	SERVIÇO PÚBLICO FEDERAL **MINISTÉRIO DA EDUCAÇÃO** **CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS** **UNIDADE DE ENSINO DESCENTRALIZADA DE DIVINÓPOLIS**				CEFET-MG
Curso:	*Engenharia de Computação*	Disciplina:	*Laboratório de Programação de Computadores II*	Período	*2º*
Professor(a):	Eduardo Habib Bechelane Maia			Valor:	***5***
Data de Entrega:	*12/01/2025*	Lista: 6			

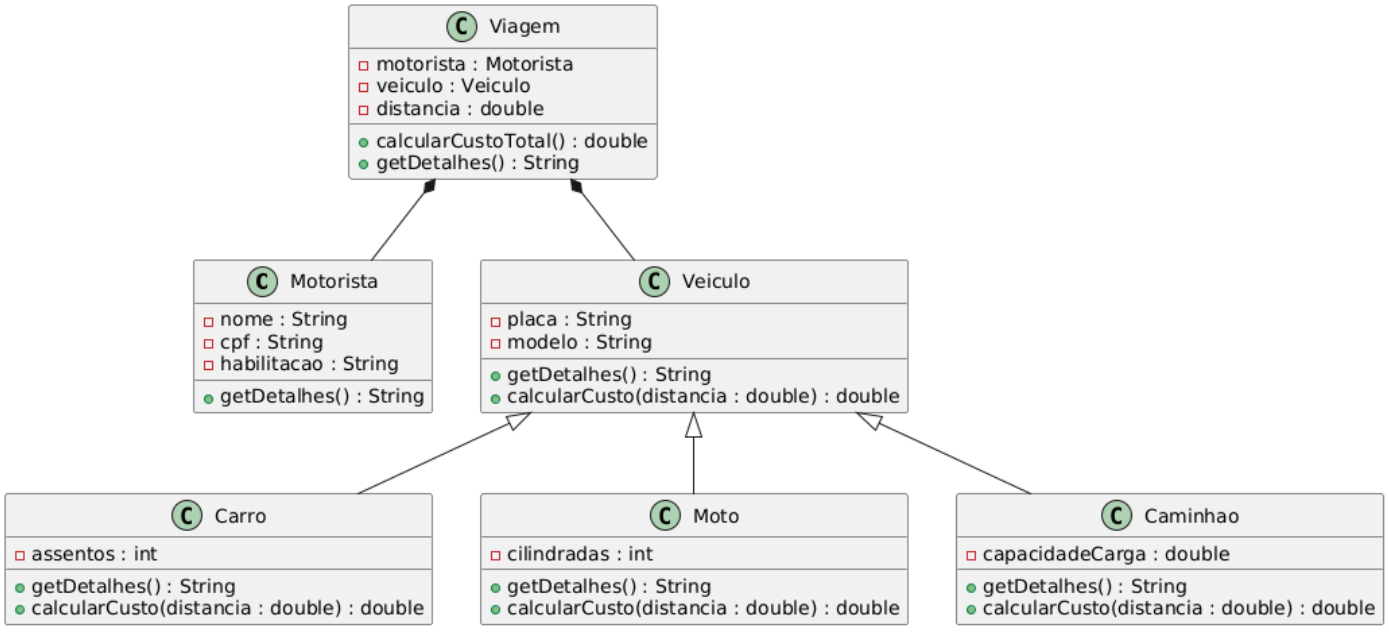
Exercício - Sistema de Gerenciamento de Veículos e Motoristas

Descrição:

Você foi contratado por uma empresa de logística chamada **RoadLink** para o desenvolvimento de um sistema. A empresa lida com motoristas especializados e uma frota diversificada, incluindo motos ágeis para entregas rápidas, carros para transporte de passageiros, e caminhões robustos para cargas pesadas.

O desafio é desenvolver um sistema inteligente que gerencie os motoristas, seus veículos e as viagens realizadas. O sistema deve garantir que cada viagem seja feita com eficiência, calculando o custo de transporte com base no tipo de veículo e na distância percorrida. Além disso, é crucial que a empresa respeite as regulamentações de habilitação, garantindo que apenas motoristas com a categoria correta possam operar os veículos designados.

Segue abaixo o diagrama de classes do sistema em questão.



Instruções para a implementação:

1. **Implemente todas as classes de acordo com o diagrama.**
2. Cada classe deve conter:
   - **Construtores para inicializar os atributos, getters e setters.**
   - **Métodos getDetalhes()** em todas as classes, que retornam uma string com os detalhes da classe.

3. A classe Viagem deve:
   - Calcular o custo total da viagem com base na distância e no tipo de veículo. Para isso, cada tipo de veículo (Carro, Moto, Caminhao) deve ter um método calcularCusto() que implemente a lógica específica do custo por quilômetro (use polimorfismo).

4. Crie um programa principal (main) que inclua o seguinte menu de opções:
   - **Cadastrar Motorista:** Insira o nome, CPF e habilitação do motorista.
   - **Cadastrar Veículo:** Permita cadastrar veículos dos tipos Carro, Moto ou Caminhao.

- **Registrar Viagem:** Associe um motorista e um veículo a uma viagem, informando a distância.
- **Exibir Detalhes de uma Viagem:** Exiba o motorista, o veículo e o custo total da viagem.
- **Excluir Motorista:** Permita remover um motorista pelo CPF.
- **Excluir Veículo:** Permita remover um veículo pela placa.
- **Sair:** Finaliza o programa.

Regras de Negócio

1. Cada motorista deve possuir uma habilitação compatível com o tipo de veículo.
   - **Moto:** Categoria A
   - **Carro:** Categoria B
   - **Caminhao:** Categoria C ou superior
2. Os custos por quilômetro para cada tipo de veículo são:
   - **Moto:** R$ 0,20/km
   - **Carro:** R$ 0,50/km
   - **Caminhao:** R$ 1,00/km

Explicação sobre o polimorfismo:

O **polimorfismo** será implementado no sistema por meio da classe base Veiculo e de suas subclasses Carro, Moto e Caminhao. A classe base define o método calcularCusto(double distancia), que é sobrescrito (override) em cada subclasse para implementar a lógica específica de cálculo do custo de transporte por quilômetro.

- Em **Java**, você usará a palavra-chave @Override para sobrescrever o método na subclasse. A chamada ao método será polimórfica: mesmo que o objeto seja referenciado como um Veiculo, o método correspondente à subclasse será executado.

Exemplo:

```
Veiculo v = new Carro("ABC-1234", "SUV", 5);
System.out.println(v.calcularCusto(100)); // Chamará a implementação de Carro
```

- Em **C++**, o método será declarado como virtual na classe base, garantindo que a implementação na subclasse seja chamada em tempo de execução.

Exemplo:

```
Veiculo* v = new Carro("ABC-1234", "SUV", 5);
cout << v->calcularCusto(100); // Chamará a implementação de Carro
```

IMPLEMENTE O LABORATÓRIO NA LINGUAGEM DE SUA PREFERÊNCIA