COM220 – Trabalho 8

Parte 1

Elon Musk é um empreendedor bilionário famoso por ter criado empresas de sucesso tais como Paypal e Tesla Motors. Um de seus empreendimentos mais recentes é a SpaceX, cuja meta é permitir que cidadãos comuns possam fazer viagens espaciais. Em setembro de 2016, Musk apresentou um projeto para colonizar Marte. Para tanto, apresentou a proposta de um foguete capaz de transportar uma grande quantidade de pessoas a uma velocidade nunca antes imaginada, o que tornaria possível realizar uma viagem da Terra a Marte (cerca de 60 milhões de Km) em aproximadamente de 100 dias.

Nesta prova você deverá implementar um sistema que permita controlar viagens a Marte. Para tanto, você deverá implementar o modelo de dados apresentado no verso. O modelo prevê 3 tipos de viajantes: passageiros, pilotos e comissários. Pilotos e comissários são considerados tripulantes, por isso não precisam de ticket de embarque. Já os pilotos precisam de uma habilitação e os comissários de uma licença. O modelo prevê também a criação de naves, que possuem um nome e devem indicar a quantidade máxima de tripulantes e passageiros que comportam, bem como a velocidade máxima de cruzeiro.

Utilizando instâncias de **Nave**, **Passageiro**, **Piloto** e **Comissario** é possível criar uma instância de **Viagem**. Além da nave e de um arrayList de viajantes (passageiros, pilotos e comissários), deve-se informar a data da viagem e sua distância. Assim, é possível calcular o número de dias dividindo a distância pela velocidade da nave (deve-se dividir o resultado por 24, caso contrário obtém-se o número de horas e não de dias).

Na criação de uma viagem é necessário adicionar tripulantes e passageiros ao arrayList de viajantes. Note que os métodos que adicionam viajantes devem levar em conta o número máximo de passageiros e tripulantes que a nova comporta. Assim, um passageiro só pode ser adicionado à lista se a nave não estiver com o número máximo de passageiros. O mesmo vale para tripulantes. Para tanto, os métodos addPassageiro e addTripulante devem lançar exceptions se o número máximo de passageiros ou tripulantes for atingido, impedindo que o novo passageiro ou tripulante seja adicionado. Após adicionar um passageiro, deve-se adicionar, utilizando o método addAlimento, todos os alimentos que o passageiro é alérgico.

Você deve seguir a risca o modelo de dados fornecido e utilizar a classe de teste mostrada a seguir.

```
import java.util.Date;
public class TestaImpl {
    public static void main(String args[]) {
        Nave n = new Nave("BFR", 3, 2, 20000);
        Viagem v = new Viagem(n, 60000000, new Date());
        Passageiro pas1 = new Passageiro("Antonio Siqueira", "111222", "1001");
        pas1.addAlimento("cafe");
        Passageiro pas2 = new Passageiro("Marcos Silva", "333888", "1002");
        pas2.addAlimento("leite");
        pas2.addAlimento("camarao");
        Passageiro pas3 = new Passageiro("Maria Carvalho", "999222", "1003");
        try{
            v.addPassageiro(pas1); v.addPassageiro(pas2); v.addPassageiro(pas3);
        } catch (Exception e){
            System.out.println(e.getMessage());
        }
        Piloto pil1 = new Piloto("Marcos Pontes", "999000", "PIL1234");
        Comissario com1 = new Comissario("Ricardo Santos", "999555", "COM1234");
        Comissario com2 = new Comissario("Maria Rosario", "999222", "COM5678");
        try{
            v.addTripulante(pil1); v.addTripulante(com1); v.addTripulante(com2);
        } catch (Exception e){
            System.out.println(e.getMessage());
            // deve imprimir: "Tripulante Maria Rosario não pode ser incluído(a)."
        }
        System.out.println("\nUtilizando a nave " + n.getNome() + " a duração "
                + "da viagem será " + v.calculaDuracao() + " dias.\n");
                                        //deve imprimir 125 dias
        System.out.println("Lista de tripulantes: \n" + v.getListaTripulantes());
                    //deve imprimir nome - identidade - habilitacao ou licença
        System.out.println("Lista de passageiros: \n" + v.getListaPassageiros());
                    //deve imprimir nome - identidade - nro ticket
    } }
```

Parte 2

Crie classes de controle e visão a fim de permitir a inserção e a consulta de viagens espaciais. Deve ser possível criar várias viagens, em diferentes datas, utilizando diferentes naves. Para cada viagem, o sistema dever permitir a alocação de uma nave e a adição de passageiros e tripulantes. O sistema deve prover um menu com as seguintes opções:

1) Cadastrar passageiros
2) Cadastrar tripulantes
3) Cadastrar naves
4) Cadastrar viagens
5) Consultar viagens
6) Sair

A opção 5 deve listar a data da viagem, sua duração, a nave utilizada e a lista de passageiros e tripulantes.