INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BRASÍLIA

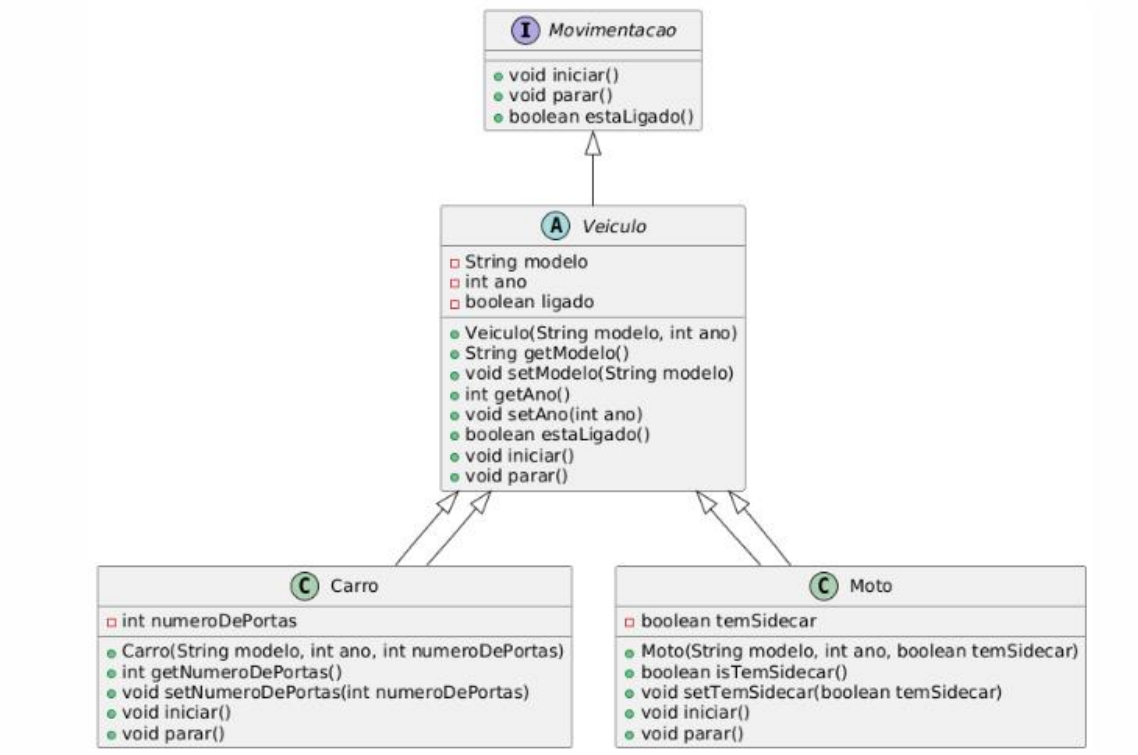
TECNÓLOGO SISTEMAS PARA INTERNET

Professor: Gustavo Henrique Dornelas de Deus

Avaliação

Encapsulamento, herança, polimorfismo, classe abstrata

Questão 1 – Você deve desenvolver um sistema simples de gerenciamento de veículos que utiliza conceitos de Programação Orientada a Objetos, incluindo herança, interfaces, métodos construtores e a implementação de um método `main` para testar os cenários. O sistema deve ser construído em Java (ou outra linguagem de sua escolha que suporte POO).



Especificação:

1. **Interface Movimentacao →** Defina uma interface chamada Movimentacao com os seguintes métodos:
    - void iniciar(): Inicia o veículo.
    - void parar(): Para o veículo.

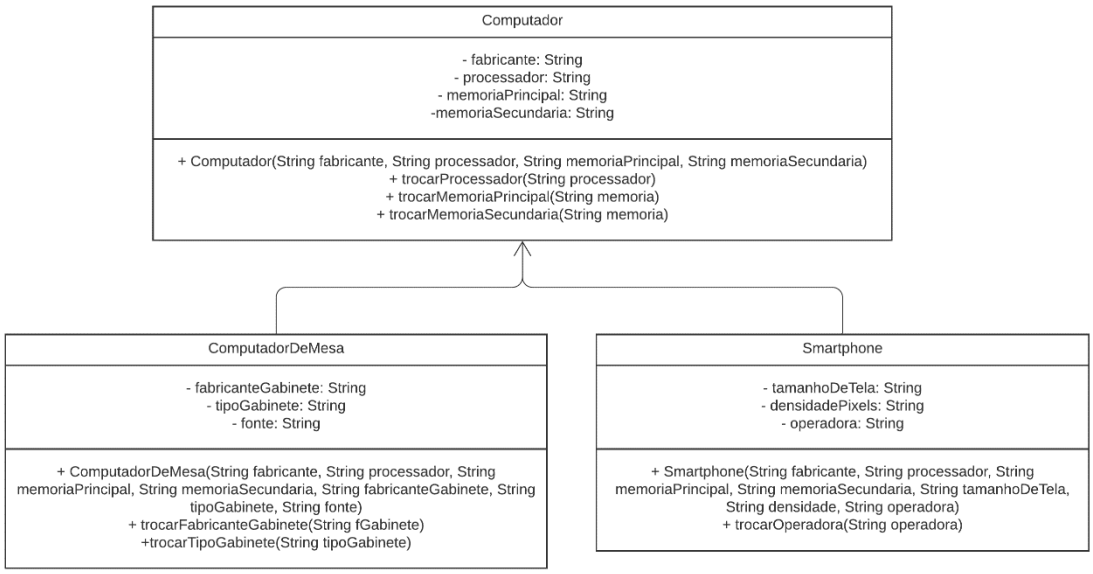
- boolean estaLigado(): Retorna um valor booleano indicando se o veículo está ligado.

2. **Classe Abstrata Veiculo →** Crie uma classe abstrata chamada Veiculo que implementa a interface Movimentacao.
   - A classe Veiculo deve ter os seguintes atributos:
     - String modelo
     - int ano
   - A classe deve ter um construtor que inicializa todos os atributos.
   - Implemente os métodos da interface Movimentacao com comportamentos padrão que podem ser sobrescritos por classes derivadas. Use um atributo boolean chamado ligado para controlar se o veículo está ligado.
   - Forneça métodos getters e setters para todos os atributos.
3. **Classe Carro**
   - Crie uma classe Carro que herda de Veiculo.
   - A classe Carro deve adicionar um atributo específico:
     - int numeroDePortas
   - A classe deve ter um construtor que inicializa todos os atributos, incluindo os da classe base.
   - Sobreponha os métodos da interface Movimentacao para implementar comportamentos específicos para carros. Por exemplo, a implementação do método iniciar() pode incluir uma mensagem indicando que o carro está pronto para dirigir.
4. **Classe Moto**
   - Crie uma classe Moto que herda de Veiculo.
   - A classe Moto deve adicionar um atributo específico:
     - boolean temSidecar
   - A classe deve ter um construtor que inicializa todos os atributos, incluindo os da classe base.
   - Sobreponha os métodos da interface Movimentacao para implementar comportamentos específicos para motos. Por exemplo, a implementação do método parar() pode incluir uma mensagem indicando que a moto está parada e pronta para a próxima viagem.
5. **Classe Main**
   - Implemente a classe Main com o método main para testar os cenários:
     - Crie pelo menos um objeto de cada tipo de veículo (Carro e Moto).
     - Teste os métodos iniciar(), parar() e estaLigado() em cada objeto.

- Exiba informações sobre cada veículo, como modelo, ano, e atributos específicos (por exemplo, número de portas para Carro e se tem sidecar para Moto).

Questão 2 – Assistência técnica

Implemente as classes do sistema da loja de informática. Observe as descrições das tabelas abaixo. Crie uma classe de teste e instancie exemplos de objetos do tipo ComputadorDeMesa e Smartphone.



Classe:	Computador			
Atributos	Identificador	Tipo	Descrição	
	fabricante	String	Nome da fabricante do computador.	
	processador	String	Identificação do processador computador.	
	memoriaPrincipal	String	Quantidade de memória principal.	
	memoriaSecundaria	String	Quantidade de memória secundária.	
Metódos	Identificador	Parâmetros	Retorno	Descrição
	Computador	String fabricante, String processador, String memoriaPrincipal e String memoriaSecundaria	-	Construtor da classe Computador. Recebe o nome da fabricante, identificação do processador e as quantidades de memórias principal e secundária como parâmetro.
	trocarProcessador	String processador	void	Altera o processador do computador.
	trocarMemoriaPrincipal	String memoria	void	Altera a memória principal do computador.
	trocarMemoriaPrincipal	String memoria	void	Altera a memória secundária do computador.

Classe:	ComputadorDeMesa		Classe pai:	Computador
Atributos	Identificador	Tipo	Descrição	
	fabricanteGabinete	String	Nome da fabricante do gabinete.	

	tipoGabinete	String	Tipo do gabinete.	
	fonte	String	Especificação da fonte do computador.	
Metódos	Identificador	Parâmetros	Retorno	Descrição
	ComputadorDeMesa	String fabricante, String processador, String memoriaPrincipal, String memoriaSecundaria, String fGabinete, String tipoGabinete e String fonte	-	Construtor da classe ComputadorDeMesa. Inicializa os atributos da classe. Utilizar o construtor da classe pai (Computador).
	trocarFabricanteGabinete	String fGabinete	void	Altera o fabricante do gabinete do computador.
	trocarTipoGabinete	String tipoGabinete	void	Altera o tipo do gabinete do computador.
	trocarFonte	String fonte	void	Altera a especificação da fonte do computador de mesa.

Classe:	Smartphone	Classe pai:	Computador
Atributos	Identificador	Tipo	Descrição
	tamanhoDeTela	String	Especificação do tamanho da tela do smartphone.
	densidadePixels	String	Densidade de pixels do smartphone.
	operadora	String	Operadora telefônica com chip presente no smartphone.
Metódos	Identificador	Parâmetros	Retorno	Descrição
	Smartphone	String fabricante, String processador, String memoriaPrincipal, String memoriaSecundaria, String tamanhoTela, String densidade e String operadora	-	Construtor da classe Smartphone. Inicializa os atributos da classe. Utilizar o construtor da classe pai (Computador).
	trocarOperadora	String operadora	void	Altera operadora telefônica do smartphone.