udi-Decola

1. O Udi-Decola é uma companhia que permite realizar buscas e compras de passagens aéreas e diárias em hotéis.
2. Para o Udi-Decola é muito importante manter um cadastro dos trechos de voos disponíveis. Para cada trecho é importante saber o seu código identificador, que é formado por um conjunto de 3 letras e 5 nros. Também é importante armazenar o local de origem, o local de destino, a hora de saída e a hora de chegada. Pense na melhor forma de representar o local de origem e saída, pois uma cidade pode ter mais de um aeroporto. Também é preciso saber a companhia aérea do trecho voo.
3. As companhias aéreas devem ser previamente cadastradas. Cada companhia aérea tem um CNPJ, nome oficial, nome de divulgação e data de criação.
4. Cada trecho de voo é realizado em vários dias do mês. Por exemplo. O trecho Uberlândia para Garulhos-SP no horário das 8h e chegada as 9h, aconteceu todos os dias desta semana, sendo cada dia um voo diferente. É importante registrar os voos de cada trecho de voo.
5. Para cada voo é preciso saber a sua data, a hora programada de partida e hora programada de chegada, que podem ser diferentes da hora do trecho do voo devido a alguma mudança da companhia. É importante também saber o nro de vagas disponíveis nesse voo para passageiros da Udi-Decola. O mesmo trecho de voo pode ter diferentes valores em diferentes voos, que dependem da época do ano e de eventos externos. Assim cada voo tem um preço da passagem.
6. Além de voos, o Udi-decola também permite a busca por hotéis. Assim, é importante armazenar cada um dos hotéis parceiros. Para cada hotel é importante registrar o seu CNPJ, nome oficial, nome de divulgação, endereço completo, ano de criação, nro de estrelas, se aceita pets ou não, nro total de quartos, horário do *check-in* e horário do *chek-out*. Também é importante uma mensagem de divulgação que será exibida na busca. Ex: “Hotel 4 estrelas com *buffet* de café da manhã”. Uma breve descrição do hotel também deve ser exibida. Lembre-se que a cidade do hotel é importante para facilitar as buscas.
7. Para cada hotel é importante registrar para cada data do ano, o nro de quartos disponíveis para locação do tipo single, duplos e triplos, bem como o valor da diária de cada quarto sem desconto, o valor do desconto e valor após aplicar o desconto. Estes são considerados os quartos *standard*. Em cada data o hotel pode aceitar ou não cancelamento.
8. Para cada hotel é também importante registrar para cada data o nro de quartos de luxo, o valor da diária sem desconto, o valor do desconto e o valor após aplicar o desconto.
9. Os funcionários da empresa precisam ser cadastrados. Para isso é importante registrar o nome, CPF, endereço completo, nro da carteira de trabalho e data de nascimento. Cada funcionário tem um salário fixo. Lembre-se que um funcionário pode ser também um cliente da empresa.
10. Há um tipo de funcionário, que são os responsáveis por nova parcerias de hotel. Cada vez que um hotel é registrado é importante saber o funcionário responsável. Também é importante saber quantos hotéis um funcionário foi responsável pelo cadastro. O salário deste tipo de funcionário é o seu salário fixo acrescida de uma porcentagem fixa para cada novo hotel que ele adiciona.
11. Para efetivar uma compra no Udi-Decola, os clientes devem ser previamente cadastrados. Para isso é preciso saber, CPF, nome, endereço completo, data de nascimento, data de cadastro e email. Quando um cliente efetua mais que um dado número de compras, ele se torna um cliente VIP e neste caso é importante armazenar o valor do desconto que o cliente terá em novas compras, bem como a data em que ele se tornou vip. O cálculo deste valor segue regras específicas da empresa, que não serão detalhadas aqui.
12. As pesquisas feitas por cliente devem ser registradas. Para cada pesquisa é importante registrar a origem, o destino, data de início e data de fim. Isso ajudará a verificar os destinos e datas mais procurados. Também é importante armazenar a data e hora que a pesquisa foi realizada.
13. Cada compra deve ser registrada. Uma compra é composta por informações sobre a passagem aérea, diárias em hotel ou ambas. Obviamente é importante armazenar a quantidade de cada um desses itens, já que uma compra pode envolver passagens para mais de uma pessoa em mais de um quarto de hotel.
14. Para cada compra é importante registrar também a data e hora em que foi realizada, bem como o cliente da compra. A compra também armazena o seu valor total, que é a soma dos itens comprados, bem como a forma de pagamento realizada.
15. Para cada compra é importante atualizar o número de compras do cliente, já que ele pode se tornar um cliente VIP após realizar uma compra e atingir o nro limite de comprar.
16. Cada hotel paga um valor fixo para o Udi-Decola por cada diária de hotel vendida.
17. Cada companhia aérea paga um valor fixo para o Udi-Decola por cada passagem aérea vendida.

1. Implemente em Java, usando técnicas de encapsulamento **todo** o seu **Sistema para a Udi-Decola**. A seguir um exemplo:

public void setNome(String nome){

if (nome.length() > 0) {

this.nome = nome;

}

}

* 1. Para validação de cpf, implemente o algoritmo real de validação de cpf (que é encontrado facilmente na Internet, implementado nas mais diversas linguagens).

1. Considerando o uso de construtores:
   1. Crie um construtor para a classe cliente que receba como parâmetro o seu nome e o seu CPF.
   2. Crie um construtor default (sem parâmetros) explicitamente para a classe cliente.
   3. Crie um construtor para a classe Compra de forma que este receba como parâmetro o cliente que fez a compra, o vôo da compra, bem como a data e hora. Desta forma, obriga-se a associação da compra com o cliente e vôo, que já foram previamente criados no sistema.
   4. Crie construtores para cada uma das classes do problema de forma a atribuir valor aos atributos da classe com os parâmetros recebidos. Toda classe deve ter pelo menos um construtor criado por você.

1. Considerando os casos de herança, implemente todas as situações de herança descritas nos requisitos.

1. Implemente todos os atributos estáticos para o funcionamento do sistema, tais como o valor fixo que os hotéis e companhias aéreas pagam para o Udi-Decola, o nro fixo de compras que um cliente deve fazer para se tornar um cliente VIP e o valor fixo pago para funcionário que registram novos hotéis.

1. Implemente a realização de uma compra. Lembre-se que a realização de uma compra tem impacto na conversão de um cliente para VIP. Lembre-se de calcular para cada compra, o valor total a ser pago pelo cliente e o valor total a ser pago para o Udi-Decola.
2. Classes Abstratas
   1. Veja se alguma classe do problema deveria ser abstrata.

1. Requisito extra
   1. Invente um requisito no problema que justifique a criação de uma nova classe ou de um novo método (podem ou não ser abstratos) ou de uma nova subclasse. Não esqueça de descrever o requisito