

Uma empresa de fretamento possui hoje uma frota de 8 vans (numeradas de 1 a 8) e faz uma rota regular do aeroporto de Congonhas para Guarulhos (ida e volta), com uma garagem em cada destino. Pretende atender a outros aeroportos em um futuro próximo, adquirindo ou não mais veículos.

Essas garagens tem um único acesso e o estacionamento é feito de ré, deixando o veículo posicionado para a saída do local, um na frente do outro.

Sempre que a lotação de um carro está completa, este sai em direção ao destino e, lá chegando, é estacionado na garagem.

Quando um estacionamento fica vazio, uma nova viagem só pode ser iniciada desta origem quando um veículo retornar à garagem.

Todo início de jornada diária, os veículos são alternadamente distribuídos entre os destinos (garagens). Uma viagem só pode ser liberada quando a jornada diária foi iniciada.

O encerramento da jornada consiste na geração de uma lista informando, veículo a veículo, a quantidade de passageiros transportados com a "limpeza" das ocorrências das viagens anteriores.

Cadastros de novos veículos e novas garagens só podem ser realizados com a jornada diária encerrada.

Elabore um programa que permita controlar o fluxo da frota com as seguintes opções:

1. Cadastrar veículo
2. Cadastrar garagem
3. Iniciar jornada
4. Encerrar jornada
5. Liberar viagem de uma determinada origem para um determinado destino
6. Listar veículos em determinada garagem
7. Informar qtde de viagens efetuadas de uma determinada origem para um determinado destino
8. Listar viagens efetuadas de uma determinada origem para um determinado destino
9. Informar qtde de passageiros transportados de uma determinada origem para um determinado destino

Utilizar o seguinte modelo para as classes do processo:

Veiculo

-id: int
-placa: string;
-lotacao: int;

Veiculos

-veiculos: List<Veiculo>
+incluir(Veiculo veiculo)

Transporte

-veiculo: Veiculo
-qtdeTransportada: int

Garagem

-id: int
-local: string
-veiculos: Stack<Veiculo>
+qtdeDeVeiculos(): int
+potencialDeTransporte(): int

Garagens

-garagens: List<Garagem>
-jornadaAtiva: bool
+incluir(Garagem garagem)
+iniciarJornada(): void
+encerrarJornada(): List<Transporte>

Viagem

-id: int
-origem: Garagem
-destino: Garagem
-veiculo: Veiculo

Viagens

-viagens: Queue<Viagem>
+incluir(Viagem viagem)