

 UNINTA	TUTOR: Alan Marques da Rocha
	CURSO: Bacharelado em Engenharia da Computação
	DISCIPLINA: Programação Orientada a Objetos (POO)
	PERÍODO: 3º e 4º semestres
	DATA: 13/11/2025
ALUNO:	

Desafio de Programação Orientada a Objetos

Sistema de Gestão de Frota e Abastecimento

Você é um aluno de Engenharia de Computação e trabalha em uma empresa de gestão de frota e abastecimento chamada **POO Frota LTDA**. Você foi contratado para projetar o sistema de controle de uma frota de veículos e o registro de seus abastecimentos.

Seu objetivo é modelar, usando os princípios de **Programação Orientada a Objetos (POO)**, as entidades principais deste sistema, focando em encapsulamento e na relação entre classes.

1. Requisitos Mínimos para a Modelagem

Classe veículo:

Esta classe representa a entidade principal da frota:

Elemento	Nome	Descrição	Conceito de POO
Atributo	Placa	Identificador único	Estado (Atributo)

Atributo	QuilometragemAtual	Distância percorrida. Deve ser iniciada em 0.0	Estado (Atributo)
Atributo	ListaAbastecimentos	Estrutura para armazenar objetos Abastecimento	Associação
Método	Acelerar()	Simula uma ação simples	Comportamento
Método	VerificarStatus()	Exibe todas as informações do veículo e seus abastecimentos	Comportamento

Classe Abastecimento:

Esta classe representa a entidade principal da frota:

Elemento	Nome	Descrição	Conceito de POO
Atributo	Data	Quando ocorreu	Estado (Atributo)
Atributo	Litros, ValorTotal, kmNoAbastecimento	Dados quantitativos do evento	Estado (Atributo)
Atributo	VeiculoAssociado	Uma referência ao objeto Veículo que foi abastecido	Associação

2. O desafio do Encapsulamento

O ponto crucial e o seu desafio é proteger a integridade do dado QuilometragemAtual. Ela NUNCA deve ser alterada diretamente ou retroceder.

Método	Classe	Responsabilidade
PRIVADO/PROTEGIDO <code>_AtualizarQuilometragem(novaKm)</code>	Veiculo	ÚNICO método com permissão para alterar QuilometragemAtual. Sua lógica deve garantir que a atualização só ocorra se novaKm for maior que o valor atual.
PÚBLICO <code>RegistrarAbastecimento(litros, valorTotal, kmNoAbastecimento)</code>	Veiculo	Dados quantitativos do evento Estado (Atributo)

3. Exemplo da Regra de Encapsulamento

Ação	QuilometragemAtual (antes)	kmNoAbastecimento Fornecida	QuilometragemAtual (depois)	Resultado da Lógica
1º Abastecimento	0.0 km	500.0 km	500.0 km	Atualizada (500.0 > 0.0)
2º Abastecimento	500.0 km	1200.0 km	1200.0 km	Atualizada (1200.0 > 500.0)
3º Abastecimento	1200.0 km	1000.0 km	1200.0 km	Não atualizada (1000.0 < 1200.0)

O que foi pedido pelo gestor:

Desenvolva código-fonte que implemente as classes e os métodos conforme as especificações. O código final deve incluir uma seção principal que **demonstre a utilização** das classes, contendo no mínimo:

1. A criação de dois (2) objetos Veiculo diferentes;
2. O registro de, pelo menos, três (3) objetos Abastecimento para um dos veículos, incluindo uma **tentativa de registro com KM menor** para provar que a lógica de encapsulamento funciona;
3. A chamada ao método VerificarStatus() para cada veículo após os registros.

Saída esperada do programa (para dois veículos):

```
... --- STATUS FINAL ---  
=====  
[STATUS DO VEÍCULO: Focus (ABC-1234)]  
=====  
Marca/Modelo: Ford/Focus  
Ano: 2022  
Placa: ABC-1234  
KM Atual: 1200.8 km  
Total de Abastecimentos Registrados: 3  
  
--- Histórico de Abastecimentos ---  
- Data: 13/11/2025 15:33:04 | Litros: 45.00 L | Valor: R$ 270.00 | KM Registrada: 500.5  
- Data: 13/11/2025 15:33:04 | Litros: 30.00 L | Valor: R$ 180.00 | KM Registrada: 1200.8  
- Data: 13/11/2025 15:33:48 | Litros: 10.00 L | Valor: R$ 60.00 | KM Registrada: 1000.0  
=====  
[STATUS DO VEÍCULO: Biz 125 (MOTO-5678)]  
=====  
Marca/Modelo: Honda/Biz 125  
Ano: 2023  
Placa: MOTO-5678  
KM Atual: 300.7 km  
Total de Abastecimentos Registrados: 2  
  
--- Histórico de Abastecimentos ---  
- Data: 13/11/2025 15:33:59 | Litros: 5.00 L | Valor: R$ 30.00 | KM Registrada: 150.0  
- Data: 13/11/2025 15:33:59 | Litros: 7.00 L | Valor: R$ 42.00 | KM Registrada: 300.7  
=====
```