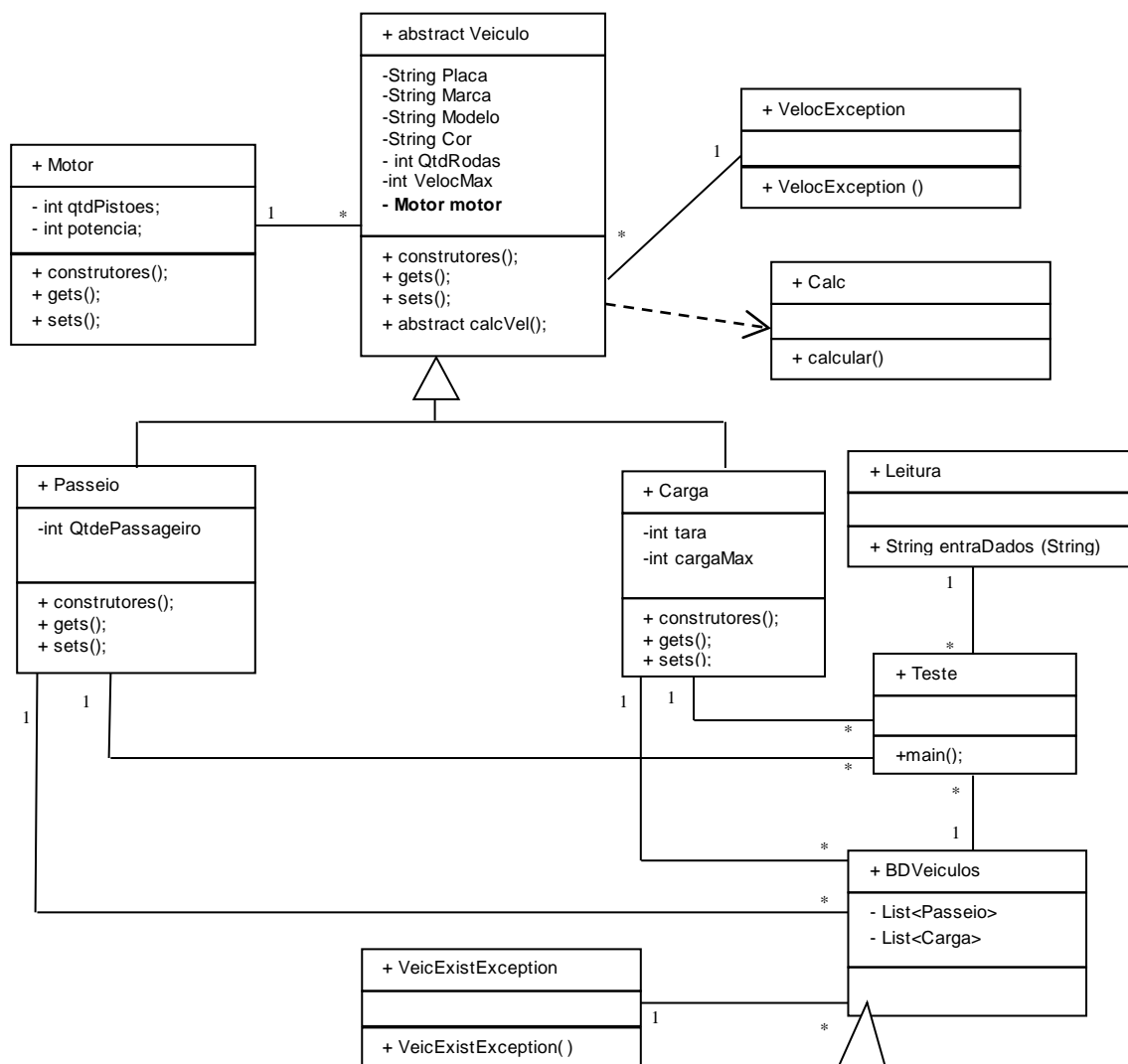


Lista de Exercícios

Forma de entrega:

- TODAS as classes deverão conter na primeira linha (de forma comentada) o nome completo do aluno;
- Compacte a pasta do projeto (que contém os arquivos .java) e nomeie o arquivo como: *Espec_Java_2016-1_nome do aluno* (exemplo: **Espec_Java_2016-1_PedroAlvaresCabral.rar** (ou .zip) e poste via Moodle



Os métodos desta deverão ser construídos de acordo com as necessidades apresentadas pelo enunciado.

SUBSÍDIOS:

Não deverá utilizar o recurso de GUI (interface gráfica)

A classe Teste deverá conter o **Menu Inicial** com as seguintes opções (e suas funcionalidade desenvolvidas):

Sistema de Gestão de Veículos - Menu Inicial

- 1)_ Cadastrar Veículo de Passeio
- 2)_ Cadastrar Veículo de Carga
- 3)_ Imprimir Todos os Veículos de Passeio
- 4)_ Imprimir Todos os Veículos de Carga
- 5)_ Imprimir Veículo de Passeio pela Placa
- 6)_ Imprimir Veículo de Carga pela Placa
- 7)_ Alterar dados do Veículo de Passeio pela Placa
- 8)_ Alterar dados do Veículo de Carga pela Placa
- 9)_ Sair do Sistema

Observações sobre as opções do Menu Inicial:

Opções 5 e 6)- O usuário informa placa do veículo, o programa executa uma busca no List do respectivo tipo de Veículo de imprime os dados deste;

Opções 7 e 8)_ O usuário informa placa do veículo, o programa executa uma busca no List do respectivo tipo de Veículo de imprime os dados deste, logo em seguida oferece condições para mudar os dados, EXCETO AS PLACAS, POIS SE FIZEREM ISSO, JÁ NÃO SERÁ O MESMO VEÍCULO

O Sistema deverá permitir:

- **A saída do sistema** só será possível caso o usuário escolha a opção 9, isto significa que a cada operação realizada no sistema, este retornará ao Menu Inicial;
- **Armazenar uma quantidade indefinida** de veículos de cada tipo (passeio e carga) e, lógico, quando necessário exibir todos eles;
- **Ao inserir um novo veículo** o sistema deverá verificar se já existe um veículo com a mesma **placa**. Caso exista deverá:
 - disparar a exceção VeicExistException (que será do tipo **verificada**);
 - informar ao usuário sobre esta existência; e
 - fornecer ao usuário a opção de sobrescrever o veículo pré-existente ou, caso o usuário opte por não sobrescrever o veículo, o sistema deverá “perguntar” se o usuário deseja cadastrar outro veículo do mesmo tipo, caso este responda **NÃO** o sistema deverá voltar ao Menu Inicial.
- Ao final do cadastro de cada veículo, deverá ser perguntado se deseja cadastrar mais um (do mesmo tipo), caso a resposta seja **NÃO** deverá voltar ao Menu Inicial;
- A “entrada” e armazenamento da velocidade máxima dos veículos sempre serão em Km/h;
- Caso a velocidade máxima de um veículo seja menor que 100 e maior 250, deverá disparar a exceção “VelocException” (que será do tipo **verificada**), informando que: “A velocidade máxima está fora dos limites brasileiros”. Neste caso (após disparar a exceção), se estiver “tentando” atribuir velocidade para um carro de passeio, este deverá assumir 150 (Km/h para velocidade máxima) se for um veículo de carga assumirá 120 Km/h;

- As velocidades sempre serão armazenadas em Km/h. Porém, **apenas para efeito de exibição por meio da classe Teste**:
 - A velocidade do carro de **passeio** deverá ser calculada em M/h (metros por hora);
 - A velocidade do **caminhão** deverá ser calculada em Cm/h (centímetros por hora);**Obs.:** Para isso o método abstrato **calcVel** deverá retornar o resultado do cálculo e este exibido na classe Teste;
- Perceba que, pelo diagrama não há uma associação entre as classes **Veículo e Teste** nem entre **Motor e Teste**. Mas deverá instanciar motor (dos Veículos), na classe Teste, **sem declarar um objeto do tipo motor nas classes**: Passeio, Carga e Teste. Somente haverá declaração na classe Veiculo;
- O método “**calcular**” da **interface “Calc”** deverá retornar um valor que será exibido pela classe Teste. Este valor será:
 - **Passeio**: será soma das quantidades de letras existentes em todos os atributos do tipo String;
 - **Carga**: será a soma de todos os valores de todos contidos em todos os atributos numéricos;
- **TODAS as impressões deverão ser feitas na classe Teste**. Antes de exibir os dados de cada veículo ou ao listar todos os veículos, será mostrada, apenas uma vez no início da exibição, a data e hora do sistema.
- **TODAS AS EXCEÇÕES DEVERÃO SER TRATADAS;**
- **EXCETO PARA OS MÉTODOS GETTERS E SETTERS, TODOS OS PARÂMETROS DEVERÃO SER OBJETOS;**
- **ALÉM DOS MÉTODOS JÁ EXPLICITADOS, PODERÃO CRIAR OUTROS QUE *REALMENTE* SEJAM NECESSÁRIOS;**
- **NÃO** PERMITA QUE:
 - QUE SEJA POSSÍVEL INSTANCIAR OBJETOS DIRETOS DA CLASSE MÃE (Veiculo)
 - SOBRESCREVAM OS MÉTODOS SETTERS DA CLASSE VEICULO;
 - SOBRESCREVAM OS MÉTODOS DAS CLASSES FILHAS (Passeio e Carga);
 - QUE AS CLASSES FILHAS (Passeio e Carga) SEJAM HERDADAS;